Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-8)

A second-order difference equation with sign-alternating coefficients

Carlos M. da Fonseca[1,2*], Can Kızılateş[3], Nazlıhan Terzioğlu[3]
[1]*Kuwait College of Science and Technology, P.O. Box 27235, Safat 13133, Kuwait*
[2]*Chair of Computational Mathematics, University of Deusto, 48007 Bilbao, Spain*
[3]*Dept. of Mathematics, Zonguldak Bülent Ecevit University, Zonguldak, 67100, Turkey*
**Corresponding author: c.dafonseca@kcst.edu.kw*

Abstract

We provide an explicit solution for the terms of the sequence $(x_{n,k})$ defined by

$$x_{n,k} = x_{n-1,k} - (-1)^{\lfloor (n-2)/k \rfloor} x_{n-2,k}\,,$$

for $n \geqslant 3$, setting $x_{1,k} = 1$ and $x_{2,k} = 0$. Several particular examples are considered.

Keywords: Chebyshev polynomials of the second kind; difference equations; Fibonacci numbers; integer sequences; tridiagonal matrices

1. Introduction

Second order linear difference equations emerge in distinct areas of mathematics. For example, solutions of constant coefficient homogeneous equations consist of many sequences of numbers, such as Fibonacci. In many instances, finding explicit forms of homogeneous second order equations is a ceaseless problem in research. Moreover, many results are focused exclusively to this aim. For several relevant references the reader is refereed to (Koshy, T., 2018; Kızılateş, C., 2021; da Fonseca, C.M., 2014).

Recently, in (Andelić, M. *et al.*, 2020) it was proposed to establish an explicit expression for the sequence $(x_{n,k})$ where

$$x_{n,k} = x_{n-1,k} - (-1)^{\lfloor (n-2)/k \rfloor} x_{n-2,k}\,, \quad \text{for } n \geqslant 3, \tag{1}$$

setting $x_{1,k} = 1$ and $x_{2,k} = 0$. This problem was motivated by similar questions originally proposed in (Trojovský, P., 2017), namely when the recurrence relations

$$y_{n,k} = (-1)^{\lfloor (n-1)/k \rfloor} y_{n-1,k} - y_{n-2,k}\,,$$

and

$$z_{n,k} = (-1)^{\lfloor (n-1)/k \rfloor} z_{n-1,k} - (-1)^{\lfloor (n-2)/k \rfloor} z_{n-2,k}$$

are satisfied. The solutions of these cases were obtained in terms of the Fibonacci sequence, defined by the standard recurrence relation $F_{n+2} = F_{n+1} + F_n$, for $n \geqslant 0$, with $F_0 = 0$ and $F_1 = 1$.

In this note, we provide a close formula for Equation (1) and discuss some particular cases. In the next section, we recall the formula of the determinant of a tridiagonal k-Toeplitz matrix. Then using a similar approach we have seen for example in (Andelić, M. *et al.*, 2020; Andelić, M *et al.*, 2011; Du, Z. *et al.*, 2022), we establish the requested formula. We also consider in particular the cases $k = 2, 3$ in detail. In the final section, we discuss several general instances.

2. The formula

From (Rózsa, P., 1969), we know that the determinant of the tridiagonal k-Toeplitz matrix

$$A_n = \begin{pmatrix} a_1 & b_1 & & & & & & & \\ 1 & \ddots & \ddots & & & & & & \\ & \ddots & a_k & b_k & & & & & \\ & & 1 & a_1 & b_1 & & & & \\ & & & 1 & \ddots & \ddots & & & \\ & & & & \ddots & a_k & b_k & & \\ & & & & & 1 & a_1 & b_1 & \\ & & & & & & 1 & \ddots & \ddots \\ & & & & & & & \ddots & \end{pmatrix}_{n\times n}, \tag{2}$$

is given by

$$\det A_n = (\sqrt{b_1}\cdots\sqrt{b_k})^q \left(\Delta_{1,\ldots,r}\, U_q(x) + \frac{\sqrt{b_k}\sqrt{b_1}\cdots\sqrt{b_r}}{\sqrt{b_{r+1}}\cdots\sqrt{b_{k-1}}}\, \Delta_{r+2,\ldots,k-1}\, U_{q-1}(x) \right),$$

where

$$x = \frac{\Delta_{1,\ldots,k} - b_k \Delta_{2,\ldots,k-1}}{2\sqrt{b_1}\cdots\sqrt{b_k}},$$

with $\Delta_{1,\ldots,k} = \det A_k$ and where $U_q(x)$ is the Chebyshev polynomials of the second kind and $n = qk+r$, with $0 \leqslant r \leqslant k-1$. In general, by $\Delta_{i,\ldots,j}$ we understand the determinant of the submatrix obtained A_k with rows and columns indexed by $\{i,\ldots,j\}$. An independent approach can be found in (Fonseca, C.M. & Petronilho, J., 2005).

Now, let us define now

$$T_{n,k} = \begin{pmatrix} 1 & 1 & & & & & & & \\ 1 & \ddots & \ddots & & & & & & \\ & \ddots & \ddots & 1 & & & & & \\ & & 1 & \ddots & -1 & & & & \\ & & & 1 & \ddots & \ddots & & & \\ & & & & \ddots & \ddots & -1 & & \\ & & & & & 1 & \ddots & 1 & \\ & & & & & & 1 & \ddots & \ddots \\ & & & & & & & \ddots & \end{pmatrix}_{n\times n}, \tag{3}$$

where the superdiagonal is of the form

$$(\underbrace{1,\ldots,1}_{k\times},\underbrace{-1,\ldots,-1}_{k\times},\underbrace{1,\ldots,1}_{k\times},\underbrace{-1,\ldots,-1}_{k\times},1,\ldots). \tag{4}$$

If we set

$$(b_1,\ldots,b_k,b_{k-1}\ldots,b_{2k}) = (\underbrace{1,\ldots,1}_{k\times},\underbrace{-1,\ldots,-1}_{k\times})$$

and replace k by $2k$ in (2), the matrix defined in Equation (3) can be seen as a tridiagonal $2k$-Toeplitz matrix. Hence, we have explicitly

$$\det T_{n,k} = x_{n,k} = i^{qk}\left(\Delta_{1,\ldots,r}\, U_q(x) + \frac{\sqrt{b_{2k}}\sqrt{b_1}\cdots\sqrt{b_r}}{\sqrt{b_{r+1}}\cdots\sqrt{b_{2k-1}}}\,\Delta_{r+2,\ldots,2k-1}\, U_{q-1}(x)\right),$$

where

$$x = \frac{\Delta_{1,\ldots,2k} + \Delta_{2,\ldots,2k-1}}{2i^k}, \tag{5}$$

with $n = 2qk + r$, for $0 \leqslant r \leqslant 2k-1$.

Notice that Equation (5) can be rewritten in terms of Chebyshev polynomials of the second kind. Namely, since

$$\Delta_{1,\ldots,2k} = i^k U_k\left(\frac{1}{2}\right) U_k\left(-\frac{i}{2}\right) - i^{k-1} U_{k-1}\left(\frac{1}{2}\right) U_{k-1}\left(-\frac{i}{2}\right),$$

one can conclude that

$$x = \frac{1}{2}\left(U_k\left(\frac{1}{2}\right) U_k\left(-\frac{i}{2}\right) + U_{k-2}\left(\frac{1}{2}\right) U_{k-2}\left(-\frac{i}{2}\right)\right).$$

3. The cases $k = 2$ and $k = 3$

As a first example, if we set $k = 2$, considering the table

r	0	1	2	3
Δ_r	1	1	0	-1
$\tilde{\Delta}_r$	0	1	1	0
ϵ_r	1	1	1	-1

we obtain

$$\det T_{n,k} = (-1)^q(\Delta_r U_q(1/2) + \epsilon_r \tilde{\Delta}_r U_{q-1}(1/2)),$$

where $n = 4q + r$, with $0 \leqslant r \leqslant 3$.

The term $x_{4q+r,2}$ is given by

	$r = 0$	$r = 1$	$r = 2$	$r = 3$
$x_{4q+r,2}$	$U_q(-1/2)$	$U_q(-1/2) - U_{q-1}(-1/2)$	$-U_{q-1}(-1/2)$	$-U_q(-1/2)$

Recall that

$$U_\ell\left(-\frac{1}{2}\right) = \begin{cases} 1 & \text{if } \ell \equiv 0 \pmod 3 \\ -1 & \text{if } \ell \equiv 1 \pmod 3 \\ 0 & \text{if } \ell \equiv 2 \pmod 3 \end{cases}.$$

Another simple example is the case when $k = 3$. Let us consider the following table:

r	0	1	2	3	4	5
Δ_r	1	1	0	-1	-1	-2
$\tilde{\Delta}_r$	-1	1	2	1	1	0
ϵ_r	$-i$	$-i$	$-i$	$-i$	i	$-i$

We obtain

$$\det T_{n,k} = (-i)^q (\Delta_r U_q(-2i) + \epsilon_r \tilde{\Delta}_r U_{q-1}(-2i)),$$

where $n = 6q + r$, with $0 \leqslant r \leqslant 5$.

Since $F_{3(n+1)} = 2i^n U_n(-2i)$ (cf. (Zhang, W., 2002)), the term $x_{6q+r,3}$ is given by

	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
$(-1)^q x_{6q+r,3}$	F_{3q+1}	F_{3q+2}	F_{3q}	$-F_{3q+1}$	$-F_{3q+2}$	$-F_{3q+3}$

4. Some general examples

In this section we obtain some other particular solutions for Equation (1). Throughout the rest of the note, we will use the Iverson bracket for a given statement S define as

$$[S] = \begin{cases} 1 & \text{if } S \text{ is true} \\ 0 & \text{otherwise} \end{cases}.$$

Our first result is straightforward.

Theorem 4.1. *For $n - k \leqslant 1$ and $n \geqslant 3$, the solution of Equation* (1) *is:*

$$x_{n,k} = \begin{cases} (-1)^m & \text{if } n \not\equiv 2 \pmod 3 \\ 0 & otherwise \end{cases},$$

where $n = 3m + t$ and $0 \leqslant t \leqslant 2$.

Proof. We will use induction on m. For $n = 3$, $t = 0$ and $m = 1$, we have

$$x_{3,k} = x_{2,k} - (-1)^{\left\lfloor \frac{1}{k} \right\rfloor} x_{1,k} = -1 = (-1)^1 = (-1)^m.$$

Assume that our assertion holds for $n = 3m + t$. In this case, using the inductive hypothesis, we have

$$\begin{aligned} x_{3m,k} &= (-1)^m, \quad \text{if } t = 0, \\ x_{3m+1,k} &= (-1)^m, \quad \text{if } t = 1, \\ x_{3m+2,k} &= 0, \quad \text{if } t = 2. \end{aligned}$$

Then, for $n = 3(m+1) + t$, we have

$$\begin{aligned} x_{3m+3,k} &= x_{3m+2,k} - (-1)^{\left\lfloor \frac{3m+1}{k} \right\rfloor} x_{3m+1,k} = (-1)^{m+1}, \quad \text{if } t = 0, \\ x_{3m+4,k} &= x_{3m+3,k} - (-1)^{\left\lfloor \frac{3m+2}{k} \right\rfloor} x_{3m+2,k} = (-1)^{m+1}, \quad \text{if } t = 1, \\ x_{3m+5,k} &= x_{3m+4,k} - (-1)^{\left\lfloor \frac{3m+3}{k} \right\rfloor} x_{3m+3,k} = 0, \quad \text{if } t = 2. \end{aligned}$$

The proof is now completed. □

The following theorem provides general solutions for (1) when $n - k > 1$ and $k \equiv 3 \pmod 6$.

Theorem 4.2. *Let $k \geqslant 1$ be an integer such that $k \equiv 3 \pmod 6$. Then, for all $n \geqslant 3$, the solution of Equation* (1) *has these forms:*

1. *For $n \equiv 0 \pmod 6$ and $n \equiv a \pmod k$,*

$$x_{n,k} = \begin{cases} (-1)^{\frac{n(n+1)}{2}+[a\equiv 6 \pmod{12}]} F_{\frac{n-a}{2}+1} & \text{if } a \equiv 0 \pmod 6 \\ (-1)^{\left(\begin{array}{c}\frac{n(n+1)}{2}+[k\equiv 3 \pmod{12} \text{ and } a\equiv 9 \pmod{12}] \\ +[k\equiv 9 \pmod{12} \text{ and } a\equiv 3 \pmod{12}]\end{array}\right)} F_{\frac{n-k+a}{2}+1} & \text{if } a \equiv 3 \pmod 6 \end{cases}.$$

2. *For $n \equiv 1 \pmod 6$ and $n \equiv a \pmod k$,*

$$x_{n,k} = \begin{cases} (-1)^{\frac{n(n+1)}{2}+1-[a\equiv 7 \pmod{12}]} F_{\frac{n-a}{2}+2} \\ \text{if } a \equiv 1 \pmod 6 \\ (-1)^{\left(\begin{array}{c}\frac{n(n+1)}{2}+1-[k\equiv 3 \pmod{12} \text{ and } a\equiv 10 \pmod{12}] \\ -[k\equiv 9 \pmod{12} \text{ and } a\equiv 4 \pmod{12}]\end{array}\right)} F_{\frac{n-k+a}{2}+1} \\ \text{if } a \equiv 4 \pmod 6 \end{cases}.$$

3. *For $n \equiv 2 \pmod 6$ and $n \equiv a \pmod k$,*

$$x_{n,k} = \begin{cases} (-1)^{\frac{n(n+1)}{2}+1-[a\equiv 8 \pmod{12}]} F_{\frac{n-a}{2}} \\ \text{if } a \equiv 2 \pmod 6 \\ (-1)^{\left(\begin{array}{c}\frac{n(n+1)}{2}+1-[k\equiv 3 \pmod{12} \text{ and } a\equiv 11 \pmod{12}] \\ -[k\equiv 9 \pmod{12} \text{ and } a\equiv 5 \pmod{12}]\end{array}\right)} F_{\frac{n-k+a}{2}+1} \\ \text{if } a \equiv 5 \pmod 6 \end{cases}.$$

4. *For $n \equiv 3 \pmod 6$ and $n \equiv a \pmod k$,*

$$x_{n,k} = \begin{cases} (-1)^{\left(\begin{array}{c}\frac{n(n+1)}{2}+[k\equiv 3 \pmod{12} \text{ and } a\equiv 0 \pmod{12}] \\ +[k\equiv 9 \pmod{12} \text{ and } a\equiv 6 \pmod{12}]\end{array}\right)} F_{\frac{n-k+a}{2}+1} & \text{if } a \equiv 0 \pmod 6 \\ (-1)^{\frac{n(n+1)}{2}+[a\equiv 3 \pmod{12}]} F_{\frac{n-a}{2}+1} & \text{if } a \equiv 3 \pmod 6 \end{cases}.$$

5. *For $n \equiv 4 \pmod 6$ and $n \equiv a \pmod k$,*

$$x_{n,k} = \begin{cases} (-1)^{\left(\begin{array}{c}\frac{n(n+1)}{2}+1-[k\equiv 3 \pmod{12} \text{ and } a\equiv 7 \pmod{12}] \\ -[k\equiv 9 \pmod{12} \text{ and } a\equiv 1 \pmod{12}]\end{array}\right)} F_{\frac{n-k+a}{2}+1} \\ \text{if } a \equiv 1 \pmod 6 \\ (-1)^{\frac{n(n+1)}{2}+1-[a\equiv 10 \pmod{12}]} F_{\frac{n-a}{2}+2} \\ \text{if } a \equiv 4 \pmod 6 \end{cases}.$$

6. For $n \equiv 5 \pmod 6$ *and* $n \equiv a \pmod k$*,*

$$x_{n,k} = \begin{cases} (-1)^{\left(\begin{array}{l} \frac{n(n+1)}{2} + [k \equiv 3 \pmod{12} \text{ and } a \equiv 8 \pmod{12}] \\ \quad + [k \equiv 9 \pmod{12} \text{ and } a \equiv 2 \pmod{12}] \end{array}\right)} F_{\frac{n-k+a}{2}+1} & \text{if } a \equiv 2 \pmod 6 \\ (-1)^{\frac{n(n+1)}{2}+[a \equiv 11 \pmod{12}]} F_{\frac{n-a}{2}} & \text{if } a \equiv 5 \pmod 6 \end{cases}.$$

Proof. Although there are several cases to be considered, we will prove here only one of them to avoid unnecessary repetitions. We shall only prove item (1). Assume that $n \equiv 0 \pmod 6$ and $n \equiv a \pmod k$. The proof will be done by strong induction on n. Let us consider the case $n = 24$, $k = 9$, namely: $n \equiv 0 \pmod 6$, $a \equiv 0 \pmod 6$ and $a \equiv 6 \pmod{12}$. Taking into account that

$$n - 1 \equiv 5 \pmod 6, \quad a - 1 \equiv 5 \pmod 6,$$

and

$$n - 2 \equiv 4 \pmod 6, \quad a - 2 \equiv 4 \pmod 6,$$

we obtain

$$x_{24,9} = (-1)^{\frac{24.25}{2}+1} F_{\frac{24-6}{2}+1} = -F_{10},$$

$$x_{23,9} = (-1)^{\frac{23.24}{2}} F_{\frac{23-5}{2}} = F_9,$$

$$x_{22,9} = (-1)^{\frac{22.23}{2}+1} F_{\frac{22-4}{2}+2} = F_{11}.$$

Thus

$$x_{24,9} = x_{23,9} - (-1)^{\left\lfloor \frac{22}{9} \right\rfloor} x_{22,9} = F_9 - F_{11} = -F_{10}.$$

Assume that our assertion holds for $n = t$. In this case, using the inductive hypothesis, we have

$$x_{t,k} = (-1)^{\frac{t(t+1)}{2}+1} F_{\frac{t-a}{2}+1},$$

$$x_{t-1,k} = (-1)^{\frac{t(t-1)}{2}} F_{\frac{t-a}{2}},$$

and

$$x_{t-2,k} = (-1)^{\frac{(t-2)(t-1)}{2}+1} F_{\frac{t-a}{2}+2}.$$

Then, for $n = t + 1$, we have

$$\begin{aligned} x_{t,k} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} x_{t-1,k} &= (-1)^{\frac{t(t+1)}{2}+1} F_{\frac{t-a}{2}+1} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} (-1)^{\frac{t(t-1)}{2}} F_{\frac{t-a}{2}} \\ &= (-1)^{\frac{(t+1)(t+2)}{2}} \left((-1)^{-t} F_{\frac{t-a}{2}+1} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} (-1)^{-2t-1} F_{\frac{t-a}{2}} \right) \\ &= (-1)^{\frac{(t+1)(t+2)}{2}} \left(F_{\frac{t-a}{2}+1} + (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} F_{\frac{t-a}{2}} \right) \\ &= (-1)^{\frac{(t+1)(t+2)}{2}} F_{\frac{t-a}{2}+2} \\ &= x_{t+1,k}, \end{aligned}$$

where

$$\begin{gathered} t + 1 \equiv 1 \pmod 6, \quad a + 1 \equiv 1 \pmod 6, \quad a + 1 \equiv 7 \pmod{12}, \\ t \equiv 0 \pmod 6, \quad a \equiv 0 \pmod 6, \quad a \equiv 6 \pmod{12}, \\ t - 1 \equiv 5 \pmod 6, \quad a - 1 \equiv 5 \pmod 6, \quad a - 1 \equiv 5 \pmod{12}, \end{gathered}$$

and $\left\lfloor \frac{t-1}{k} \right\rfloor$ is even. So, the first part of item (1) is proved. We now prove the second part. For this, we will again use the induction on n. Let us consider the case $n = 48$, $k = 27$, namely: $n \equiv 0 \pmod 6$, $a \equiv 3 \pmod 6$, $a \equiv 9 \pmod{12}$. Using

$$n-1 \equiv 5 \pmod 6, \quad a-1 \equiv 2 \pmod 6, \quad a-1 \equiv 8 \pmod{12},$$

and

$$n-2 \equiv 4 \pmod 6, \quad a-2 \equiv 1 \pmod 6, \quad a-2 \equiv 7 \pmod{12},$$

we have

$$x_{48,27} = (-1)^{\frac{48.49}{2}+1} F_{\frac{48-27+21}{2}+1} = -F_{22},$$
$$x_{47,27} = (-1)^{\frac{47.48}{2}+1} F_{\frac{47-27+20}{2}+1} = -F_{21},$$
$$x_{46,27} = (-1)^{\frac{46.47}{2}} F_{\frac{46-27+19}{2}+1} = -F_{20}.$$

So, we get

$$x_{47,27} - (-1)^{\left\lfloor \frac{46}{27} \right\rfloor} x_{46,27} = -F_{21} - F_{20} = -F_{22} = x_{48,27}.$$

Assume that our claim holds for $n = t$. In this case, using the induction hypothesis, we obtain

$$x_{t,k} = (-1)^{\left(\frac{t(t+1)}{2}+1\right)} F_{\frac{t-k+a}{2}+1},$$
$$x_{t-1,k} = (-1)^{\left(\frac{(t-1)t}{2}+1\right)} F_{\frac{t-k+a}{2}},$$
$$x_{t-2,k} = (-1)^{\left(\frac{(t-2)(t-1)}{2}\right)} F_{\frac{t-k+a}{2}-1}.$$

Then, for $n = t+1$, we get

$$\begin{aligned}
x_{t,k} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} x_{t-1,k} &= (-1)^{\left(\frac{t(t+1)}{2}+1\right)} F_{\frac{t-k+a}{2}+1} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} (-1)^{\left(\frac{(t-1)t}{2}+1\right)} F_{\frac{t-k+a}{2}} \\
&= (-1)^{\frac{(t+1)(t+2)}{2}} \left((-1)^{-t} F_{\frac{t-k+a}{2}+1} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} (-1)^{-2t} F_{\frac{t-k+a}{2}} \right) \\
&= (-1)^{\frac{(t+1)(t+2)}{2}} \left(F_{\frac{t-k+a}{2}+1} - (-1)^{\left\lfloor \frac{t-1}{k} \right\rfloor} F_{\frac{t-k+a}{2}} \right) \\
&= (-1)^{\frac{(t+1)(t+2)}{2}} \left(F_{\frac{t-k+a}{2}+1} + F_{\frac{t-k+a}{2}} \right) \\
&= (-1)^{\frac{(t+1)(t+2)}{2}} F_{\frac{t-k+a}{2}+2} \\
&= x_{t+1,k},
\end{aligned}$$

where

$$t+1 \equiv 1 \pmod 6, \quad a+1 \equiv 4 \pmod 6, \quad a+1 \equiv 10 \pmod{12},$$
$$t \equiv 0 \pmod 6, \quad a \equiv 3 \pmod 6, \quad a \equiv 9 \pmod{12},$$
$$t-1 \equiv 5 \pmod 6, \quad a-1 \equiv 2 \pmod 6, \quad a-1 \equiv 8 \pmod{12},$$

and $\left\lfloor \frac{t-1}{k} \right\rfloor$ is odd. Therefore, our proof is completed. □

Remark 4.1. In Section 3, we obtained the solutions of Equation (1) for $n = 6q + r$ and $k = 3$ based on the determinant of the matrix $T_{n,k}$. Now, if we take $n = 6q + r$ and $k = 3$ in Theorem 4.2, we can obtain the solutions of Equation (1) in a different method. Namely, if q is even, we have

Also, if q is odd, we get

Remark 4.2. For $k \not\equiv 0 \pmod 3$, the solutions of Equation (1) do not necessarily belong to the Fibonacci sequence. For example

$$x_{34,7} = 169,$$

or

$$x_{99,32} = 3010349,$$

are not Fibonacci numbers.

	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
$x_{6q+r,3}$	F_{3q+1}	F_{3q+2}	F_{3q}	$-F_{3q+1}$	$-F_{3q+2}$	$-F_{3q+3}$

	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
$-x_{6q+r,3}$	F_{3q+1}	F_{3q+2}	F_{3q}	$-F_{3q+1}$	$-F_{3q+2}$	$-F_{3q+3}$

References

Andelić, M., Du, Z., da Fonseca, C.M., Kılıç, E. (2020). Second-order difference equations with sign-alternating coefficients. Journal of Difference Equations and Applications, 26, 149-162.

Andelić, M., da Fonseca, C.M., Mamede, R. (2011). On the number of P-vertices of some graphs. Linear Algebra and its Applications, 434, 514-525.

Du, Z., Dimitrov, D., da Fonseca, C.M. (2022). New strong divisibility sequences. Ars Mathematica Contemporanea, 22, #P1.08.

da Fonseca, C.M. (2014). Unifying some Pell and Fibonacci identities. Applied Mathematics and Computation, 236, 41-42.

da Fonseca, C.M., Petronilho, J. (2005). Explicit inverse of a tridiagonal k-Toeplitz matrix. Numerische Mathematik, 100(3), 457-482.

Kızılateş, C. (2021). New families of Horadam numbers associated with finite operators and their applications. Mathematical Methods in the Applied Sciences, 44(18), 14371-14381.

Koshy, T. (2018). Fibonacci and Lucas Numbers with Applications. Vol. 1, Second edition, Pure and Applied Mathematics (Hoboken), John Wiley & Sons, Inc., Hoboken, NJ.

Rózsa, P. (1969). On periodic continuants. Linear Algebra and its Applications, 2, 267-274.

Trojovský, P. (2017). On a difference equation of the second order with an exponential coefficient. Journal of Difference Equations and Applications, 23, 1737-1746.

Zhang, W. (2002). On Chebyshev polynomials and Fibonacci numbers. Fibonacci Quarterly, 40, 424-428.

Submitted: 06/05/2022
Revised: 19/07/2022
Accepted: 19/07/2022
DOI: 10.48129/kjs.20425

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-15)

An optimal fourth-order second derivative free iterative method for nonlinear scientific equations

Ghulam Akbar Nadeem[1], Waqas Aslam[2], Faisal Ali[1,*]

[1]*Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan, Pakistan,*
[2]*Government College for Elementary Teachers, Rangeelpur Multan, Pakistan*
[*]*Corresponding Author: faisalali@bzu.edu.pk*

Abstract

In the present paper, we develop an efficient second derivative free two-step optimal fourth-order iterative method for nonlinear equations. We explore the convergence criteria of the proposed method and also exhibit its validity and efficiency by considering some test problems. We present both numerical as well as graphical comparisons. Further, the dynamical behavior of the proposed method is explored.

Keywords: Approximation scheme; iterative methods; nonlinear equations; order of convergence; polynomiography.

1. Introduction

In everyday life, most of the phenomena lead to some kind of nonlinear equations. In general, these nonlinear equations cannot be dealt analytically or for exact solutions. So, naturally, the scientists focused on numerical methods for solving such type of equations, (Dogan, 2013; Uddin & Imdad, 2015; Bayat *et al.*, 2015; Karakaya *et al.*, 2016; Demiray & Bulut, 2017; Al-jawary & Nabi, 2020; Ozer, 2021; Eze, 2022). Particularly, finding the numerical solution of the nonlinear equation

$$f(x) = 0, \tag{1}$$

has been a sweltering problem in the fields of science and engineering, e.g.(Babolian & Baizar, 2002; Abbasbandy, 2003; Bhalekar & Daftardar-Gejji, 2011; Chun, 2018; He, 2016; He *et al.*, 2020; He *et al.*, 2021; He & El-Dib, 2020) and references therein. In the construction and performance of a numerical method for nonlinear equations, its convergence order and the number of evaluations per iteration are significant.

Definition 1. Let $\{x_n\}$ be the sequence of approximations that converges to the root α of $f(x) = 0$ i.e. $\lim_{n\to\infty} x_n = \alpha$. If there exist positive real numbers p and k such that $\lim_{n\to\infty} \frac{|x_{n+1}-\alpha|}{|x_n-\alpha|^p} = k$, then p is called the order of convergence of the method.

Definition 2. If p is the convergence order of an iterative method and m denotes the number of function evaluations per iteration, then the efficiency index (E.I) of the method is $p^{\frac{1}{m}}$.

One of the most significant and well-known techniques, for solving nonlinear equations, is the Newton's method which converges quadratically, Ostrowski, (1973):

$$x_{n+1} = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right), \quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \ldots . \tag{2}$$

Noor & Gupta, (2007) modified Householder iterative method and developed the following fourth-order method which requires four evaluations per iteration i.e. its EI = 1.4141.

$$x_{n+1}=y_n-\left(\frac{f(y_n)}{f'(y_n)}\right)-\left(\frac{1}{2}\right)\left[\left(\frac{f(y_n)}{f'(y_n)}\right)\right]^2\left[\left(\frac{f'(x_n)}{f(x_n)}\right)\right]\left[\left(\frac{f'(x_n)+f'(y_n)}{f'(y_n)}\right)\right], n=0,1,2,\ldots, \quad (3)$$

where, $y_n=x_n-\left(\frac{f(x_n)}{f'(x_n)}\right), \quad f'(x_n)\neq 0, \quad n=0,1,2,\ldots$.

An immense literature is available regarding third-order and fourth-order iterative methods for solving nonlinear equations, (Chun, 2007; Herceg & Herceg, 2008; Saeed & Khthr, 2011; Thukral, 2013; Singh & Jaiswal, 2014; Jaiswal, 2014; Ali *et al.*, 2018; Huang *et al.*, 2018; Naseem *et al.*, 2020; Sana *et al.* 2021) and references therein.

Keeping in view the importance of convergence order and the number of evaluations per iteration required in an iterative method, Kung & Turab, (1974) gave a conjecture. According to this conjecture, an iterative method is said to be an optimal one if it needs $(n+1)$ evaluations per iteration and posses convergence order 2^n. Some useful optimal fourth-order iterative methods have been constructed by various researchers (Sharma *et al.*, 2020; Ali *et al.*, 2020; Shams *et al.*, 2020; Cordero *et al.*, 2021; Hafiz & Khirallah, 2021).

Cordero *et al.*, (2010) introduced the following optimal fourth-order method:

$$x_{n+1}=x_n-\left(\frac{f(x_n)+f(y_n)}{f'(x_n)}\right)-\left[\frac{f(y_n)}{f'(x_n)}\right]^2\left[\frac{2f(x_n)+f(y_n)}{f'(x_n)}\right], \quad (4)$$

where, $y_n=x_n-\left(\frac{f(x_n)}{f'(x_n)}\right), \quad f'(x_n)\neq 0, \quad n=0,1,2,\ldots$.

Obviously, the above method is of order $2^2=4$ and requires $(2+1)=3$ evaluations per iteration, so it is an optimal fourth-order iterative method with EI = 1.5874 .

Sherma & Bahl, (2015) also developed an optimal fourth-order method:

$$x_{n+1}=x_n-\left[-\left(\frac{1}{2}\right)+\left(\frac{9f'(x_n)}{8f'(y_n)}\right)+\left(\frac{3f'(y_n)}{8f'(x_n)}\right)\right]\left(\frac{f(x_n)}{f'(x_n)}\right), \quad n=0,1,2,\ldots, \quad (5)$$

where, $y_n=x_n-\left(\frac{2}{3}\right)\left(\frac{f(x_n)}{f'(x_n)}\right), \quad f'(x_n)\neq 0, \quad n=0,1,2,\ldots$.

A second derivative free optimal fourth-order method was introduced by Shengfeng Li, (2019):

$$x_{n+1}=x_n-\left(\frac{[f(x_n)-f(y_n)]f(x_n)}{[f(x_n)-2f(y_n)]f'(x_n)}\right), \quad n=0,1,2,\ldots, \quad (6)$$

where, $y_n=x_n-\left(\frac{f(x_n)}{f'(x_n)}\right), \quad f'(x_n)\neq 0, \quad n=0,1,2,\ldots$.

In this paper, being inspired from the literature regarding optimal iterative methods for nonlinear equations, we present a rapidly convergent and efficient optimal fourth-order iterative method. In order to demonstrate the validity and effectiveness of the proposed method, we explore the numerical as well as graphical comparisons. We also consider some complex polynomials to study the dynamics of the suggested method. We present four polynomiographs against four different polynomials. These polynomiographs clearly exhibit the corresponding roots along with the region of convergence according to the chosen initial guess.

2. Construction of Iterative Method

Noor *et al.*, (2007) presented the following sixth-order iterative method which involves the second derivative of the function and needs five evaluations per iteration, i.e. its EI = 1.4309.

$$x_{n+1}=y_n-\frac{f(y_n)}{f'(y_n)}-\left(\frac{(f(y_n))^2 f''(y_n)}{2\left(f'(y_n)\right)^3}\right), \tag{7}$$

where, $y_n=x_n-\left(\frac{f(x_n)}{f'(x_n)}\right)$, $\quad f'(x_n)\neq 0, \quad n=0,1,2,\ldots$.

We consider the following interpolation scheme, Rehman *et al.*, (2021):

$$H(t)=a+b(t-y_k)+c(t-y_k)^2+d(t-y_k)^3, \tag{8}$$

where a, b, c and d are unknowns and can be determined by applying the following conditions:

$$H(x_k)=f(x_k),\ H(y_k)=f(y_k),\ H'(x_k)=f'(x_k),\ H'(y_k)=f'(y_k),\ H''(y_k)=f''(y_k). \tag{9}$$

Using the above conditions, we obtain the following system of equations:

$$f(y_k)=a, \tag{10}$$

$$f(x_k)=a+b(x_k-y_k)+c(x_k-y_k)^2+d(x_k-y_k)^3, \tag{11}$$

$$f'(x_k)=b+2c(x_k-y_k)+3d(x_k-y_k)^2, \tag{12}$$

$$f'(y_k)=b, \tag{13}$$

$$f''(y_k)=2c+6d(x_k-y_k). \tag{14}$$

Solving equations (10)-(14), simultaneously, we obtain

$$f''(y_k)=\frac{6[f(x_k)-f(y_k)]-2(x_k-y_k)[2f'(x_k)+f'(y_k)]}{(x_k-y_k)^2}=P(x_k,\ y_k). \tag{15}$$

Let us now consider the following interpolation scheme:

$$G(t)=a+b(t-y_k)+c(t-y_k)^2, \tag{16}$$

where a, b and c are the unknowns, which can be determined by applying the following conditions:

$$G(x_k)=f(x_k),\ G(y_k)=f(y_k),\ G'(x_k)=f'(x_k),\ G'(y_k)=f'(y_k). \tag{17}$$

Using the above conditions, we obtain the following system of equations:

$$f(y_k) = a, \tag{18}$$

$$f(x_k) = a + b(x_k - y_k) + c(x_k - y_k)^2, \tag{19}$$

$$f'(x_k) = b + 2c(x_k - y_k), \tag{20}$$

$$f'(y_k) = b. \tag{21}$$

Solving equations (18)-(21), simultaneously, we get

$$f'(y_k) = \frac{2[f(x_k) - f(y_k)]}{(x_k - y_k)} - f'(x_k) = Q(x_k\ ,\ y_k). \tag{22}$$

From equations (15) and (22), we get

$$f''(y_k) = \frac{6[f(x_k) - f(y_k)] - 2(x_k - y_k)[2f'(x_k) + Q(x_k\ ,\ y_k)]}{(x_k - y_k)^2} = R(x_k\ ,\ y_k). \tag{23}$$

Thus, using equations (7), (22) and (23), we are in a position to formulate the following optimal fourth-order second derivative free iterative method for solving nonlinear equation (1).

2.1 Algorithm For a given x_0, compute the approximate solution x_{n+1} by the following iterative scheme:

$$x_{n+1} = y_n - \frac{f(y_n)}{Q(x_n\ ,\ y_n)} - \left(\frac{f^2(y_n) R(x_n\ ,\ y_n)}{2Q^3(x_n\ ,\ y_n)}\right), \tag{24}$$

where, $y_n = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right)$, $\quad f'(x_n) \neq 0, \quad n = 0, 1, 2, \ldots$.

3. Convergence Analysis

The convergence criteria for the newly proposed iterative method i.e. algorithm 2.1 is described in the following theorem.

3.1 Theorem

Assume that the function $f: I \subset \mathbb{R} \to \mathbb{R}$ (where I is an open interval) has a simple root $\alpha \in I$. If $f(x)$ is a sufficiently differentiable function in the neighborhood of α, then the method given in algorithm 2.1 has the convergence order at least 4.

Proof Since $f(x)$ is sufficiently differentiable, therefore, the Taylor's series expansions of $f(x_n)$ and $f'(x_n)$ about α are given by:

$$f(x_n) = f'(\alpha)\left\{e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_2 e_n^5 + O\left(e_n^6\right)\right\}, \tag{25}$$

and

$$f'(x_n) = f'(\alpha)\left\{1+2c_2e_n+3c_3e_n^2+4c_4e_n^3+5c_5e_n^4+6c_6e_n^5+O\left(e_n^6\right)\right\}, \tag{26}$$

where, $e_n{=}x_n{-}\alpha$ *and* $c_j{=}\left(\frac{1}{j!}\right)\left(\frac{f^{(j)}(\alpha)}{f'(\alpha)}\right), \qquad j{=}\,2,3,\ldots.$

From equations (25) and (26), we get

$$\frac{f(x_n)}{f'(x_n)}{=}e_n{-}c_2e_n^2{+}2\left(c_2^2{-}c_3\right)e_n^3{+}\left(-4c_2^3{+}7c_2c_3{-}3c_4\right)e_n^4{+}(8c_2^4{-}20c_2^2c_3{+}10c_2c_4{+}6c_3^2{-}4c_5)e_n^5{+}O\left(e_n^6\right). \tag{27}$$

Using equation (27), we get

$$\begin{aligned} y_n &= x_n-\frac{f(x_n)}{f'(x_n)}{=}\alpha{+}c_2c_n^2{-}2\left(c_2^2{-}c_3\right)e_n^3+\left(4c_2^3{-}7c_2c_3{+}3c_4\right)e_n^4 \\ &\quad +(-8c_2^4{+}20c_2^2c_3{-}10c_2c_4-6c_3^2{+}4c_5)e_n^5+O\left(e_n^6\right). \end{aligned} \tag{28}$$

Using equation (28), the Taylor's series of $f(y_n)$ is given by

$$f(y_n) {=}c_2e_n^2{-}2\left(c_2^2{-}c_3\right)e_n^3{+}\left(5c_2^3{-}7c_2c_3{+}3c_4\right)e_n^4{+}(-12c_2^4{+}24c_2^2c_3{-}10c_2c_4{-}6c_3^2{+}4c_5)e_n^5{+}O\left(e_n^6\right). \tag{29}$$

Using equations (25), (26), (28) and (29), we get

$$Q(x_n\ ,\ y_n){=}\,1{+}\left(2c_2^2{-}c_3\right)e_n^2{+}\left(-4c_2^3{+}6c_2c_3{-}2c_4\right)e_n^3{+}(8c_2^4{-}16e_2^2c_3{+}8c_2c_4{+}4c_3^2{-}3c_5)e_n^4{+}O\left(e_n^5\right). \tag{30}$$

Using equations (25), (26), (28) and (29), we get

$$R(x_n\ ,\ y_n){=}2c_2{+}4c_3e_{\rm n}{+}\left(2c_2c_3{+}6c_4\right)e_n^2{+}\left(-4c_2^2c_3{+}4c_2c_4{+}4c_3^2{+}8c_5\right)e_n^3{+}O\left(e_n^4\right). \tag{31}$$

Thus, using equations (28)-(31), the error term for algorithm 2.1 becomes:

$$e_{n+1}{=}\,-c_2c_3e_n^4{+}\left(2c_2^2c_3{-}2c_2c_4{-}2c_3^2\right)e_n^5{+}O\left(e_n^6\right). \tag{32}$$

This completes the proof.

4. Numerical Examples

In order to exhibit the validity and effectiveness of the newly proposed optimal fourth-order iterative method given in algorithm 2.1 (AM), we compare the same, numerically as well as graphically, with the standard Newton's method (NM), Sharma & Bahl, (2015) method (equation 5) (RM), Noor & Gupta, (2007) method (equation 3) (HM), Cardero *et al.*, (2010) method (equation 4) (CM), recently developed method by Shengfeng Li, (2019) (equation 6) (LM) and Noor *et al.*, (2000) method (equation 7) (NR) in the context of standard nonlinear equations. The numerical comparison, by using the software Maple-18, is presented in the following table, in which NFE column represents the number of function evaluations, whereas figure 1 to figure 8 exhibit the graphical comparison, performed with the help of Matlab software. In order to stop the iteration process, we use the condition $|x_{n+1}{-}x_n|{<}\varepsilon$ with $\varepsilon = 10^{-15}$. Both comparative studies clearly indicate that the newly developed method performs more efficiently.

Table 1: Numerical Examples

$f(x)$	x_0	*Method*	n	x_k	$\lvert f(x_n)\rvert$	$\lvert x_{n+1}-x_n\rvert$	**NFE**
x^3-x-8	0.5	NM	21	2.1663127473977890	$2.906126e^{-16}$	$1.213442e^{-10}$	42
		RM	7	2.1663127473977890	$2.906126e^{-16}$	$3.999760e^{-10}$	28
		HM	14	2.1663127473977890	$2.906126e^{-16}$	$1.721625e^{-7}$	56
		LM	14	2.1663127473977890	$2.906126e^{-16}$	$1.604659e^{-8}$	42
		CM	50	-22713.26588077619	$2.906126e^{-16}$	$2.264950e^{4}$	150
		NR	8	2.1663127473977890	$2.906126e^{-16}$	$8.622847e^{-9}$	40
		AM	6	2.1663127473977890	$2.906126e^{-16}$	$2.998342e^{-10}$	18
$x^3-e^{(sinx)}-1.3$	0.43	NM	13	1.5897513629099752	$1.028154e^{-16}$	$1.597951e^{-9}$	26
		RM	7	1.5897513629099752	$1.028154e^{-16}$	$5.554208e^{-12}$	21
		HM	6	1.5897513629099752	$1.028154e^{-16}$	$2.063160e^{-10}$	24
		LM	21	1.5897513629099752	$1.028154e^{-16}$	$3.616346e^{-6}$	23
		CM	41	1.5897513629099752	$2.187217e^{-15}$	$1.130202e^{-4}$	123
		NR	6	1.5897513629099752	$1.028154e^{-16}$	$1.769922e^{-14}$	30
		AM	5	1.5897513629099752	$1.028154e^{-16}$	$1.728701e^{-12}$	15
$x^3+sinx-0.5$	-.70	NM	11	0.4324702259081946	$1.486126e^{-18}$	$1.290499e^{-14}$	22
		RM	20	0.4324702259081946	$1.486126e^{-18}$	$1.379866e^{-12}$	60
		HM	5	0.4324702259081946	$1.486126e^{-18}$	$1.284167e^{-8}$	20
		LM	5	0.4324702259081946	$1.486126e^{-18}$	$2.552682e^{-6}$	15
		CM	6	0.4324702259081946	$1.486126e^{-18}$	$1.626361e^{-6}$	18
		NR	5	0.4324702259081946	$1.486126e^{-18}$	$7.327305e^{-15}$	25
		AM	4	0.4324702259081946	$1.486126e^{-18}$	$9.966395e^{-08}$	12
$x^4-2tan^{-1}(x)$-1	-5.5	NM	11	-0.5048496838915417	$1.013082e^{-17}$	$4.613243e^{-11}$	22
		RM	43	-0.5048496838915417	$1.095395e^{-17}$	$5.949597e^{-9}$	129
		HM	5	-0.5048496838915417	$1.013082e^{-17}$	$2.401892e^{-13}$	20
		LM	6	-0.5048496838915417	$1.013082e^{-17}$	$9.637054e^{-15}$	18
		CM	6	-0.5048496838915417	$1.095395e^{-17}$	$5.956829e^{-05}$	18
		NR	5	-0.5048496838915417	$1.013082e^{-17}$	$2.898906e^{-11}$	25
		AM	4	-0.5048496838915417	$1.013082e^{-17}$	$1.850485e^{-11}$	12
$x^5+xsin(x-1)$	-.65	NM	17	0.7230912060028413	$8.893281e^{-18}$	$5.426442e^{-11}$	34
		RM	9	0.7230912060028413	$8.893281e^{-18}$	$1.796728e^{-14}$	36
		HM	7	0.7230912060028413	$8.893281e^{-18}$	$3.322140e^{-07}$	28
		LM	7	0.7230912060028413	$8.893281e^{-18}$	$3.440059e^{-07}$	21
		CM	50	227793617.43987802	$6.133491e^{41}$	$1.050057e^{08}$	150
		NR	7	0.7230912060028413	$8.893281e^{-18}$	$9.934670e^{-6}$	35
		AM	5	0.7230912060028452	$6.950625e^{-15}$	$7.280912e^{-06}$	15

$f(x)$	x_0	*Method*	n	x_k	$\|f(x_n)\|$	$\|x_{n+1}-x_n\|$	**NFE**
x^3+x+e^x+5	9.00	NM	14	-1.5426515636094549	$2.718112e^{-15}$	$2.413960e^{-8}$	28
		RM	25	-1.5426515636094549	$2.121712e^{-16}$	$3.274434e^{-12}$	75
		HM	6	-1.5426515636094549	$2.121712e^{-16}$	$1.070996e^{-6}$	24
		LM	7	-1.5426515636094549	$5.224052e^{-15}$	$2.834313e^{-4}$	21
		CM	8	-1.5426515636094549	$2.121712e^{-16}$	$8.797138e^{-15}$	24
		NR	6	-1.5426515636094549	$2.121712e^{-16}$	$8.510917e^{-9}$	30
		AM	5	-1.5426515636094549	$2.121712e^{-16}$	$2.213423e^{-12}$	15
x^3+4x^2+1	0.7	NM	21	-4.0606470275541425	$2.121712e^{-16}$	$1.345687e^{-08}$	42
		RM	7	-4.0606470275541425	$7.712486e^{-16}$	$1.241081e^{-13}$	28
		HM	12	-4.0606470275541425	$7.712486e^{-16}$	$2.098410e^{-10}$	48
		LM	32	-4.0606470275541425	$7.712486e^{-16}$	$1.607614e^{-12}$	96
		CM	27	-4.0606470275541425	$7.712486e^{-16}$	$6.873173e^{-12}$	81
		NR	12	-4.0606470275541425	$7.712486e^{-16}$	$1.353735e^{-10}$	60
		AM	6	-4.0606470275541425	$7.712486e^{-16}$	$4.036553e^{-15}$	18
$x^{10}-1$	2.50	NM	14	1.0000000000000000	$0.000000e^{0}$	$1.203329e^{-14}$	28
		RM	50	390114007548790110	$8.164234e^{645}$	$1.334660e^{64}$	150
		HM	6	1.0000000000000000	$0.000000e^{0}$	$2.197544e^{-11}$	24
		LM	6	1.0000000000000000	$0.000000e^{0}$	$1.922172e^{-05}$	18
		CM	8	1.0000000000000000	$0.000000e^{0}$	$9.125819e^{-09}$	24
		NR	6	1.0000000000000000	$0.000000e^{0}$	$5.712945e^{-11}$	30
		AM	5	1.0000000000000000	$0.000000e^{0}$	$7.647115e^{-09}$	15

In the above Table 1, we consider 8 examples involving different types of functions i.e. trigonometric functions, inverse trigonometric functions and exponential function. It is notable that, in each case, proposed method converges to the actual root in least number of iterations with minimum number of function evaluations.
The following figures i.e. figure 1 to figure 8 exhibit the plots of number of iterations against log of residuals in the context of each example considered in Table 1. Clearly, the proposed method converges in least number of iterations.

4.1 Graphical Comparison

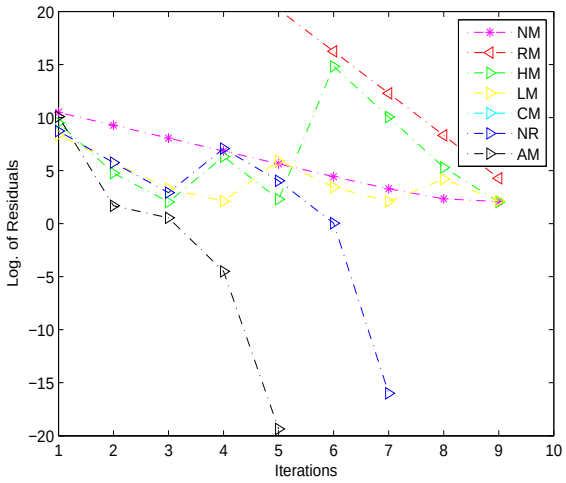


Fig. 1. $(f(x) = x^2 - x - 8)$

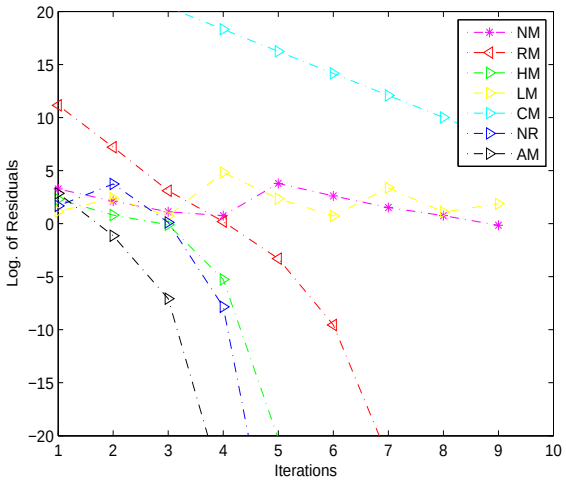


Fig. 2. $(f(x) = x^3 - e^{sinx} - 1.3)$

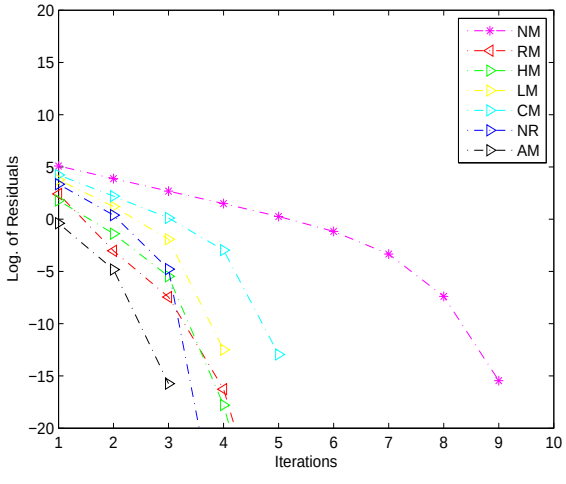


Fig. 3. $(f(x) = x^3 + sinx - 0.5)$

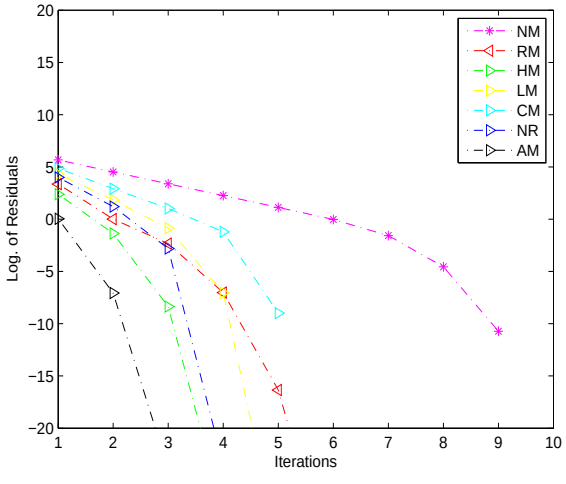


Fig. 4. $(f(x) = x^4 - 2tan^{-1}(x) - 1)$

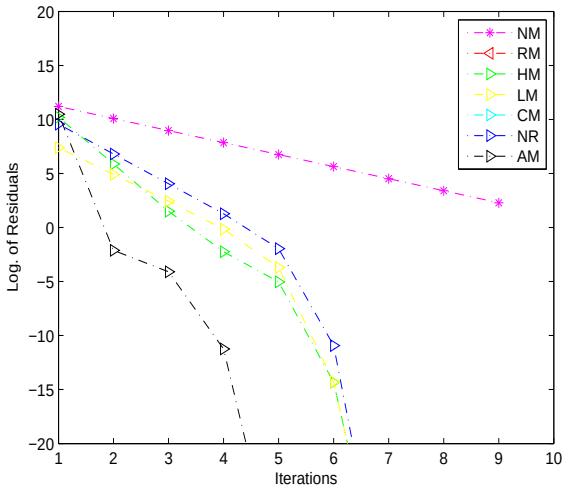


Fig. 5. $(f(x) = x^5 + xsin(x-1))$

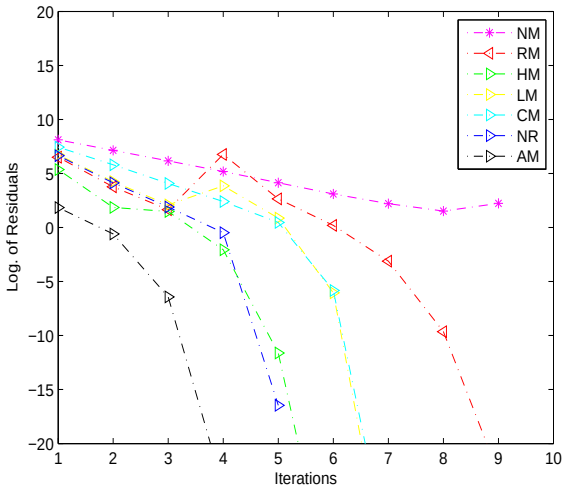


Fig. 6. $(f(x) = x^3 + x + e^x + 5)$

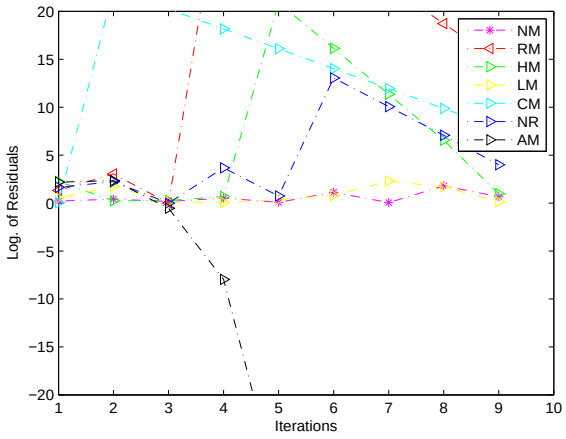


Fig. 7. $(f(x) = x^3 + 4x^2 + 1)$

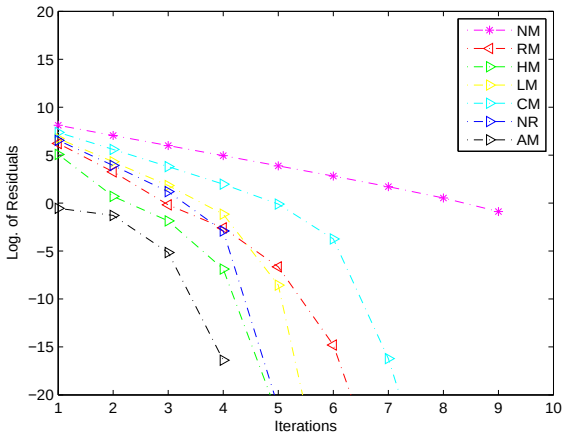


Fig. 8. $(f(x) = x^{10} - 1)$

5. Polynomiography

Kalantary, 2009, introduced the concept of polynomiography, which is an art and science of visualization of the zeros of complex polynomials through fractal and non-fractal images obtained by using the convergence properties of iteration functions. Through polynomiography, nice looking graphics are generated. An individual image is known as polynomiograph. Polynomiography is a modern technique to solve problems with the help of computer technology. It has vast and diverse applications in science, art, design, industry; especially in textile industry. Fundamental theorem of algebra describes that a polynomial of degree $n \geqslant 1$ has n roots. In the study of polynomiography, the degree of a polynomial describes the number of basins of attraction. The colors of polynomiograph indicate the number of iterations required to achieve the approximate root of a certain polynomial with a given accuracy corresponding to a chosen initial guess. For further description and applications of polynomiography, we refer (Kalantary, 2005a; Kalantary,

2005b; Kotarski *et al.*, 2012) The following figures i.e. figure 9 to figure 12 display the polynomiographs and basins of attraction of some standard complex polynomials in the context of newly proposed method. We have used Matlab software for the purpose.

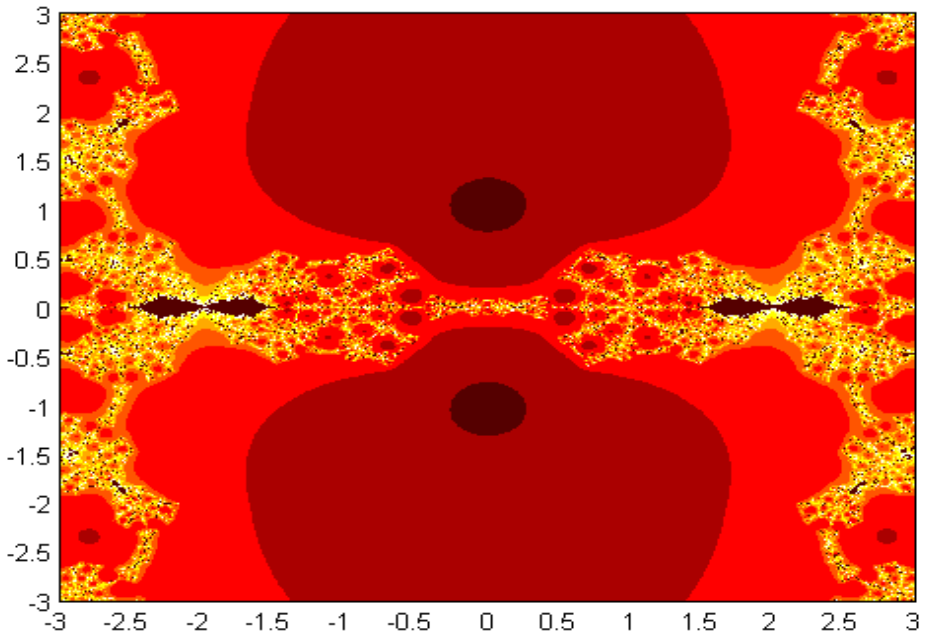


Fig. 9. Polynomiograph of z^2+1

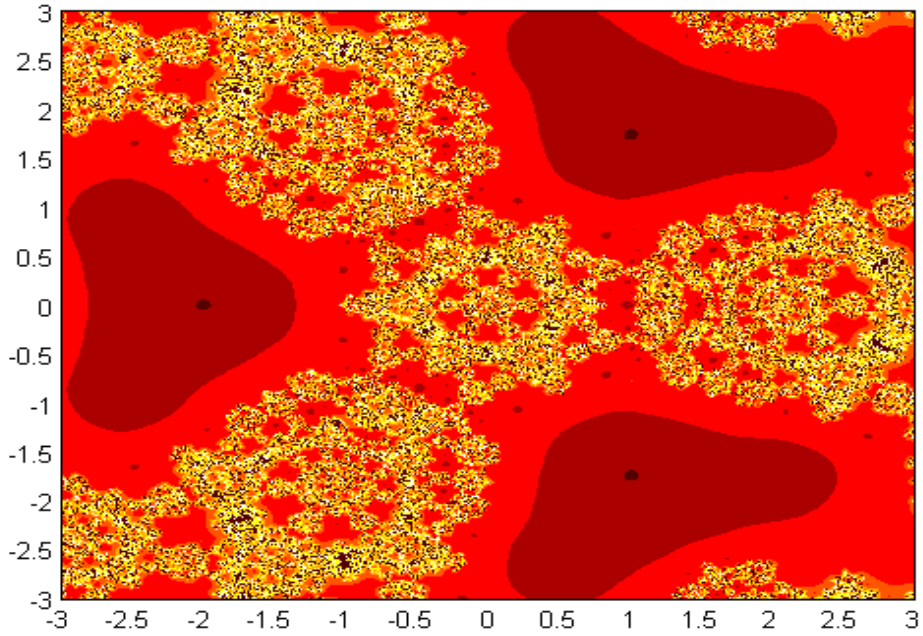


Fig. 10. Polynomiograph of z^3+8

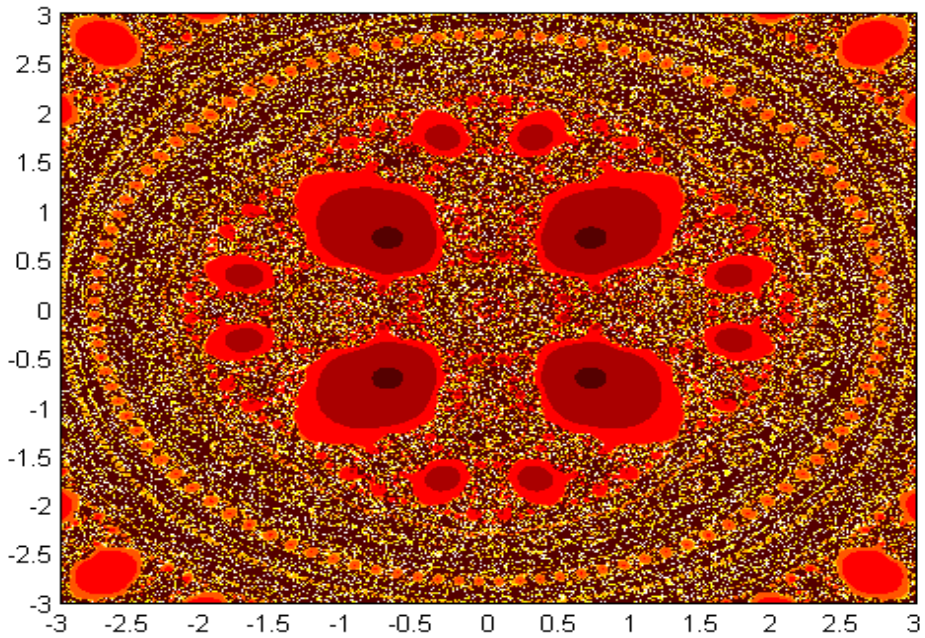


Fig. 11. Polynomiograph of z^4+1

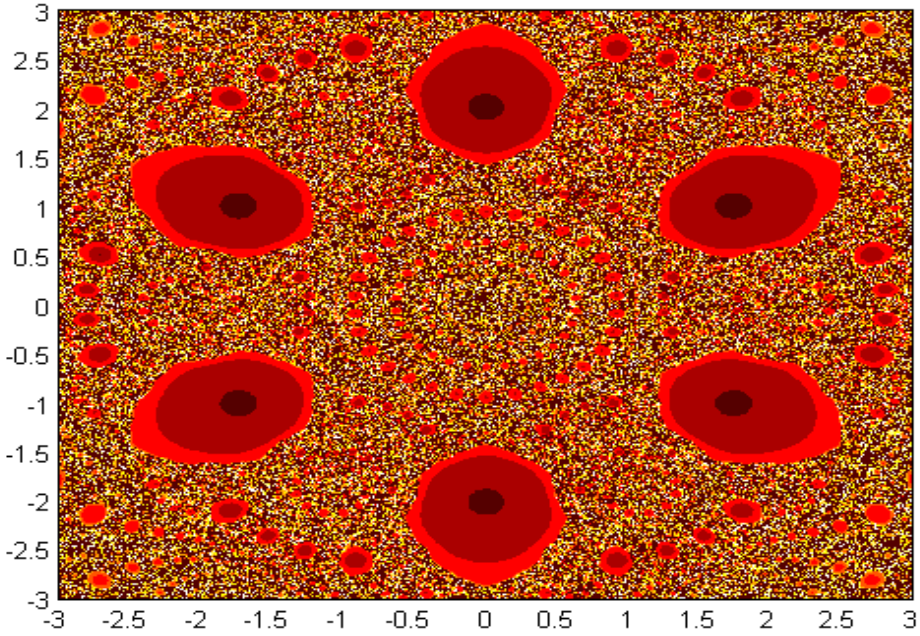


Fig. 12. Polynomiograph of z^6+64

6. Conclusion

A second derivative free optimal fourth-order iterative method has been established in this paper. The numerical and graphical comparisons clearly indicate that the newly constructed method performs efficiently with least computational cost compared to other existing iterative methods. Dynamical behavior of the developed method has also been explored to envisage the visualization of the roots of complex polynomials and is of significant interest.

References

Abbasbundy, S., (2003) Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method. Applied Mathematics and Computation, 145: 887–893.

Ali, F., Aslam, W. and Huang, S., (2020) New technique for the approximation of the zeros of nonlinear scientific models. International Journal of Nonlinear Science and Numerical Simulation, 22(2020): 705-719.

Ali, F., Aslam, W., Ali, K., Anwar, M. A. and Nadeem, A., (2018) New family of iterative methods for solving nonlinear models. Discrete Dynamics in Nature and Society, 2018, Article ID 9619680, 12 pages.

AL-Jawary, M. A. & Nabi, Z. J., (2020) Three iterative methods for solving Jeffery-Hamel flow problem. Kuwait Journal of Science, 47(1): 1-13.

Bayat, M., Pakar, I. and Bayat, M., (2015) Nonlinear vibration of mechanical systems by means of homotopy perturbation method. Kuwait Journal of Science, 42(3): 64-85.

Babolian, E. & Baizar, J., (2002) Solution of nonlinear equations by modified Adomian decomposition method. Applied Mathematics and Computation, 132: 167–172.

Bhalekar, S. & Daftardar-Gejji, V., (2011) Convergence of new iterative method. International Journal of Differential Equations, Article ID 989065, 10 pages.

Chun, C., (2018) Some fourth-order iterative methods for solving nonlinear equations. Applied Mathematics and Computation, 195(2): 454–459.

Chun, C., (2007) A method for obtaining iterative formulas of order three. Applied Mathematics Letters, 20(2007): 1103–1109.

Cordero, A., Hueso, J. L., Martinez, E. and Torregrosa J. R., (2010) New modifications of Potra-Ptak s method with optimal fourth and eighth orders of convergence. Journal of Computational and Applied Mathematics, 234(10): 2969-2976.

Cordero, A., Torregrosa, J. R. and Triguero, N.P., (2021) A general optimal iterative scheme with arbitrary order of convergence. Symmetry, 13(5): 884.

Demiray, S. T. & Bulut, H., (2017) New exact solutions for generalized Gardner equation. Kuwait Journal of Science, 44(1): 1-8.

Dogan, N., (2013) Numerical solution of chaotic Genesio system withh with multi-step Laplace Adomian decomposition method. Kuwait Journal of Science, 40(1): 109-121.

Eze, S. C., (2022) Nonlinear fractional arctic systems. Kuwait Journal of Science, 49(1): 1-23.

Hafiz, M. A. & Khirallah, M. Q., (2021) An optimal fourth order method for solving nonlinear equations. Journal of Mathematics and Computer Science, 23(2021): 86-97.

He, C. H., (2016) An introduction to an ancient Chinese algorithm and its modification. International Journal of Numerical Methods for Heat Fluid Flow, 26(8): 2486–2491.

He, C. H., Liu, C., He, J. H. and Gepreel, K. A., (2021) Low frequency property of a fractal vibration model for a concrete beam, Fractals. 29(5): 2150117.

He, C. H., Shen, Y. and Ji, F. Y., (2020) Taylor series solution for fractal Bratu-type equation arising in electrospinning process. Fractals, 28(1): Article ID 2050011, 7 pages.

He, J. H. & El-Dib, Y. O., (2020) Homotopy perturbation method for Fangzhu oscillator. Journal of Mathematical Chemistry, 58(10): 2245–2253.

Herceg, D. & Herceg, D., (2008) A method for obtaining third order iterative formulas. Novi Sad Journal of Mathematics, 38(2): 195-207.

Huang, S., Rafiq, A., Shahzad, M. R. and Ali, F., (2018) New Higher Order Iterative Methods for Solving Nonlinear Equations. Hacettepe Journal of Mathematics and Statistics, 47 (1): 77-91.

Jaiswal, J. P., (2014) Some class of third and fourth order iterative methods for solving nonlinear equations. Journal of Applied Mathematics, (2014): Article ID 817656, 17 pages.

Kalantari, B., (2005a) Method of creating graphical works based on polynomials, United States Patent 6(2005): 894–705.

Kalantari, B., (2005b) Polynomiography: From the Fundamental Theorem of Algebra to Art. Leonardo, 38(3): 233-238.

Kotarski, W., Gdawiec, K. and Lisowska, A., (2012) Polynomiography via Ishikawa and Mann iterations. Advances in Visual Computing, Part I, G. Bebis, R. Boyle, B. Parvin et al., Eds., vol. 7431 of Lecture Notes in Computer Science, pp. 305-313, Springer, Berlin, Germany, 2012.

Kalantary, B., (2009) Polynomial Root-Finding and Polynomiography. World Science Publishing Company, Hackensack.

Karakaya, V., Gursoy, F. and Erturk, M., (2016) Some convergence and data dependence results for various fixed point iterative methods. Kuwait Journal of Science, 43(1): 112-128.

Kung, H. T. & Traub, J. F., (1974) Optimal order of one-point and multi-point iteration. Applied Mathematics and Computation, 21(1974): 643-651.

Naseem, A., Rehman, M. A. and Abdeljawad, T., (2020) Higher order root finding algorithms and their basins of attraction. Journal of Mathematics, (2020): Article ID 5070363, 11 pages.

Noor, K. I., Noor, M. A. and Momani, S., (2007) Modified Householder iterative method for nonlinear equations. Applied Mathematics and Computation, 190(2007): 1534-1539.

Noor, M. A. & Gupta, V., (2007) Modified Householder iterative method free from second derivatives for nonlinear equations. Applied Mathematics and Computation, 190(2007): 1701-1706.

Ostrowski, A. M., (1973) Solution of equations in Euclidean and Banach Space. Third Eddition, Academic Press, New York.

Ozer, S., (2021) Two efficient numerical methods for solving Rosenau-Kdv-RLW equation. Kuwait Journal of Science, 48(1): 14-24.

Rehman, M. A., Naseem, A. and Abdeljawad, T., (2021) Some novel sixth-order iteration schemes for computing zeros of nonlinear scalar equations and their applications in engineering. Journal of Function Spaces, (2021): Article ID 5566379, 11 pages.

Sana, G., Mohammed, P. O., Shin, D. Y., Noor, M. A. and Oudat, M. S., (2021) On iterative methods for solving nonlinear equations in quantum calculus. Fractal and Fractional, 5(3): 60.

Sharma, R. & Bahl, A., (2015) An optimal fourth order iterative method for solving nonlinear equations and its dynamics. Journal of Complex Analysis, 9(2015): 259167-259176.

Sharma, E., Panday, S. and Dwivedi, M., (2020) New optimal fourth order iterative method for solving nonlinear equations. Iternational Journal on Emerging Technologies, 11(3): 755-758.

Shams, M., Mir, N. A., Rafiq, N., Almatroud, A. O. and Akram, S., (2020) On dynamics of iterative techniques for nonlinear equation with applications in engineering. Mathematical Problems in Engineering, (2020): Article ID 5853296, 17 pages.

Shengfeng, L., (2019) Fourth-order iterative method without calculating the higher derivatives for nonlinear equation. Journal of Algorithms and Computational Technology, 13(2019): 1-8.

Singh, A. & Jaiswal, J. P., (2014) Several new third order and fourth order iterative methods for solving nonlinear equations. International Journal of Engineering Mathematics, (2014): Article ID 828409, 11 pages.

Saeed, R. K. & Khthr, F. W., (2011) New third order iterative method for solving nonlinear equations. Journal of Applied Sciences Research, 7(6): 916-921.

Thukral, R., (2013) Introduction to higher-order iterative methods for finding multiple roots of nonlinear equations. Journal of Mathematics, (2013): Article ID 404635, 3 pages.

Uddin, I. & Imdad, M., (2015) On certain convergence of S-iteration scheme in CAT(0) spaces. Kuwait Journal of Science, 42(2): 93-106.

Submitted: 16/01/2022
Revised: 02/05/2022
Accepted: 29/05/2022
DOI: 10.48129/kjs.18253

Four dimensional matrix mappings and applications

Mehmet Ali Sarıgöl*
Dept. of Mathematics, Pamukkale University, Turkey
**Corresponding author: msarigol@pau.edu.tr*

Abstract

In this paper, we characterize the classes $(\mathcal{L}, \mathcal{L}_k)$, $(\mathcal{L}_k, \mathcal{L})$ and $(\mathcal{L}_\infty, \mathcal{L}_k)$, $1 \leq k < \infty$, of all four dimensional infinite matrices, where $\mathcal{L}_k$ and $\mathcal{L}_\infty$ are the spaces of all absolutely k-summable and bounded double sequences, respectively. Using them, we establish some relations between $\left|\overline{N}, p_n, q_n\right|$ and $\left|\overline{N}, p'_n, q'_n\right|_k$ summability methods which extend some results of Bosanquet (1950), Sarıgöl (1993), Sarıgöl & Bor (1995), and Sunouchi (1949) to double summability methods, and give a relation between single and double summability methods.

Keywords: Banach space, double matrix mapping, double summability, four dimensional matrix, inclusion theorem

1. Introduction

Let us consider an infinite single series Σx_v of complex (or real) numbers with partial sums s_n, and let (σ_n^α) denote the n-th Cesàro means of order α with $\alpha > -1$ of the sequence (s_n). The series Σx_v is said to be summable $|C, \alpha|_k$, $k \geq 1$, in Flett's notation (Flett, 1957), if $\left(n^{1/k^*}\Delta\sigma_n^\alpha\right) \in \ell_k$, where ℓ_k is the space of the set of absolutely k-summable single sequences and $1/k^* + 1/k = 1$. Let (p_n) be a sequence of positive numbers satisfying

$$P_n = \sum_{v=0}^{n} p_v \to \infty \; as \; n \to \infty, \; P_{-1} = p_{-1} = 0. \tag{1}$$

The sequence-to-sequence transformation $u_n = \sum_{v=0}^{n} p_v s_v / P_n$ defines the sequence (u_n) of the weighted mean or simply $\left(\overline{N}, p_n\right)$ mean of the sequence (s_n), generated by the sequence of coefficients (p_n) (Hardy, 1949). The series Σx_v is said to be summable $\left|\overline{N}, p_n\right|_k$, $k \geq 1$, if $\left\{\left(p_n^{-1}P_n\right)^{1/k^*}\Delta u_n\right\} \in \ell_k$, where $\Delta u_n = p_n\left(P_n P_{n-1}\right)^{-1}\sum_{v=1}^{n} P_{v-1}x_v$ (Bor, 2016), which, for $p_n = 1$, includes the method $|C, 1|_k$.

Throughout the paper, (p_n), (q_n), (p'_n) and (q'_n) will denote the sequences of positive numbers satisfying equation 1 and

$$\mu_{mn}(k) = \begin{cases} \frac{1}{P_{m-1}}\left(\frac{p_m}{P_m}\right)^{1/k}, & n = 0, m \geq 1 \\ \frac{1}{Q_{n-1}}\left(\frac{q_n}{Q_n}\right)^{1/k}, & m = 0, n \geq 1 \\ \frac{1}{P_{m-1}Q_{n-1}}\left(\frac{p_m q_n}{P_m Q_n}\right)^{1/k}, & m \geq 1, n \geq 1. \end{cases} \tag{2}$$

A summability method Y is stronger than another method X if each series summable by X implies its summability by Y (not necessarily to the same sum). Hereof, there are many papers in the literature done by various authors, e.g. (see, (Bor, 2016), (Bor & Thorpe, 1987), (Borwein & Cass, 1968), (Bosanquet, 1950), (Das *et al.*, 1967), (Flett, 1957), (Hardy, 1949), (Güleç, 2019), (Mazhar, 1972), (Mishra

et al., 2018), (Mohapatra, 1967), (Rhoades, 1998), (Rhoades, 1999), (Rhoades, 2003), (Sarıgöl, 1991), (Sarıgöl, 1992), (Sarıgöl, 1993), (Sarıgöl & Bor, 1995), (Sarıgöl, 2021), (Sarıgöl & Mursaleen, 2021), (Sunouchi, 1949), (Thorpe, 1972), (Zraiqat, 2019)). Among them, in the special case $k = 1$ the following known result is due to Sunouchi (Sunouchi, 1949).

Theorem 1.1. In order that every $\left|\overline{N}, p_n\right|$ summable series should be $\left|\overline{N}, p'_n\right|$ summable, it is sufficient that

$$\frac{p'_n P_n}{P'_n p_n} = O(1). \tag{3}$$

Reviewing this paper, Bosanquet observed that equation 3 is also necessary for the conclusion and so completed Theorem 1.1 in necessary and sufficient form (see (Bosanquet, 1950)).

In (Sarıgöl, 1993), Theorem 1.1 has been extended to the case $1 \leq k < \infty$ as follows.

Theorem 1.2. Let $1 \leq k < \infty$. Then, in order that every $\left|\overline{N}, p_n\right|$ summable series should be $\left|\overline{N}, p'_n\right|_k$ summable, it is necessary and sufficient that

$$\frac{p'_n}{P'_n}\left(\frac{P_n}{p_n}\right)^k = O(1).$$

Also, it has been showed in (Sarıgöl & Bor, 1995) that the converse of the implication is not true.

Theorem 1.3. Let $1 < k < \infty$. Then, for every sequences (p_n) and (p'_n), there exists a series which is summable $\left|\overline{N}, p_n\right|_k$ but is not summable by $\left|\overline{N}, p'_n\right|$.

First, we recall related notations. Let $\sum_{r=0}^{\infty}\sum_{s=0}^{\infty} x_{rs}$ be an infinite double series of real or complex numbers with partial sums s_{mn}, *i.e.*,

$$s_{mn} = \sum_{r=0}^{m}\sum_{s=0}^{n} x_{rs}. \tag{4}$$

For the sake of brevity, we denote the summations $\sum_{r=0}^{\infty}\sum_{s=0}^{\infty}$ and $\sum_{r=0}^{m}\sum_{s=0}^{n}$ by $\sum_{r,s=0}^{\infty}$ and $\sum_{r,s=0}^{m,n}$, respectively. By T_{mn}, we denote the double Riesz mean transformation $\left(\overline{N}, p_m, q_n\right)$ of the double sequence (s_{mn}), *i.e.*,

$$T_{mn} = \frac{1}{P_m Q_n}\sum_{r,s=0}^{m,n} p_r q_s s_{rs}. \tag{5}$$

The series $\sum_{r,s=0}^{\infty} x_{rs}$ is said to be summable $\left|\overline{N}, p_m, q_n\right|_k$, $k \geq 1$, if (see (Sarıgöl, 2021))

$$\sum_{m,n=0}^{\infty}\left(\frac{P_m Q_n}{p_m q_n}\right)^{k-1}\left|\overline{\Delta} T_{mn}\right|^k < \infty \tag{6}$$

where $\overline{\Delta} T_{00} = s_{00} = x_{00}$, and, for $m, n \geq 1$,

$$\begin{aligned}\overline{\Delta} T_{m0} &= T_{m0} - T_{m-1,0},\ \overline{\Delta} T_{0n} = T_{0n} - T_{0,n-1,}\\ \overline{\Delta} T_{mn} &= T_{mn} - T_{m-1,n} - T_{m,n-1} + T_{m-1,n-1}.\end{aligned}$$

We note that, in the special case $p_n = q_n = 1$, the summability $\left|\overline{N}, p_m, q_n\right|_k$ reduces to the absolute double Cesàro summability $|C, 1, 1|_k$, given by Rhoades (1998).

There is a close relationship between the method $\left|\overline{N}, p_m, q_n\right|_k$ and the space $\mathcal{L}_k$, $1 \leq k < \infty$, defined by the set of all double sequences $x = (x_{rs})$ of complex numbers such that $\sum_{r,s=0}^{\infty}|x_{rs}|^k < \infty$, which reduces to $\mathcal{L}$ for $k = 1$, studied by Zeltser (2001). Also, $\mathcal{L}_k$ is the Banach space (Başar & Sever, 2009) according to its natural norm

$$\|x\|_{\mathcal{L}_k} = \left(\sum_{r,s=0}^{\infty}|x_{rs}|^k\right)^{1/k}, 1 \leq k < \infty.$$

Further, the space $\mathcal{L}_\infty$ consists of all bounded double sequences and it is a Banach space with the norm $\|x\|_{\mathcal{L}_\infty} = \sup_{r,s} |x_{rs}|$.

Let $x = (x_{rs})$ be a double sequence of complex numbers. If for every $\varepsilon > 0$ there exists a natural integer $n_0(\varepsilon)$ and real number l such that $|x_{rs} - l| < \varepsilon$ for all $r, s \geq n_0(\varepsilon)$, then, the double sequence x is said to be convergent in the Pringsheim sense. Also, a double series $\sum_{r,s=0}^{\infty} x_{rs}$ is convergent if and only if the double sequence (s_{mn}) in equation 4 is convergent.

Let U and V be two double sequence spaces, and $A = (a_{mnrs})$ be a four dimensional infinite matrix of complex (or, real) numbers. Then, A defines a matrix transformation from U into V, written $A \in (U, V)$, if for every sequence $x = (x_{rs}) \in U$, the A-transform $A(x) = (A_{mn}(x))$ of x exists and belongs to V, where

$$A_{mn}(x) = \sum_{r,s=0}^{\infty} a_{mnrs} x_{rs}$$

provided the double series on right side converges for $m, n \geq 0$.

The transpose $A^t = (a_{rsmn})$ of the matrix $A = (a_{mnrs})$ is defined by

$$A^t_{rs}(x) = \sum_{m,n=0}^{\infty} a_{mnrs} x_{mn} \text{ for } m, n \geq 0.$$

The β-dual U^β of the space U is the set of all double sequences (b_{rs}) such that $\sum_{r,s=0}^{\infty} b_{rs} x_{rs}$ converges for all $x \in U$.

In this paper we characterize the classes $(\mathcal{L}, \mathcal{L}_k)$, $(\mathcal{L}_k, \mathcal{L})$ and $(\mathcal{L}_\infty, \mathcal{L}_k)$, $k \geq 1$, of all four dimensional infinite matrices, and extend Theorem 1.1, Theorem 1.2 and Theorem 1.3 to double summability methods, and also establish a relation between single and double summability methods.

2. Needed Lemmas

We require the following lemmas for the proofs of our theorems.

Lemma 2.1 (Zaanen 1953, p.134) A linear mapping T from a Banach space U into another Banach space V is continuous if and only if it is bounded, i.e., there exists a constant L such that $\|T(x)\|_V \leq L \|x\|_U$ for all $x \in U$.

Lemma 2.2 (Sarıgöl, 1991) Let $k > 0$. Then, there exists two strictly positive constans M_1 and M_2, depending only on k, such that

$$\frac{M_1}{P^k_{r-1}} \leq \sum_{m=r}^{\infty} \mu^k_{m0}(k) \leq \frac{M_2}{P^k_{r-1}} \tag{7}$$

for all $r \geq 1$, where M_1 and M_2 are independent of (p_n).

Lemma 2.3 (Sarıgöl, 2021) Let $k > 0$. Then, there exists two strictly positive constans N_1 and N_2, depending only on k, such that

$$\frac{N_1}{P^k_{r-1} Q^k_{s-1}} \leq \sum_{m,n=r,s}^{\infty} \mu^k_{mn}(k) \leq \frac{N_2}{P^k_{r-1} Q^k_{s-1}} \tag{8}$$

for all $r, s \geq 1$, where N_1 and N_2 are independent of (p_n) and (q_n).

3. Main Result

Our results are as follows.

Theorem 3.1 Let $k \geq 1$ and $A = (a_{mnrs})$ be a four dimensional infinite matrix of complex numbers. Then, in order that $A \in (\mathcal{L}, \mathcal{L}_k)$ it is necessary and sufficient that

$$\sum_{m,n=0}^{\infty} |a_{mnrs}|^k = O(1). \tag{9}$$

Proof. Assume equation 9 holds. Then, we should show that $A(x) = (A_{mn}(x)) \in \mathcal{L}_k$ for every $x = (x_{rs}) \in \mathcal{L}$. Now, using equation 9, it follows from Minkowski's inequality that

$$\begin{aligned} \|A(x)\|_{\mathcal{L}_k} &= \left(\sum_{m,n=0}^{\infty} |A_{mn}(x)|^k \right)^{1/k} \leq \left(\sum_{m,n=0}^{\infty} \left(\sum_{r,s=0}^{\infty} |a_{mnrs} x_{rs}| \right)^k \right)^{1/k} \\ &= \sum_{r,s=0}^{\infty} |x_{rs}| \left(\sum_{m,n=0}^{\infty} |a_{mnrs}|^k \right)^{1/k} = O(1) \|x\|_{\mathcal{L}} < \infty. \end{aligned}$$

which gives the desired conclusion.

Conversely, let $A \in (\mathcal{L}, \mathcal{L}_k)$. Then, for $k \geq 1$, since $\mathcal{L}_k$ is a Banach space (see (Başar & Sever, 2009)), by Lemma 2.1, there exists a constant K such that $\|A(x)\|_{\mathcal{L}_k} \leq K \|x\|_{\mathcal{L}}$, *i.e.*,

$$\left(\sum_{m,n=0}^{\infty} \left| \sum_{r,s=0}^{\infty} a_{mnrs} x_{rs} \right|^k \right)^{1/k} \leq K \|x\|_{\mathcal{L}} \tag{10}$$

for all $x \in \mathcal{L}$. So, by applying the double sequence $x \in \mathcal{L}$ to equation 10, where $x_{ij} = 1$ for $i = r, j = s$, zero otherwise, we obtain

$$\sum_{m,n=0}^{\infty} |a_{mnrs}|^k \leq K, \text{ for } r, s \geq 0, \tag{11}$$

which gives equation 9.

This step concludes the proof.

Theorem 3.2 Let $1 < k < \infty$ and $A = (a_{mnij})$ be an four dimensional infinite matrix of complex numbers. Define $W_k(A)$ and $w_k(A)$ by

$$W_k(A) = \sum_{r,s=0}^{\infty} \left(\sum_{m,n=0}^{\infty} |a_{mnrs}| \right)^k, \tag{12}$$

$$w_k(A) = \sup_{MXN} \sum_{r,s=0}^{\infty} \left| \sum_{(m,n) \in MXN} a_{mnrs} \right|^k \tag{13}$$

where M and N are finite subsets of natural numbers. Then, the following statements are equivalent:

$$(i) \ W_{k^*}(A) < \infty \qquad (ii) \ A \in (\mathcal{L}_k, \mathcal{L})$$

$$(iii) \ A^t \in (\mathcal{L}_\infty, \mathcal{L}_{k^*}) \qquad (iv) \ w_{k^*}(A) < \infty.$$

where k^* is the conjugate of k, *i.e.*, $1/k + 1/k^* = 1$.

Proof. To prove the Theorem, it is enough to show that $(i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (iv) \Rightarrow (i)$.

$(i) \Rightarrow (ii)$. Assume (i) holds. Then, for all $x \in \mathcal{L}_k$, it follows from Hölder's inequality that

$$\begin{aligned} \|A(x)\|_{\mathcal{L}} &= \sum_{m,n=0}^{\infty} \left| \sum_{r,s=0}^{\infty} a_{mnrs} x_{rs} \right| \leq \sum_{r,s=0}^{\infty} \sum_{m,n=0}^{\infty} |a_{mnrs} x_{rs}| \\ &\leq \left\{ \sum_{r,s=0}^{\infty} \left(\sum_{m,n=0}^{\infty} |a_{mnrs}| \right)^{k^*} \right\}^{1/k^*} \|x\|_{\mathcal{L}_k} \\ &\leq (W_{k^*}(A))^{1/k^*} \|x\|_{\mathcal{L}_k} < \infty, \end{aligned} \tag{14}$$

which gives (ii).

$(ii) \Rightarrow (iii)$. Suppose $A \in (\mathcal{L}_k, \mathcal{L})$. Then, since $\mathcal{L}_k$ is a Banach space, where $k \geq 1$, by Lemma 2.1, there exists a constant L such that

$$\|A(x)\|_{\mathcal{L}} = \sum_{m,n=0}^{\infty} \left| \sum_{r,s=0}^{\infty} a_{mnrs} x_{rs} \right| \leq L \|x\|_{\mathcal{L}_k} \tag{15}$$

for all $x \in \mathcal{L}_k$. Also, it is observed by putting $x_{rs}\text{sgn}a_{mnrs}$ instead of x_{rs} that

$$\sum_{m,n=0}^{\infty} \sum_{r,s=0}^{\infty} |a_{mnrs} x_{rs}| \leq L \|x\|_{\mathcal{L}_k}. \tag{16}$$

Now, let $u \in \mathcal{L}_\infty$ be given. Then, by equation 15,

$$\begin{aligned} \left| \sum_{m,n=0}^{\infty} \sum_{r,s=0}^{\infty} u_{mn} a_{mnrs} x_{rs} \right| &\leq \|u\|_{\mathcal{L}\infty} \sum_{m,n=0}^{\infty} \sum_{r,s=0}^{\infty} |a_{mnrs} x_{rs}| \\ &\leq L \|u\|_{\mathcal{L}\infty} \|x\|_{\mathcal{L}_k}. \end{aligned} \tag{17}$$

In equation 17, taking $x_{rs} = 1$ for $(r, s) = (i, j)$, and zero otherwise, it is easily seen that

$$\left| \sum_{m,n=0}^{\infty} a_{mnrs} u_{mn} \right| \leq \sum_{m,n=0}^{\infty} |a_{mnrs} u_{mn}| \leq L \|u\|_{\mathcal{L}_\infty},$$

which gives that $A^t(u)$ is defined for all $r, s \geq 0$, where the double sequence $A^t(u) = \left(A^t_{rs}(u)\right)$ is given by

$$A^t_{rs}(u) = \sum_{m,n=0}^{\infty} a_{mnrs} u_{mn} : m, n \geq 0 \tag{18}$$

Again, it follows by considering equation 17 that

$$\left| \sum_{r=0}^{\infty} \sum_{s=0}^{\infty} A^t_{rs}(u) x_{rs} \right| \leq L \|u\|_{\mathcal{L}\infty} \|x\|_{\mathcal{L}_k} \tag{19}$$

which implies that the series in the left side hand of equation 19 converges. Therefore, since the dual of space $\mathcal{L}_k$ is the space $\mathcal{L}_{k^*}$(see (Başar & Sever, 2009)), we obtain $A^t(u) \in \mathcal{L}_{k^*}, i.e., A^t \in (\mathcal{L}_\infty, \mathcal{L}_{k^*})$.

$(iii) \Rightarrow (iv)$. If $A^t \in (\mathcal{L}_\infty, \mathcal{L}_{k^*})$, then, by Lemma 2.1, there exists a constant K such that $\left\|A^t(x)\right\|_{\mathcal{L}_{k^*}} \leq K \|x\|_{\mathcal{L}_\infty}$ for all $x \in \mathcal{L}_\infty, i.e.,$

$$\left(\sum_{r,s=0}^{\infty} \left| \sum_{m,n=0}^{\infty} a_{mnrs} x_{mn} \right|^{k^*} \right)^{1/k^*} \leq K \|x\|_{\mathcal{L}_\infty}. \tag{20}$$

Let M and N be any finite subsets of all nature numbers. Take a sequence $x = (x_{mn})$ as $x_{mn} = 1$ for $(r, s) \in MXN$, and zero otherwise. Then, equation 20 is reduced to.

$$\left(\sum_{r,s=0}^{\infty} \left| \sum_{(m,n)\in MXN} a_{mnrs} \right|^{k^*} \right)^{1/k^*} \leq K$$

which proves $w_{k^*}(A) < \infty$.

$(iii) \Rightarrow (iv)$. Suppose (iii) is satisfied and a_{mnrs} are real numbers. Then, for every finite subsets M and N of nature numbers,

$$\sum_{r,s=0}^{\infty} \left| \sum_{(m,n)\in MXN} a_{mnrs} \right|^{k^*} \leq w_{k^*}(A).$$

Let $H^+ = \{(m,n) \in MXN : a_{mnrs} \geq 0\}$ and $H^- = \{(m,n) \in MXN : a_{mnrs} < 0\}$. Then, by considering the inequality $|a+b|^{k^*} \leq 2^{k^*}\left(|a|^{k^*} + |b|^{k^*}\right)$, where a and b are complex numbers, we have

$$\begin{aligned} W_{k^*}(A) &= \sum_{r,s=0}^{\infty} \left(\sum_{m,n=0}^{\infty} |a_{mnrs}| \right)^{k^*} \\ &= \sum_{r,s=0}^{\infty} \left\{ \sum_{(m,n)\in H^+}^{\infty} a_{mnrs} + \sum_{(m,n)\in H^-}^{\infty} -a_{mnrs} \right\}^{k^*} \\ &\leq 2^{k^*} \sum_{r,s=0}^{\infty} \left\{ \left(\sum_{(m,n)\in H^+}^{\infty} a_{mnrs} \right)^{k^*} + \left(\sum_{(m,n)\in H^-}^{\infty} -a_{mnrs} \right)^{k^*} \right\} \\ &\leq 2^{k^*+1} w_k(A). \end{aligned}$$

If a_{mnrs} is complex number for $m,n,r,s \geq 0$, it is easily seen that $W_{k^*}(A) \leq 2^{2k^*+3} w_k(A) < \infty$, which implies (iv).

Thus the proof of the Theorem is completed.

Theorem 3.3 Let $k \geq 1$. Then, in order that every $\left|\overline{N}, p_m, q_n\right|$ summable double series should be summable $\left|\overline{N}, p'_m, q'_n\right|_k$, it is necessary and sufficient that

$$(i) \quad \frac{p'_m}{P'_m}\left(\frac{P_m}{p_m}\right)^k = O(1) \quad \text{and} \quad (ii) \quad \frac{q'_n}{Q'_n}\left(\frac{Q_n}{q_n}\right)^k = O(1). \tag{21}$$

Proof. Suppose that euation 21i and equation 21ii are satisfied. Let (T_{mn}) and (T'_{mn}) be the double sequences of $\left(\overline{N}, p_n, q_n\right)$ and $\left(\overline{N}, p'_n, q'_n\right)$ means of the series $\sum_{r,s=0}^{\infty} x_{rs}$, respectively, i.e.,

$$T_{mn} = \frac{1}{P_m Q_n} \sum_{r,s=0}^{m,n} p_r q_s \sum_{v,\mu=0}^{r,s} x_{v\mu}, \tag{22}$$

$$T'_{mn} = \frac{1}{P'_m Q'_n} \sum_{r,s=0}^{m,n} p'_r q'_s \sum_{v,\mu=0}^{r,s} x_{v\mu}. \tag{23}$$

Then, since $P_{-1} = Q_{-1} = 0$, it can be written that

$$\begin{aligned} T_{mn} &= \frac{1}{P_m Q_n} \sum_{v,\mu=0}^{m,n} p_v q_\mu \sum_{r,s=0}^{v,\mu} x_{r,s} \\ &= \frac{1}{P_m Q_n} \sum_{r,s=0}^{m,n} x_{r,s} \sum_{v,\mu=r,s}^{m,n} p_v q_\mu \\ &= \frac{1}{P_m Q_n} \sum_{r,s=0}^{m,n} x_{r,s} \left(P_m - P_{r-1}\right)\left(Q_n - Q_{s-1}\right) \\ &= \sum_{r,s=0}^{m,n} x_{rs} \left(1 - \frac{P_{r-1}}{P_m}\right)\left(1 - \frac{Q_{s-1}}{Q_n}\right), \end{aligned}$$

which implies

$$\begin{aligned} y_{00} &= \overline{\Delta}T_{00} = x_{00} \\ y_{m0} &= \overline{\Delta}T_{m0} = \frac{p_m}{P_m P_{m-1}} \sum_{r=1}^{m} P_{r-1} x_{r0} \\ y_{0n} &= \overline{\Delta}T_{0n} = \frac{q_n}{Q_n Q_{n-1}} \sum_{s=1}^{n} Q_{s-1} x_{0s} \\ y_{mn} &= \overline{\Delta}T_{mn} = \frac{p_m q_n}{P_m P_{m-1} Q_n Q_{n-1}} \sum_{r=1,s}^{m,n} P_{r-1} Q_{s-1} x_{rs}. \end{aligned} \tag{24}$$

Also, similarly, we get

$$\overline{\Delta}T'_{m,n} = \frac{p'_m q'_n}{P'_m P'_{m-1} Q'_n Q'_{n-1}} \sum_{r,s=1}^{m,n} P'_{r-1} Q'_{s-1} x_{rs}. \tag{25}$$

The double series $\sum_{r,s=0}^{\infty} x_{r,s}$ is summable $\left|\overline{N}, p_m, q_n\right|$ iff $y = (y_{mn}) \in \mathcal{L}$, and also we obtain by solving equation 25 for x_{rs} that, for $m, n \geq 1$,

$$\begin{aligned} x_{00} &= y_{00} \\ x_{m0} &= \frac{P_m}{p_m} y_{m0} - \frac{P_{m-2}}{p_{m-1}} y_{m-1,0} \\ x_{0n} &= \frac{Q_n}{q_n} y_{0n} - \frac{Q_{n-2}}{q_{n-1}} y_{0,n-1} \\ x_{mn} &= \frac{P_m Q_n}{p_m q_n} y_{mn} - \frac{P_{m-2} Q_n}{p_{m-1} q_n} y_{m-1,n} - \\ &\quad \frac{Q_{n-2} P_m}{q_{n-1} p_m} y_{m,n-1} + \frac{P_{m-2} Q_{n-2}}{p_{m-1} q_{n-1}} y_{m-1,n-1} \end{aligned} \tag{26}$$

Let

$$y'_{mn} = \left(\frac{P'_m Q'_n}{p'_m q'_n}\right)^{1-1/k} \overline{\Delta}T'_{mn} = \mu'_{mn}(k) \sum_{r,s=1}^{m,n} P'_{r-1} Q'_{s-1} x_{rs} \tag{27}$$

where $\overline{\Delta}T'_{mn}$ is defined by equation 25, and $\mu'_{mn}(k)$ is obtained from $\mu_{mn}(k)$ interchanging p_m and p_m by p'_m and q'_n, respectively . Then, by equation 27, the double series $\sum_{r,s=0}^{\infty} x_{rs}$ is summable $\left|\overline{N}, p'_n, p'_n\right|_k$ iff $y' = (y'_{mn}) \in \mathcal{L}_k$. Further, it follows from equation 26 and equation 27 that, for $m, n \geq 1$,

$$\begin{aligned} y'_{m0} &= \mu'_{m0}(k) \sum_{r=1}^{m-1} \frac{p_r P'_r - p'_r P_r}{p_r} y_{r0} + \frac{\mu'_{m0}(k) P'_{m-1} P_m}{p_m} y_{m0}, \\ y'_{0n} &= \mu'_{0n}(k) \sum_{s=1}^{n-1} \frac{q_s Q'_s - q'_s Q_s}{q_s} y_{0s} + \frac{\mu'_{0n}(k) Q'_{n-1} Q_n}{q_n} y_{0n}, \end{aligned}$$

$$
\begin{aligned}
y'_{mn} &= \mu'_{mn}(k)\sum_{r,s=1}^{m,n} P'_{r-1}Q'_{s-1}\left(\frac{P_rQ_s}{p_rq_s}y_{rs}-\frac{P_{r-2}Q_s}{p_{r-1}q_s}y_{r-1,s}\right.\\
&\left.-\frac{P_rQ_{s-2}}{p_rq_{s-1}}y_{r,s-1}+\frac{P_{r-2}Q_{s-2}}{p_{r-1}q_{s-1}}y_{r-1,s-1}\right)\\
&= \mu'_{mn}(k)\left\{\sum_{r,s=1}^{m,n} P'_{r-1}Q'_{s-1}\frac{P_rQ_s}{p_rq_s}y_{rs}-\sum_{r,s=1}^{m-1,n} P'_rQ'_{s-1}\frac{P_{r-1}Q_s}{p_rq_s}y_{rs}\right.\\
&\left.-\sum_{r,s=1}^{m,n-1} P'_{r-1}Q'_s\frac{P_rQ_{s-1}}{p_rq_s}y_{rs}+\sum_{r,s=1}^{m-1,n-1} P'_rQ'_s\frac{P_{r-1}Q_{s-1}}{p_rq_s}y_{rs}\right\}\\
&= \mu'_{mn}(k)\left\{\frac{P'_{m-1}P_mQ'_{n-1}Q_n}{p_mq_n}y_{mn}+\frac{P'_{m-1}P_m}{p_m}\sum_{s=1}^{n-1}\frac{q_sQ'_{s-1}-q'_sQ_{s-1}}{q_s}y_{ms}\right.\\
&+\frac{Q'_{n-1}Q_n}{q_n}\sum_{r=1}^{m-1}\frac{p_rP'_{r-1}-p'_rP_{r-1}}{p_r}y_{rn}+\\
&\left.\sum_{r,s=1}^{m-1,n-1}\frac{\left(q_sQ'_{s-1}-q'_sQ_{s-1}\right)\left(p_rP'_{r-1}-p'_rP_{r-1}\right)}{q_sp_r}\right\}y_{rs}.
\end{aligned}
$$

Therefore we can state

$$y'_{mn}=\sum_{r,s=0}^{m,n}a_{mnrs}y_{rs}=A_{mn}(y),$$

that is, $y\prime=(y'_{mn})$ is the $A-$transform sequence of the sequence $y=(y_{rs})$, where the matrix $A=(a_{mnrs})$ is defined by

$$
a_{mnrs}=\begin{cases}
\frac{\mu'_{0n}(k)Q'_{n-1}Q_n}{q_n}, & s=n,\ m=r=0\\
\frac{\mu'_{0n}(k)(q_sQ'_s-q'_sQ_s)}{q_s}, & 1\le s<n,\ m=r=0\\
\frac{\mu'_{m0}(k)P'_{m-1}P_m}{p_m}, & r=m,\ n=s=0\\
\frac{\mu'_{m0}(k)(p_rP'_r-p'_rP_r)}{p_r}, & 1\le r<m,\ n=s=0\\
\frac{\mu'_{mn}(k)P'_{m-1}P_m\left(q_sQ'_{s-1}-q'_sQ_{s-1}\right)}{p_m}, & 1\le s<n\\
\frac{\mu'_{mn}(k)Q'_{n-1}Q_n\left(p_rP'_{r-1}-p'_rP_{r-1}\right)}{q_np_r}, & 1\le r<m\\
\frac{\mu'_{mn}(k)\left(q_sQ'_{s-1}-q'_sQ_{s-1}\right)\left(p_rP'_{r-1}-p'_rP_{r-1}\right)}{q_sp_r}, & 1\le s<n, 1\le r<m\\
\frac{\mu'_{mn}(k)P'_{m-1}P_mQ'_{n-1}Q_n}{p_mq_n}, & r=m,\ s=n\\
0, \text{ otherwise}
\end{cases}
$$

This gives that $\left|\overline{N},p_m,q_n\right|\Rightarrow\left|\overline{N},p'_m,q'_n\right|_k$ iff $(y'_{mn})\in\mathcal{L}_k$ for every $(y_{mn})\in\mathcal{L}, i.e., \mathcal{A}\in(\mathcal{L},\mathcal{L}_k)$. Now, by Theorem 3.1, we should show that equation 21i and equation 21ii are equivalent to the equation 9. To do this, let us write

$$
\begin{aligned}
\sum_{m,n=r,s}^{\infty}|a_{mnrs}|^k &= \sum_{m=r}^{\infty}\left(|a_{msrs}|^k+\sum_{n=s+1}^{\infty}|a_{mnrs}|^k\right)\\
&= |a_{rsrs}|^k+\sum_{m=r+1}^{\infty}|a_{msrs}|^k+\sum_{n=s+1}^{\infty}|a_{rnrs}|^k+\sum_{m,n=r+1,s+1}^{\infty}|a_{mnrs}|^k\\
&= L_1+L_2+L_3+L_4, \text{ say.}
\end{aligned}
$$

Then, equation 9 holds iff $L_1 = O(1),\ L_2 = O(1),\ L_3 = O(1)$ and $L_4 = O(1)$. Now, it is written that

$$\begin{aligned} L_1' &= |a_{0s0s}| = \left(\frac{q_s'}{Q_s'}\right)^{1/k} \frac{Q_s}{q_s} \\ L_1'' &= |a_{r0r0}| = \left(\frac{p_r'}{P_r'}\right)^{1/k} \frac{P_r}{p_r} \\ L_1''' &= |a_{rsrs}| = \left(\frac{p_r'q_s'}{P_r'Q_s'}\right)^{1/k} \frac{P_rQ_s}{p_rq_s}. \end{aligned}$$

Hence, if $L_1' = O(1)$ and $L_1'' = O(1)$, then, since $p_r \leq P_r$ and $q_s \leq Q_s$ for all r, s, then, $p_r'P_r/P_r'p_r = O(1)$ and $q_s'Q_s/Q_s'q_s = O(1)$, and so we have $L_1''' = O(1)$. This shows that $L_1 = O(1)$ if and only if $L_1' = O(1)$ and $L_1'' = O(1)$, or, equivalently, equation 21i and equation 21ii hold. Also, using equation 21i and equation 21ii, it follows from Lemma 2.2 and Lemma 2.3 that

$$\begin{aligned} L_2 &= \sum_{m=r+1}^{\infty} |a_{msrs}|^k \leq \sum_{m=r+1}^{\infty} \left(|a_{m0r0}|^k + |a_{msrs}|^k\right) \\ &= \left\{ \left| \left(P_r' - p_r' \frac{P_r}{p_r}\right) \right|^k + \left| \left(\frac{q_s'}{Q_s'}\right)^{1/k} \frac{Q_s}{q_s} \left(P_{r-1}' - \frac{p_r'P_{r-1}}{p_r}\right) \right|^k \right\} \frac{1}{P_r'^k} \\ &= \left| \left(1 - \frac{p_r'P_r}{P_r'p_r}\right) \right|^k + \frac{q_s'}{Q_s'} \left(\frac{Q_s}{q_s}\right)^k \left| \left(1 - \frac{p_r'P_r}{P_r'p_r}\right) \right|^k = O(1), \end{aligned}$$

$$\begin{aligned} L_3 &= \sum_{n=s+1}^{\infty} |a_{rnrs}|^k \leq \sum_{n=s+1}^{\infty} \left(|a_{0n0s}|^k + |a_{rnrs}|^k\right) \\ &= \left\{ \left| Q_s' - q_s' \frac{Q_s}{q_s} \right|^k + \left| \left(\frac{p_r'}{P_r'}\right)^{1/k} \frac{P_r}{p_r} \left(Q_{s-1}' - \frac{q_s'Q_{s-1}}{q_s}\right) \right|^k \right\} \frac{1}{Q_s'^k} \\ &= \left| 1 - \frac{q_s'Q_s}{Q_s'q_s} \right|^k + \frac{p_r'}{P_r'} \left(\frac{P_r}{p_r}\right)^k \left| \left(1 - \frac{q_s'Q_s}{Q_s'q_s}\right) \right|^k = O(1), \end{aligned}$$

$$\begin{aligned} L_4 &= \sum_{m,n=r+1,s+1}^{\infty} |a_{mnrs}|^k \\ &= \sum_{m,n=r+1,s+1}^{\infty} \left| \mu_{mn}'(k) \left(Q_{s-1}' - \frac{q_s'Q_{s-1}}{q_s}\right) \left(P_{r-1}' - \frac{p_r'P_{r-1}}{p_r}\right) \right|^k \\ &= \left| \left(Q_{s-1}' - \frac{q_s'Q_{s-1}}{q_s}\right) \left(P_{r-1}' - \frac{p_r'P_{r-1}}{p_r}\right) \right|^k \sum_{m,n=r+1,s+1}^{\infty} \mu_{mn}'^k(k) \\ &= \left| \left(Q_{s-1}' - \frac{q_s'Q_{s-1}}{q_s}\right) \left(P_{r-1}' - \frac{p_r'P_{r-1}}{p_r}\right) \right|^k \frac{1}{P_r'^k Q_s'^k} \\ &= O(1) \left(\frac{q_s'Q_s}{Q_s'q_s} \frac{p_r'P_r}{P_r'p_r}\right)^k = O(1). \end{aligned}$$

This completes the proof.

Theorem 1.2 and Theorem 3.3 lead to the following result which gives a important relation between single and double absolute Riesz summability methods.

Corollary 3.4 Let $k \geq 1$. Then, in order that every $\left|\overline{N}, p_m, q_n\right|$ summable double series should be summable $\left|\overline{N}, p_m', q_n'\right|_k$ it is necessary and sufficient that every $\left|\overline{N}, p_m\right|$ and $\left|\overline{N}, q_n\right|$ summable simple series are summable $\left|\overline{N}, p_m'\right|_k$ and $\left|\overline{N}, q_n'\right|_k$, respectively.

For $k=1$, Theorem 3.3 also extends the result of Bosanquet (1950) and Sunouchi (1949) to double summability as follows.

Corollary 3.5 In order that every $\left|\overline{N},p_m,q_n\right|$ summable double series should be summable $\left|\overline{N},p'_m,q'_n\right|_k$ it is necessary and sufficient that

$$(i)\quad \frac{p'_m P_m}{P'_m p_m}=O(1) \qquad \text{and} \qquad (ii)\quad \frac{q'_n Q_n}{Q'_n q_n}=O(1).$$

For $p_n=q_n=1$, $\left|\overline{N},p_n,p_n\right|_k$ reduces to $|C,1,1|_k$ and hence one can obtain some new results as:

Corollary 3.6 Let $k\geq 1$.Then, in order that every $\left|\overline{N},p_m,q_n\right|$ summable double series should be summable $|C,1,1|_k$ it is necessary and sufficient that

$$(i)\quad \frac{1}{m}\left(\frac{P_m}{p_m}\right)^k=O(1) \qquad \text{and} \qquad (ii)\quad \frac{1}{n}\left(\frac{Q_n}{q_n}\right)^k=O(1).$$

Corollary 3.7 Let $k\geq 1$.Then, in order that every $|C,1,1|$ summable double series should be summable $\left|\overline{N},p_m,q_n\right|_k$ it is necessary and sufficient that

$$(i)\quad m^k\frac{p_m}{P_m}=O(1) \qquad \text{and} \qquad (ii)\quad n^k\frac{q_n}{Q_n}=O(1).$$

However the following result shows that converse implication of Theorem 3.3 is not true.

Theorem 3.8 Let $k>1$. Then, for every sequences (p_m), (q_n), (p'_m) and (q'_n), there exists a series which is summable $\left|\overline{N},p_m,q_n\right|_k$ but not summable $\left|\overline{N},p'_m,q'_n\right|$.

Proof. Let us consider (T_{mn}) and (T'_{mn}) defined by equation 22 and equation 23. Write

$$Y_{mn}=\mu_{mn}(k)\overline{\Delta}T_{mn} \text{ for } m,n\geq 0 \tag{28}$$

where $\overline{\Delta}T=\left(\overline{\Delta}T_{mn}\right)$ is defined by equation 24. Then the double series $\sum_{r,s=0}^{\infty}x_{r,s}$ is summable $\left|\overline{N},p_m,q_n\right|_k$ and $\left|\overline{N},p'_m,q'_n\right|$ if and only if $Y=(Y_{mn})\in\mathcal{L}_k$ and $\overline{\Delta}T'=\left(\overline{\Delta}T'_{m,n}\right)\in\mathcal{L}$, respectively, where $\overline{\Delta}T'_{m,n}$ is given by equation 25 . Further, by equation 2 and equation 28, for $m,n\geq 1$,

$$\begin{aligned}
\overline{\Delta}T'_{m,0} &= \mu'_{m0}(1)\sum_{r=1}^{m-1}\frac{\left(P'_{r-1}P_r-P'_rP_{r-1}\right)Y_{r0}}{p_r\mu_{r0}(k)}+\frac{P'_{m-1}P_m\mu'_{m0}(1)Y_{m0}}{p_m\mu_{m0}(k)}\\
\overline{\Delta}T'_{0,n} &= \mu'_{0n}(1)\sum_{s=1}^{n-1}\frac{\left(Q'_{s-1}Q_s-Q'_sQ_{s-1}\right)Y_{0s}}{q_s\mu_{0s}(k)}+\frac{Q'_{n-1}Q_n\mu'_{0n}(1)Y_{0n}}{q_n\mu_{0n}(k)}
\end{aligned}$$

and

$$\begin{aligned}
\overline{\Delta}T'_{m,n} = \mu'_{mn}(1)\Bigg\{ & \frac{P'_{m-1}P_mQ'_{n-1}Q_n}{p_mq_n\mu_{mn}(k)}Y_{mn}+\frac{P'_{m-1}P_m}{p_m}\sum_{s=1}^{n-1}\frac{\left(Q'_{s-1}Q_s-Q'_sQ_{s-1}\right)Y_{ms}}{q_s\mu_{ms}(k)}\\
&+\frac{Q'_{n-1}Q_n}{q_n}\sum_{r=1}^{m-1}\frac{\left(P'_{r-1}P_r-P'_rP_{r-1}\right)Y_{rn}}{p_r\mu_{rn}(k)}\\
&+\sum_{r,s=1}^{m-1,n-1}\frac{\left\{P'_rP_{r-1}\left(Q'_sQ_{s-1}-Q'_{s-1}Q_s\right)-P'_{r-1}P_r\left(Q'_sQ_{s-1}-Q'_{s-1}Q_s\right)\right\}Y_{rs}}{p_rq_s\mu_{rs}(k)}\Bigg\}
\end{aligned}$$

Therefore it can be written that

$$\overline{\Delta}T'_{m,n}=\sum_{r,s=0}^{m,n}a_{mnrs}Y_{rs},=A_{mn}(Y)$$

where the matrix $A=(a_{mnrs})$ is given by

$$a_{mnrs}=\begin{cases}\frac{\mu'_{m0}(1)P'_{m-1}P_m}{p_m\mu_{m0}(k)}, & r=m,\ n=s=0\\ \frac{\mu'_{m0}(1)\left(P'_{r-1}P_r-P'_rP_{r-1}\right)}{p_r\mu_{r0}(k)}, & 1\leq r<m,\ n=s=0\\ \frac{\mu'_{0n}(1)Q'_{n-1}Q_n}{q_n\mu_{0n}(k)}, & s=n,\ m=r=0\\ \frac{\mu'_{0n}(1)\left(Q'_{s-1}Q_s-Q'_sQ_{s-1}\right)}{q_s\mu_{0s}(k)}, & 1\leq s<n, m=r=0\\ \frac{\mu'_{mn}(1)P'_{m-1}P_m\left(Q'_{s-1}Q_s-Q'_sQ_{s-1}\right)}{p_mq_s\mu_{ms}(k)}, & 1\leq s<n, m\geq 1\\ \frac{\mu'_{mn}(1)Q'_{n-1}Q_n\left(P'_{r-1}P_r-P'_rP_{r-1}\right)Y_{rn}}{q_np_r\mu_{rn}(k)}, & 1\leq r<m, n\geq 1\\ \frac{\mu'_{mn}(1)\left\{P'_rP_{r-1}\left(Q'_sQ_{s-1}-Q'_{s-1}Q_s\right)-P'_{r-1}P_r\left(Q'_sQ_{s-1}-Q'_{s-1}Q_s\right)\right\}}{p_rq_s\mu_{rs}(k)}, & 1\leq s<n, 1\leq r<m\\ \frac{\mu'_{mn}(1)P'_{m-1}P_mQ'_{n-1}Q_n}{p_mq_n\mu_{mn}(k)}, & s=n,\ r=m,\\ 0, & \text{otherwise}\end{cases}$$

This gives that $\left|\overline{N},p_m,q_n\right|_k\Rightarrow\left|\overline{N},p'_m,q'_n\right|$ if and only if $A\in(\mathcal{L}_k,\mathcal{L})$. But, it follows from the definition of the matrix that

$$\begin{aligned}W_{k^*}(A) &= \sum_{r,s=0}^{\infty}\left(\sum_{m,n=0}^{\infty}|a_{mnrs}|\right)^{k^*}\geq\sum_{r=0}^{\infty}|a_{r0r0}|^{k^*}\\ &= \sum_{r=0}^{\infty}\left|\left(\frac{p'_rP_r}{P'_rp_r}\right)\left(\frac{P_r}{p_r}\right)^{1/k}P_{r-1}\right|^{k^*}\geq\sum_{r=0}^{\infty}P_{r-1}^{k^*}=\infty.\end{aligned}$$

Therefore, the proof is completed by Theorem 3.2.

References

Başar, F. & **Sever, Y. (2009).** The space $\mathcal{L}_q$ of double sequences, Mathematical Journal of Okayama University, 51: 149–157.

Bor, H. (2016). Some equivalence theorems on absolute summability methods, Acta Mathematica Hungarica, 149 (1): 208–214.

Bor, H. & **Thorpe, B. (1987).** On some absolute summability methods, Analysis, 7: 145-152.

Borwein, D. & **Cass, F.T. (1968).** On strong Nörlund summability, Mathematische Zeitschrift, 103: 94-111.

Bosanquet, L.S. (1950). Mathematical Reviews, 11: 654.

Das, G. Srivastava, V.P. & **Mohapatra, R.N. (1967).** On absolute summability factors of infinite series, Journal of the Indian Mathematical Society, 31: 189-200

Flett, T.M. (1957). On an extension of absolute summability and theorems of Littlewood and Paley, Proceedings of the London Mathematical Society, 7: 113-141.

Hardy, G.H. (1949). Divergent Series, Clarendon Press, Oxford.

Güleç, G.C.H. (2019). Summability factor relations between absolute weighted and Cesàro means, Mathematical Methods in the Applied Sciences, 42: 5398-5402.

Mazhar, S.M. (1972). On the absolute Nörlund summability factors of infinite series, Proceedings of the American Mathematical Society, 32: 232-236.

Mishra, L.N. Das, P.K., Samanta, P., Misra, M. and Misra, U.K. (2018). On Indexed Absolute Matrix Summability of an Infinite Series, Applications and Applied Mathematics, 13: 274-285.

Mohapatra R. N.(1967). A note on summability factors, Journal of the Indian Mathematical Society, 31: 213-224.

Rhoades, B.R. (1998). Absolute comparison theorems for double weighted mean and double Cesàro means, Mathematica Slovaca 48: 285-291.

Rhoades, B.R. (1999). Inclusion theorems for absolute matrix summability methods, Journal of Mathematical Analysis and Application, 238: 82-90.

Rhoades, B.R. (2003). On absolute normal double matrix summability methods, Glasnik Matematicki, 38 (58): 57- 73.

Sarıgöl, M.A. (1991). Necessary and sufficient condition for the equivalence of the summability methods $\left|\overline{N}, p_n\right|_k$ and $|C,1|_k$, Indian Journal of Pure and Applied Mathematics, 22: 483-489.

Sarıgöl, M.A.(1992). On absolute weighted mean summability methods, Proceedings of the American Mathematical Society, 115: 157-160.

Sarıgöl, M.A.(1993). A note summability, Studia Scientiarum Mathematicarum Hungarica, 28: 395-401.

Sarıgöl, M.A. & **Bor, H. (1995).** Characterization of absolute summability factors, Journal of Mathematical Analysis and Application, 195: 537-545.

Sarıgöl, M.A. (2021). On absolute weighted mean summability methods, Quaestiones Mathematicae, 44: 755-764.

Sarıgöl, M.A. & **Mursaleen, M. (2021).** Almost absolute weighted summability with index k and matrix transformations, Journal of Inequalities and Applications, 2021:108.

Sunouchi, G. (1949). Notes on Fourier analysis, XVIII, Absolute summability of series with constant terms, Tohoku Mathematical Journal, (2)1: 57–65.

Thorpe, B. (1972). An Inclusion theorem and consistency of real regular Nörlund methods of summability, Journal of the London Mathematical Society, 2-5, , 519–525.

Zaanen, A.C. (1953). Linear Analysis, Amsterdam.

Zeltser, M. (2001). Investigation of double sequence spaces by soft and hard analytical methods, Dissertationes Mathematicae Universitatis Tartuensis 25, Tartu University Press, Univ. of Tartu, Faculty of Mathematics and Computer Science, Tartu.

Zraiqat, A. (2019). Inclusion and equivalence relations between absolute Nörlund and absolute weighted mean summability methods, Boletim da Sociedade Paranaense de Matemática, 37: 103–117.

Submitted: 18/12/2021
Revised: 11/03/2022
Accepted: 15/03/2022
DOI: 10.48129/kjs.17649

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-14)

Minimum total irregularity index of tricyclic graphs

Hassan Ahmed*, Akhlaq Ahmad Bhatti
Dept. of Sciences and Humanities
National University of Computer and Emerging Sciences,
b-block, Faisal Town, Lahore, Pakistan
**Corresponding author: hassanms5664@gmail.com*

Abstract

The quantitative characterization of the topological structures of irregular graphs has been demonstrated through several irregularity measures. In the literature, not only different chemical and physical properties can be well comprehended but also quantitative structure-activity relationship (QSPR) and quantitative structure-property relationship (QSAR) are documented through these measures. A simple graph $G = (V, E)$ is a collection of V and E as vertex and edge sets respectively, with no multiple edges or loops. Keeping in view the importance of various irregularity measures, in (Abdo *et al.*, 2014a) the authors defined the total irregularity of a simple graph $G = G(V, E)$ as

$$irr_t(G) = \frac{1}{2}\sum_{u,v\in V} |d_G(u) - d_G(v)|,$$

where $d_G(u)$ indicates the degree of the vertex u, where $u \in V(G)$. In this paper, we have determined the first minimum, second minimum and third minimum total irregularity index of the tricyclic graphs on the n vertices.

Keywords: Irregularity; topological index; total irregularity index; λ-transformation; tricyclic graphs.

1. Introduction

Let $G = (V, E)$ be a graph with edge and vertex sets as denoted by E and V respectively. The number of edges attached on a vertex v of a graph G is the degree $d_G(v)$ of vertex v. If $V = \{v_i\}_{i=1}^n$, then sequence $(d_1, d_2, d_3, \ldots d_n)$ is called degree sequence of G (Bondy & Murty, 1976), where d_i is the degree of i^{th} vertex of G. We assume the sequence $(d_G(v_i))_{i=1}^n$ is in decreasing order $i.e.$ for $i < z$, $(d(v_z) \leq d(v_i))$. For convenience, we will use $\mathcal{DS}$ as the notation for degree sequence of a graph G.

With recent advances in graph theory in different areas, chemical graph theory is one of the most active area of research. Chemical graph theory or the theory of chemical graphs is a sub-branch of mathematical chemistry that describes non-trivial graph theory applications for solving molecular problems where the chemical structure is transformed into a mathematical structure. A representation of an object only provides information on the number of elements it

comprises, and its connectivity is defined as the graph's topological representation.
A topological index is a numerical value that is used primarily for predicting chemical and physical properties of various compounds and structures. A molecular graph is called a topological representation of a molecule. Significant number of topological indices during the last two decades have been documented. Many existing topological indices based on degrees can be classified as BID index, whose general form is

$$BID(G) = \sum_{uv \in E} f(d_u, d_v), \tag{1}$$

where uv is the edge connecting vertices u and v of the graph. There are numerous indices introduced such as the ABC index, Zagreb index, Randic index, etc. Some information can be found in the articles ((Akbar & Akhlaq, 2016), (Akbar & Akhlaq, 2017), (Hassan *et al.*, 2019) cited therein. Currently, the study of such types of indices has become a very active research area in the theory of chemical graphs. One such area is the quantitative analysis of different topological structures of irregular graphs.
The graph that has the same degree of all its vertices is *regular*, otherwise, it is *irregular*. Several approaches have been proposed which characterize the irregularity of a graph. Albertson in (Albertson, 1997) introduced $|d_G(u) - d_G(v)|$ as an imbalance of an edge $e = uv \in E$ and defined

$$irr(G) = \sum_{uv \in E} |d_G(u) - d_G(v)| \tag{2}$$

as an irregularity of a graph G. More results about the above-mentioned concepts are mentioned in ((Dimitrov & Skrekovski, 2015), (Abdo *et al.*, 2014b), (L.H. You *et al.*, 2014a), (L.H. You *et al.*, 2014b), (Henning & Rautenbach, 2007), (Albertson, 1997), (Hensen & $M\acute{e}lot$, 2005)). Taking inspiration from the structure and significance of Equation 2, a new irregularity measure was introduced by the authors in (Abdo *et al.*, 2014a) termed the total irregularity index, defined as

$$irr_t(G) = \frac{1}{2} \sum_{u,v \in V} |d_G(u) - d_G(v)| \tag{3}$$

Even though both graph invariants compute irregularity, the irregularity is captured by one parameter, i.e. the vertex degree, but in some respects the later is preferable to the old one. For instance, equation (3) has the known characteristic of an irregularity computation that the graphs with identical total irregularity have the same $\mathcal{DS}$, whereas equation (2) does not possess this property. Clearly, equation (3) is an upper bound of equation (2). In (Dimitrov & Skrekovski, 2015), the relationship between $irr(G)$ and $irr_t(G)$ for the connected graph on n vertices have been derived, that is, $irr_t(G) \leq n^2\{\frac{irr(G)}{4}\}$. Furthermore, for any tree, they also computed that $irr_t(T) \leq (n-2)irr(T)$. In (Abdo *et al.*, 2014a) the bounds on $irr_t(G)$ on cycle, path, and the star graph, denoted as C_n, P_n, and S_n, on the n vertices respectively, were computed. They also proved that the graph with maximal total irregularity on n vertices between all the trees is the star graph. Following result is due to (Abdo *et al.*, 2014a).

Theorem 1.1. *Let G be an n-vertex simple and undirected graph. Then*

(i) $irr_t(G) \leq (2n^3 - 3n^2 - 2n + 3)$.

(ii) $irr_t(G) \leq (n-1)(n-2)$ *if G is a tree, with equality iff* $G \cong S_n$.

The authors in (L.H. You *et al.*, 2014a) and (Hensen & $M\acute{e}lot$, 2005) examined the total irregularity of the unicyclic and bicyclic graphs and defined graphs with $n^2 - n - 6$ as maximum total irregularity among all the unicyclic graphs and graphs with $n^2 + n - 16$ as maximum total irregularity among all bicyclic graphs on n vertices respectively. By using the Gini index in (M. Eliasi, 2015), the author obtained the ordering of the total irregularity index for some classes of connected graphs, with the same number of vertices. Recently, the authors in (F. Gao *et al.*, 2021) characterized trees T of order n and triangulation graphs with respect to difference of Mostar index and irregularity of graphs. For more related research, readers are requested to see (Xu & Das, 2016).
In Section 2, we have described an important transformation in the current note to examine the minimum total irregularity of tricyclic graphs. We have also determined first, second and third minimum total irregularity of tricyclic graphs on n vertices in Section 3. Lastly, summary of the note is mentioned in Section 4.

2. λ-Transformation

An important transformation in this section is explained to explore the minimum total irregularity of graphs. Before introduction of transformation, let us define induced subgraph and hanging tree (Yingxue Zhu *et al.*, 2014).
Let G be an n-vertex graph then a subset of the vertices of G having edges incident on the vertices in the subset as endpoints is known as vertex-induced or simply induced subgraph of G. Let T be induced sub-tree of G, if G can be obtained back by connecting T to a vertex of $G \setminus T$. Then T is a hanging tree of G. Now we introduce the λ-Transformation as:
λ-Transformation: Let G be a simple graph with at least two leaves. Let u be a vertex of $d_G(u) \geq 3$ and T be hanging tree of G connecting to u with $|V(T)| \geq 1$, and v be the leaf of G with $v \notin T$. By removing T from u and connecting it to the vertex v and the graph obtained be denoted as G^*. Then this transformation from vertex u to v is a λ-transformation on G (see Figure 1).

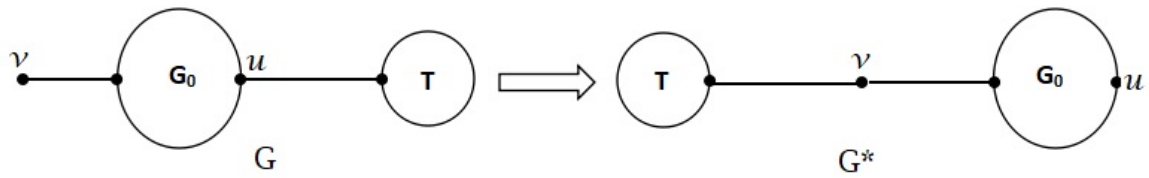


Fig. 1. G and G^*(obtained from λ-Transformation)

The following result is due to (Yingxue Zhu *et al.*, 2014), after λ-Transformation and it will be used in the main results as it will help us to compute total irregularity index of tricyclic graphs.

Lemma 2.1. *(Yingxue Zhu et al., 2014) Let G be an $n-vertex$ graph then $irr_t(G) > irr_t(G^*)$, where G^* is the graph obtained from G, after λ-Transformation from u to v.*

Proof. Let $G = (V, E)$, consider the vertex set $V = V^1 \cup V^2 \cup V^3$ such that

$$V^1 = \{x | d_G(x) \geq d_G(u), x \in V\}$$

$$V^2 = \{x | d_G(x) = 1, x \in V\}$$

$$V^3 = \{x | 2 \leq d_G(x) < d_G(u), x \in V\}$$

Clearly, $u \in V^1$, $v \in V^2$. Let $|V^1| = j$, $|V^2| = k$, $|V^3| = l$, then $j \geq 1, k \geq 2$ and $j+k+l = n$. Note by λ-transformation, the degrees of v and u become $d_{G^*}(v) = d_G(v) + 1 = 2$, $d_{G^*}(u) = d_G(u) - 1$ and $d_{G^*}(w) = d_G(w)$ for any $w \in V \setminus \{u, v\}$. Let $U = V \setminus \{u, v\}$. Then

$$|d_{G^*}(u) - d_{G^*}(v)| - |d_G(u) - d_G(v)| = -2,$$

$$\sum_{w \in U} (|d_{G^*}(u) - d_{G^*}(w)| - |d_G(u) - d_G(w)|) = (j-1) - (l+k-1) = j - l - k,$$

$$\sum_{w \in U} (|d_{G^*}(v) - d_{G^*}(w)| - |d_G(v) - d_G(w)|) = -(j-1) - l + (k-1)$$
$$= -j - l + k.$$

Thus, we have $irr_t(G^*) - irr_t(G) = -2 + (j - l - k) + (-j - l + k) = -2l - 2 < 0$.

Remark. *Let λ-transformation be performed on G from the vertex u to v and G^* be the resulting graph. Then by λ-transformation and Lemma 2.1, we have $d_{G^*}(u) = d_G(u) - 1 \geq 2$ and $d_{G^*}(v) = d_G(v) + 1 = 2$. If $d_{G^*}(u) \geq 3$, G^* has at least two leaves, and there's a hanging tree of G^* connecting to vertex u, we can repeat λ-transformation from vertex u on G^*, till the degree of u equals 2, or the resulting graph consists of just one leaf, or no hanging tree connects to vertex u.*

We can see from the above arguments that λ-transformation can be achieved on G iff three conditions hold mentioned below:

(i) There exists a vertex u with degree greater or equal to 3;

(ii) There is a hanging tree of G, connecting to vertex u;

(iii) G has at least two leaves.

Following trivial result will be useful to establish our main results.

Lemma 2.2. *((Bondy & Murty, 1976)) Let $G = (V, E)$ be a graph and $|E| = m$. Then $\sum_{v \in V} d_G(v) = 2m$.*

■

In the following section, we establish the main results by describing different classes in tricyclic graphs on n vertices.

3. The Total Irregularity of Tricyclic Graphs

A connected (n, m) graph G is said to be a tricyclic graph if $m = n + 2$. Within this section, the extremal graphs are described by computing, the first, second and third minimum total irregularity of n-vertex tricyclic graphs.
Tricyclic graphs can be divided into three types: $\xi - graph$, $\Omega - graph$, and $\vartheta - graph$.
The class of $\xi - graph$, denoted by $\xi(p, q, r, s, i)$ contains three types of tricyclic graphs (see Figure 2). The first one is obtained from three cycles C_p, C_q, and C_r having one common vertex (say u), between C_p and C_q, and one (say v), between C_q and C_r (i.e. having no paths between

the cycles see Figure 2(a)). It is denoted by $\xi_1(p,q,r,s,i) = \xi_1$. The second is obtained having one common vertex u between C_p and C_q a path between C_q and C_r to any vertex $w \in V \setminus u$ (see Figure 2(b). It is denoted by $\xi_2(p,q,r,s,i) = \xi_2$. Lastly, third is obtained by attaching two disjoint paths P_s and P_i between C_p and C_q and one between C_q and C_r respectively (see Figure 2(c)), where $p,q,r \geq 3$. It is denoted by $\xi_3(p,q,r,s,i) = \xi_3$.

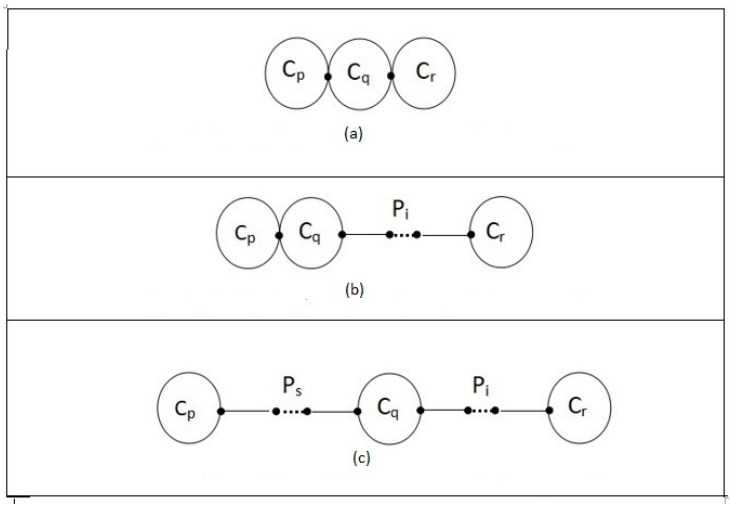


Fig. 2. Tricyclic graphs: (a) $\xi_1(p,q,r,s,i)$; (b) $\xi_2(p,q,r,s,i)$; (c) $\xi_3(p,q,r,s,i)$

An $\Omega - graph$ denoted by $\Omega(p,q,r,s,i,y)$, contains four types of tricyclic graphs (see Figure 3 and 4). The first graph, denoted by $\Omega_1 = \Omega_1(p,q,r,s,i,y)$, with only one common vertex, (say u), attached to C_p, C_q and C_r (see Figure 3(a)). The second graph, denoted by $\Omega_2 = \Omega_2(p,q,r,s,i,y)$ is obtained from Ω_1 by attaching a path P_y of length $y \geq 1$ between vertex u and C_r (see Figure 3(b)). The third graph, denoted by $\Omega_3 = \Omega_3(p,q,r,s,i,y)$, obtained from Ω_2 by attaching a path P_i of length $i \geq 1$ between vertex u and C_q (see Figure 4(a)). Lastly, the fourth graph, denoted by $\Omega_4 = \Omega_4(p,q,r,s,i,y)$ is obtained from Ω_3 by attaching a path P_s of length $s \geq 1$ between vertex u and C_p (see Figure 4(b)), where $p,q,r \geq 3$.

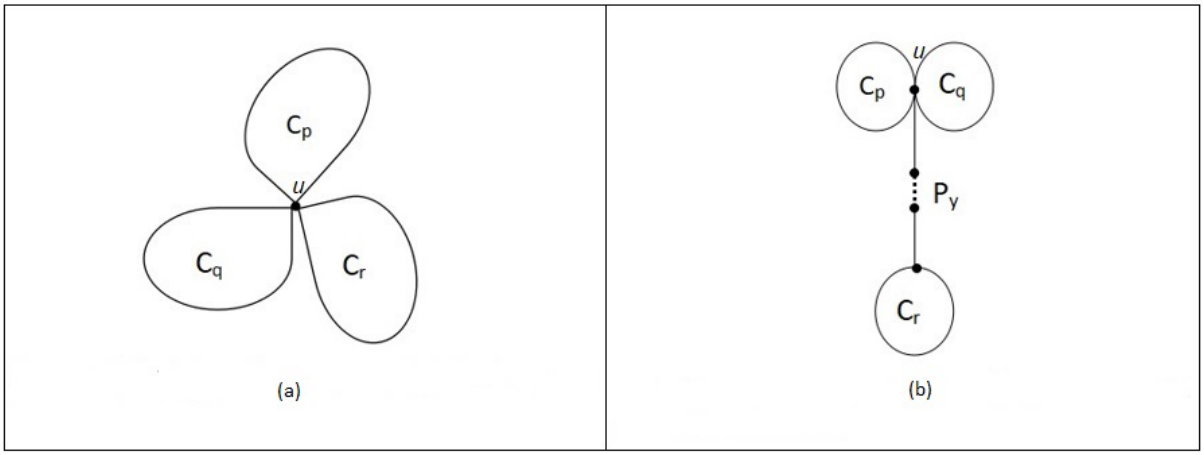


Fig. 3. Tricyclic graphs: (a) Ω_1; (b) Ω_2

A $\vartheta - graph$, denoted by $\vartheta(p,q,r,s,i)$ contains four types of tricyclic graphs (see Figure 5). The first graph, denoted by $\vartheta_1 = \vartheta_1(p,q,r,s,i)$, is a graph with three cycles (namely, C_p, C_q, C_r) on $p+q+r-s-i$ vertices, having $(s+i)$ vertices as common with each other (see Figure 5(a)). In the second case, the graph denoted by $\vartheta_2 = \vartheta_2(p,q,r,s,i)$, is obtained

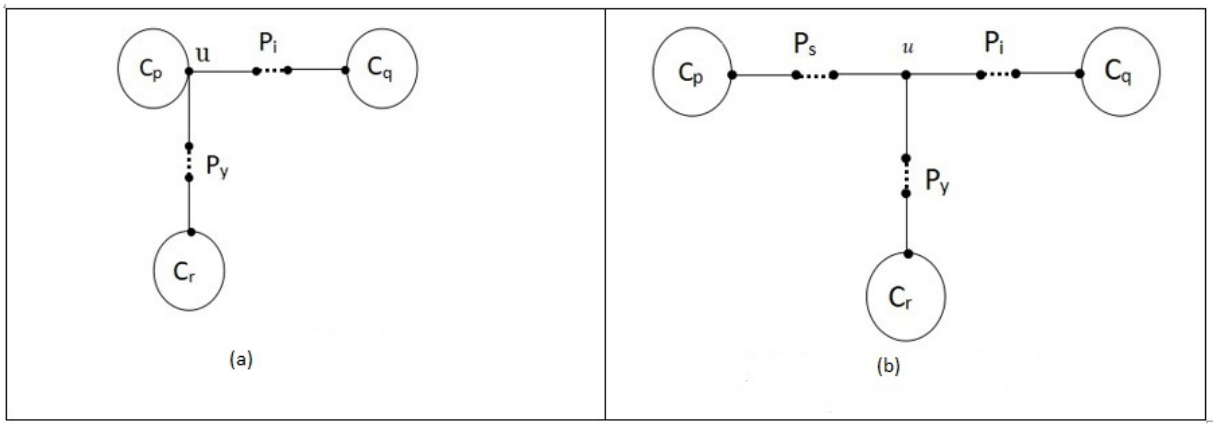


Fig. 4. Tricyclic graphs: (a) Ω_3; (b) Ω_4

from ϑ_1 by removing C_r from C_q and attaching it to one of the end vertices $\{f_1, f_s\}$ (see Figure 5(b)). In the third case, the graph is obtained from ϑ_1 by attaching a path P_{r-i} from one of the end vertices $\{e_1, e_{p-s}, h_1, h_i\}$ with a vertex of disjoint cycle C_r(see Figure 5(c)), let it be denoted by $\vartheta_3 = \vartheta_3(p, q, r, s, i)$. Lastly, the graph denoted by $\vartheta_4 = \vartheta_4(p, q, r, s, i)$ is obtained by attaching a path between the cycle C_r and one of the end vertices $\{f_1, f_s\}$ (see Figure 5(d)), where $p, q, r \geq 3$ and $s, i \geq 2$.

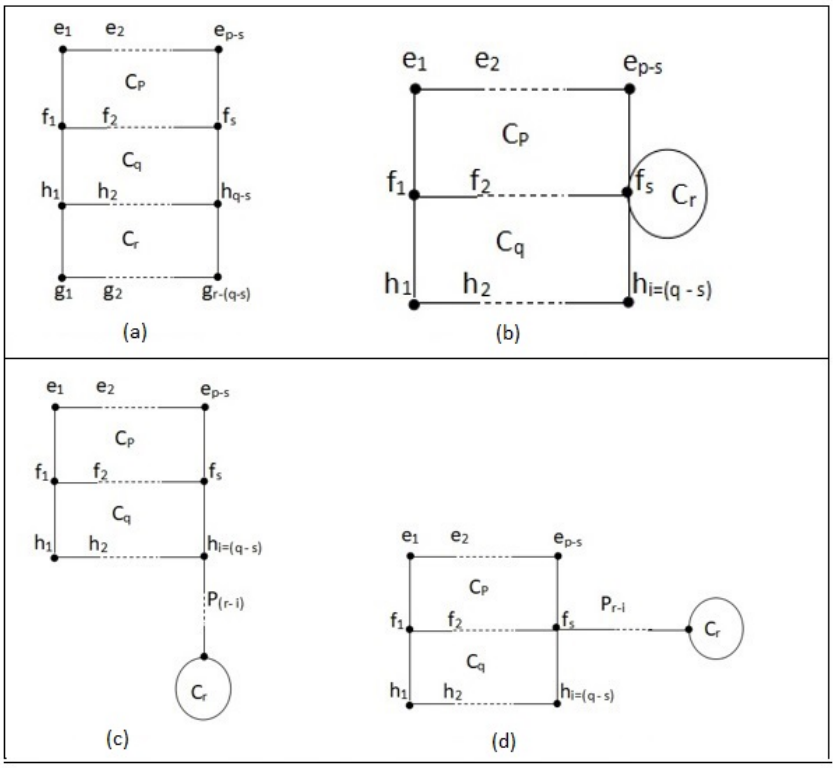


Fig. 5. Tricyclic graphs: (a) ϑ_1; (b) ϑ_2; (c) ϑ_3; (d) ϑ_4;

Let the set of all tricyclic graphs on n vertices be denoted by $\mathcal{T}_n$. As defined above $\mathcal{T}_n$ is based on three types of graphs $\xi - graph$, $\Omega - graphs$, and $\vartheta - graph$.

3.1. Graphs having minimum total irregularity in $\xi(p, q, r, s, i)$

In this section, we determine the minimum total irregularity of tricyclic graphs in $\xi(p, q, r, s, i)$. Let $\xi_1 = \xi_1(p, q, r, s, i)$ having no paths (see Figure 2(a)), $\xi_2 = \xi_2(p, q, r, s, i)$ with a one path P_i with length $i \geq 1$ (see Figure 2(b)) and $\xi_3 = \xi_3(p, q, r, s, i)$ with two paths P_s and P_i with lengths $s, i \geq 1$ respectively (see Figure 2(c)).

Theorem 3.1. *Let $n \geq 7$, $G \in \xi_1 = \xi_1(p, q, r, s, i)$ then*

(i) $irr_t(G) \geq 4n - 8$ and equality holds iff $(4, 4, 2, 2, \ldots, 2)$ is the $\mathcal{DS}$ of G.

(ii) If $(4, 4, 2, 2, \ldots, 2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n - 14$, with equality iff the $\mathcal{DS}$ of G is $(4, 4, 3, 2, 2, \ldots, 2, 1)$.

Proof. We know that $\sum_{v \in V} d_G(v) = 2(n + 2)$ from Lemma 2.2. Let us divide the vertex set as follows,

$$j = |\{x | d_G(x) \geq 3, x \in V\}|,$$

$$k = |\{x | d_G(x) = 1, x \in V\}|,$$

$$t = |\{x | d_G(x) = \Delta_G, x \in V\}|.$$

Since $G \in \xi_1 = \xi_1(p, q, r, s, i)$, then $j \geq 2, k \geq 0, 1 \leq t \leq j$ and $\Delta_G \geq 4$. Note $G \in \xi_1$ if $j = 2, \Delta_G \geq 5$ or $j \geq 3$ so vertex u with $d_G(u) \geq 3$ exists and hanging tree of G which connects to u exists. We complete the proof by considering following cases:

Case 1. If $j = 2$, then there are three subcases mentioned below:

Subcase (i): If $\Delta_G = 4$, then $k = 0$ and the $\mathcal{DS}$ is $(4, 4, 2, 2, \ldots, 2)$ as $2(n + 2) = \sum_{v \in V} d_G(v) = 8 + 2(n - 2 - k) + k$, then $irr_t(G) = 4n - 8$.

Subcase (ii): If $\Delta_G = 5$, then $k = 1$ and the $\mathcal{DS}$ is $(5, 4, 2, 2, \ldots, 2, 1)$ as $2(n + 2) = \sum_{v \in V} d_G(v) = 5 + 4 + 2(n - 2 - k) + k$, then $irr_t(G) = 6n - 10 > 6n - 14$.

Subcase (iii): If $\Delta_G \geq 6$, then $k \geq \Delta_G - 4 \geq 2$ as $2(n + 2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 4 + 2(n - 2 - k) + k$ and λ-transformation can be done $(k - 1)-$ times on G till the $\mathcal{DS}$ of the graph obtained becomes $(5, 4, 2, 2, \ldots, 2, 1)$. Let the graph obtained be denoted as F_1, then $irr_t(G) > irr_t(F_1) = 6n - 10 > 6n - 14$ by Lemma 2.1.

Case 2. Now if $j \geq 3$, then consider following subcases:

Subcase (i): If $j + \Delta_G = 7$, then $j = 3, \Delta_G = 4, 2 \leq t \leq 3$.

If $t = 2$, then $k = 1$ and the $\mathcal{DS}$ is $(4, 4, 3, 2, 2, \ldots, 2, 1)$ as $2(n + 2) = \sum_{v \in V} d_G(v) = 4 + 4 + 3 + 2(n - 3 - k) + k = 11 + 2(n - 3 - k) + k$, so $irr_t(G) = 6n - 14$.

If $t = 3$, then $k = 2$ as $2(n + 2) = \sum_{v \in V} d_G(v) = 4t + 2(n - 3 - k) + k$, and λ-transformation can be done once on G so the $\mathcal{DS}$ of obtained graph is $(4, 4, 3, 2, 2, \ldots, 2, 1)$. Let the obtained graph be denoted as F_2, then $irr_t(G) > irr_t(F_2) = 6n - 14$ by Lemma 2.1.

Subcase (ii): If $j + \Delta_G \geq 8$, then $k \geq \Delta_G + j - 6 \geq 2$ as $2(n + 2) = \sum_{v \in V} d_G(v) \geq$

$\Delta_G+3(j-1)+2(n-j-k)+k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(4,4,3,2,2,\ldots,2,1)$. Let the obtained graph be denoted as F_3, then $irr_t(G) > irr_t(F_3) = 6n-14$ by Lemma 2.1.

□

Theorem 3.2. *Let $n \geq 8$, $G \in \xi_2 = \xi_2(p,q,r,s,i)$ then*

(i) $irr_t(G) \geq 4n-10$ and equality holds iff $(4,3,3,2,2,\ldots,2)$ is the $\mathcal{DS}$ of G.

(ii) If $(4,3,3,2,2,\ldots,2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n-18$, with equality iff the $\mathcal{DS}$ of G is $(4,3,3,3,2,2,\ldots,2,1)$.

Proof. It is easy to see that $\sum_{v\in V} d_G(v) = 2(n+2)$ from Lemma 2.2.
Let us divide the vertex set as,

$$j = |\{x|d_G(x) \geq 3, x \in V\}|,$$

$$k = |\{x|d_G(x) = 1, x \in V\}|,$$

$$t = |\{x|d_G(x) = \Delta_G, x \in V\}|.$$

Since $G \in \xi_2 = \xi_2(p,q,r,s,i)$ then $j \geq 3,\ k \geq 0, 1 \leq t \leq j$ and $\Delta_G \geq 4$.
Note $G \in \xi_2$ if $j = 3,\ \Delta_G \geq 4$ or $j \geq 4$ so there exists a vertex u with $d_G(u) \geq 3$ and there exists a hanging tree of G which connects to u. We complete the proof by considering following cases:

Case 1. If $j = 3$, then consider following subcases:
Subcase (i): If $\Delta_G = 4$, then $k = 0$ and the $\mathcal{DS}$ is $(4,3,3,2,2,\ldots,2)$ as $2(n+2) = \sum_{v\in V} d_G(v) = 4+3+3+2(n-3-k)+k$, then $irr_t(G) = 4n-10$.
Subcase (ii): If $\Delta_G = 5$, then $1 \leq t \leq 3$
If $t = 1$, then $k = 1$ and $k = 2$. For $k = 1$ the $\mathcal{DS}$ is $(5,3,3,2,2,\ldots,2,1)$ as $2(n+2) = \sum_{v\in V} d_G(v) \geq \Delta_G+3+3+2(n-3-k)+k$ and $irr_t(G) = 6n-12 > 6n-18$. For $k = 2$ λ-transformation can be done on G once and the $\mathcal{DS}$ of the graph obtained becomes $(5,3,3,2,2,\ldots,2,1)$. Let the obtained graph denoted by F_4, then $irr_t(G) > irr_t(F_4) = 6n-12 > 6n-18$ from Lemma 2.1.
If $t \geq 2$, then $k \geq 3$ as $2(n+2) = \sum_{v\in V} d_G(v) \geq 5+5+3+2(n-3-k)+k$ λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph becomes $(5,3,3,2,2,\ldots,2,1)$. Let the obtained graph denoted by F_5, then $irr_t(G) > irr_t(F_5) = 6n-12 > 6n-18$ by Lemma 2.1.
Subcase (iii): If $\Delta_G \geq 6$, then $k \geq \Delta_G+j-7 \geq 2$ as $2(n+2) = \sum_{v\in V} d_G(v) \geq \Delta_G+3(j-1)+2(n-j-k)+k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph is $(5,4,2,2,\ldots,2,1)$. Let the obtained graph be denoted as F_6, then $irr_t(G) > irr_t(F_6) = 6n-10 > 6n-14$ by Lemma 2.1.
Case 2. If $j \geq 4$, then consider following subcases:
Subcase (i): If $j+\Delta_G = 8$, then $k = 1$, and the $\mathcal{DS}$ of G is $(4,3,3,3,2,2,\ldots,2,1)$ as $2(n+2) = \sum_{v\in V} d_G(v) \geq \Delta_G+3(j-1)+2(n-j-k)+k$, then $irr_t(G) = 6n-18$.

Subcase (ii): If $j + \Delta_G \geq 9$, then $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph is $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$. Let the obtained graph be denoted as F_7, then $irr_t(G) > irr_t(F_7) = 6n - 18$ by Lemma 2.1.

□

Theorem 3.3. *Let $n \geq 9$, $G \in \xi_3 = \xi_3(p, q, r, s, i)$ then*

(i) $irr_t(G) \geq 4n - 16$ and equality holds iff $(3, 3, 3, 3, 2, 2, \ldots, 2)$ is the $\mathcal{DS}$ of G.

(ii) If $(3, 3, 3, 3, 2, 2, \ldots, 2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n - 26$, with equality iff the $\mathcal{DS}$ of G is $(3, 3, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

Proof. It is easy to see that $\sum_{v \in V} d_G(v) = 2(n+2)$ from Lemma 2.2.
Let us divide vertex set as below,

$$j = |\{x | d_G(x) \geq 3, x \in V\}|,$$

$$k = |\{x | d_G(x) = 1, x \in V\}|,$$

$$t = |\{x | d_G(x) = \Delta_G, x \in V\}|.$$

Since $G \in \xi_3 = \xi_3(p, q, r, s, i)$ then $j \geq 4,\ k \geq 0, 1 \leq t \leq j$ and $\Delta_G \geq 3$.
Note $G \in \xi_3 = \xi_3(p, q, r, s, i)$ if $j = 4,\ \Delta_G \geq 3$ or $j \geq 5$ so there exists a vertex u with $d_G(u) \geq 3$ and there exists hanging tree of G which connects to u. We have completed the proof by considering the following cases:
Case 1. If $j = 4$, then consider following subcases:
Subcase (i): If $\Delta_G = 3$, then $k = 0$ and the $\mathcal{DS}$ is $(3, 3, 3, 3, 2, 2, \ldots, 2)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, then $irr_t(G) = 4n - 16$.
Subcase (ii): If $\Delta_G = 4$, then $1 \leq t \leq 4$.
If $t = 1$, then $k = 1$. For $k = 1$ the $\mathcal{DS}$ is $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and $irr_t(G) = 6n - 18 > 6n - 26$.
If $t \geq 2$, then $k \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph becomes $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$. Let the obtained graph denoted by F_8, thus $irr_t(G) > irr_t(F_8) = 6n - 18 > 6n - 26$ by Lemma 2.1.
Subcase (iii): If $\Delta_G \geq 5$,
then $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph is $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$. Let the obtained graph be denoted as F_9, thus $irr_t(G) > irr_t(F_9) = 6n - 18 > 6n - 26$ by Lemma 2.1.
Case 2. If $j \geq 5$, then consider the following subcases:
Subcase (i): If $j + \Delta_G = 8$, then $k = 1$, and the $\mathcal{DS}$ of G is $(3, 3, 3, 3, 3, 2, 2, \ldots, 2, 1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, then $irr_t(G) = 6n - 26$.

Subcase (ii): If $j + \Delta_G \geq 9$, then $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of obtained graph is $(3,3,3,3,3,2,2,\ldots,2,1)$. Let the graph obtained be denoted as F_{10}, then $irr_t(G) > irr_t(F_{10}) = 6n - 26$ by Lemma 2.1.

□

3.2. The graphs with minimum total irregularity in $\Omega - graph$

In this section, we determine the first minimum, second minimum, and third minimum total irregularity of tricyclic graphs in $\Omega(p,q,r,s,i,y)$.

Theorem 3.4. *Let $n \geq 7$, $G \in \Omega_1 = \Omega_1(p,q,r,s,i,y)$ then*

(i) $irr_t(G) \geq 4n-4$ and equality holds iff $(6,2,2,\ldots,2)$ is the $\mathcal{DS}$ of G.

(ii) If $(6,2,2,\ldots,2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n-8$, with equality iff the $\mathcal{DS}$ of G is $(6,3,2,2,\ldots,2,1)$.

Proof. It is obvious that $\sum_{v \in V} d_G(v) = 2(n+2)$ from Lemma 2.2.
Let us consider the vertex set as,

$$j = |\{x | d_G(x) \geq 3, x \in V\}|,$$

$$k = |\{x | d_G(x) = 1, x \in V\}|,$$

$$t = |\{x | d_G(x) = \Delta_G, x \in V\}|.$$

Since $G \in \Omega_1 = \Omega_1(p,q,r,s,i,y)$, then $j \geq 1,\ k \geq 0, 1 \leq t \leq j$ and $\Delta_G \geq 6$.
Note $G \in \Omega_1$ if $j = 1,\ \Delta_G \geq 6$ or $j \geq 2$ so there exists a vertex u with $d_G(u) \geq 3$ and there exists hanging tree of G which connects to u. We complete the proof by considering the following cases:
Case 1. If $j = 1$, then consider the following subcases:
Subcase (i): If $\Delta_G = 6$, then $k = 0$ and the $\mathcal{DS}$ is $(6,2,2,\ldots,2)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, thus $irr_t(G) = 4n-4$.
Subcase (ii): If $\Delta_G = 7$, then $k = 1$. For $k = 1$, $\mathcal{DS}$ is $(7,2,2,\ldots,2,1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and $irr_t(G) = 6n-6 > 6n-8$.
Subcase (iii): If $\Delta_G \geq 7$, then $k \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(7,2,2,\ldots,2,1)$. Let the graph obtained be denoted as F_{11}, then $irr_t(G) > irr_t(F_{11}) = 6n-6 > 6n-8$ by Lemma 2.1.
Case 2. If $j \geq 2$, then consider the following subcases:
Subcase (i): If $\Delta_G = 6$, then $1 \leq t \leq 2$,
If $t = 1$ then $1 \leq k \leq 3$,. For $k = 1$ the $\mathcal{DS}$ of G is $(6,3,2,2,\ldots,2,1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, thus $irr_t(G) = 6n-8$. For $k \geq 2$ and we can do λ-transformation $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(6,3,2,2,\ldots,2,1)$. Let the graph obtained be denoted as F_{12}, then $irr_t(G) > irr_t(F_{12}) = 6n-8$ by Lemma 2.1.

Subcase (ii): If $\Delta_G \geq 7$, then $1 \leq t \leq 2$ and $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(6, 3, 2, 2, \ldots, 2, 1)$. Let the graph obtained be denoted as F_{13}, then $irr_t(G) > irr_t(F_{13}) = 6n - 8$ by Lemma 2.1.

□

By following the same pattern as above we get the following results by direct calculations.

Theorem 3.5. *Let $n \geq 8$, $G \in \Omega_2 = \Omega_2(p, q, r, s, i, y)$ then*

(i) $irr_t(G) \geq 4n - 6$ *and equality holds iff* $(5, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(5, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 12$*, with equality iff the $\mathcal{DS}$ of G is* $(5, 3, 3, 2, 2, \ldots, 2, 1)$.

□

Theorem 3.6. *Let $n \geq 9$, $G \in \Omega_3 = \Omega_3(p, q, r, s, i, y)$ then*

(i) $irr_t(G) \geq 4n - 10$ *and equality holds iff* $(4, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(4, 3, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 18$*, with equality iff the $\mathcal{DS}$ of G is* $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

□

Theorem 3.7. *Let $n \geq 10$, $G \in \Omega_4 = \Omega_4(p, q, r, s, i, y)$*

(i) $irr_t(G) \geq 4n - 16$ *and equality holds in case* $(3, 3, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(3, 3, 3, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 12$*, with equality iff the $\mathcal{DS}$ of G is* $(3, 3, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

□

3.3. The graphs with minimum total irregularity in $\vartheta - graph$

In this section, we have determined first minimum, second minimum, and third minimum total irregularity of tricyclic graphs in $\vartheta(p, q, r, s, i)$.

Theorem 3.8. *Let $n \geq 5$, $G \in \vartheta_1 = \vartheta_1(p, q, r, s, i)$*

(i) $irr_t(G) \geq 4n - 10$ *and equality holds iff* $(4, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(4, 3, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 18$*, with equality iff the $\mathcal{DS}$ of G is* $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

Proof. We know that $\sum_{v \in V} d_G(v) = 2(n+2)$ from Lemma 2.2.
Consider the following distribution of vertex set as,

$$j = |\{x | d_G(x) \geq 3, x \in V\}|,$$

$$k = |\{x | d_G(x) = 1, x \in V\}|,$$

$$t = |\{x | d_G(x) = \Delta_G, x \in V\}|.$$

Since $G \in \vartheta_1 = \vartheta_1(p, q, r, s, i,)$ then $j \geq 3,\ k \geq 0, 1 \leq t \leq j$ and $\Delta_G \geq 4$.
Note $G \in \vartheta_1$ if $j = 3,\ \Delta_G \geq 4$ or $j \geq 4$ so there exists a vertex u with $d_G(u) \geq 3$ and there exists hanging tree of G which connects to u. We prove by considering the following cases:
Case 1. If $j = 3$, then consider the following cases:
Subcase (i): If $\Delta_G = 4$,
then $1 \leq t \leq 3$. If $t = 1$ then $k = 0$ and the $\mathcal{DS}$ is $(4, 3, 3, 2, 2, \ldots, 2)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, then $irr_t(G) = 4n - 10$.
If $t = 2$ then $k = 1$ and the $\mathcal{DS}$ is $(4, 4, 3, 2, 2, \ldots, 2, 1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, thus $irr_t(G) = 6n - 14 > 6n - 18$.
If $t = 3$ then $k = 2$ and λ-transformation can be done once on G s.t. the $\mathcal{DS}$ of graph obtained is $(4, 4, 3, 2, 2, \ldots, 2, 1)$. Let the graph obtained be denoted by F_{14}, thus $irr_t(G) \geq irr_t(F_{14}) = 6n - 14 > 6n - 18$.
Subcase (ii): If $\Delta_G = 5$, then $k = 1$. For $k = 1$ $\mathcal{DS}$ is $(5, 3, 3, 2, 2, \ldots, 2, 1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and $irr_t(G) = 6n - 12 > 6n - 18$.
Subcase (iii): If $\Delta_G \geq 6$, then $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(5, 3, 3, 2, 2, \ldots, 2, 1)$. Let the graph obtained be denoted as F_{15}, then $irr_t(G) > irr_t(F_{15}) = 6n - 12 > 6n - 18$ by Lemma 2.1.
Case 2. If $j \geq 4$, then consider the following subcases:
Subcase (i): If $j + \Delta_G = 8$, then $k = 1$. For $k = 1$ the $\mathcal{DS}$ of G is $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$, thus $irr_t(G) = 6n - 18$.
Subcase (ii): If $j + \Delta_G \geq 9$, then $k \geq \Delta_G + j - 7 \geq 2$ as $2(n+2) = \sum_{v \in V} d_G(v) \geq \Delta_G + 3(j-1) + 2(n-j-k) + k$ and λ-transformation can be done $(k-1)-$times on G till the $\mathcal{DS}$ of graph obtained is $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$. Let the graph obtained be denoted as F_{16}, thus $irr_t(G) > irr_t(F_{16}) = 6n - 18$ by Lemma 2.1.

□

Similarly, by direct calculation, we have the following results.

Theorem 3.9. *Let $n \geq 6$, $G \in \vartheta_2 = \vartheta_2(p, q, r, s, i)$ then*

(i) $irr_t(G) \geq 4n - 6$ and equality holds iff $(5, 3, 2, 2, \ldots, 2)$ is the $\mathcal{DS}$ of G.

(ii) If $(5, 3, 2, 2, \ldots, 2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n - 12$, with equality iff the $\mathcal{DS}$ of G is $(5, 3, 3, 2, 2, \ldots, 2, 1)$.

□

Theorem 3.10. *Let $n \geq 7$, $G \in \vartheta_3 = \vartheta_3(p, q, r, s, i)$ then*

(i) $irr_t(G) \geq 4n - 16$ and equality holds iff $(3, 3, 3, 3, 2, 2, \ldots, 2)$ is the $\mathcal{DS}$ of G.

(ii) If $(3, 3, 3, 3, 2, 2, \ldots, 2)$ is not the $\mathcal{DS}$ of G, then $irr_t(G) \geq 6n - 26$, with equality iff the $\mathcal{DS}$ of G is $(3, 3, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

□

Theorem 3.11. *Let $n \geq 7$, $G \in \vartheta_4 = \vartheta_4(p, q, r, s, i)$ then*

(i) $irr_t(G) \geq 4n - 10$ *and equality holds iff* $(4, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(4, 3, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 18$*, with equality iff the $\mathcal{DS}$ of G is* $(4, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

□

4. The graphs with minimum total irregularity in $\mathcal{T}_n$

By section 3 we have determined first minimum, second minimum and the third minimum total irregularity in $\mathcal{T}_n$ immediately.

Theorem 4.1. *Let $n \geq 7$, $G \in \mathcal{T}_n$ then*

(i) $irr_t(G) \geq 4n - 16$ *and equality holds iff* $(3, 3, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G.*

(ii) If $(3, 3, 3, 3, 2, 2, \ldots, 2)$ *is not the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 4n - 10$*, with equality iff the $\mathcal{DS}$ of G is* $(4, 3, 3, 3, 2, 2, \ldots, 2)$.

(iii) If neither $(3, 3, 3, 3, 2, 2, \ldots, 2)$ *nor* $(4, 3, 3, 3, 2, 2, \ldots, 2)$ *is the $\mathcal{DS}$ of G, then* $irr_t(G) \geq 6n - 26$*, with equality iff the $\mathcal{DS}$ of G is* $(3, 3, 3, 3, 2, 2, \ldots, 2, 1)$.

□

References

Abdo, H., Cohen, N., Dimitrov, D. (2014a) The total irregularity of a graph, Discrete Mathematics and Theoretical Computer Science. **16**(1), 201-206.

Abdo, H., Cohen, N., Dimitrov, D. (2014b) Bounds and computation of irregularity of a graph, arXiv:1207.4804 [cs.DM].

Albertson, M.O. (1997) The total irregularity of a graph, *Ars Combination.* **46**, 219-225.

Akbar, A., Bhatti, A. A. (2016) A note on the augmented Zagreb index of cacti with fixed number of vertices and cycles, Kuwait J. Sci. 43 (**4**) pp. 11-17.

Akbar, A., Bhatti, A. A. (2017) A note on the minimum reduced reciprocal $Randić$ index of $n - vertex$ unicyclic graphs, Kuwait J. Sci. 44 (**2**)pp. 27-33.

Bondy, J.A. and Murty, U.S. (1976) Graph Theory and its Applications, The Macmillan Press, London.

Dimitrov, D., Skrekovski, R. (2015) Comparing the irregularity and the total irregularity of graphs, Ars. Math. Contemp. **9**, 45-50.

Eliasi, M. (2015) The Maximal Total Irregularity of Some Connected Graphs, Iranian Journal of Mathematical Chem. 6 no. 2, 121-128.

Gao, F., Xu, K., Došlić, T. (2021) On the difference of Mostar index and irregularity of graphs, Bulletin of the Malaysian Mathematical Sciences Society, 44 905-926.

Hassan, A., Bhatti, A. A., Ali, A. (2019) *Zeroth-order general Randić index of cactus graphs*, AKCE International Journal of Graphs and Combinatorics 16, 182-189

Henning, M.A., Rautenbach, D. (2007) On the irregularity of bipartite graphs, *Discrete Math.* **307** 1467-1472.

Hansen, P., $Mélot$**, H., (2005)** Variable neighborhood search for extremal graphs. 9. Bounding the irregularity of a graph, DIMACS Ser. *Discrete Math. Theoret. Comput. Sci.* **69** 253-264.

Xu, K., Das, K. C. (2016) Some extremal graphs with respect to inverse degree, Discrete Appl.Math. 203 171183.

You,L.H., Yang, J.S. and You, Z.F. (2014a) The maximal total irregularity of unicyclic graphs, *Ars Comb.,* **114** 153-160.

You, L.H., Yang, J.S., Zhu, Y.X. and You, Z.F. (2014b) The maximal total irregularity of bicyclic graphs, J. Appl. Math. 1-9, DOI: 10.1155/2014/785084.

Zhu, Y., You, L., Yang, J. (2016) The minimal total irregularity of some classes of graphs, Filomat **30** no. 5, 1203-1211.

Submitted: 20/03/2021
Revised: 23/12/2021
Accepted: 29/12/2021
DOI: 10.48129/kjs.13063

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-9)

On left restriction semigroups with zero

Baddi-Ul-Zaman*
School of Mathematical Sciences, South China Normal University,
Guangzhou 510631, P. R. China
**Corresponding author: amath1141@gmail.com*

Abstract

In this article, we give the notion of left restriction meet-semigroup, and establish some results regarding atomistic left restriction semigroups. Then we discuss decompositions of (non-zero) semigroups with zero by proving a decomposition theorem. We also show that every atomistic left restriction semigroup S can be decomposed as an orthogonal sum of atomistic left restriction semigroups N_i, where each summand N_i is an irreducible ideal of S. Finally, properties of the summands N_i, when S embeds in some $\mathcal{PT}_X$ the partial transformation monoid on a set X, are investigated.

Keywords: Atomistic left restriction semigroup; irreducible ideal; left restriction meet-semigroup; left restriction semigroup; orthogonal sum.

1. Introduction

A semigroup T is an inverse semigroup, if for all $v \in T$, there is a unique element w in T such that $vwv = v$ and $wvw = w$. Recently, in (FitzGerald, 2020), the author presented the theory of representations of inverse semigroups via homomorphisms into complete atomistic inverse meet-semigroups. The class of inverse meet-semigroups contains $\mathcal{I}_X$ the symmetric inverse monoid on X, $\mathcal{I}_X^*$ (the dual of $\mathcal{I}_X$) and partial automorphism monoids of structures, namely modules, vector spaces and graphs. Some remarkable theorems of decompositions of various representations were proved in (FitzGerald, 2020). Another motivation of the FitzGerald's work is that the representation of T in $\mathcal{I}_X^*$ is more influential than that of in $\mathcal{I}_X$ (see, e.g., (FitzGerald, 2020)).

Left restriction semigroups are non-regular semigroups and are generalizations of inverse semigroups. They arise very naturally from partial transformation monoids in the same way that inverse semigroups arise from symmetric inverse monoids. Since the 1960s, left restriction semigroups occurred with various names and from diverse points of view in literature. For the first time in 1973, left restriction semigroups appeared in their own right in the paper (Trokhimenko, 1973). Also, they were studied in the setting of *SL2 γ-semigroups* in (Batbedat, 1981; Batbedat & Fountain, 1981). These semigroups were also studied as the idempotent connected Ehresmann semigroups in (Lawson, 1991). Later, left restriction semigroups arose in (Jackson & Stokes, 2001) as *(left) twisted C-semigroups*. In (Manes, 2006), they were studied as *guarded semigroups*, which appeared from the restriction categories in (Cockett & Lack, 2002). Recall that for any set X, the partial transformation monoid $\mathcal{PT}_X$ becomes left restriction semigroup under the unary operation $\alpha \mapsto I_{\mathrm{dom}\,\alpha}$. We also recall that left restriction semigroups are precisely the $(2,1)$-subalgebras of some $\mathcal{PT}_X$. Left restriction semigroups were termed as weakly left E-ample semigroups—the (former) York terminology. For weakly left E-ample semigroups, see, e.g., (Hollings, 2007). The reader is referred to (Gould, 2010) for the history of (left) restriction semigroups and their basic properties.

We shall make use of LR-semigroup, ALR-semigroup, LR-meet-semigroup and CALR-meet-semigroup as the abbreviations of left restriction semigroup, atomistic left restriction semigroup, left restriction meet-semigroup and complete atomistic left restriction meet-semigroup respectively unless stated otherwise.

The remaining article is adorned with four more sections. In Section 2, some helpful definitions, related facts are provided. In Section 3, the notion of LR-meet-semigroup is given, and some results associated with ALR-semigroups are proved. Note that LR-semigroups and LR-meet-semigroups generalize inverse semigroups and inverse meet-semigroups respectively. In Section 4, we establish a decomposition theorem for (non-zero) semigroups with zero, and then we prove that every ALR-semigroup S can be decomposed as an orthogonal sum of ALR-semigroups N_i, where each summand N_i is an irreducible ideal of S. In Section 5, we explore properties of the summands N_i, when S is an LR-subsemigroup of some $\mathcal{PT}_X$.

2. Preliminaries

For rudimentary notions related to semigroup theory, and Green's relations $\mathcal{R}$, $\mathcal{L}$, we suggest (Howie, 1995). First, we recall generalized Green's relations.

In (Lawson, 1991), the author introduced the *generalized Green's relations*, i.e., $\widetilde{\mathcal{R}}_F$, $\widetilde{\mathcal{L}}_F$ on a semigroup S, where F is a subset of $E(S)$ the set of idempotents of S. For any $v, w \in S$, $\widetilde{\mathcal{R}}_F$ can be defined as:

$$v \,\widetilde{\mathcal{R}}_F\, w \iff \Big[(\,\forall\, f \in F\,)\; fv = v \Leftrightarrow fw = w\Big].$$

The relation $\widetilde{\mathcal{L}}_F$ is defined dually. The relation $\widetilde{\mathcal{R}}_F$ ($\widetilde{\mathcal{L}}_F$) is an equivalence relation. Green's relation $\mathcal{R}$ ($\mathcal{L}$) is left (right) compatible. On the contrary, $\widetilde{\mathcal{R}}_F$ ($\widetilde{\mathcal{L}}_F$) needs not be left (right) compatible. Note that $\mathcal{R} \subseteq \widetilde{\mathcal{R}}_F$ ($\mathcal{L} \subseteq \widetilde{\mathcal{L}}_F$).

Let $v \in S$ and $f \in F$. Let $v \,\widetilde{\mathcal{R}}_F\, f$. Then as $f \in F$,

$$ff = f \Rightarrow fv = v. \tag{1}$$

Moreover, for any $v \in S$, $f \in F$,

$$v \,\widetilde{\mathcal{R}}_F\, f \iff fv = v \text{ and } \forall\, h \in F\; [hv = v \Rightarrow hf = f]. \tag{2}$$

Therefore, f is the minimum element of $\mathsf{Ll}_v(F)$, where $\mathsf{Ll}_v(F)$ is the set of all left identities of v belonging to F.

Let F be a semilattice (a semigroup of idempotents in which every two elements commute) such that $f, g \in F$. If $v \,\widetilde{\mathcal{R}}_F\, f$ and $v \,\widetilde{\mathcal{R}}_F\, g$, then $f \,\widetilde{\mathcal{R}}_F\, g$. Since $gg = g$, by Equation 1, we have $gf = f$. Since $g \,\widetilde{\mathcal{R}}_F\, f$ and $ff = f$, by Equation 1, we have $fg = g$. Since $gf = fg$, we deduce $f = g$. Therefore, f is unique in the $\widetilde{\mathcal{R}}_F$-class of v if F is a semilattice. For $\widetilde{\mathcal{R}}_F$, $\widetilde{\mathcal{L}}_F$, see, e.g., (Zenab, 2018).

Second, our necessity is to remind the notion of LR-semigroup and related facts. For LR-semigroups, their right sided and two-sided versions, we prescribe (Gould, 2010; Zenab, 2018).

Definition 2.1. *(Zenab, 2018) An* LR*-semigroup is a unary semigroup $(S, \cdot, ^\dagger)$ such that the unary operation † satisfies the following identities:*

$$v^\dagger v = v, \tag{3}$$

$$v^\dagger w^\dagger = w^\dagger v^\dagger, \tag{4}$$

$$(v^\dagger w)^\dagger = v^\dagger w^\dagger, \tag{5}$$

$$vw^\dagger = (vw)^\dagger v. \tag{6}$$

If we put $E_S = S^\dagger = \{w^\dagger \mid w \in S\}$, then one can check that E_S is a semilattice. For every $w^\dagger \in E_S$, $(w^\dagger)^\dagger = w^\dagger$. Each element of E_S is called a projection of S. The set E_S is known as the *semilattice of projections* of S. A partial order $\leq$ on S is defined by the rule that for all $v, w \in S$, $v \leq w$ if and only if $v = v^\dagger w$. This relation is the natural partial order on S, and restricts to the usual partial order on E_S. Moreover, $\leq$ is compatible with multiplication. If $\mathscr{V}$ is the class of all LR-semigroups, then $\mathscr{V}$ is a variety of algebras of type $(2, 1)$. An inverse semigroup Y is an LR-semigroup, if † is defined by $y^\dagger = yy^{-1}$.

Now we define LR-semigroup with zero as follows.

Definition 2.2. *An* LR*-semigroup with zero is a unary semigroup $(S, \cdot, ^\dagger)$, where $(S, \cdot)$ is a semigroup with zero 0_S, † is a unary operation with $0_S^\dagger = 0_S$, and † satisfies Equation 3−Equation 6.*

In the above definition, for all $w \in S$ such that $w \neq 0_S$, $w^\dagger \neq 0_S$. Also, for all $w \in S$, $0_S \leq w$.

An alternative characterization for LR-semigroups is given by Lemma 2.3.

Lemma 2.3. *(Zenab, 2018) Suppose that $(S, \cdot, ^\dagger)$ is a unary semigroup. Then S is an* LR*-semigroup with semilattice of projections E_S if and only if*

(i) *E_S is a semilattice;*

(ii) *every $\widetilde{\mathcal{R}}_{E_S}$-class has an idempotent of E_S;*

(iii) *$\widetilde{\mathcal{R}}_{E_S}$ is a left congruence;*

(iv) *the left ample condition holds, i.e., for all $t \in S$, $e \in E_S$, $te = (te)^\dagger t$.*

Note that, by Lemma 2.3, the LR-semigroup S with semilattice of projections E_S is a weakly left E_S-ample semigroup, and vice versa. Also, in S, for any $t \in S$, the $\widetilde{\mathcal{R}}_{E_S}$-class of t contains a unique idempotent of E_S, which we denote by $t^\dagger$. Then by Equation 2, $t^\dagger t = t$. Remember that $t^\dagger$ is the minimum element of $\mathsf{Ll}_t(E_S)$ the set of all left identities of t in E_S. It can be observed that in S,

$$s \,\widetilde{\mathcal{R}}_{E_S} t \iff s^\dagger = t^\dagger. \tag{7}$$

Example 2.4. (Hollings, 2007) Suppose that T is a weakly left E-ample semigroup, namely LR-semigroup T with semilattice of projections E, and suppose that J is a non-empty set. Denote by P the $J \times J$ identity matrix and consider the Rees matrix semigroup $\mathcal{M} := \mathcal{M}^0(T; J, J; P)$. Define a multiplication on $\mathcal{M}$ by

$$(j, t, k)0 = 0(j, t, k) = 00 = 0$$

and

$$(j, t, k)(l, u, m) = \begin{cases} (j, tu, m) & \text{if } k = l, \\ 0 & \text{if } k \neq l. \end{cases}$$

The set of idempotents of $\mathcal{M}$ is $E(\mathcal{M}) = \{(j, f, j) \mid f \in E(T)\} \cup \{0\}$. In (Hollings, 2007), Example 2.7.3 shows that $\mathcal{M}$ is a weakly left $\mathcal{E}$-ample semigroup such that $0^\dagger = 0$ and $(j, t, k)^\dagger = (j, t^\dagger, j)$, where $\mathcal{E} = \{(j, f, j) \in E(\mathcal{M}) \mid f \in E\} \cup \{0\}$.

Definition 2.5. *(FitzGerald, 2020; Petrich, 1984) Let W be a semigroup containing zero. Let $\{W_\lambda\}_{\lambda \in I}$ be the class of subsemigroups such that $W = \bigcup_{\lambda \in I} W_\lambda$. If for all $\lambda, \mu \in I$ with $\lambda \neq \mu$, $W_\lambda \cap W_\mu = W_\lambda W_\mu = \{0\}$, then W is an orthogonal sum of subsemigroups W_λ, denoted by $W = \sum_{\lambda \in I} W_\lambda$.*

In the above definition, each W_λ is said to be a *summand* in the orthogonal sum W.

Next we remind the following definitions, utmost useful, and taken from (Erné & Joshi, 2015; Howie, 1995).

Let $(P, \leq)$ be a partial ordered set (poset). Then P is called a *meet-semilattice* if for any $m, n \in P$, $m \wedge n$ (meet of m and n) exists in P. Let $\overline{P} = P \cup \{0\}$ be a poset, where 0 is the least element of $\overline{P}$.

If $0 \neq w \in \overline{P}$, then w is called an *atom* if w is a minimal element of $\overline{P} \setminus \{0\}$. The set $\overline{P}$ is an *atomistic poset* if for all $0 \neq w \in \overline{P}$, w is a join of a set of atoms (i.e., of the set of all atoms it dominates).

In the rest of the paper, every LR-semigroup S is an LR-semigroup with zero 0_S unless explicitly stated. Denote by E_S the semilattice of projections of an LR-semigroup S. Moreover, $r \wedge s$ ($r \vee s$) means the meet (join) of a set $\{r, s\}$, while $\bigwedge A$ ($\bigvee A$) means the meet (join) of a non-empty set A.

3. Left restriction meet-semigroups

We furnish the notion of LR-meet-semigroup, and prove some results associated with ALR-semigroups.

In the beginning, let us define the following.

Definition 3.1. *An* LR*-meet-semigroup* $(M, \cdot, ^\dagger, \wedge)$ *is an* LR*-semigroup* $(M, \cdot, ^\dagger)$ *such that* M *is a meet-semilattice with respect to (w.r.t.) the natural partial order* $\leq$ *on* $(M, \cdot, ^\dagger)$.

In the above definition, $(M, \wedge)$ is a semilattice, and for any $u_1, u_2 \in M$,

$$u_1 \leq u_2 \iff u_1 \wedge u_2 = u_1.$$

Hence, $\leq$ is also a natural ordering on $(M, \wedge)$.

Definition 3.2. *A complete left restriction meet-semigroup* $(M, \cdot, ^\dagger, \wedge)$ *is an* LR*-semigroup* $(M, \cdot, ^\dagger)$ *such that for any* $\emptyset \neq B \subseteq M$, $\bigwedge B$ *exists w.r.t.* $\leq$ *on* $(M, \cdot, ^\dagger)$.

Definition 3.3. *An* LR*-semigroup* $(M, \cdot, ^\dagger)$ *is an* ALR*-semigroup if* M *is an atomistic poset w.r.t. its natural partial order.*

Definition 3.4. *Let* $(M, \cdot, ^\dagger)$ *be an* ALR*-semigroup. If for any* $\emptyset \neq B \subseteq M$, $\bigwedge B$ *exists w.r.t.* $\leq$ *on* $(M, \cdot, ^\dagger)$, *then* M *is called a* CALR*-meet-semigroup.*

Proposition 3.5. *Let* M *be an* ALR*-semigroup with zero* 0_M. *Let* $P_{t^\dagger} = t^\dagger M t^\dagger$, *where* $t^\dagger \in E_M \setminus \{0_M\}$. *Then*

(i) $P_{t^\dagger}$ *is an* LR*-subsemigroup of* M *with zero, and containing an identity* $t^\dagger$;

(ii) *every non-zero element of* $P_{t^\dagger}$ *dominates an atom of* $P_{t^\dagger}$;

(iii) *for all non-zero* $x, y \in P_{t^\dagger}$ *such that* $x \not\leq y$, *a non-zero element* k *exists in* $P_{t^\dagger}$ *such that* $k \leq x$ *and* $k \wedge y = 0_{P_{t^\dagger}}$;

(iv) $P_{t^\dagger}$ *is an* ALR*-subsemigroup of* M *with zero, and containing an identity* $t^\dagger$.

Proof. (i) It is simple to verify that $P_{t^\dagger}$ is a subsemigroup of M with zero $0_{P_{t^\dagger}} = 0_M$. We put $0 = 0_{P_{t^\dagger}} = 0_M$. It can be seen that $t^\dagger$ is an identity element of $P_{t^\dagger}$. Now we show that $P_{t^\dagger}$ is closed under †. If $d \in P_{t^\dagger}$ is such that $d \neq 0$, then $t^\dagger d = d$. We can write $(t^\dagger d)^\dagger = d^\dagger$. Since M is an LR-semigroup, by Equation 5, we deduce $t^\dagger d^\dagger = d^\dagger$. Then we have $d^\dagger = t^\dagger t^\dagger d^\dagger$. Since projections of M commute, we deduce $d^\dagger = t^\dagger d^\dagger t^\dagger$. Therefore, $d^\dagger \in P_{t^\dagger}$. Also, $0^\dagger = 0$. So $P_{t^\dagger}$ is closed under †. Hence, $P_{t^\dagger}$ is an LR-subsemigroup of M with zero, and containing an identity $t^\dagger$.

(ii) Let $x \in P_{t^\dagger}$ be such that $x \neq 0$. Since $x \in M$ and M is atomistic, there exists an atom a of M such that $a \leq x$. Since $\leq$ is compatible with multiplication, we obtain $t^\dagger a t^\dagger \leq t^\dagger x t^\dagger$. Since $x \in P_{t^\dagger}$, we have $t^\dagger a t^\dagger \leq x$. Now we prove that $t^\dagger a t^\dagger$ is an atom of $P_{t^\dagger}$. As $a \leq x$, we have $t^\dagger a t^\dagger = t^\dagger a^\dagger x t^\dagger$. Then $t^\dagger a t^\dagger = a^\dagger t^\dagger x t^\dagger = a^\dagger x = a$. Since $a > 0$, $t^\dagger a t^\dagger > 0$. Suppose that for all $r \in P_{t^\dagger}$, $0 \leq r < t^\dagger a t^\dagger$. Since $a = t^\dagger a t^\dagger$, we have $0 \leq r < a$. Since a is an atom of M and $r \in M$, we obtain $r = 0$. Consequently, $t^\dagger a t^\dagger$ is an atom of $P_{t^\dagger}$. Thus, every non-zero element of $P_{t^\dagger}$ dominates an atom of $P_{t^\dagger}$.

(iii) For any non-zero $v \in M$, let $M_v = \{m \mid m \text{ is an atom of } M, m \leq v\}$. Let $x, y \in P_{t^\dagger}$ be such that $x, y \neq 0$ and $x \not\leq y$. Since $x, y \in M$, there exists an atom (a non-zero element) $c \in M$ such that $c \in M_x$ and $c \notin M_y$. Therefore, we have $c \leq x$ and $c \wedge y = 0$. Since $c \leq x$, by compatibility, we have $t^\dagger c t^\dagger \leq t^\dagger x t^\dagger$. As $x \in P_{t^\dagger}$, we obtain $t^\dagger c t^\dagger \leq x$. Now we prove that $t^\dagger c t^\dagger \neq 0$. Suppose that

$t^\dagger ct^\dagger = 0$. As $c \leq x$, we obtain $t^\dagger c^\dagger x t^\dagger = 0$. Then we have $c^\dagger t^\dagger x t^\dagger = 0$. Then $c^\dagger x = 0$. Therefore, $c = 0$—a contradiction. Hence, $t^\dagger ct^\dagger \neq 0$. Next we prove that $t^\dagger ct^\dagger \wedge y = 0$. Certainly, one lower bound of $\{t^\dagger ct^\dagger, y\}$ is 0. If ℓ is any lower bound of $\{t^\dagger ct^\dagger, y\}$, then $\ell \leq t^\dagger ct^\dagger$ and $\ell \leq y$. By Equation 5, $(t^\dagger ct^\dagger)^\dagger c = t^\dagger (ct^\dagger)^\dagger c$. By Equation 6, we have $(t^\dagger ct^\dagger)^\dagger c = t^\dagger c(t^\dagger)^\dagger$. Then $(t^\dagger ct^\dagger)^\dagger c = t^\dagger ct^\dagger$. So $t^\dagger ct^\dagger \leq c$. Since $\ell \leq t^\dagger ct^\dagger$, we have $\ell \leq c$. Since ℓ is the lower bound of $\{c, y\}$ and $c \wedge y = 0$, we deduce $\ell = 0$. Thus, $t^\dagger ct^\dagger \wedge y = 0$. Hence, for all non-zero $x, y \in P_{t^\dagger}$ such that $x \not\leq y$, a non-zero element k exists in $P_{t^\dagger}$ such that $k \leq x$ and $k \wedge y = 0$.

(iv) By (i), $P_{t^\dagger}$ is an LR-subsemigroup of M with zero, and containing an identity $t^\dagger$. Now we prove that $P_{t^\dagger}$ is atomistic. For this purpose, we show that every non-zero element of $P_{t^\dagger}$ is a join of a set of atoms of $P_{t^\dagger}$. For any non-zero $x \in P_{t^\dagger}$, let $\mathscr{P}_x = \{p \mid p \text{ is an atom of } P_{t^\dagger}, p \leq x\}$. We require to show that for any non-zero $x \in P_{t^\dagger}$, $x \leq y$, where $y \in P_{t^\dagger}$ such that y is any upper bound of $\mathscr{P}_x$. On the contrary, suppose that $x \not\leq y$. By (iii), there exists a non-zero $c \in P_{t^\dagger}$ such that $c \leq x$ and $c \wedge y = 0$. By (ii), there exists an atom $\overline{p}$ of $P_{t^\dagger}$ such that $\overline{p} \leq c$. Then we have $\overline{p} \leq x$. Therefore, $\overline{p} \in \mathscr{P}_x$. Since $c \wedge y = 0$, we deduce $\overline{p} \wedge y = 0$. Since $\overline{p} \in \mathscr{P}_x$ and y is any upper bound of $\mathscr{P}_x$, we deduce $\overline{p} \leq y$. Since $\overline{p} \wedge y = 0$, we have $\overline{p} \not\leq y$—a contradiction. Hence, $x \leq y$. Therefore, $x = \bigvee \mathscr{P}_x$. Therefore, $P_{t^\dagger}$ is atomistic. Thus, $P_{t^\dagger}$ is an ALR-subsemigroup of M with zero, and containing an identity $t^\dagger$. □

Proposition 3.6. *Let M be a* CALR*-meet-semigroup with zero $0_{_M}$. Let $P_{t^\dagger} = t^\dagger M t^\dagger$, where $t^\dagger \in E_{_M} \setminus \{0_{_M}\}$. Then $P_{t^\dagger}$ is a* CALR*-meet-subsemigroup of M with zero, and containing an identity $t^\dagger$.*

Proof. By Proposition 3.5 (iv), $P_{t^\dagger}$ is an ALR-subsemigroup of M with zero $0_{_{P_{t^\dagger}}} = 0_{_M}$ and an identity $t^\dagger$. We put $0 = 0_{_{P_{t^\dagger}}} = 0_{_M}$. Let $\emptyset \neq B \subseteq P_{t^\dagger}$. If $0 \in B$, then $\bigwedge B = 0$. Suppose that $0 \notin B$. Since $P_{t^\dagger} \subseteq M$ and M is a CALR-meet-semigroup with zero, it follows that $\bigwedge B$ exists in M. Let $g = \bigwedge B$, where $g \in M$. Then for all $b \in B$, $g \leq b$. Since $\leq$ is compatible with multiplication, we obtain $t^\dagger g t^\dagger \leq t^\dagger b t^\dagger$. Since $b \in P_{t^\dagger}$, we have $t^\dagger g t^\dagger \leq b$. Accordingly, $t^\dagger g t^\dagger$ is a lower bound of B, belonging to $P_{t^\dagger}$. Let ℓ be any lower bound of B such that $\ell \in P_{t^\dagger}$. Since $\ell \in M$ and g is a meet of B in M, we deduce $\ell \leq g$. By compatibility, we have $t^\dagger \ell t^\dagger \leq t^\dagger g t^\dagger$. Since $\ell \in P_{t^\dagger}$, we have $\ell \leq t^\dagger g t^\dagger$. Consequently, $t^\dagger g t^\dagger = \bigwedge B$. Hence, $P_{t^\dagger}$ is a CALR-meet-subsemigroup of M with zero, and containing an identity $t^\dagger$. □

From now on, for ease of notation, for any semigroup A with zero, we will drop the subscript from zero element $0_{_A}$ and write simply 0.

4. Decompositions of semigroups with zero

In this section, we prove a theorem of decomposition for (non-zero) semigroups with zero.

Let us define the following.

Definition 4.1. *Let S be a semigroup with zero. Let N be a non-zero ideal of S. Then N is called reducible if there exist non-zero ideals N_1, N_2 of S such that $N = N_1 \cup N_2$ and $N_1 \cap N_2 = \{0\}$, in this case, we denote it by $N = N_1 \overset{0}{\coprod} N_2$; otherwise N is called irreducible.*

Lemma 4.2. *Let S be a semigroup with zero. Let $\{N_i\}_{i \in I}$ be a family of irreducible ideals of S. Suppose that $\bigcap_{i \in I} N_i \neq \{0\}$. Then $\bigcup_{i \in I} N_i$ is an irreducible ideal of S.*

Proof. Clearly, $\bigcup_{i \in I} N_i$ is an ideal of S. On the contrary, suppose that $\bigcup_{i \in I} N_i = C \overset{0}{\coprod} D$ such that C and D are non-zero ideals of S. By Definition 4.1, we have $\bigcup_{i \in I} N_i = C \cup D$ and $C \cap D = \{0\}$. Take $N_0 \in \{N_i \mid i \in I\}$. This implies that $N_0 = N_0 \cap \Big[\bigcup_{i \in I} N_i\Big]$. Since $\bigcup_{i \in I} N_i = C \cup D$, we have $N_0 = N_0 \cap (C \cup D)$. Then we have $N_0 = (N_0 \cap C) \cup (N_0 \cap D)$. Since N_0 is irreducible, it follows that either $N_0 \cap C = \{0\}$ or $N_0 \cap D = \{0\}$. Assume that $N_0 \cap D = \{0\}$. Then $N_0 = N_0 \cap C$. Then $N_0 \subseteq C$. Now assume that there exist i, j such that $i \neq j$ with $N_i \subseteq C$ and $N_j \subseteq D$. Then we have

$\{0\} \neq \bigcap_{i\in I} N_i \subseteq N_i \cap N_j \subseteq C \cap D = \{0\}$—a contradiction. Then either $\bigcup_{i\in I} N_i \subseteq C$ or $\bigcup_{i\in I} N_i \subseteq D$. So either $\bigcup_{i\in I} N_i = C$ or $\bigcup_{i\in I} N_i = D$. If $\bigcup_{i\in I} N_i = C$, then $D = 0$, which is a contradiction, or if $\bigcup_{i\in I} N_i = D$, then $C = 0$—a contradiction. Thus, $\bigcup_{i\in I} N_i$ is an irreducible ideal of S. □

Theorem 4.3. *Let S be a semigroup with zero. Then S has a unique decomposition $S = \sum_{i\in I} N_i$, where each N_i is an irreducible ideal of S.*

Proof. We divide our proof into the following steps.

Step (1). We know that for all $0 \neq x \in S$, $\langle x\rangle := \{x\} \cup xS \cup Sx \cup SxS$ is the ideal of S generated by x. First, we need to show that $\langle x\rangle$ is irreducible. On the contrary, suppose that $\langle x\rangle = A \overset{0}{\coprod} B$, where A and B are non-zero ideals of S. Then $x \in A \cup B$ and either $x \in A$ or $x \in B$. Without loss of generality, assume that $x \in A$. As A is an ideal of S, it follows that $\{x\}, xS, Sx, SxS \subseteq A$. Therefore, $\langle x\rangle \subseteq A$. Since $A \cap B = \{0\}$, we obtain $B = \{0\}$—a contradiction. Hence, $\langle x\rangle$ is irreducible.

Step (2). For all $0 \neq x \in S$, define

$$\Omega_x = \{V \mid x \in V \text{ and } V \text{ is an irreducible ideal of } S\}.$$

By the proof of Step (1), $\langle x\rangle \in \Omega_x$. Therefore, $\Omega_x \neq \emptyset$. Let $T_x = \bigcup_{V\in\Omega_x} V$. Since $\bigcap_{V\in\Omega_x} V \neq \{0\}$, by Lemma 4.2, T_x is an irreducible ideal of S.

Step (3). Now we show that for all $x, y \in S$, either $T_x \cap T_y = \{0\}$ or $T_x = T_y$. If $T_x \cap T_y = \{0\}$, then we are done. If $T_x \cap T_y \neq \{0\}$, then by Lemma 4.2, $T_x \cup T_y$ is an irreducible ideal of S. Since $x \in T_x \cup T_y$, it follows that $T_x \cup T_y \in \Omega_x$. Since $T_x = \bigcup_{V\in\Omega_x} V$, we have $T_x \cup T_y \subseteq T_x$. As $T_x \subseteq T_x \cup T_y$, we obtain $T_x = T_x \cup T_y$. Similarly, $T_y = T_x \cup T_y$. Hence $T_x = T_y$.

Step (4). By the proof of Step (3), there exists an index set I such that $S = \bigcup_{i\in I} T_{x_i}$ and for any $i, j \in I$ with $i \neq j$, $T_{x_i} \cap T_{x_j} = \{0\}$. In particular, for $i \neq j$, we have $T_{x_i}T_{x_j} \subseteq T_{x_i} \cap T_{x_j} = \{0\}$. Thus,

$$S = \sum_{i\in I} T_{x_i}.$$

Step (5). Suppose that S has another decomposition $S = \sum_{j\in J} M_j$. For all $i \in I$, $T_{x_i} = T_{x_i} \cap S = T_{x_i} \cap \Big[\bigcup_{j\in J} M_j\Big] = \bigcup_{j\in J}(T_{x_i} \cap M_j)$. Since T_{x_i} is irreducible, it follows that there exists exactly one $k \in J$ such that

$$T_{x_i} \cap M_k \neq \{0\}. \tag{8}$$

Then we have $T_{x_i} = T_{x_i} \cap M_k$. Then $T_{x_i} \subseteq M_k$. Now $M_k = M_k \cap S = \bigcup_{i\in I}(M_k \cap T_{x_i})$. Since M_k is irreducible, it follows that there exists exactly one $l \in I$ such that $M_k \cap T_{x_l} \neq \{0\}$. By Equation 8, we deduce $l = i$. Thus, $M_k = M_k \cap T_{x_i}$. Then we have $M_k \subseteq T_{x_i}$. Hence $T_{x_i} = M_k$. The proof is completed. □

Now we explore some properties of the orthogonal sum $S = \sum_{i\in I} N_i$ as in the above theorem when S is an LR-semigroup.

Proposition 4.4. *Suppose that S is an* LR*-semigroup with zero, where $S = \sum_{i\in I} N_i$, the orthogonal sum as in Theorem 4.3. Then the following hold:*

(i) *every N_i is an* LR*-subsemigroup of S;*

(ii) *for all $i \in I$, $0 \neq x \in N_i$ and $0 \neq y \in S$, if $y \leq x$, then $y \in N_i$;*

(iii) *for all $i \in I$ and $0 \neq c \in N_i$, c is an atom of N_i if and only if c is an atom of S;*

(iv) *for all* $i \in I$ *and* $0 \neq x \in N_i$, *define* $A_x = \{c \mid c \text{ is an atom of } S, c \leq x\}$ *and* $B_x = \{c \mid c \text{ is an atom of } N_i, c \leq x\}$. *Then* $A_x = B_x$.

Proof. (i) It is clear that every N_i is a subsemigroup of S. Now we prove that every N_i is an LR-subsemigroup of S. We need to prove that for any $i \in I$, and for any $0 \neq x \in N_i$, $x^\dagger \in N_i$. On the contrary, suppose that for $i \neq k$, $x^\dagger \in N_k$. Since $N_k N_i = \{0\}$, we deduce $x = x^\dagger x = 0$—a contradiction. Therefore, $x^\dagger \in N_i$. Also, $0^\dagger = 0$. Hence, every N_i is an LR-subsemigroup of S.

(ii) On the contrary, assume that for $i \neq k$, $y \in N_k$. As $y \leq x$, we have $y = y^\dagger x$. By (i), $y^\dagger \in N_k$. Since $N_k N_i = \{0\}$, we deduce $y = y^\dagger x = 0$—a contradiction. Hence, $y \in N_i$.

(iii) Let c be any non-zero element of N_i. Suppose that c is an atom of S. Then it is clear that c is an atom of N_i. Conversely, suppose that c is an atom of N_i. For every non-zero $s \in S$ such that $0 < s \leq c$, by (ii), $s \in N_i$. As c is an atom of N_i, it follows that $s = c$. Thus, c is an atom of S.

(iv) Let $a \in A_x$. Then a is an atom of S with $a \leq x$. Since $x \in N_i$, by (ii), it follows that $a \in N_i$. So a is also an atom of N_i. Therefore, $a \in B_x$. So $A_x \subseteq B_x$. If $b \in B_x$, then b is an atom of N_i with $b \leq x$. By (iii), b is also an atom of S. Therefore, $b \in A_x$. So $A_x = B_x$. □

As a corollary of Theorem 4.3 and Proposition 4.4, we obtain the following theorem.

Theorem 4.5. *Let S be a semigroup with zero. Let $S = \sum_{i \in I} N_i$ be as in Theorem 4.3. Then*

(a) *S is an* LR*-semigroup if and only if every N_i $(i \in I)$ is an* LR*-semigroup;*

(b) *S is an* ALR*-semigroup if and only if every N_i $(i \in I)$ is an* ALR*-semigroup.*
In particular, every ALR*-semigroup S is an orthogonal sum of* ALR*-subsemigroups such that each summand is an irreducible ideal of S.*

Proof. (a) If S is an LR-semigroup, then by Proposition 4.4 (i), each N_i is an LR-semigroup. Conversely, if each N_i is an LR-semigroup, then we need to show that Equation 3−Equation 6 hold in S. If all the letters involved lie in the same N_i for some $i \in I$, then Equation 3−Equation 6 hold. On the other hand, in Equation 4−Equation 6, if v and w lie in N_i and N_j $(i \neq j)$ respectively, then all the involved products are zero. Therefore, S is an LR-semigroup.
(b) Let S be an ALR-semigroup. By (a), each N_i is an LR-semigroup. Now we show that N_i is atomistic. For all $i \in I$ and $0 \neq x \in N_i$, define $A_x = \{c \mid c \text{ is an atom of } S, c \leq x\}$ and $B_x = \{c \mid c \text{ is an atom of } N_i, c \leq x\}$. By Proposition 4.4 (iv), $A_x = B_x$. Since S is atomistic, it follows that $x = \bigvee A_x = \bigvee B_x$. Hence, each N_i is an ALR-semigroup. Conversely, suppose that each N_i is an ALR-semigroup. Then by (a), S is an LR-semigroup. For every $0 \neq x \in S$, we have $x \in N_i$ for some $i \in I$. Let A_x, B_x be as above. Then we have $x = \bigvee B_x = \bigvee A_x$. Hence, S is atomistic. The proof is completed. □

5. Properties of the N_i when S embeds in some $\mathcal{PT}_X$

It is known that any LR-semigroup S embeds in some $\mathcal{PT}_X$, which is an ALR-semigroup, and that in any such embedding, for $\sigma \in S$, $\sigma^\dagger$ is the identity map on the domain $\mathrm{d}(\sigma)$ of σ.

Therefore it is of interest to examine the properties of the N_i when $S = \sum_{i \in I} N_i$ is an LR-subsemigroup of $\mathcal{PT}_X$. Without loss of generality, we need consider only the case where the zero of S is the zero of $\mathcal{PT}_X$, namely the empty partial mapping $\emptyset$. This is because of the Proposition 5.2.

Lemma 5.1. *If S is an* LR*-subsemigroup of $\mathcal{PT}_X$ with zero element ζ, and suppose that $\alpha \in S$, and if $(x, y) \in \alpha$ and $x \in \mathrm{d}(\zeta)$, then $x = y$.*

Proof. Since $\zeta = \zeta^\dagger$ is the identity map on its domain, it follows that $(x, y) \in \zeta \circ \alpha = \zeta$ whence $x = y$. □

Proposition 5.2. *If S is an* LR*-subsemigroup of $\mathcal{PT}_X$ with zero element ζ, then the map*

$$\alpha \mapsto \alpha \setminus \zeta$$

is an injective morphism of S into $\mathcal{PT}_Y$ such that $\zeta \mapsto \emptyset$, where $Y = X \setminus \mathrm{d}(\zeta)$.

Proof. Since $\zeta \leq \alpha$, i.e., $\zeta \subseteq \alpha$, the map is injective, and clearly $\zeta \mapsto \emptyset$. Then $(\alpha\setminus\zeta)\circ(\beta\setminus\zeta) = \alpha\circ\beta\setminus\zeta$, as can be shown in the usual manner, together with the aid of the Lemma 5.1. □

If we put $D_i = \bigcup\{\mathrm{d}(\alpha)\colon \alpha \in N_i\}$, $R_i = \bigcup\{\mathrm{r}(\alpha)\colon \alpha \in N_i\}$ and $X_i = D_i \cup R_i$, then we see that N_i is an LR-subsemigroup of $\mathcal{PT}_{X_i}$; and N_i is irreducible since Theorem 4.3 still applies. For $i \neq j$, the sets X_i and X_j need not be disjoint, but must be distinct.

Next, if $R_i \cap D_j \neq \emptyset$, then there are $\alpha \in N_i$, $\beta \in N_j$ such that $\alpha\beta \neq \emptyset$, thus, $i = j$. The converse is true since non-trivial N_i always contains a non-zero $\alpha^\dagger$ and $\mathrm{d}(\alpha^\dagger) = \mathrm{r}(\alpha^\dagger)$ whence $R_i \cap D_i \neq \emptyset$.

In fact, if $\mathrm{r}(\alpha) = \mathrm{d}(\beta)$, then α and β are in the same component, N_i say. So if we say that α,β are Φ-related if $\mathrm{r}(\alpha) = \mathrm{d}(\beta)$, and let Ψ be the smallest equivalence relation containing Φ, then Ψ must partition S into its irreducible components N_i.

ACKNOWLEDGEMENTS

The author would like to thank professors Yuqun Chen and Zerui Zhang for helpful discussions on Section 4.

References

Batbedat, A. (1981). γ-demi-groupes, demi-modules, produit demi-direct. In Semigroups Proceedings (pp. 1-18). Springer. Oberwolfach.

Batbedat, A., & **Fountain, J. B. (1981)**. Connections between left adequate semigroups and γ-semigroups. Semigroup Forum, 22(1), 59–65.

Cockett, J. R. B., & **Lack, S. (2002)**. Restriction categories I: categories of partial maps. Theoretical Computer Science, 270(1-2), 223–259.

Erné, M., & **Joshi, V. (2015)**. Ideals in atomic posets. Discrete Mathematics, 338(6), 954–971.

FitzGerald, D. G. (2020). Representations of inverse semigroups in complete atomistic inverse meet-semigroups. Semigroup Forum, 101(1), 87–101.

Gould, V. (2010). Notes on restriction semigroups and related structures; formerly (weakly) left E-ample semigroups. See `http://www-users.york.ac.uk/~varg1/restriction.pdf`

Hollings, C. (2007). Partial Actions of Semigroups and Monoids. Ph.D. thesis, University of York, Heslington, York YO10 5DD, UK.

Howie, J. M. (1995). Fundamentals of Semigroup Theory. Oxford University Press, Oxford.

Jackson, M., & **Stokes, T. (2001)**. An invitation to C-semigroups. Semigroup Forum, 62(2), 279–310.

Lawson, M. V. (1991). Semigroups and ordered categories. I: The reduced case. Journal of Algebra, 141(2), 422–462.

Manes, E. (2006). Guarded and banded semigroups. Semigroup Forum, 72(1), 94–120.

Petrich, M. (1984). Inverse Semigroups. Wiley, New York.

Trokhimenko, V. S. (1973). Menger's function systems. Izvestiya Vysshikh Uchebnykh Zavedenii. Matematika, 11(138), 71–78. (Russian)

Zenab, R. (2018). Algebraic properties of Zappa–Sz´ep products of semigroups and monoids. Semigroup Forum, 96(2), 316–332.

Submitted: 27/10/2021
Revised: 13/03/2022
Accepted: 16/03/2022
DOI: 10.48129/kjs.16921

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-12)

On weighted noncorona graphs with properties $\mathcal{R}$ and $-\mathcal{SR}$

Uzma Ahmad, Saira Hameed, Sadia Akhter*
Dept. of Mathematics, University of the Punjab, Lahore, Pakistan
**Corresponding author: uzma.math@pu.edu.pk*

Abstract

Let G_w be a simple weighted graph with adjacency matrix $A(G_w)$. The set of all eigenvalues of $A(G_w)$ is called the spectrum of weighted graph G_w denoted by $\sigma(G_w)$. The reciprocal eigenvalue property (or property $\mathcal{R}$) for a connected weighted nonsingular graph G_w is defined as, if $\eta \in \sigma(G_w)$ then $\frac{1}{\eta} \in \sigma(G_w)$. Further, if η and $\frac{1}{\eta}$ have the same multiplicities for each $\eta \in \sigma(G_w)$ then this graph is said to have strong reciprocal eigenvalue property (or property $\mathcal{SR}$). Similarly, a connected weighted nonsingular graph G_w is said to have anti-reciprocal eigenvalue property (or property $-\mathcal{R}$) if $\eta \in \sigma(G_w)$ then $-\frac{1}{\eta} \in \sigma(G_w)$. Furthermore, if η and $-\frac{1}{\eta}$ have the same multiplicities for each $\eta \in \sigma(G_w)$ then strong anti-reciprocal eigenvalue property (or property $-\mathcal{SR}$) holds for the weighted graph G_w. In this article, classes of weighted noncorona graphs satisfying property $\mathcal{R}$ and property $-\mathcal{SR}$ are studied.

Keywords: Adjacency matrix; anti-reciprocal eigenvalue property; corona graphs; strong anti-reciprocal eigenvalue property; weighted graphs

1. Introduction

Spectral graph theory is the branch of mathematics that deals with the properties of graphs in contact with the characteristic polynomial, eigenvectors and eigenvalues of matrices associated with the graphs. Spectral graph theory emerged during 1950s and 1960s. Cvetković summed up virtually all examination to date nearby (Cvetković, 1980). Later on, it was updated by an overview of recent results in the Theory of Graph Spectra (Cvetković *et al.*, 1988). In 2012, discrete geometric analysis was created and developed by Sunada, that dealt with spectral graph theory in terms of discrete Laplacians associated with weighted graphs and discovered applications in different fields, including shape investigation (Sunada, 2012). Nowadays, the spectral graph theory has expanded to vertexvarying graphs often encountered in many real life applications. Also, there are many simple properties of graphs that can be obtained from the eigenvalues of the matrices e.g., the number of edges, the number of connected components (using the adjacency matrix).

Let G be any simple connected graph comprised of the vertex set $V(G)$ and the edge set $E(G)$. Two vertices are called adjacent if there is an edge between them and if one of the vertices of an edge of a graph is a pendant vertex, the edge is said to be pendant. Let G be any graph of order n then the adjacency matrix of the graph G is a matrix of order $n \times n$ defined as, $A(G) = [n_{ij}]$, where n_{ij} is the number of edges between the vertices i and j. A graph G is classified as, singular or nonsingular depending on whether its adjacency matrix is singular or nonsingular. The

characteristic polynomial of a graph G can be written, as
$f(G;t) = \det(tI - A(G))$ and its roots are called the eigenvalues of graph G and the set of all eigenvalues of graph G is called the spectrum of G denoted as $\sigma(G)$.
Let w be a positive weight function defined on edge set of simple connected graph G, which is used to assign weights to the edges and $W(G)$ is the collection of all positive weight functions defined on the edge set of G. A graph G in which the positive weight function w is used to assign weights to the edges of graph is known as weighted graph, denoted by G_w. We use $V(G_w)$ and $E(G_w)$ to denote the vertex set and edge set of weighted graph G_w. Ordinary graphs can be seen as a particular case of weighted graphs in which all the edges are assigned weight 1. An edge between the vertices i and j is denoted by $[i, j]$. Let $A(G_w)$ denotes the adjacency matrix of weighted graph G_w, defined as

$$A(G_w) = [a_{ij}] = \begin{cases} w[i,j], & \text{if } [i,j] \in E(G_w) \\ \\ 0, & \text{otherwise.} \end{cases}$$

The investigation of a graph's structure by associating different matrices to it is a long-standing and fascinating field of study for researchers. The reader can get some initial concepts from (Cvetković, 1980). It would be useful to take a small picture of a large graph that contains information about the graph in a concise way. Studying the spectrum of various matrices, such as the adjacency matrix, the Laplacian matrix, etc. that can be associated with the graph has proven to be one of the most useful ways of doing so.
It is possible to obtain information about a graph by looking at these eigenvalues that might otherwise be difficult to obtain. For instance, a connected graph G is bipartite if and only if $-\eta$ is an eigenvalue of G whenever η is an eigenvalue of G (Godsil & Royle, 2004). In addition η and $-\eta$ have the same multiplicites.

Definition 1.1 *A connected weighted nonsingular graph G_w is said to satisfy the strong reciprocal eigenvalue property (or property $\mathcal{SR}$) if $\frac{1}{\eta} \in \sigma(G)$ whenever $\eta \in \sigma(G)$ and both have the same multiplicities. Weighted Graph G_w has the reciprocal eigenvalue property (property $\mathcal{R}$) when the multiplicity constraint is removed.*

Definition 1.2 *A connected weighted nonsingular graph G_w is said to satisfy the strong anti-reciprocal eigenvalue property (property $-\mathcal{SR}$) if $-\frac{1}{\eta} \in \sigma(G_w)$ whenever $\eta \in \sigma(G_w)$ and both have the same multiplicities. Moreover, if the multiplicity constraint is removed the weighted graph G_w is said to satisfy anti-reciprocal eigenvalue property (property $-\mathcal{R}$).*

Definition 1.3 *A polynomial $f(t) = \sum_{i=0}^{n} a_i t^i$ of degree n is called palindromic polynomial if $a_i = a_{n-i}$ and anti-palindromic polynomial if $a_i = -a_{n-i}$ for $i = 0, 1, \ldots, n$. Property $\mathcal{SR}$ is satisfied by a polynomial $f(t)$ if and only if it is palindromic or anti-palindromic.*

(Frucht & Harary, 1970) defined the corona product of graphs which plays an important role in constructing and characterizing graphs with reciprocal eigenvalue property.

Definition 1.4 *Let L_1 and L_2 be two connected graphs of order n and m, respectively. The corona product $L_1 \circ L_2$ is a graph formed by one copy of graph L_1 and n-copies of L_2 and by connecting each vertex of jth copy of L_2 with the jth vertex of L_1, for $1 \leq j \leq n$.*

We proceed with some previous results. In 1978, graphs with property $\mathcal{SR}$ were investigated for nonsingular trees under the names symmetric property (Godsil & Mckay, 1978) and property C (Cvetković *et al.*, 1978). This property was renamed "property $\mathcal{SR}$" by Barik *et al.* in

2006, and they also introduced property $\mathcal{R}$. They showed that for nonsingular trees, these two properties are the same (Barik *et al.*, 2006).
If specific limits on the weight function are implemented, these properties are similar for weighted trees (Neumann & Pati, 2013), as well as a subclass of connected bipartite graphs with unique perfect matching (Panda & Pati, 2015). In general, however, these properties are not identical (Panda & Pati, 2016).
In 2012, J. D. Lagrange investigated property $-\mathcal{SR}$ first time for the zero-divisor graphs of finite commutative rings with non-zero divisors (Lagrange, 2012).
Authors investigated (Bapat *et al.*, 2016) that if G is a connected bipartite graph having a unique perfect matching M, then weighted graph G_w satisfies property $\mathcal{SR}$, for all $w \in W(G)$ if and only if G is corona.
(Hameed & Ahmad, 2020) analyzed noncorona graphs with zero diagonal entries of the inverse of their adjacency matrix and a single perfect matching, and discovered that they do not meet property $-\mathcal{SR}$ even for a single weight function w.
Property $-\mathcal{SR}$ for the class of connected simple weighted graphs having unique perfect matching M, denoted by G_M, was investigated by (Ahmad *et al.*, 2020). They showed that the weighted graph G_w satisfies property $-\mathcal{SR}$ for all $w \in W(G)$ if and only if G is corona. They also verified property $-\mathcal{SR}$ for some families of noncorona graphs (Ahmad *et al.*, 2021) and authors of (Barik *et al.*, 2021) further generalized these families. They constructed the classes of noncorona graphs by taking a connected corona graph M and by joining each vertex of finite number of copies of corona cycles of different finite length to non-pendant vertices of M, in such a way that no corona cycle is attached to more than one non-pendant vertex.
Until now, the properties $\mathcal{R}$ and $-\mathcal{SR}$ are not studied for weighted noncorona graphs. So, the question arises 'are there any weighted noncorona graphs with these eigenvalue properties?' With the required properties, we constructed families of weighted noncorona graphs. In Section 2, a family of weighted noncorona graphs satisfying property $\mathcal{R}$ and in Section 3 two family of weighted noncorona graphs satisfying property $-\mathcal{SR}$ are constructed. Throughout the paper simple and undirected graphs will be discussed and e_i is the standard unit vector whose i-th entry is equal to 1. Following Lemma gives necessary and sufficient condition for a polynomial to satisfy property $-\mathcal{SR}$.

Lemma 1.1 *(Ahmad et al., 2020) A polynomial $f(t) = \sum_{i=0}^{2n} a_i t^i$ satisfies property $-\mathcal{SR}$ if and only if*

$$a_{2n-i} = \begin{cases} a_i, & \text{if } i \text{ and } n \text{ have the same parity,} \\ -a_i, & \text{otherwise.} \end{cases} \qquad i = 0, 1, 2, \ldots, 2n.$$

Lemma 1.2 and Lemma 1.3 on determinant and inverse of a block matrix involving the Schur complement are used in the proofs of our main results.

Lemma 1.2 *(Bapat, 2010) If A is a block matrix i.e, $A = \begin{bmatrix} K & L \\ M & N \end{bmatrix}$ where K and N are square matrices. Then*

$$\det(A) = \begin{cases} \det(K)\det(N - MK^{-1}L), & \text{if } K \text{ is invertible} \\ \det(N)\det(K - LN^{-1}M), & \text{if } N \text{ is invertible.} \end{cases}$$

Lemma 1.3 *(Bapat, 2010) If A is a block matrix and $A = \begin{bmatrix} K & L \\ M & N \end{bmatrix}$ where K and N are square matrices and N is invertible. Then A is invertible if and only if the Schur complement of N is invertible i.e, $A_N = K - LN^{-1}M$ is invertible, and*

$$A^{-1} = \begin{bmatrix} A_N^{-1} & -A_N^{-1}LN^{-1} \\ -NMA_N^{-1} & N^{-1} + N^{-1}MA_N^{-1}LN^{-1} \end{bmatrix}.$$

The Lemma 1.4 is used in the proof of Theorem 3.1.

Lemma 1.4 *(Barik et al., 2021) Let G be a regular graph of order m and regularity r, and $G_1 = G \circ K_1$. Then*

$$\mathbf{1}^t(tI_{2m} - A(G_1))^{-1}\mathbf{1} = \frac{(2t - r + 2)m}{t^2 - rt - 1}.$$

2. Weighted noncorona graphs satisfying property $\mathcal{R}$

In this Section, we construct a class of weighted noncorona graphs which satisfy property $\mathcal{R}$ but not property $\mathcal{SR}$. In (Panda, 2016) and (Panda & Pati, 2016), authors constructed a class of unweighted noncorona graphs satisfying property $\mathcal{R}$. Now the question arises that 'is it possible to assign weights to some edges so that this class still satisfies property $\mathcal{R}$?' To answer this question, we assign weights to some particular edges of the family of unweighted graphs constructed in (Panda, 2016) and (Panda & Pati, 2016). The new family of weighted noncorna graphs with property $\mathcal{R}$ is as follows.

Consider one copy of P_4, join every vertex of this copy to a new vertex a and name graph as $\acute{G}$

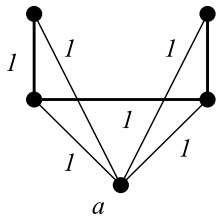


Fig. 1. Graph $\acute{G}$

as shown in Figure 1. Now take k ($k \geq 1$) copies of P_4 named as $P_4^1, P_4^2, \ldots, P_4^k$. With the help of $\acute{G}$ and these k copies of P_4 construct a family $\aleph$ of weighted noncorona graphs in which each weighted graph H_w^k is created by joining every non-pendant vertex in the k copies of P_4 to the vertex a and assigning weights $w_i > 0$ to the joining edges of a and each P_4^i for $i = 1, 2, \ldots, k$ respectively and then add a new vertex b at a. The edges in all k copies of P_4 and $\acute{G}$ are assigned weight 1. A weighted noncorona graph H_w^2 belonging to this family is shown in Figure 2.

The following result proves that weighted noncorona graph $H_w^k \in \aleph$ satisfies property $\mathcal{R}$ but not $\mathcal{SR}$.

Theorem 2.1 *The weighted noncorna graph $H_w^k \in \aleph$ satisfies property $\mathcal{R}$ but not $\mathcal{SR}$.*

Proof:

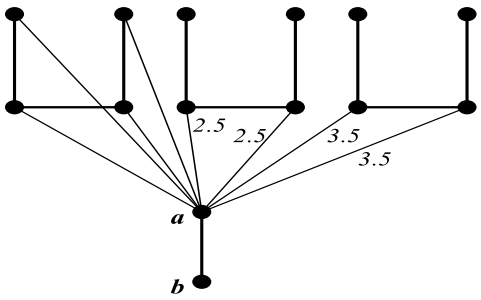


Fig. 2. Weighted noncorona graph H_w^2

The adjacency matrix $A(H_w^k)$ of the graph H_w^k can be written, as

$$A(H_w^k) = \begin{pmatrix} A(\acute{G}) & e_1 & w_1K_{5,4} & \cdots & w_kK_{5,4} \\ e_1^t & 0 & \mathbf{0}^t & \cdots & \mathbf{0}^t \\ w_1K_{5,4}^t & \mathbf{0} & A(P_4^1) & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_kK_{5,4}^t & \mathbf{0} & O & \cdots & A(P_4^k) \end{pmatrix},$$

where

$$K_{5,4} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Suppose that

$$\mathcal{B} = \begin{pmatrix} tI_4 - A(P_4^1) & \cdots & O \\ \vdots & \ddots & \vdots \\ O & \cdots & tI_4 - A(P_4^k) \end{pmatrix}.$$

Then the characteristic polynomial of H_w^k can be written, as

$f(H_w^k; t) = \det(tI - A(H_w^k))$

$$= \det \begin{pmatrix} tI_5 - A(\acute{G}) & -e_1 & -w_1K_{5,4} & \cdots & -w_kK_{5,4} \\ -e_1^t & t & \mathbf{0}^t & \cdots & \mathbf{0}^t \\ -w_1K_{5,4}^t & \mathbf{0} & tI_4 - A(P_4^1) & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -w_kK_{5,4}^t & \mathbf{0} & O & \cdots & tI_4 - A(P_4^k) \end{pmatrix},$$

using Lemma 1.2

$$= \det(\mathcal{B}) \det\Bigg(\begin{bmatrix} tI_5 - A(\acute{G}) & -e_1 \\ -e_1^t & t \end{bmatrix} - \begin{bmatrix} -w_1K_{5,4} & \cdots & -w_kK_{5,4} \\ \mathbf{0}^t & \cdots & \mathbf{0}^t \end{bmatrix} \mathcal{B}^{-1} \begin{bmatrix} -w_1K_{5,4}^t & \mathbf{0} \\ \vdots & \vdots \\ -w_kK_{5,4}^t & \mathbf{0} \end{bmatrix} \Bigg),$$

where

$$\mathcal{B}^{-1} = \begin{pmatrix} (tI_4 - A(P_4^1))^{-1} & \cdots & O \\ \vdots & \ddots & \vdots \\ O & \cdots & (tI_4 - A(P_4^k))^{-1} \end{pmatrix},$$

and

$$(tI_4 - A(P_4))^{-1} = \frac{1}{t^4 - 3t^2 + 1} \begin{pmatrix} (t^2-1)t & t^2 & t^2-1 & t \\ t^2 & (t^2-1)t & t & t^2-1 \\ t^2-1 & t & t(t^2-2) & 1 \\ t & t^2-1 & 1 & t(t^2-2) \end{pmatrix}.$$

Thus,

$$\begin{aligned} f(H_w^k;t) &= (\textstyle\prod_{i=1}^k f(P_4;t)) \det\left(\begin{bmatrix} tI_5 - A(\acute{G}) & -e_1 \\ -e_1^t & t \end{bmatrix} - \begin{bmatrix} \frac{2t}{t^2-t-1} \sum_{i=1}^k w_i^2 K_{5,5} & \mathbf{0} \\ \mathbf{0}^t & 0 \end{bmatrix} \right) \\ &= (t^4 - 3t^2 + 1)^k \det\left(\begin{bmatrix} tI_5 - A(\acute{G}) - \frac{2t}{t^2-t-1} \sum_{i=1}^k w_i^2 K_{5,5} & -e_1 \\ -e_1^t & t \end{bmatrix} \right) \\ &= (t^2-t-1)^k (t^2+t-1)^k (t^4 - t^3 - 2(\textstyle\sum_{i=1}^k w_i^2 + 3)t^2 - t + 1)(t^2+t-1). \end{aligned}$$

Here notice that, $\{1.618033, -0.618033\}$ are the roots of polynomial (t^2-t-1) then $\{0.618033 = \frac{1}{1.618033}, -1.618033 = \frac{1}{-0.618033}\}$ are the roots of polynomial (t^2+t-1) and the polynomial $(t^4 - t^3 - 2(\sum_{i=1}^k w_i^2 + 3)t^2 - t + 1)$ is palindromic as a result this polynomial satisfies property $\mathcal{SR}$. However, because $f(H_w^k;t)$ has an additional factor (t^2+t-1), we can see that every eigenvalue of H_w^k has its reciprocal as an eigenvalue of H_w^k but multiplicities are different so weighted noncorna graph H_w^k satisfies property $\mathcal{R}$ but not $\mathcal{SR}$.
Following example is an illustration of the weighted noncorna graph belonging to the family $\aleph$, it can be seen from Table 1 that weighted noncorona graph H_w^2 satisfies property $\mathcal{R}$ but not $\mathcal{SR}$.

Example 2.1 *The weighted noncorna graph H_w^2, is shown in Figure 2. The eigenvalues of H_w^2, their reciprocals and their multiplicities are given in the following Table:*

Table 1. Eigenvalues of H_w^2, their reciprocals and their multiplicities

Sr. No.	η	Multiplicity of η	$\frac{1}{\eta}$	Multiplicity of $\frac{1}{\eta}$
1	-2.61803	1	-0.38196	1
2	-1.61803	3	-0.61803	2
3	-0.61803	2	-1.61803	3
4	-0.38196	1	-2.61803	1
5	0.26794	1	3.73205	1
6	0.61803	3	1.61803	2
7	1.61803	2	0.61803	3
8	3.73205	1	0.26794	1

3. Weighted noncorona graphs satisfying property $-\mathcal{SR}$

In this Section, some classes of weighted noncorona graphs are constructed which satisfy property $-\mathcal{SR}$. Consider a connected weighted graph G_w, $w > 0$ of order n and $G_w^1 = G_w \circ K_1$ be its weighted corona graph in which pendant edges are assigned weight 1. Let $F^p = C_p \circ K_1$ be corona cycle where C_p is a cycle of order p, $p \geq 3$. Now, with the help of weighted graph G_w^1 and corona cycles with edges assigned weight 1, we construct families of weighted noncorona

graphs as follows:
Take a copy weighted graph of G_w^1 and k corona cycles $F_1^{p_1}, F_2^{p_2}, \ldots, F_k^{p_k}$ (where p_i's not necessarily same, for $i = 1, 2, \ldots, k$) with edges assigned weight 1. Consider any number of non-pendant vertices $v_1, v_2, \ldots, v_l$, $(1 \leq l \leq n)$ of weighted graph G_w^1. Join each v_j, $(j \leq l)$ to all the vertices of each corona cycle $F_i^{p_i}, i = 1, 2, \ldots, k$. Assign weight w_i to the edges joining a cycle $F_i^{p_i}$, $(i = 1, 2, \ldots, k)$ to all the vertices $v_1, v_2, \ldots, v_l$ and name this weighted graph as $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)}$ as shown in Figure 3. We denote the family containing all weighted noncorona graphs $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)}$ by $\mathfrak{G}$. Now, instead of assigning weight w_i to the edges joining a cycle $F_i^{p_i}$, $(i = 1, 2, \ldots, k)$ to all the vertices $v_1, v_2, \ldots, v_l$, if we assign weight w_j to the edges joining the vertex v_j to each corona cycle for $j = 1, 2, \ldots, l$ we obtain a new weighted graph named as, $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1;w_2;\ldots;w_l)}$ as shown in Figure 5. We denote the family containing all weighted noncorona graphs $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1;w_2;\ldots;w_l)}$ by $\mathfrak{H}$.

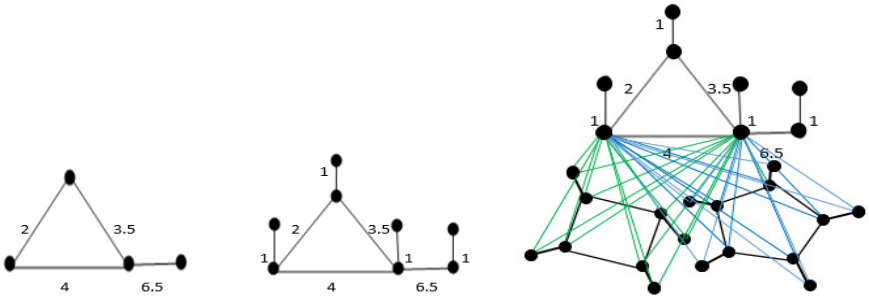


Fig. 3. Weighted graph U, weighted corona graph U_w^1 and $S^{(4,5;2)}_{(3.5,6.5)}$.

Observation 3.1 *For a weighted corona graph G_w^1 of order $2n$, the sum of first $n \times n$ entries of cofactor matrix of $tI - A(G_w^1)$ can be written, as*

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(-1)^{i+j}C_{ij} = ct^k g(t),$$

where c is any constant and $g(t)$ is a polynomial of degree $2n - 2k$, $1 \leq k \leq n$, satisfying property $-\mathcal{SR}$. Then note that $f(t) + c\, t^k g(t)$ also satisfies property $-\mathcal{SR}$, where $f(t)$ is the characteristic polynomial of the weighted corona graph G_w^1 of weighted graph G_w and $g(t)$ is the polynomial obtained from the sum of first $n \times n$ entries of the cofactor matrix of $tI - A(G_w^1)$.

We can see this observation with the help of Example 3.1.

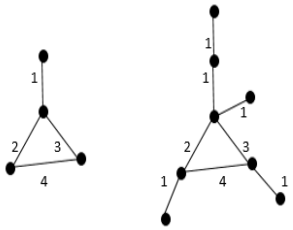


Fig. 4. Weighted graph Z_w and its weighted corona graph Z_w^1

Example 3.1 *Consider a connected weighted graph Z_w of order $n = 4$ and its corona graph as shown in the Figure 4. Then characteristic polynomial of $Z_w^1 = Z_w \circ K_1$ can be determined, as*
$f(Z_w^1; t) = det(tI - A(Z_w^1)) = t^8 - 34t^6 - 48t^5 + 82t^4 + 48t^3 - 34t^2 + 1.$
We can see that it is a polynomial of order $2n = 8$ which satisfies property $-\mathcal{SR}$ as Z_w^1 is weighted corona graph. Now the sum of first 4×4 entries of cofactor matrix of $tI - A(Z_w^1)$ can be written, as

$$\begin{aligned} tg(t) &= 4t^7 + 20t^6 - 10t^5 - 88t^4 + 10t^3 + 20t^2 - 4t \\ &= 2t(2t^6 + 10t^5 - 5t^4 - 44t^3 + 5t^2 + 10t - 2), \end{aligned}$$

which satisfies property $-\mathcal{SR}$ by Lemma 1.1.
Now

$$f(t) + tg(t) = t^8 + 4t^7 - 14t^6 - 58t^5 - 6t^4 + 58t^3 - 14t^2 - 4t + 1,$$

which also satisfies property $-\mathcal{SR}$ by Lemma 1.1.

By Laplace expansion, we can easily obtain the following result.

Lemma 3.1 *Let A be any $2n \times 2n$ matrix, then*

$$\det(A + \begin{bmatrix} J_n & O_n \\ O_n & O_n \end{bmatrix}) = \det(A) + \sum_{i=1}^{n}\sum_{j=1}^{n}(-1)^{(i+j)}\det(A[i,j]),$$

where J_n is the matrix of ones, O_n is the matrix of zeros and $A[i,j]$ is the sub-matrix of matrix A obtained by deleting ith row and jth column.

The following result proves that weighted noncorona graph $S_{(w_1,w_2,\ldots,w_k)}^{(p_1,p_2,\ldots,p_k;l)}$ satisfies property $-\mathcal{SR}$.

Theorem 3.1 *The weighted noncorona graph $S_{(w_1,w_2,\ldots,w_k)}^{(p_1,p_2,\ldots,p_k;l)} \in \mathfrak{G}$ for $1 \leq l \leq n$ satisfies property $-\mathcal{SR}$.*

Proof:
The adjacency matrix $A(S_{(w_1,w_2,\ldots,w_k)}^{(p_1,p_2,\ldots,p_k;l)})$ of the weighted noncorona graph $S_{(w_1,w_2,\ldots,w_k)}^{(p_1,p_2,\ldots,p_k;l)}$ can be written, as

$$A(S_{(w_1,w_2,\ldots,w_k)}^{(p_1,p_2,\ldots,p_k;l)}) = \begin{pmatrix} A(G_w) & I_n & w_1N_{n,2p_1} & \cdots & w_kN_{n,2p_k} \\ I_n & O & O & \cdots & O \\ w_1N_{n,2p_1}^t & O & A(F_1^{p_1}) & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_kN_{n,2p_k}^t & O & O & \cdots & A(F_k^{p_k}) \end{pmatrix},$$

where $N_{n,2p_k} = \begin{bmatrix} J_{l,2p_k} \\ O_{n-l,2p_k} \end{bmatrix}$ for $1 \leq l \leq n$ is a block matrix in which $J_{l,2p_k}$ is the matrix with all entries 1 of order $l \times 2p_k$ and $O_{n-l,2p_k}$ is the Null matrix of order $(n-l) \times 2p_k$. Let us suppose that

$$\mathcal{D} = \begin{pmatrix} tI_{2p_1} - A(F_1^{p_1}) & \cdots & O \\ \vdots & \ddots & \vdots \\ O & \cdots & tI_{2p_k} - A(F_k^{p_k}) \end{pmatrix}.$$

Then the characteristic polynomial of $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)}$ can be written, as

$$f(S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)};t) = \det(tI - A(S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)}))$$

$$= \det\begin{pmatrix} tI_n - A(G_w) & -I_n & -w_1N_{n,2p_1} & \cdots & -w_kN_{n,2p_k} \\ -I_n & O & O & \cdots & O \\ -w_1N^t_{n,2p_1} & O & tI_{2p_1} - A(F_1^{p_1}) & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -w_kN^t_{n,2p_k} & O & O & \cdots & tI_{2p_k} - A(F_k^{p_k}) \end{pmatrix},$$

using Lemma 1.2

$$= \det(\mathcal{D})det\left(\begin{bmatrix} tI_n - A(G_w) & -I_n \\ -I_n & tI_n \end{bmatrix} - \begin{bmatrix} -w_1N_{n,2p_1} & \cdots & -w_kN_{n,2p_k} \\ O & \cdots & O \end{bmatrix}\right.$$

$$\left.\mathcal{D}^{-1}\begin{bmatrix} -w_1N^t_{n,2p_1} & O \\ \vdots & \vdots \\ -w_kN^t_{n,2p_k} & O \end{bmatrix}\right)$$

$$= (\textstyle\prod_{i=1}^k f(F_i^{p_i};t))\det\left(\begin{bmatrix} tI_n - A(G_w) & -I_n \\ -I_n & tI_n \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^k w_i^2\mathbf{1}^t\mathcal{D}^{-1}\mathbf{1}N_n & O \\ O & O \end{bmatrix}\right).$$

Now, from Lemma 1.4,

$$\mathbf{1}^t\mathcal{D}^{-1}\mathbf{1} = \frac{2t}{t^2-2t-1}\sum_{i=1}^k p_i,$$

Thus,

$$= (\textstyle\prod_{i=1}^k f(F_i^{p_i};t))\det\left(\begin{bmatrix} tI_n - A(G_w) & -I_n \\ -I_n & tI_n \end{bmatrix} - \begin{bmatrix} \frac{2t}{t^2-2t-1}\sum_{i=i}^k p_iw_i^2\ N_n & O \\ O & O \end{bmatrix}\right)$$

$$= (\textstyle\prod_{i=1}^k f(F_i^{p_i};t))\det((tI_{2n} - A(G_w^1)) + \begin{bmatrix} aN_n & O \\ O & O \end{bmatrix}), \text{ where } a = -\frac{2t}{t^2-2t-1}\sum_{i=1}^k p_iw_i^2.$$

Now by using Lemma 3.1

$$= (\textstyle\prod_{i=1}^k f(F_i^{p_i};t))[\det(tI_{2n} - A(G_w^1)) + a\sum_{i=1}^l\sum_{j=1}^l(-1)^{i+j}\det((tI_{2n} - A(G_w^1)[i,j])],$$

and by Observation 3.1

$$f(S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)};t) = \frac{\prod_{i=1}^k f(F_i^{p_i};t)}{t^2-2t-1}(f(t) + ct^kg(t)),$$

where $f(t) = (t^2-2t-1)f(G_w \circ K_1;t)$ satisfies property $-\mathcal{SR}$ and by Observation 3.1, $f(t) + ct^kg(t)$ satisfies property $-\mathcal{SR}$ also for $i = 1, 2, \ldots k$, $\frac{f(F_i^{p_i};t)}{(t^2-2t-1)}$ satisfies property $-\mathcal{SR}$. Thus, $f(S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1,w_2,\ldots,w_k)};t)$ satisfies property $-\mathcal{SR}$.

Following example is an illustration of the weighted noncorona graph $S^{(4,5;2)}_{(3.5,6.5)}$ for $p_1 = 4,\ p_2 = 5,\ w_1 = 3.5,\ w_2 = 6.5$ and $l = 2$, it can be seen from Table 2 that weighted noncorona graph $S^{(4,5;2)}_{(3.5,6.5)}$ satisfies property $-\mathcal{SR}$.

Example 3.2 *Let M_w be any connected weighted graph of order 4 and $M_w^1 = M_w \circ K_1$ be its weighted corona graph in which pendant edge has weight 1 as shown in Figure 3. Now, construct the weighted noncorona graph $S^{(4,5;2)}_{(3.5,6.5)}$ by using M_w^1 and the corona cycles F_1^4 and*

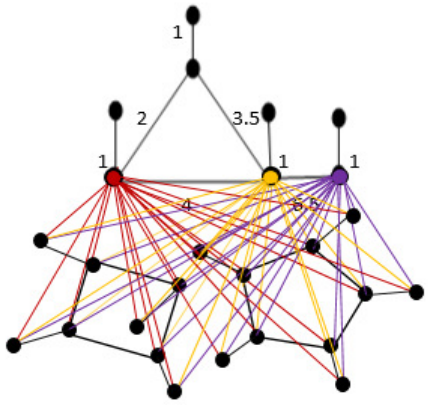


Fig. 5. Weighted noncorona graph $S^{(4,5;3)}_{(0.5;1.5;2.5)}$ in which red edges are assigned weight 0.5, yellow edges are assigned weight 1.5 and purple edges are assigned weight 2.5.

F^5_2, as shown in Figure 3. The weights assigned to the joining edges of corona cycles F^4_1, F^5_2 to 2 selected vertices of M_w are 3.5 and 6.5 represented by green and blue edges respectively. The eigenvalues of $S^{(4,5;2)}_{(3.5,6.5)}$ and with their multiplicities are mentioned in the following table.

Table 2. Eigenvalues of $S^{(4,5;2)}_{(3.5,6.5)}$, their reciprocals and their multiplicities

Sr. No.	η	multiplicity of η	$-\frac{1}{\eta}$	multiplicity of $-\frac{1}{\eta}$
1	-42.194	1	0.0237	1
2	-7.2208	1	0.13849	1
3	-2.4142	1	0.41421	1
4	-2.0953	2	0.47726	2
5	-1	2	1	2
6	-0.99623	1	1.0038	1
7	-0.73764	2	1.3557	2
8	-0.41421	1	2.4142	1
9	-0.2936	1	3.4060	1
10	-0.020767	1	48.154	1
11	0.0237	1	-42.194	1
12	0.13849	1	-7.2208	1
13	0.41421	1	-2.4142	1
14	0.47726	2	-2.0953	2
15	1	2	-1	2
16	1.0038	1	-0.99623	1
17	1.3557	2	-0.73764	2
18	2.4142	1	-0.41421	1
19	3.4060	1	-0.2936	1
20	48.154	1	-0.020767	1

The following theorem can be proved with the same strategy as in Theorem 3.1.

Theorem 3.2 *Weighted noncorona graph $S^{(p_1,p_2,\ldots,p_k;l)}_{(w_1;w_2;\ldots;w_l)}$ satisfies property $-\mathcal{SR}$.*

4. Conclusion

In this article, we constructed three classes of weighted noncorona graphs namely $\aleph$, $\mathfrak{G}$ and $\mathfrak{H}$ which satisfy property $\mathcal{R}$ or $-\mathcal{SR}$. The family of weighted noncorona $\aleph$ satisfies property $\mathcal{R}$ but not $\mathcal{SR}$. The other two families $\mathfrak{G}$ and $\mathfrak{H}$ satisfy property $-\mathcal{SR}$.

References

Ahmad, U., Hameed, S., & **Jabeen, S. (2020).** Class of weighted graphs with strong anti-reciprocal eigenvalue property. Linear and Multilinear Algebra, 68(**6**):1129-1139.

Ahmad, U., Hameed, S., & **Jabeen, S. (2021).** Noncorona graphs with strong anti-reciprocal eigenvalue property. Linear and Multilinear Algebra, 69(**10**):1878-1888.

Bapat, R. B. (2010). Graphs and Matrices (Vol. **27**). London: Springer. https://doi.org/10.1007/978-1-4471-6569-9.

Bapat, R. B., Panda, S. K., & **Pati, S. (2016).** Strong reciprocal eigenvalue property of a class of weighted graphs. Linear Algebra and its Applications, 511, 460-475.

Barik, S., Neumann, M., & **Pati, S. (2006).** On nonsingular trees and a reciprocal eigenvalue property. Linear and Multilinear Algebra, 54(**6**):453-465.

Barik, S., Ghosh, S., & **Mondal, D. (2021).** On graphs with strong anti-reciprocal eigenvalue property. Linear and Multilinear Algebra, 1-14.

Cvetković, D. M. (1980). Spectra of Graphs: Theory and Applications, Pure and Applied Mathematics, 87. Academic Press, Inc., New York-London.

Cvetković, D. M., Doob, M., Gutman, I., & **Torgasev, A. (1988).** Recent Results in the Theory of Graph Spectra. Annals of Discrete mathematics.

Cvetković, D. M., Gutman, I., & **Simić, S. K. (1978).** On self pseudo-inverse graphs. Publikacije Elektrotehničkog fakulteta. Serija Matematika i fizika, (**602/633**):111-117.

Frucht, R., & **Harary, F. (1970).** On the corona of two graphs (pp. 322-325). Valparaiso, Chile; Ann Arbor, Mich., U.S.A.

Godsil, C. D., & **Mckay, B. D. (1978).** A new graph product and its spectrum. Bulletin of the Australian Mathematical Society, 18(**1**):21-28.

Godsil, C. & **Royle, G. F. (2004)** Algebraic Graph Theory (Vol. 207). Springer Science & Business Media. New York.

Hameed, S., & **Ahmad, U. (2020).** Inverse of the adjacency matrices and strong anti-reciprocal eigenvalue property. Linear and Multilinear Algebra, 1-26.

Lagrange, J. D. (2012). Boolean rings and reciprocal eigenvalue properties. Linear algebra and its applications, 436(**7**):1863-1871.

Neumann, M. & **Pati, S. (2013).** On reciprocal eigenvalue property of weighted trees. Linear Algebra and its Applications, 438(**10**):3817-3828.

Panda, S. K. (2016). On the Inverse of Bipartite Graphs with Unique Perfect Matchings and Reciprocal Eigenvalue Properties (Doctoral dissertation).

Panda, S. & **Pati, S. (2015).** On the inverse of a class of bipartite graphs with unique perfect matchings. The Electronic Journal of Linear Algebra, 29, 89-101.

Panda, S., & **Pati, S. (2016).** Graphs with reciprocal eigenvalue properties. The Electronic Journal of Linear Algebra, 31, 511-514.

Sunada, T. (2012) Topological crystallography: with a view towards discrete geometric analysis (Vol. 6). Springer Science & Business Media.

Submitted: 01/12/2021
Revised: 30/05/2022
Accepted: 05/06/2022
DOI: 10.48129/kjs.17497

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-12)

Recovery of coefficients of a heat equation by Ritz collocation method

Kamal Rashedi*
Dept. of Mathematics, University of Science and Technology of Mazandaran, Behshahr, Iran
**Corresponding author: k.rashedi@mazust.ac.ir*

Abstract

In this work, we discuss a one dimensional inverse problem for the heat equation where the unknown functions are solely time-dependent lower order coefficient and multiplicative source term. We use as data two integral overdetermination conditions along with the initial and Dirichlet boundary conditions. In the first step, the lower order term is eliminated by applying a transformation and the problem is converted to an equivalent inverse problem of determining a heat source with initial and boundary conditions, as well as a nonlocal energy over-specification. Then, we propose a Ritz approximation as the solution of the unknown temperature distribution and consider a truncated series as the approximation of unknown time-dependent coefficient in the heat source. The collocation method is utilized to reduce the inverse problem to the solution of a linear system of algebraic equations. Since the problem is ill-posed, numerical discretization of the reformulated problem may produce ill-conditioned system of equations. Therefore, the Tikhonov regularization technique is employed in order to obtain stable solutions. For the perturbed measurements, we employ the mollification method to derive stable numerical derivatives. Numerical simulations while solving two test examples are presented to show the applicability of the proposed method.

Keywords: Inverse coefficient problem; mollification method; parabolic equation; Ritz approximation; Tikhonov regularization

1. Introduction

In this paper, we consider the inverse problem of finding $\Big(u(x,t), c(t), d(t)\Big)$ in the parabolic equation (Shekarpaz & Azari, 2018)

$$u_t - a(x,t)u_{xx} + b(x,t)u_x + c(t)u = d(t)g(x,t), \quad (x,t) \in Q, \tag{1}$$

with the initial condition

$$u(x,0) = u_0(x), \quad -L < x < L, \tag{2}$$

boundary conditions

$$u(-L,t) = u(L,t) = 0, \quad 0 < t < T, \tag{3}$$

and subject to the integral over-specifications of the functions $\omega_1(x)u(x,t)$ and $\omega_2(x)u(x,t)$ over the spatial domain (energy over-specifications)

$$\int_{-L}^{L} \omega_1(x)u(x,t)dx = \mu_1(t), \quad t \in [0,T], \tag{4}$$

$$\int_{-L}^{L} \omega_2(x)u(x,t)dx = \mu_2(t), \quad t \in [0,T], \tag{5}$$

where $Q = [-L, L] \times [0, T]$ and $a(x,t),\ b(x,t),\ g(x,t),\ \mu_1(t),\ \mu_2(t),\ u_0(x),\ \omega_1(x),\ \omega_2(x)$ are given functions with appropriate conditions. The additional Equations 4-5 are interpreted as the measurements of function $u(x,t)$ by sensor averaging over the segment $[-L, L]$ of space variable. Furthermore, we assume that the following compatibility conditions hold:

$$u_0(-L) = u_0(L) = 0, \quad \int_{-L}^{L} \omega_1(x)u_0(x)dx = \mu_1(0), \qquad \int_{-L}^{L} \omega_2(x)u_0(x)dx = \mu_2(0). \tag{6}$$

Integral overdetermination conditions are employed to establish an integral or integro-differential equation of the Fredholm or Volterra type and then the analysis of the existence, uniqueness and continuous dependence of the solution is given for the new reformulated problem. The properties of the kernel functions $\omega_1(x)$ and $\omega_2(x)$ included in the integral boundary conditions can directly affect the solvability constraints of the problem and further may complicate the application of the numerical techniques to obtain accurate solutions.

As a special class of the inverse problems, the inverse coefficient problems (ICPs) appear in studying various physical phenomena in order to determine some unknown properties of a region in parabolic and hyperbolic equations. The unknown coefficients can be a function of only time variable if the spatial change in the solution of the direct problem is small in comparison with the change in time (see (Dehghan & Shamsi, 2006; Shamsi & Dehghan, 2012) and (Shamsi & Dehghan, 2006) and many references therein). Moreover, if the property of the medium under study does not change rapidly, the unknown coefficient can be space-wise dependent solely (Liao, 2011). However, in the general form it depends on the solution of the direct problem (Rashedi, 2021; Samarskii & Vabishchevich, 2008).

Although the ICPs in the heat equations are well-studied, the particular problem of determining multiple unknown time-dependent coefficients in heat transfer is less investigated (Hussein & Lesnic, 2014; Lesnic *et al.*, 2016). In (Ivanchov & Pabyrivs'ka , 2001) and (Ivanchov & Pabyrivs'ka , 2002), the authors established conditions for the existence and uniqueness of a solution of the inverse problems for a parabolic equation with two unknown time-dependent coefficients. In (Hussein *et al.*, 2014), the authors investigated the numerical approximation of time-dependent thermal conductivity and convection coefficients in a one-dimensional parabolic equation from boundary temperature and heat flux. In (Huntul *et al.*, 2017), the authors studied simultaneous reconstruction of time-dependent coefficients including the thermal conductivity, convection or absorption coefficients in the parabolic heat equation from heat moments. In (Lingde *et al.*, 2017), the authors studied an inverse problem of the simultaneous determination of the right-hand side and the lowest coefficients in parabolic equations and proposed linearized approximations in time using the fully implicit scheme and standard finite difference procedures in space.

In (Shekarpaz & Azari, 2018), a numerical approach based on the forward finite difference and backward finite difference methods was presented for solving the problem given by Equations 1-5. Even though this approach is effective for solving various kinds of partial differential equations, the high computational cost of FD schemes is a difficulty of this method. Moreover, they can often achieve only two or three digits of accuracy (Dehghan & Shamsi, 2006; Shamsi & Dehghan, 2012, 2006). In this paper we use a collocation technique (Canuto *et al.*, 2006; Jahangiri *et al.*, 2016) to provide more accurate and stable numerical solution for the inverse problem 1-5.

The organization of this article is as follows. In Section 2, we review theoretical results concerning the uniqueness of the solution for the inverse problem 1-5 and use new variables to derive the equivalent problem. Section 3, presents the application of Ritz collocation method to the solution of the reformulated problem. In Section 4, some numerical examples are presented to demonstrate the effectiveness of the proposed method. In Section 5, we present some concluding remarks.

2. Uniqueness

In (Kamynin, 2015), the authors established the situations under which the system of Equations 1-5 possesses a unique solution.

Theorem 2.1 *Suppose that all the functions appearing in the Equations 1-5 are measurable and the compatibility conditions of Equation 6 among the boundary and initial conditions hold. Moreover, as-*

sume that there exist the constants

$$C_{1a},\, C_{2a},\, C_{u_0},\, C_g,\, C_{\omega_1},\, C_{\omega_2},\, C_{\mu_1},\, C_{\mu_2} > 0,\, C_a^*,\, C_a^{**},\, C_b,\, C_b^*,\, C_{\omega_1}^*,\, C_{\omega_1}^{**},\, C_{\omega_2}^*,\, C_{\omega_2}^{**},\, C_{\mu_1}^*,\, C_{\mu_2}^* \geq 0,$$

subject to

- $\forall\, (x,t) \in Q,\ C_{1a} \leq a(x,t) \leq C_{2a},\ |a_x(x,t)| \leq C_a^*,\ |a_{xx}(x,t)| \leq C_a^{**},$
- $\forall\, (x,t) \in Q,\ |b(x,t)| \leq C_b,\ |b_x(x,t)| \leq C_b^*,\ |g(x,t)| \leq C_g,$
- $\forall\, x \in [-L,L],\ |\omega_1(x)| \leq C_{\omega_1},\ |\omega_1'(x)| \leq C_{\omega_1}^*,\ |\omega_1''(x)| \leq C_{\omega_1}^{**},\ \omega_1(\mp L) = 0,$ $\omega_1(x) \in W_2^2([-L,L]),$
- $\forall\, x \in [-L,L],\ |\omega_2(x)| \leq C_{\omega_2},\ |\omega_2'(x)| \leq C_{\omega_2}^*,\ |\omega_2''(x)| \leq C_{\omega_2}^{**},\ \omega_2(\mp L) = 0,$ $\omega_2(x) \in W_2^2([-L,L]),$
- $\forall\, x \in [-L,L],\ |u_0(x)| \leq C_{u_0},\ u_0(x) \in W_2^1([-L,L]),$
- $\forall\, t\ \in [0,T],\ |\mu_1(t)| \leq C_{\mu_1},\ |\mu_1'(t)| \leq C_{\mu_1}^*,\ |\mu_2(t)| \leq C_{\mu_2},\ |\mu_2'(t)| \leq C_{\mu_2}^*,$

and denoting $G_{\omega_1}(t) := \int_{-L}^{L} g(x,t)\omega_1(x)dx,\ G_{\omega_2}(t) := \int_{-L}^{L} g(x,t)\omega_2(x)dx$*, then there exists* C_Δ *such that if*

$$\forall\, t \in [0,T],\ Det \begin{pmatrix} \mu_1(t) & -G_{\omega_1}(t) \\ \mu_2(t) & -G_{\omega_2}(t) \end{pmatrix} \geq C_\Delta > 0,$$

then, the inverse problem given by Equations 1-5 has a unique solution.

Proof. Please refer to (Kamynin, 2015; Shekarpaz & Azari, 2018).

Next, we employ a method to transform problem 1-5 into a problem of finding an unknown heat source from one additional measurement. Let

$$v(x,t) = r(t)u(x,t), \quad r(t) = e^{\int_0^t c(z)dz}, \tag{7}$$

then, applying transformation 7 in Equations 1-5 results the following system of equations

$$v_t - a(x,t)v_{xx} + b(x,t)v_x = r(t)d(t)g(x,t), \quad (x,t) \in Q, \tag{8}$$

$$v(x,0) = u_0(x), \quad -L < x < L, \tag{9}$$

$$v(-L,t) = v(L,t) = 0, \quad 0 < t < T, \tag{10}$$

$$\int_{-L}^{L} \omega_1(x)v(x,t)dx = \mu_1(t)r(t), \quad t \in [0,T], \tag{11}$$

$$\int_{-L}^{L} \omega_2(x)v(x,t)dx = \mu_2(t)r(t), \quad t \in [0,T]. \tag{12}$$

The unknown function $r(t)$ can be disappeared in Equations 11-12 if either one of the functions $\mu_1(t)$ or $\mu_2(t)$ is nonzero on the interval $[0,T]$. Without loss of generality, we assume that $\forall\, t \in [0,T],\ \mu_1(t) \neq 0$. From Equation 11 we have

$$r(t) = \frac{\int_{-L}^{L} \omega_1(x)v(x,t)dx}{\mu_1(t)}, \tag{13}$$

which by substituting the Equation 13 in Equation 12, the following equation is achieved:

$$\int_{-L}^{L} \omega_2(x)v(x,t)dx = \frac{\mu_2(t)}{\mu_1(t)} \int_{-L}^{L} \omega_1(x)v(x,t)dx, \quad t \in [0,T]. \tag{14}$$

Now by defining

$$H(t) := r(t)d(t), \tag{15}$$

the main problem is reduced to the simplified problem of identifying $\Big(v(x,t), H(t)\Big)$ using the following system of equations

$$v_t - a(x,t)v_{xx} + b(x,t)v_x = H(t)g(x,t), \quad (x,t) \in Q, \tag{16}$$

$$v(x,0) = u_0(x), \quad -L < x < L, \tag{17}$$

$$v(-L,t) = v(L,t) = 0, \quad 0 < t < T, \tag{18}$$

and

$$\mu_2(t)\int_{-L}^{L} \omega_1(x)v(x,t)dx - \mu_1(t)\int_{-L}^{L} \omega_2(x)v(x,t)dx = 0, \quad t \in [0,T]. \tag{19}$$

Theorem 2.2 *Assume that at least one of the functions $\mu_1(t)$ or $\mu_2(t)$ is nonzero over the interval $[0,T]$. Then, the problems given by Equations 1-5 and 16-19 are equivalent.*

Proof. Obviously, if $\Big(u(x,t), c(t), d(t)\Big)$ is a solution of problem 1-5, then from Equations 7 and 15, $\Big(v(x,t), H(t)\Big)$ is a sloution of problem 16-19. Conversely, assuming that $\Big(v(x,t), H(t)\Big)$ is a solution of problem 16-19, the function $r(t)$ is verified from Equation 13 provided that $\mu_1(t) \neq 0$. Then, Equation 15 yields $d(t) = \frac{H(t)}{r(t)}$. Utilizing Equation 7 and differentiating $r(t) = e^{\int_0^t c(z)dz}$ with respect to t we get

$$c(t) = \frac{r'(t)}{r(t)}, \quad u(x,t) = \frac{v(x,t)}{r(t)}. \tag{20}$$

Therefore, we will consider problem 16-19 instead of problem 1-5.

3. Solution method

Suppose that $P_m(z),\ m = 0,1,2,3,...$ denote the well-known Legendre polynomials of order m which are defined on the interval $[-1,1]$ and can be determined via the following recurrence formula:

$$P_0(z) = 1,\ P_1(z) = z,\ P_{m+1}(z) = \frac{2m+1}{m+1}zP_m(z) - \frac{m}{m+1}P_{m-1}(z),\ m = 1,2,3,....$$

Then, we consider $\phi_i(x) := P_i(\frac{x}{L})$ as the shifted Legendre polynomial of degree i in the interval $[-L,L]$ and $\psi_j(t) := P_j(\frac{2t}{T} - 1)$ as the shifted Legendre polynomial of degree j in the interval $[0,T]$. The Ritz approximation $v_{N,N'}(x,t)$ based on polynomial basis functions is sought in the form of the following truncated series

$$v_{N,N'}(x,t) = \sum_{i=0}^{N}\sum_{j=0}^{N'} c_{ij}t(x+L)(x-L)\phi_i(x)\psi_j(t) + u_0(x), \tag{21}$$

and the approximation of $H(t)$ is considered as

$$H_{N''}(t) = \sum_{j=0}^{N''} \alpha_j\psi_j(t). \tag{22}$$

Substituting the approximations $v_{N,N'}(x,t)$ and $H_{N''}(t)$ in Equations 16 and 19 respectively, the following residual functions are constructed

$$Res_1(x,t) = \sum_{i=0}^{N}\sum_{j=0}^{N'} c_{ij}\Big\{(x^2-L^2)\phi_i(x)(\psi_j(t)+t\psi_j'(t)) - a(x,t)\psi_j(t)\Big(2\phi_i(x)+(x^2-L^2)\phi_i{}''(x) \tag{23}$$

$$+4x\phi_i'(x)\Big)+b(x,t)\Big(2x\phi_i(x)+(x^2-L^2)\phi_i'(x)\Big)\Big\}-\sum_{i=0}^{N''}\alpha_i g(x,t)\psi_i(t)+b(x,t)u_0'(x)-a(x,t)u_0''(x), \tag{24}$$

$$Res_2(t)=\sum_{i=0}^{N}\sum_{j=0}^{N'}c_{ij}t\Big\{\mu_2(t)\psi_j(t)\Delta_i^*-\mu_1(t)\psi_j(t)\Delta_i^{**}\Big\}+\mu_2(t)\mu_1(0)-\mu_1(t)\mu_2(0), \tag{25}$$

where

$$\Delta_i^*=\int_{-L}^{L}\omega_1(x)\phi_i(x)dx,\quad \Delta_i^{**}=\int_{-L}^{L}\omega_2(x)\phi_i(x)dx.$$

Collocating the residual functions $Res_1(x_i,t_j)=0$ and $Res_2(t_k^*)=0$ at the points

$$(x_i,t_j)=\Big(\frac{(2i-2-N)L}{N+2},\frac{jT}{N'+2}\Big),\ t_k^*=\frac{kT}{N''+2}\quad i=\overline{1,N+1},\ j=\overline{1,N'+1},\ k=\overline{1,N''+1}, \tag{26}$$

forms a linear system of algebraic equations

$$AC=g, \tag{27}$$

where C is the vector of unknown constants $c_{ij},\ \alpha_k$. Generally, A is an ill-conditioned matrix, therefore we require using regularization techniques to obtain stable solution. Hence, instead of Equation 27, according to the Tikhonov regularization method we solve the modified system of equations

$$(A^{tr}A+\lambda I)c=A^{tr}g, \tag{28}$$

where I is the identity matrix, A^{tr} denotes the transpose of the matrix A and $\lambda>0$ is the regularization parameter (Hansen, 1992). Therefore, the approximations of functions $v(x,t)$ and $H(t)$ are specified.

It is worthy to note that the approximation given by Equation 21 satisfies the initial and boundary conditions 17-18 exactly, provided that the compatibility conditions of Equation 6 hold. Thus by increasing the parameters N, N' and N'', if the residual functions $Res_1(x,t),\ Res_2(t)\longrightarrow 0$, then the Equations 16 and 19 are satisfied and the approximations $v_{N,N'}(x,t)$ and $H_{N''}(t)$ converge to the exact solutions $v(x,t)$ and $H(t)$, respectively.

In the following, we consider the approximation of the function $r(t)$ as

$$G_{N,N'}(t):=\frac{\int_{-L}^{L}\omega_1(x)v_{N,N'}(x,t)dx}{\mu_1(t)}, \tag{29}$$

and calculate the approximation of the unknown functions $c(t),\ d(t)$ and $u(x,t)$ in two different situations.

Case 1: Suppose that all the initial and boundary conditions 17-19 are given accurately. By substituting the approximations 22 and 29 in Equations 15 and 20, the following approximations are obtained

$$c_{approx}(t)=\frac{\frac{d}{dt}\Big(G_{N,N'}(t)\Big)}{G_{N,N'}(t)},\ d_{approx}(t)=\frac{H_{N''}(t)}{G_{N,N'}(t)},\ u_{approx}(x,t)=\frac{v_{N,N'}(x,t)}{G_{N,N'}(t)}. \tag{30}$$

Case 2: In real applications, due to the presence of inaccuracies in the input data we need to perform the regularization procedure to deal with the derivative of the perturbed data such as $G'(t)$ since it involves perturbed function $\mu_1'(t)$. Therefore, regarding the perturbed boundary data, let $\mu_1^\sigma(t)$ and $G_{N,N'}^\sigma(t)=\frac{\int_{-L}^{L}\omega_1(x)v_{N,N'}(x,t)dx}{\mu_1^\sigma(t)}$ be perturbations such that

$$\max\{\|G_{N,N'}^\sigma(t)-G(t)\|_\infty,\|\mu_1(t)-\mu_1^\sigma(t)\|_\infty\}\leq\sigma.$$

Then, we employ the mollification method of (Murio, 1993) by taking into account the Gaussian mollifier $F_\delta(t) = \frac{\exp(-\frac{t^2}{\delta^2})}{\delta\sqrt{\pi}}$ where $\delta > 0$ is the radius of mollification. The mollification of the perturbed data $(G^\sigma_{N,N'}(t))'$ is performed using the convolution

$$\Big\{F_\delta * (G^\sigma_{N,N'})'\Big\}(t) := \int_{-\infty}^{+\infty} F_\delta(r)(G^\sigma_{N,N'})'(t-r)dr. \tag{31}$$

We use

$$\Big\{F_\delta * (G^\sigma_{N,N'})'\Big\}(t) = \Big\{F'_\delta * (G^\sigma_{N,N'})\Big\}(t), \tag{32}$$

such that for a given $\delta > 0$ the function $\Big\{F'_\delta * (G^\sigma_{N,N'})\Big\}(t)$ is calculated numerically using the mid-point integration rule, that is

$$\Big\{F'_\delta * (G^\sigma_{N,N'})\Big\}(t) \simeq \frac{\pi}{m_\delta}\sum_{i=0}^{m_\delta-1} Q(t, -\frac{\pi}{2} + \frac{\pi i}{m_\delta} + \frac{\pi}{2m_\delta}),\ Q(t,r) = F'_\delta(\tan r)G^\sigma_{N,N'}(t-\tan r)\sec^2 r. \tag{33}$$

Then, we consider the following

$$(G^\sigma_{N,N'})'(t) = \Big\{F'_\delta * (G^\sigma_{N,N'})\Big\}(t) \simeq \sum_{i=0}^{N''} \beta_i^{\delta,\sigma}\psi_i(t), \tag{34}$$

and consequently

$$(G^\sigma_{N,N'})(t) \simeq \sum_{i=0}^{N''} \beta_i^{\delta,\sigma}\int_0^t \psi_i(z)dz + G^\sigma_{N,N'}(0), \quad G^\sigma_{N,N'}(0) \approx \frac{\int_{-L}^{L}\omega_1(x)u_0(x)dx}{\mu_1^\sigma(0)}. \tag{35}$$

The strategy given by Equations 32-35 is admissible if for small value $\epsilon > 0$, and the appropriate given values δ and m_δ we find

$$\|\sum_{i=0}^{N''} \beta_i^{\delta,\sigma}\int_0^t \psi_i(z)dz + \frac{\int_{-L}^{L}\omega_1(x)u_0(x)dx}{\mu_1^\sigma(0)} - \frac{\int_{-L}^{L}\omega_1(x)v_{N,N'}(x,t)dx}{\mu_1^\sigma(t)}\|_\infty \leq \epsilon. \tag{36}$$

If so, the approximate solution for $c(t)$ is given by

$$c_{approx}(t) = \frac{\mu_1^\sigma(t)\sum_{i=0}^{N''}\beta_i^{\delta,\sigma}\psi_i(t)}{\int_{-L}^{L}\omega_1(x)v_{N,N'}(x,t)dx}, \tag{37}$$

and the approximations of $u(x,t)$ and $d(t)$ are derived as follows

$$d_{approx}(t) = \frac{H_{N''}(t)}{G^\sigma_{N,N'}(t)},\ u_{approx}(x,t) = \frac{v_{N,N'}(x,t)}{G^\sigma_{N,N'}(t)}. \tag{38}$$

4. Numerical experiments

To test the applicability of the proposed technique, we solve two examples. The notations

$$E(u(x,t)) = |u_{exact}(x,t) - u_{approx}(x,t)|, \quad E(d(t)) = |d_{exact}(t) - d_{approx}(t)|$$

and

$$E(c(t)) = |c_{exact}(t) - c_{approx}(t)|,$$

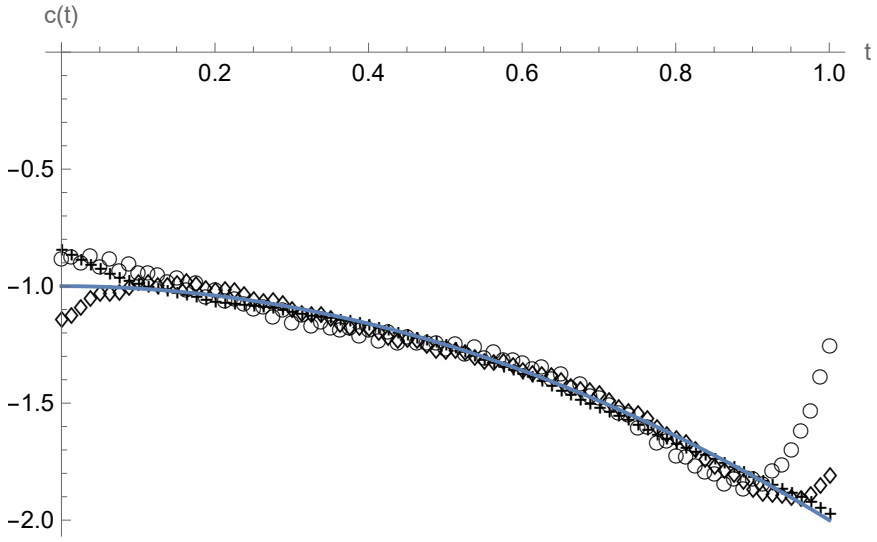


Fig. 1. Representation of the exact (blue line) and approximate solutions for $c(t)$ obtained by applying the proposed method with $N = N' = N'' = 5$ and $\lambda = 10^{-5}$, $\delta = 0.01,\ m_\delta = 600,\ \epsilon = 0.25$ in the presence of the perturbed boundary data subject to different values of σ, i.e. $+++$: corresponding to $\sigma = 1 \times 10^{-2}$, $\diamondsuit\diamondsuit\diamondsuit$: corresponding to $\sigma = 3 \times 10^{-2}$, $\circ\circ\circ$: corresponding to $\sigma = 6 \times 10^{-2}$, discussed in Example 4.0.2.

Table 1. The results of l^2-norm of functions $Res_1(x,t)$ and $Res_2(t)$ and the relative root-mean square error for functions $c(t)$, $d(t)$ and $u(x,t)$ with $M = 50$, discussed in Example 4.0.1.

(N, N', N'')	$\|Res_1(x,t)\|_2$	$\|Res_2(t)\|_2$	$RRMSE(c)$	$RRMSE(d)$	$RRMSE(u)$
$(6,6,4)$	8.2×10^{-1}	1.3×10^{-4}	2.6×10^{-2}	1.3×10^{-3}	2.2×10^{-3}
$(8,8,5)$	1.9×10^{-1}	3.9×10^{-5}	2.4×10^{-3}	6.4×10^{-4}	6.4×10^{-4}
$(9,9,6)$	3.1×10^{-2}	8.83×10^{-7}	5×10^{-4}	8×10^{-5}	1.2×10^{-4}
$(10,10,7)$	6×10^{-3}	6.86×10^{-7}	7.4×10^{-5}	1.72×10^{-5}	1.3×10^{-5}

are defined as the absolute error for functions $u(x,t)$, $d(t)$ and $c(t)$ respectively. Moreover, we define the relative root-mean square error for functions $c(t)$, $d(t)$ and $u(x,t)$ as follows

$$RRMSE(c) := \sqrt{\frac{\sum_{i=0}^{M} E^2(c(\frac{iT}{M}))}{\sum_{i=0}^{M} c^2(\frac{iT}{M})}}, \quad RRMSE(d) := \sqrt{\frac{\sum_{i=0}^{M} E^2(d(\frac{iT}{M}))}{\sum_{i=0}^{M} d^2(\frac{iT}{M})}},$$

$$RRMSE(u) := \sqrt{\frac{\sum_{i,j=0}^{M} E^2(u(\frac{2Li}{M} - L, \frac{jT}{M}))}{\sum_{i,j=0}^{M} u^2(\frac{2Li}{M} - L, \frac{jT}{M})}}.$$

Throughout this work, we select the regularization parameters λ by applying the L-Curve criterion (Hansen, 1992) and find the appropriate values for δ and m_δ by trial and error. Numerical implementation is carried out with Wolfram Mathematica software in a personal computer.

Table 2. The results of the infinity norm of errors for the approximations of unknown functions $c(t)$, $d(t)$ and $u(x,t)$ in the presence of exact boundary data, discussed in Example 4.0.1.

(N, N', N'')	$\|E(c(t))\|_\infty$	$\|E(d(t))\|_\infty$	$\|E(u(x,t))\|_\infty$	λ
$(6,6,4)$	6.9×10^{-2}	5.8×10^{-3}	8.5×10^{-3}	10^{-11}
$(8,8,5)$	3.61×10^{-3}	3.87×10^{-3}	4×10^{-3}	10^{-12}
$(9,9,6)$	7.1×10^{-4}	3.9×10^{-4}	8.7×10^{-4}	10^{-13}
$(10,10,7)$	5×10^{-5}	5.3×10^{-5}	5.7×10^{-5}	10^{-13}

4.0.1 Example 1

Consider the inverse problem

$$u_t - xtu_{xx} + (x^2 + t^2)u_x + c(t)u = d(t)g(x,t), \quad in \quad [-1,1] \times [0,1], \tag{39}$$

where

$$g(x,t) = \sin(\pi x)e^x\Big(1 + e^{-t^2} + (t^2 + x^2) - tx(1 - \pi^2)\Big) + \pi\cos(\pi x)e^x(t - x)^2,$$

with initial condition

$$u_0(x) = e^x \sin(\pi x), \quad -1 \le x \le 1, \tag{40}$$

and homogeneous boundary conditions

$$u(-1,t) = u(1,t) = 0, \quad 0 \le t \le 1, \tag{41}$$

and overspecifications

$$\int_{-1}^{1}(1 - x^2)u(x,t)dx = \frac{2\pi e^{t-1}\Big(5 + \pi^2 + e^2(-1 + 3\pi^2)\Big)}{(1 + \pi^2)^3}, \tag{42}$$

and

$$\int_{-1}^{1} x^2(x^2-1)u(x,t)dx = \frac{-2\pi e^{t-1}\Big(125 - 89\pi^2 - 25\pi^4 - 3\pi^6 + e^2(-25 + 101\pi^2 - 59\pi^4 + 7\pi^6)\Big)}{(1 + \pi^2)^5}. \tag{43}$$

The exact solutions of this problem are

$$c(t) = e^{-t^2},\ d(t) = e^t,\ u(x,t) = e^{t+x}\sin(\pi x).$$

We solve the problem by applying the numerical scheme discussed in Section 3 in the presence of exact boundary data and use the approximations given by Equation 30. The results for relative root-mean square error for functions $c(t)$, $d(t)$ and $u(x,t)$ together with l^2-norm of fuctions $Res_1(x,t)$ and $Res_2(t)$ are presented in Table 1. Moreover, in Tables 2-3 we report the infinity norm and l^2-norm of errors for the approximations of unknown functions $c(t)$, $d(t)$ and $u(x,t)$ per different number of basis functions which indicate that the accuracy is improved by increasing the number of basis functions.

Table 3. The results of the l^2-norm of errors for the approximations of unknown functions $c(t),\ d(t)$ and $u(x,t)$ in the presence of exact boundary data, discussed in Example 4.0.1.

(N, N', N'')	$\|E(c(t))\|_2$	$\|E(d(t))\|_2$	$\|E(u(x,t))\|_2$	λ
$(6,6,4)$	2×10^{-2}	2.4×10^{-3}	4.9×10^{-3}	10^{-11}
$(8,8,5)$	1.81×10^{-3}	1.07×10^{-3}	1.4×10^{-3}	10^{-12}
$(9,9,6)$	3.8×10^{-4}	1.3×10^{-4}	2.7×10^{-4}	10^{-13}
$(10,10,7)$	5.2×10^{-5}	2.8×10^{-5}	3×10^{-5}	10^{-13}

Table 4. The results of the infinity norm of errors for the approximations of unknown functions $c(t),\ d(t)$ and $u(x,t)$ in the presence of exact boundary data, discussed in Example 4.0.2.

(N, N', N'')	$\|E(c(t))\|_\infty$	$\|E(d(t))\|_\infty$	$\|E(u(x,t))\|_\infty$	λ
$(4,4,4)$	0.051	0.056	0.024	10^{-4}
$(6,6,5)$	0.0034	0.0027	0.0009	10^{-6}
$(8,8,6)$	0.0001	0.00067	0.00021	10^{-9}
$(10,10,7)$	2×10^{-6}	1.1×10^{-7}	1.6×10^{-5}	10^{-11}

Table 5. The results of l^2-norm of functions $Res_1(x,t)$ and $Res_2(t)$ and the relative root-mean square error for functions $c(t)$, $d(t)$ and $u(x,t)$ with $M = 50$, discussed in Example 4.0.2.

(N, N', N'')	$\|Res_1(x,t)\|_2$	$\|Res_2(t)\|_2$	$RRMSE(c)$	$RRMSE(d)$	$RRMSE(u)$
$(4,4,4)$	1.9×10^{-1}	2.54×10^{-3}	2.4×10^{-2}	7.2×10^{-3}	2×10^{-3}
$(6,6,5)$	1.8×10^{-2}	5.2×10^{-5}	1.51×10^{-3}	3.3×10^{-4}	1.4×10^{-4}
$(8,8,6)$	1.09×10^{-3}	1.6×10^{-5}	6×10^{-4}	6.7×10^{-5}	1.2×10^{-5}
$(10,10,7)$	5.03×10^{-5}	1.58×10^{-6}	2.24×10^{-6}	6.9×10^{-6}	7.7×10^{-7}

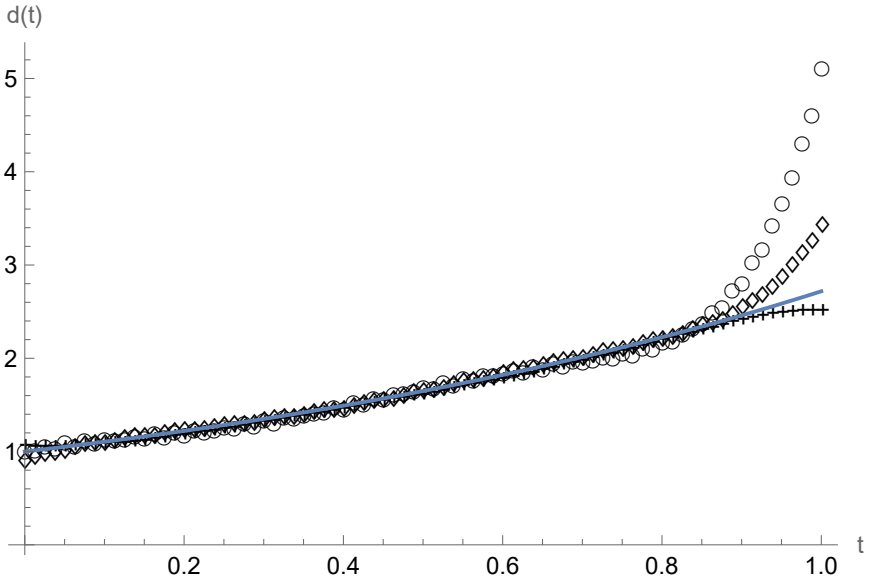


Fig. 2. Representation of the exact (blue line) and approximate solutions for $d(t)$ obtained by applying the proposed method with $N = N' = N'' = 5$ and $\lambda = 10^{-5}$, $\delta = 0.01,\ m_\delta = 600,\ \epsilon = 0.25$ in the presence of the perturbed boundary data subject to different values of σ, i.e. $+++$: corresponding to $\sigma = 1 \times 10^{-2}$, $\diamondsuit\diamondsuit\diamondsuit$: corresponding to $\sigma = 3 \times 10^{-2}$, $\circ\circ\circ$: corresponding to $\sigma = 6 \times 10^{-2}$, discussed in Example 4.0.2.

4.0.2 Example 2

Consider (Shekarpaz & Azari, 2018) the problem given by Equations 1-5 defined over the bounded domain $Q = [-1, 1] \times [0, 1]$ with the following properties:

$$a(x,t) = 1,\ b(x,t) = 1,\ g(x,t) = -2t + (\pi^2 - 2t)\cos(\pi x) + t(2-t)(1+\cos(\pi x)), \tag{44}$$

$$u_0(x) = 1 + \cos(\pi x),\ \omega_1(x) = 1 + x^2,\ \omega_2(x) = 1 - x,\ \mu_1(t) = (\frac{8}{3} - \frac{4}{\pi^2})e^t,\ \mu_2(t) = 2e^t, \tag{45}$$

and the exact solutions

$$c(t) = -1 - t^2,\ d(t) = e^t,\ u(x,t) = e^t(\cos(\pi x) + 1).$$

By using the approximations 30 presented in Section 3 with different values $N,\ N',\ N''$, we produce the results tabulated in Tables 4-5. From the numerical findings it can be seen that the infinity norm of errors as well as the relative root-mean square errors are decreased as the number of basis functions increases gradually which indicate that our method is convergent. Next, we study the numerical stability of the solution with respect to the boundary conditions. Thus, we generate the perturbed boundary data using the following rules (Kirsch, 2011)

$$\mu_1^\sigma(t) = \mu_1(t) + \sigma\sin(\frac{t}{\sigma^2}), \quad \sigma = r \times 10^{-2},\ r \in \mathbf{N}, \tag{46}$$

$$\mu_2^\sigma(t) = \mu_2(t) + \sigma\sin(\frac{t}{\sigma^2}), \quad \sigma = r \times 10^{-2},\ r \in \mathbf{N}. \tag{47}$$

By employing the investigated method with $N = N' = N'' = 5$ and $\sigma \in \{1, 3, 6\} \times 10^{-2}$ and taking the approximations 37 and 38, we obtain the results as shown in Figures 1-2. From the illustrations, it can be seen that the performance of the method is good and the proposed technique finds the stable solution while the amount of noise tends to zero. Indeed, the fair agreement between the exact and approximate solutions holds since the errors imposed to the additional data and propagated with the approximations are of the same order.

5. Conclusion

This article gives a stable numerical solution of an inverse coefficient problem in the one-dimensional heat equation from integral overdetermination conditions. By utilizing new variables, the main problem is converted to a problem of reconstructing an unknown heat source from one additional measurement. We propose a Ritz approximation as the solution of the unknown temperature distribution and consider some truncated series as the approximation of unknown time-dependent function in the heat source. Then, the collocation technique is employed to reduce the inverse problem to the solution of algebraic equations. We take advantage of the mollification method to derive the stable numerical derivatives and solve the ill-conditioned system of equations by using the Tikhonov regularization technique in order to obtain the stable solutions. Following the numerical simulations, it is confirmed that our method proposes a robust approach in dealing with introduced artificial errors in the input boundary data and performs quite well in the presence of exact boundary data since the approximate solutions converge to the exact solutions numerically. Compared to the results presented in (Shekarpaz & Azari, 2018), it can be observed that the algorithm proposed in the present paper yields better results because of providing higher accuracy with lower computational cost. This technique can be extended to solve similar problems in higher dimensions.

References

Canuto, C., Hussaini, M. Y., Quarteroni, A., & **Zang, T. A. (2006)**. Spectral methods: fundamentals in single domains. *Springer*.

Dehghan, M., & **Shamsi, M. (2006)**. Numerical solution of two-dimensional parabolic equation subject to nonstandard boundary specifications using the pseudospectral Legendre method. *Numerical Methods for Partial Differential Equations*, **22**(6), 1255-1266.

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, **34**(4), 561-580.

Huntul, M. J., Lesnic, D., & **Hussein, M. S. (2017)**. Reconstruction of time-dependent coefficients from heat moments. *Applied Mathematics and Computation*, **301**, 233-253.

Hussein, M. S., Lesnic, D., & **Ivanchov, M. I. (2014)**. Reconstruction of time-dependent coefficients from heat moments. *Computer and Mathematics with Applications*, **67**(5), 1065-1091.

Hussein, S. O., & **Lesnic, D. (2014)**. Determination of a space-dependent force function in the onedimensional wave equation. *Electronic Journal of Boundary Elements*, arXiv preprint arXiv:1410.5498, 2014.

Ivanchov, M. I., & **Pabyrivs'ka, N. V. (2001)**. Simultaneous determination of two coefficients of a parabolic equation in the case of nonlocal and integral conditions. *Ukrainian Mathematical Journal*, **53**(5), 674-684.

Ivanchov, M. I., & **Pabyrivs'ka, N. V. (2002)**. On determination of two time-dependent coefficients in a parabolic equation. *Siberian Mathematical Journal*, **43**(2), 323-329.

Jahangiri, S., Maleknejad, K., & **Kajani, M. T. (2016)**. A hybrid collocation method based on combining the third kind Chebyshev polynomials and block-pulse functions for solving higher-order initial value problems. *Kuwait Journal of Science*, **43**(4), 1-10.

Kamynin, V. L. (2015). The inverse problem of the simultaneous determination of the right-hand side and the lowest coefficient in a parabolic equation with many space variables. *Mathematical Notes*, **97**(3-4), 349-361.

Kirsch, A. (2011). An introduction to the mathematical theory of inverse problems. Vol. 120. *Springer*.

Lesnic, D., Hussein, B. T., & **Johansson, B. T. (2016)**. Inverse space-dependent force problems for the wave equation. *Journal of Computational and Applied Mathematics*, **306**, 10-39.

Liao, W. (2011). A computational method to estimate the unknown coefficient in a wave equation using boundary measurements. *Inverse Problems in Science and Engineering*, **19**(6), 855-877.

Lingde, Su., Vabishchevish , P. N., & **Vasil'ev, V. I. (2017)**. The inverse problem of the simultaneous determination of the right-hand side and the lowest coefficients in parabolic equations. *NAA2016, Lecture Notes in Computer Science*, **10187**, 633-639.

Murio, D. A. (1993). The Mollification Method and the Numerical Solution of Ill-Posed Problems. *Wiley*.

Rashedi, K. (2021). A numerical solution of an inverse diffusion problem based on operational matrices of orthonormal polynomials. *Mathematical methods in the applied sciences* , **44**(17), 12980-12997.

Samarskii, A. A., & **Vabishchevich, P. N. (2008)**. Numerical methods for solving inverse problems of mathematical physics. *De Gruyter*.

Shamsi, M., & **Dehghan, M. (2006)**. Recovering a time-dependent coefficient in a parabolic equation from overspecified boundary data using the pseudospectral Legendre method . *Numerical Methods for Partial Differential Equations*, **23**(1), 196-210.

Shamsi, M., & **Dehghan, M. (2012)**. Determination of a control function in three-dimensional parabolic equations by Legendre pseudospectral method. *Numerical Methods for Partial Differential Equations*, **28**(1), 74-93.

Shekarpaz, S., & **Azari, H. (2018)**. An inverse problem of identifying two unknown parameters in parabolic differential equations. *Iranian Journal of Science and Technology: Science, Transaction A*, **42**, 2045-2052.

Submitted: 02/08/2021
Revised: 15/11/2021
Accepted: 16/11/2021
DOI: 10.48129/kjs.18581

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-15)

Some results on Steiner decomposition number of graphs

E.Ebin Raja Merly[1], M.Mahiba[2,*]
[1,2] *Dept. of Mathematics, Nesamony Memorial Christian College, Manonmanium Sundaranar University, Tamilnadu, India.*

**Corresponding author: mahibakala@gmail.com*

Abstract

Let G be a connected graph with Steiner number $s(G)$. A decomposition $\pi = \{G_1, G_2, \dots, G_n\}$ is said to be a Steiner decomposition if $s(G_i) = s(G)$ for all i $(1 \leq i \leq n)$. The maximum cardinality obtained for the Steiner decomposition π of G is called the Steiner decomposition number of G and is denoted by $\pi_{st}(G)$. In this paper we present a relation between Steiner decomposition number and independence number of G. Steiner decomposition number for some power of paths are discussed. It is also shown that given any pair m, n of positive integers with $m \geq 2$ there exists a connected graph G such that $s(G) = m$ and $\pi_{st}(G) = n$.

Keywords: Independence number; power of path; realization theorem; steiner decomposition number; steiner number.

1. Introduction

All graphs considered in this paper are connected, simple and undirected. For basic graph theoretic terminologies we refer to (Harary, 1988). The concept of Steiner number of a graph is introduced by Chartrand and Zhang (Chartrand & Zhang, 2002). Let G be a connected graph. For a set $W \subseteq V(G)$, a tree T contained in G is a Steiner tree with respect to W if T is a tree of minimum order with $W \subseteq V(T)$. The set $S(W)$ consists of all vertices in G that lie on some Steiner tree with respect to W. The set W is a Steiner set for G if $S(W) = V(G)$. The minimum cardinality among the Steiner sets of G is the Steiner number $s(G)$. Steiner concept is considered to be the extension of geodesic concept and hence it provides a new way to study the structure of graphs based on distance. Further investigation on this concept is seen in the works (Pelayo, 2004; Hernando *et al.,* 2005; Yero & Rodriguez-Velazquez, 2015).

Decomposition of graphs is considered as one of the most prominent areas of research because of its significant contribution towards Structural graph theory and Combinatorics. A decomposition of graph G is the collection of connected edge disjoint subgraphs $G_1, G_2, \dots, G_n$ such that $E(G_1) \cup E(G_2) \cup \dots \cup E(G_n) = E(G)$. In literature, different types of decomposition of graph have been studied by imposing conditions on the subgraphs G_i such as decompositions given in (Merly & Jothi, 2018; Romero-Valencia *et al.,* 2019). A parameter called decomposition number is also studied along with the decomposition techniques. Some of these parameters are found in (Nagarajan *et al.,* 2009; Abraham & Hamid, 2010; Arumugam *et al.,* 2013; John & Stalin, 2021).

Motivated by the results and applications of the decomposition parameters stated in those papers, we introduced a new decomposition technique called Steiner decomposition of graphs (Merly & Mahiba, 2021a) and initiated the study of the parameter Steiner decomposition number of graphs. In (Merly & Mahiba, 2021b), Steiner decomposition number of Complete $n-$ Sun graph is presented. A Steiner decomposition is a decomposition $\pi = \{G_1, G_2, \dots, G_n\}$ such that $s(G_i) = s(G), (1 \leq i \leq n)$. The maximum cardinality of a Steiner decomposition π is called the Steiner decomposition number of G and is denoted as $\pi_{st}(G)$. A graph G is said to be Steiner decomposable graph if $\pi_{st}(G) \geq 2$. A graph G is said to be non Steiner decomposable graph if $\pi_{st}(G) = 1$.

For a connected graph G, a set $S \subseteq V(G)$ is said to be an independent set of G if no two vertices of S are adjacent in G. An independent set S is said to be maximum if G has no independent set S' with $|S'| > |S|$. The cardinality of the maximum independent set is called the independence number of G and is denoted by $\alpha(G)$. In a connected graph G, a vertex of degree one is said to be pendant vertex and a vertex whose removal makes the graph disconnected is said to be cutvertex. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two simple graphs. The union of G_1 and G_2 denoted by $G_1 \cup G_2$ is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. Star graph $K_{1,n}$ is a tree of order $n+1$ with one vertex having degree n and all other vertices having degree one. Bistar denoted by $B_{m,n}(m, n \geq 2)$ is a graph obtained by joining the central vertices of star graphs $K_{1,m}$ and $K_{1,n}$ with an edge. A spider tree is a tree with atmost one vertex of degree ≥ 3 and the vertex of degree ≥ 3 is called as branch vertex. A leg of spider tree is a path from the branch vertex to a pendant vertex of the tree. $S_n(m)$ denote a spider tree of n legs with one leg having length $m \geq 2$ and other $(n-1)$ legs having length one. $U_3(k)$ denote a unicyclic graph created from the cycle C_3 by attaching k pendant vertices to a vertex of C_3. $U_3(k_1, k_2)$ denote a unicyclic graph created from the cycle C_3 by attaching k_1 pendant vertices to a vertex of C_3 and attaching k_2 pendant vertices to another vertex of C_3.

2. Main Results

In this section we derive a relation between $\pi_{st}(G)$ and $\alpha(G)$.

Theorem 2.1. *(Merly & Mahiba, 2021a) For any graph G with q edges, $s(G) = 2$ if and only if $\pi_{st}(G) = q$.*

Theorem 2.2. *(Merly & Mahiba, 2021a) For any Steiner decomposable graph G with $s(G) \geq 3$, $\pi_{st}(G) \leq \left\lfloor \frac{q}{s(G)} \right\rfloor$.*

Theorem 2.3. *(Merly & Mahiba, 2021a) Let G be a connected graph of size q.*

a) For any Steiner decomposable graph G with $s(G) > 3$, $\pi_{st}(G) = \frac{q}{s(G)}$ if and only if $G_i = K_{1,s(G)}$ $\forall$ i.

b) For any Steiner decomposable graph G with $s(G) = 3$, $\pi_{st}(G) = \frac{q}{3}$ if and only if $G_i = K_{1,3}$ or K_3 $\forall$ i.

Theorem 2.4. *Let G be a connected graph such that $|V(G)| = p, |E(G)| = q$ and $s(G) \geq 4$. If $\pi_{st}(G) = \frac{q}{s(G)}$ then $\alpha(G) \geq |V(G) - S|$ where S is the collection of cutvertices of all the subgraphs in the Steiner decomposition of maximum cardinality.*

Proof. Let G be a connected graph on p vertices, q edges and Steiner number $s(G) \geq 4$. Assume $\pi_{st}(G) = \frac{q}{s(G)}$. This implies that $\pi = \{G_i = K_{1,s(G)} \,/\, 1 \leq i \leq \frac{q}{s(G)}\}$ is the Steiner decomposition of maximum cardinality for G. Let S be the collection of all cutvertices of G_i, $1 \leq i \leq \frac{q}{s(G)}$. Any pair of vertices in $V(G) - S$ is non adjacent in G, if not it contradicts π is a decomposition for G. Therefore $V(G) - S$ is an independent set and hence $\alpha(G) \geq |V(G) - S|$. □

Corollary 2.5. *Let G be a connected graph with $p > \frac{q}{s(G)}$. Then $\pi_{st}(G) \neq \frac{q}{s(G)}$ if $\alpha(G) < p - \frac{q}{s(G)}$.*

Proof. Assume $\alpha(G) < p - \frac{q}{s(G)}$. To prove $\pi_{st}(G) \neq \frac{q}{s(G)}$. Suppose $\pi_{st}(G) = \frac{q}{s(G)}$ then $\pi = \{G_1, G_2, \dots, G_{\frac{q}{s(G)}}\}$ is a Steiner decomposition for G. By the above theorem, $\alpha(G) \geq |V(G) - S|$ where S is the collection of all cutvertices in the decomposition π. Since $p > \frac{q}{s(G)}$, $|S| \leq \frac{q}{s(G)}$.

$$\begin{aligned}\alpha(G) &\geq |V(G) - S| \\ &= |V(G)| - |S| \ \ (since\ V(G) \supseteq S) \\ &= p - |S| \\ &\geq p - \frac{q}{s(G)}\end{aligned}$$

which is a contradiction to our assumption. Therefore $\pi_{st}(G) \neq \frac{q}{s(G)}$.

3. Steiner decomposition of power of path

Definition 3.1. *(Lin et al., 2011) The k^{th} power of the graph G denoted by G^k has the same vertex set as G and two distinct vertices u and v of G are adjacent in G^k if and only if their distance in G is atmost k.*

Definition 3.2. *Let G be a simple graph. For $S \subset V(G)$, graph $G - S$ is obtained by removing each vertex of S and all its associated incident edges from G. For $T \subset E(G)$, $G - T$ denote the graph obtained from G by deleting each edge of T.*

Let P_{n+1} denote the path of order $n + 1$. P_{n+1}^k denote the k^{th} power of path P_{n+1}. The number of edges of the graph P_{n+1}^k is $k\left((n+1) - \left(\frac{k+1}{2}\right)\right)$. If $k \geq n$ then P_{n+1}^k is the complete graph on $n + 1$ vertices. We proved that complete graph is non Steiner decomposable graph *(Merly & Mahiba, 2021a)*. Hence in this section we consider only the graphs P_{n+1}^k where $2 \leq k < n$ for our discussion.

Theorem 3.3. *(AbuGhneim et al., 2014) If $n = qk + r$ where q is a positive integer and $0 < r \leq k$, then $s\left(P_{n+1}^k\right) = r + 1$.*

Result 3.4. *If P_{n+1}^k is the graph with $n = qk + 1$ then $\pi_{st}\left(P_{n+1}^k\right) = \frac{k}{2}(2n - k + 1)$.*

Since $n = qk + 1$, $s\left(P_{n+1}^k\right) = 2$. By theorem 2.1, the result is attained.

Result 3.5. *For P_{mk}^k where $m > 1$, $s\left(P_{mk}^k\right) = k$.*

Since $mk - 1 = (m-1)k + (k-1)$ by theorem 3.3, $s\left(P_{mk}^k\right) = k$.

Lemma 3.6. $\alpha\left(P_{n+1}^k\right) = \left\lceil\frac{n+1}{k+1}\right\rceil$.

Proof. Let $V\left(P_{n+1}^k\right) = \{v_1, v_2, \dots, v_{n+1}\}$. Let $V_j = \left\{v_{(j-1)k+j}, v_{(j-1)k+j+1}, \dots, v_{jk+j}\right\}, 1 \leq j \leq \left\lceil\frac{n+1}{k+1}\right\rceil - 1$ and $V_{\left\lceil\frac{n+1}{k+1}\right\rceil} = \left\{v_{\left(\left\lceil\frac{n+1}{k+1}\right\rceil-1\right)k+\left\lceil\frac{n+1}{k+1}\right\rceil}, v_{\left(\left\lceil\frac{n+1}{k+1}\right\rceil-1\right)k+\left\lceil\frac{n+1}{k+1}\right\rceil+1}, \dots, v_{n+1}\right\}$. We have, $\left|V_j\right| = k + 1, 1 \leq j \leq \left\lceil\frac{n+1}{k+1}\right\rceil - 1$ and $\left|V_{\left\lceil\frac{n+1}{k+1}\right\rceil}\right| \leq k + 1$. Generate the set S by choosing the first vertex from the vertex subsets V_j, $1 \leq j \leq \left\lceil\frac{n+1}{k+1}\right\rceil$. The set thus formed will be $S = \left\{v_1, v_{k+2}, v_{2k+3}, \dots, v_{\left(\left\lceil\frac{n+1}{k+1}\right\rceil-1\right)k+\left\lceil\frac{n+1}{k+1}\right\rceil}\right\}$. For any two distinct vertices of S, their distance in P_{n+1} is atleast $k + 1$ and so they are non adjacent in P_{n+1}^k. Therefore S is an independent set. Suppose there exists an independent set S' with $|S'| > |S|$ then atleast two vertices of S' belong to the same vertex subset V_m (say). In P_{n+1}^k, any pair of vertices of V_j, $1 \leq j \leq \left\lceil\frac{n+1}{k+1}\right\rceil$ is adjacent. This contradicts that S' is an independent set. Hence S is a maximum independent set. Thus $\alpha\left(P_{n+1}^k\right) = \left\lceil\frac{n+1}{k+1}\right\rceil$.

Throughout the section we consider the vertex set of $G = P_{n+1}^k$ as $V(G) = \{v_1, v_2, \dots, v_{n+1}\}$. Define the set A_i for $1 \leq i \leq n$ as $A_i = \left\{v_{i+j} / 1 \leq j \leq k, i + j \leq n + 1\right\}$. Construct the decomposition $\psi = \{H_1, H_2, \dots, H_n\}$ such that H_i, $1 \leq i \leq n$ is a star graph with cut vertex as v_i and the vertices of A_i as pendant vertices. Construction of the subgraphs $H_i \in \psi, 1 \leq i \leq n$ is shown in figure 1. By making necessary alterations on H_i's belonging to ψ, we obtain the desired Steiner decomposition.

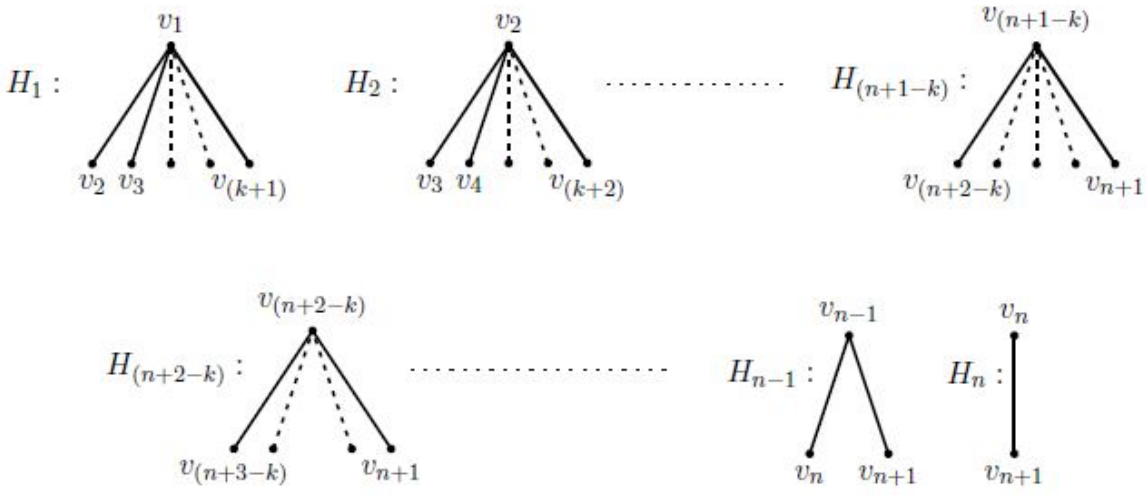


Fig. 1. Decomposition ψ of P_{n+1}^k

Theorem 3.7. *For the graph* $G = P_{mk}^k$ *with* $k = 2n, n > 1, \pi_{st}(G) = mk - n - 1$.

Proof. Let $G = P_{mk}^k$ and $k = 2n, n > 1$. The decomposition ψ can be reframed and written as $\psi = \{H_1, H_2, \dots, H_{mk-3n-1}\} \cup \{H_{mk-3n}, H_{mk-3n+1}, \dots, H_{mk-2n-1}\} \cup \{H_{mk-2n}, H_{mk-2n+1}, \dots, H_{mk-n-1}\} \cup \{H_{mk-n}, H_{mk-n+1}, \dots, H_{mk-1}\}$.

Let us define $G_s^* = H_{mk-3n+s} \cup H_{mk-n+s}, 0 \le s \le n-2$. Let $V'(G_s^*) \subset V(G_s^*), 0 \le s \le n-2$ such that $V'(G_s^*) = \{v_{mk-2n+(s+1)}, v_{mk-2n+(s+2)}, \dots, v_{mk-2n+(n-1)}\}$.

To obtain the Steiner decomposition of the graph, define

$$G_l = H_l, 1 \le l \le mk - 3n - 1$$

$$G_{mk-3n+s} = G_s^* - V'(G_s^*), 0 \le s \le n-2$$

$$G_{mk-2n-1} = H_{mk-2n-1} \cup H_{mk-1}$$
$$G_{mk-2n} = H_{mk-2n}$$

Construct $G_{mk-2n+r}, 1 \le r \le n-1$ from the graph $H_{mk-2n+r}$ by attaching the edges removed from $G_s^*, 0 \le s \le n-2$ in the process of constructing $G_{mk-3n+s}, 0 \le s \le n-2$ with one of the end vertex as $v_{mk-2n+r}$.

Consider the decomposition of P_{mk}^k, k even as $\pi = \{G_l / 1 \le l \le mk - 3n - 1\} \cup \{G_{mk-3n+s} / 0 \le s \le n-2\} \cup \{G_{mk-2n-1}, G_{mk-2n}\} \cup \{G_{mk-2n+r} / 1 \le r \le n-1\}$.

$$G_l, G_{mk-2n} \cong K_{1,k}, 1 \le l \le mk - 3n - 1$$

$$G_{mk-2n-1} \cong S_k(2)$$

$$G_{mk-3n+s} \cong B_{n+s,n-s}, 0 \le s \le n-2$$

$$G_{mk-2n+r} \cong K_{1,k}, 1 \leq r \leq n-1$$

By the result 3.5, Steiner number of G is k. Since $s(K_{1,k}) = s(S_k(2)) = k$ and $s(B_{n+s,n-s}) = 2n = k$, decomposition $\pi = \{G_1, G_2, \dots, G_{mk-n-1}\}$ is a Steiner decomposition for G. Now to prove $\pi_{st}(G) = mk - n - 1$. By theorem 2.2, $\pi_{st}(G) \leq \left\lfloor \frac{q}{s(G)} \right\rfloor$. On calculating the value of $\left\lfloor \frac{q}{s(G)} \right\rfloor$,

$$\begin{aligned}
\left\lfloor \frac{q}{s(G)} \right\rfloor &= \left\lfloor mk - \left(\frac{k+1}{2}\right) \right\rfloor \\
&= \left\lfloor \frac{2mk-(k+1)}{2} \right\rfloor \\
&= \frac{2mk-(k+1)-1}{2} \quad (since\ 2mk-(k+1)\ is\ odd) \\
&= \frac{4mn-(2n+1)-1}{2} \\
&= 2mn - n - 1 \\
&= mk - n - 1 \\
&= cardinality\ of\ \pi
\end{aligned}$$

Therefore π is a Steiner decomposition of maximum cardinality for G and so $\pi_{st}(G) = mk - n - 1$.

Theorem 3.8. *Let $G = P_{mk}^k$. If k is odd and $1 < m < \frac{k+1}{2}$ then $\pi_{st}(G) = mk - n - 1$.*

Proof. Let $G = P_{mk}^k$, where $k = 2n-1, n \geq 3$ and $1 < m < \frac{k+1}{2}$. The decomposition ψ can be reframed and written as $\psi = \{H_1, H_2, \dots, H_{mk-3n}\} \cup \{H_{mk-3n+1}, H_{mk-3n+2}, \dots, H_{mk-2n-2}\} \cup \{H_{mk-2n-1}, H_{mk-2n}, H_{mk-2n+1}\} \cup \{H_{mk-2n+2}, H_{mk-2n+3}, \dots, H_{mk-n-1}\} \cup \{H_{mk-n}, H_{mk-n+1}, \dots, H_{mk-3}, H_{mk-2}, H_{mk-1}\}$.

Let us define $G_s^* = H_{mk-3n+(s+1)} \cup H_{mk-n+s}$, $0 \leq s \leq n-3$. Let $V'(G_s^*) \subset V(G_s^*)$, $0 \leq s \leq n-3$ such that $V'(G_s^*) = \{v_{mk-3n+(s+2)}, v_{mk-3n+(s+3)}, \dots, v_{mk-2n-1}, v_{mk-n-(s+1)}\}$.

To obtain the Steiner decomposition of the graph, define

$$G_l = H_l, 1 \leq l \leq mk - 3n$$

$$G_{mk-3n+(s+1)} = G_s^* - V'(G_s^*), 0 \leq s \leq n-3$$

$$G_{mk-2n-1} = H_{mk-2n-1}$$

$$G_{mk-2n} = H_{mk-2n} \cup H_{mk-2}$$

$$G_{mk-2n+1} = H_{mk-2n+1} \cup H_{mk-1}$$

Let $E'(G_s^*)$, $0 \leq s \leq n-3$ be the set of edges removed from $E(G_s^*)$ while constructing $G_{mk-3n+(s+1)}$.

Construct $G_{mk-n-(s+1)}$, $0 \leq s \leq n-3$ from the graph $H_{mk-n-(s+1)}$ by attaching the edges in the set $E'(G_s^*)$.

Consider the decomposition of G as $\pi = \{G_l / \ 1 \leq l \leq mk-3n\} \cup \{G_{mk-3n+(s+1)} / \ 0 \leq s \leq n-3\} \cup \{G_{mk-2n-1}, G_{mk-2n}, G_{mk-2n+1}\} \cup \{G_{mk-n-(s+1)} / \ 0 \leq s \leq n-3\}$.

$$G_l, G_{mk-2n-1} \cong K_{1,k}, 1 \leq l \leq mk-3n$$

$$G_{mk-2n} \cong U_3(1, k-2)$$

$$G_{mk-2n+1} \cong U_3(k-2)$$

$$G_{mk-3n+(s+1)} \cong B_{n-s,n-1+s}, 0 \leq s \leq n-3$$

$$G_{mk-n-(s+1)} \cong B_{n-(s+2),n+(s+1)}, 0 \leq s \leq n-4$$

$$G_{mk-2n+2} \cong S_k(2)$$

Since $s(K_{1,k}) = s(U_3(1,k-2)) = s(U_3(k-2)) = s(B_{n-s,n-1+s}) = s(B_{n-(s+2),n+(s+1)}) = s(S_k(2)) = k = s(G)$, π is a Steiner decomposition for G. The cardinality of π is $mk-n-1$. Now, we have to prove $\pi_{st}(G) = mk-n-1$. From lemma 3.6, $S = \left\{v_1, v_{k+2}, v_{2k+3}, \dots, v_{\left(\left\lfloor \frac{mk}{k+1} \right\rfloor - 1\right)k + \left\lfloor \frac{mk}{k+1} \right\rfloor}\right\}$ is a maximum independent set for G. We have, $(m-1)k + m = mk - (k-m)$. Since $m < \frac{k+1}{2}, k-m > k - \left(\frac{k+1}{2}\right)$. For $k > 1, k - \left(\frac{k+1}{2}\right) > 0$ and so $k - m > 0$. This implies $(m-1)k + m < mk$. We know that distance between any pair of vertices belonging to S is atleast $k+1$ in the graph G and clearly $mk + (m+1) > mk$. Hence we can conclude $\left\lceil \frac{mk}{k+1} \right\rceil = m$ and so $\alpha(G) = m$.

$$\frac{q}{s(G)} = mk - \left(\frac{k+1}{2}\right) \quad (since\ k\ is\ odd, mk - \left(\frac{k+1}{2}\right)\ is\ an\ integer)$$

$$p - \frac{q}{s(G)} = \frac{k+1}{2}$$

$$> m$$

$$= \alpha(G)$$

Therefore, $\alpha(G) < p - \frac{q}{s(G)}$

Also we have, $p > \frac{q}{s(G)}$. Hence by corollary 2.5, $\pi_{st}(G) \neq mk - \left(\frac{k+1}{2}\right)$. That is $\pi_{st}(G) \neq mk - n$. Hence π is a Steiner decomposition for G with maximum cardinality. Therefore $\pi_{st}(G) = mk - n - 1$.

Theorem 3.9. *For* $G = P^4_{5^2+20m}$ *with* $m \geq 0, \pi_{st}(G) = 17 + 16m$.

Proof. Let $G = P^4_{5^2+20m}$, $m \geq 0$ be the graph with order p and size q.

$$p - 1 = 5^2 + 20m - 1$$
$$= 24 + 20m$$
$$= 4\big(5(1+m)\big) + 4$$

By theorem 3.3, $s(G) = 5$. The decomposition ψ can be reframed and written as $\psi = \{H_1, H_2, \dots, H_{20(1+m)}\} \cup \{H_{21+20m}, H_{22+20m}, H_{23+20m}, H_{24+20m}\}$

Define

$$G_{jk} = H_{k+5j} \cup < \{v_{k+5j}v_{1+5j}\} >;\ 0 \leq \mathrm{j} \leq 3 + 4m, k = 2,3,4,5$$

$$G^* = H_{21+20m} \cup H_{22+20m} \cup H_{23+20m} \cup H_{24+20m}$$

Clearly $\pi = \{G_{jk} / \ 0 \leq j \leq 3 + 4m, k = 2,3,4,5\} \cup \{G^*\}$ is a decomposition for G.

$$G_{jk} \cong K_{1,5}\ ;\ 0 \leq \mathrm{j} \leq 3 + 4m, k = 2,3,4,5$$

$$G^* \cong K_5$$

Since $s(K_{1,5}) = s(G^*) = 5, \pi$ is a Steiner decomposition for G. The cardinality of π is $17 + 16m$.

Now,

$$\frac{q}{s(G)} = \frac{4\left(\left(5^2+20m\right)-\frac{5}{2}\right)}{5}$$
$$= 18 + 16m$$
$$\frac{q}{s(G)} = 18 + 16m < 5^2 + 20m = p$$

Therefore, $p > \frac{q}{s(G)}$.

$$\alpha(G) = \left\lceil \frac{5^2 + 20m}{5} \right\rceil$$
$$= 5 + 4m \qquad (1)$$

$$p - \frac{q}{s(G)} = 5^2 + 20m - (18 + 16m)$$
$$= 7 + 4m \qquad (2)$$

From Equations (1) & (2),

$$\alpha(G) < p - \frac{q}{s(G)}$$

Hence by corollary 2.5, $\pi_{st}(G) \neq 18 + 16m$. Therefore π is a Steiner decomposition with maximum cardinality and so $\pi_{st}(G) = 17 + 16m$.

4. Realization Theorem

Definition 4.1. *The contraction of pair of vertices v_i and v_j of a graph produces a graph in which the two vertices v_i and v_j are replaced by the new vertex v such that v is adjacent to the union of vertices to which v_i, v_j were originally adjacent.*

Definition 4.2. *(Ghosh et al., 2021) Globe graph (Gl_n) is obtained from two isolated vertices that are joined by n paths of length two.*

Theorem 4.3. *For any positive integer m, n $(m \geq 2)$ there exists a connected graph G such that $s(G) = m$ and $\pi_{st}(G) = n$.*

Proof. **Case 1:** $m \leq n$

Subcase 1: $m = 2$

Path graph on $n + 1$ vertices, P_{n+1} satisfies the required properties.

Subcase 2: $m > 2$

For $2 < m \leq n$, the Complete bipartite graph $G = K_{m,n}$ has the properties $s(G) = m$ and $\pi_{st}(G) = n$.

Case 2: $m > n$

Subcase 1: $n = 1$

Star graph $K_{1,m}$ is a non Steiner decomposable graph with $s(K_{1,m}) = m$. Therefore it satisfies the required properties.

Subcase 2: m, n both odd and $n \geq 3$

Construct the graph with the desired properties as follows:

- Take $\frac{n-1}{2}$ copies of the globe graph $Gl_{\frac{m+1}{2}}$. Label the two vertices of degree $\frac{m+1}{2}$ in each copy of $Gl_{\frac{m+1}{2}}$ as u_i and $v_i, 1 \leq \text{i} \leq \frac{n-1}{2}$ respectively.
- Take $\frac{n-1}{2}$ copies of the globe graph $Gl_{\frac{m-1}{2}}$. Label the two vertices of degree $\frac{m-1}{2}$ in each copy of $Gl_{\frac{m-1}{2}}$ as x_i and $y_i, 1 \leq \text{i} \leq \frac{n-1}{2}$ respectively.
- Consider the set $S = \{(v_i, x_i)/\ 1 \leq i \leq \frac{n-1}{2}\} \cup \{(y_i, u_{i+1})/\ 1 \leq i \leq \frac{n-3}{2}\}$. By vertex contraction process, contract the pair of vertices given in each ordered pair of S.

- Take a copy of the star graph $K_{1,\frac{m-1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex u_1.
- Take a copy of the star graph $K_{1,\frac{m+1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex $y_{\frac{n-1}{2}}$.

In figure 2, the resultant graph G and its Steiner decomposition indicated by horizantal lines is given.

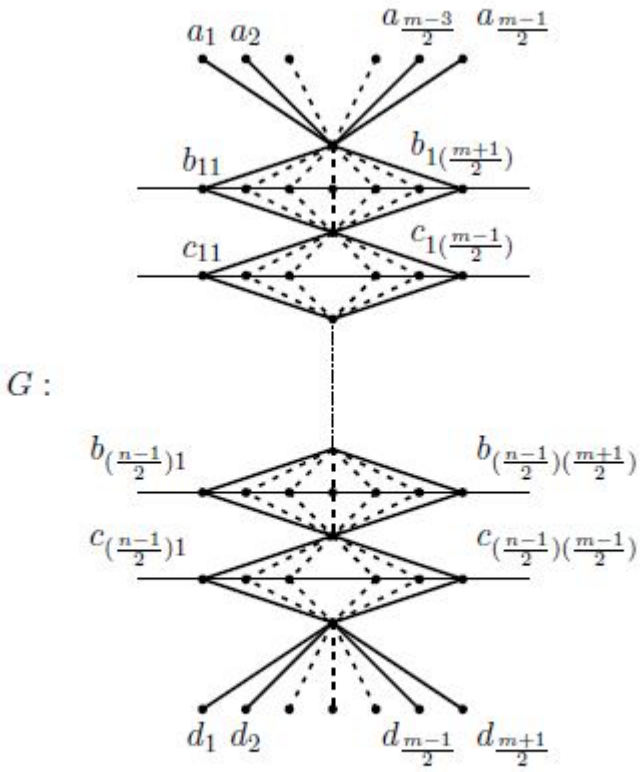


Fig. 2. Graph G with m, n both odd, $n \geq 3$ and $m > n$

Total number of edges of G is mn. Minimum Steiner set of $G = \left\{a_i / \ 1 \leq i \leq \frac{m-1}{2}\right\} \cup \{d_i / \ 1 \leq i \leq \frac{m+1}{2}\}$ and so $s(G) = m$. Since each subgraph in the decomposition is the star graph $K_{1,m}$ by theorem 2.3, $\pi_{st}(G) = n$.

Subcase 3: m even

Construct the graph with the desired properties as follows:

- Take $(n-1)$ copies of the globe graph $Gl_{\frac{m}{2}}$. Label the two vertices of degree $\frac{m}{2}$ in each copy of $Gl_{\frac{m}{2}}$ as u_i and $v_i, 1 \leq i \leq n-1$ respectively.
- Consider the set $S = \{(v_i, u_{i+1})/\ 1 \leq i \leq n-2\}$. By vertex contraction process, contract the pair of vertices given in each ordered pair of S.
- Take a copy of the star graph $K_{1,\frac{m}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex u_1.
- Take another copy of the star graph $K_{1,\frac{m}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex v_{n-1}.

In figure 3, the resultant graph G and its Steiner decomposition indicated by horizantal lines is given.

Total number of edges of G is mn. Minimum Steiner set of $G = \{a_i/\ 1 \leq i \leq \frac{m}{2}\} \cup \{c_i/\ 1 \leq i \leq \frac{m}{2}\}$ and so $s(G) = m$. Since each subgraph in the decomposition is the star graph $K_{1,m}$ by theorem 2.3, $\pi_{st}(G) = n$.

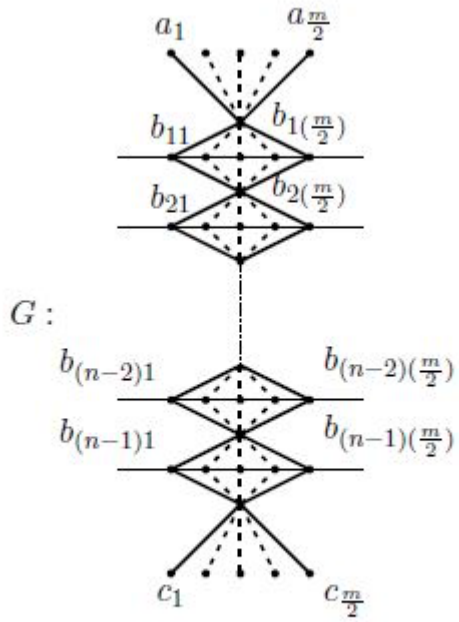


Fig. 3. Graph G with m even and $m > n$

Subcase 4: m odd and n even $(n > 2)$

Construct the graph with the desired properties as follows:

- Take a copy of the globe graph $Gl_{\frac{m+1}{2}}$. Label the two vertices of degree $\frac{m+1}{2}$ as u_1 and v_1 respectively.

- Take a copy of the star graph $K_{1,\frac{m-1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex u_1. Label the new vertex as u_1^*.
- Take another copy of the star graph $K_{1,\frac{m-1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex v_1. Label the new vertex as v_1^*.
- Take ($\frac{n}{2}-1$) copies of the globe graph Gl_m. Label the two vertices of degree m in each copy of Gl_m as x_i and y_i, $1 \leq \mathrm{i} \leq \frac{n}{2}-1$ respectively.
- Consider the set $S=\{(y_i, x_{i+1})/\ 1 \leq i \leq \frac{n}{2}-2\} \cup \{(v_1^*, x_1)\}$. By vertex contraction process, contract the pair of vertices given in each ordered pair of S.

In figure 4, the resultant graph G and its Steiner decomposition indicated by horizantal lines is given.

Total number of edges of G is mn. Minimum Steiner set of $G=\{a_i/\ 1 \leq i \leq \frac{m-1}{2}\} \cup \{c_i/\ 1 \leq i \leq \frac{m-1}{2}\} \cup \{y_{\frac{n}{2}-1}\}$ and so $s(G)=m$. Since each subgraph in the decomposition is the star graph $K_{1,m}$ by theorem 2.3, $\pi_{st}(G)=n$.

Subcase 5: m odd and $n=2$

Construct the graph with the desired properties as follows:

- Take a copy of the globe graph $Gl_{\frac{m+1}{2}}$. Label the two vertices of degree $\frac{m+1}{2}$ as u_1 and v_1 respectively.

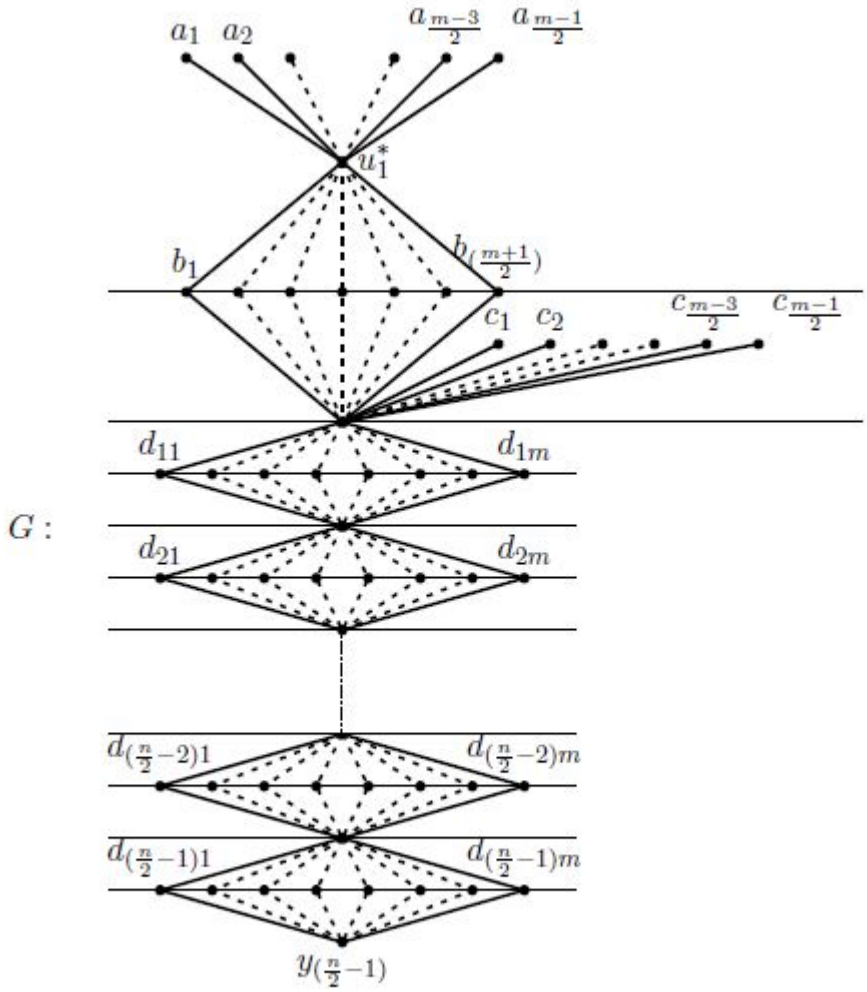


Fig. 4. Graph G with m odd, n even $(n > 2)$ and $m > n$

- Take a copy of the star graph $K_{1,\frac{m-1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex u_1.
- Take a copy of the star graph $K_{1,\frac{m+1}{2}}$ and by vertex contraction process, contract its cut vertex with the vertex v_1.

In figure 5, the resultant graph G and its Steiner decomposition is given.

Total number of edges of G is $2m + 1$. Minimum Steiner set of $G = \{a_i / \ 1 \leq i \leq \frac{m-1}{2}\} \cup \{c_i / \ 1 \leq i \leq \frac{m+1}{2}\}$ and so $s(G) = m$. By theorem 2.2, $\pi_{st}(G) \leq 2$ and since $\pi = \{G_1, G_2\}$ is a Steiner decomposition of cardinality 2, $\pi_{st}(G) = 2$.

Thus for any positive integers m, n $(m \geq 2)$ there exists a connected graph G such that $s(G) = m$ and $\pi_{st}(G) = n$.

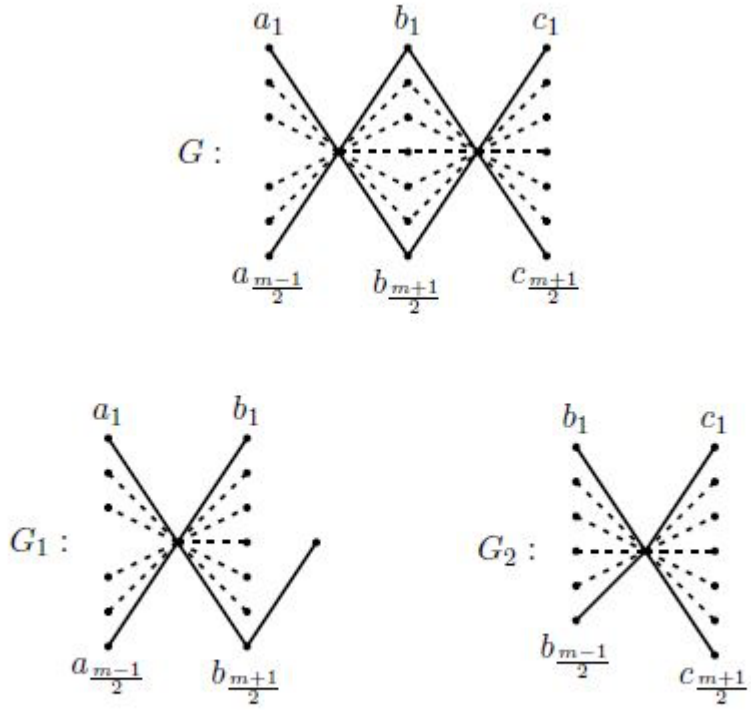


Fig. 5. Graph G with m odd, $n = 2$, $m > n$ and its Steiner decomposition

5. Conclusion

This paper is an extensive study of the decomposition parameter Steiner decomposition number of graphs. Here, a relation between independence number and Steiner decomposition number is obtained. This result plays a vital role in justifying the value of the parameter for some graph families. Also, Steiner decomposition number of some power of paths and a realization theorem is presented. Future works can be carried out on obtaining the Steiner decomposition number related bounds for any power of path and investigating the value of the parameter for other graph classes. Bounds of Steiner decomposition number of graphs based on various graph theoretical parameters can also be studied.

References

Abraham, V.M., & **Hamid, I.S. (2010)**. Induced acyclic path decomposition in graphs. International Journal of Mathematical and Computer Sciences, **6(3)**, 166-169.

AbuGhneim, O. A., Al-Khamaiseh, B., & **Al-Ezeh, H. (2014)**. The geodetic, hull, and Steiner numbers of powers of paths. Utilitas Mathematica, **95**, 289-294.

Arumugam, S., Hamid, I., & **Abraham, V.M. (2013)**. Decomposition of graphs into paths and cycles. Journal of Discrete Mathematics, **2013**, DOI: https://doi.org/10.1155/2013/721051.

Chartrand, G., & **Zhang, P. (2002)**. The Steiner number of a graph. Discrete Mathematics, **242**, 41-54.

Ghosh, P., Ghosh, S., & **Pal, A. (2017)**. 3-Total Sum Cordial Labeling on Some New Graphs. Journal of Informatics and Mathematical Sciences, **9**, 665-673.

Harary, F. (1988). Graph Theory. Narosa Publishing House, New Delhi.

Hernando, C., Jiang, T., Mora, M., Pelayo, I.M., & **Seara, C. (2005)**. On the Steiner, geodetic and hull numbers of graphs. Discrete Mathematics, **293**, 139-154.

John, J., & **Stalin, D. (2021)**. The edge geodetic self decomposition number of a graph. RAIRO Oper. Res., **55**, S1935-S1947.

Lin, M. C., Rautenbach, D., Soulignac, F. J., & **Szwarcfiter, J. L. (2021)**. Powers of cycles, powers of paths, and distance graphs. Discrete Applied Mathematics, **159**, 621-627.

Merly, E. E. R., & **Jothi, D. J. (2018)**. Connected Domination decomposition of helm graph. International Journal of Scientific Research and Review, **7(10)**, 327-331.

Merly, E. E. R., & **Mahiba, M. (2021a)**. Steiner decomposition number of graphs. Malaya Journal of Matematik, **Special Issue**, 560-563.

Merly, E. E. R., & **Mahiba, M. (2021b)**. Steiner Decomposition Number of Complete $n-$ Sun graph. Journal of Physics: Conference series, **1947**, DOI: https://doi.org/10.1088/1742-6596/1947/1/012002.

Nagarajan, K., Nagarajan, A., & **Hamid, I.S. (2009)**. Equiparity path decomposition number of a graph. International Journal of Mathematical Combinatorics, **1**, 61-76.

Pelayo, I.M. (2004). Comment on "The Steiner number of a graph" by G. Chartrand and P. Zhang:[Discrete Mathematics 242 (2002) 41-54]. Discrete Mathematics, **280**, 259-263.

Romero-Valencia, J., Hernández-Gómez, J.C., & **Reyna-Hernández, G. (2019)**. On the inverse degree index and decompositions in graphs. Kuwait Journal of Science, **46(4)**, 14-22.

Yero, I.G., & **Rodriguez-Velazquez, J.A. (2015)**. Analogies between the geodetic number and the Steiner number of some classes of graphs. Filomat, **29**, 1781-1788.

Submitted: 23/10/2021
Revised: 29/03/2022
Accepted: 10/04/2022
DOI: 10.48129/kjs.16863

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-8)

$\mathcal{Z}$-graphic topology on undirected graph

Hanan Omer Zomam[1,2], Makkia Dammak[3,*]
[1]*Dept. of Mathematics, College of Science, Taibah University, Al-Madinah Al-Munawarah, Saudi Arabia.*
[2] *Dept. of Mathematics, Faculty of Science, Shendi University, Sudan.*
[3] *Dept. of Mathematics, Faculty of Sciences, University of Sfax, Tunisia.*
**Corresponding author: makkia.dammak@gmail.com*

Abstract

In this work, we define $\mathcal{Z}_G$ a topology on the vertex set of a graph G which preserves the connectivity of the graph, called $\mathcal{Z}$-graphic topology. We prove that two isomorphic graphs have homeomorphic and symmetric $\mathcal{Z}$-graphic topologies. We show that $\mathcal{Z}_G$ is an Alexandroff topology and we give a necessary and sufficient condition for a topology to be $\mathcal{Z}$-graphic.

Keywords: Connected components; homeomorphism; graph; symmetric topologies; topology.

1. Introduction

Graph theory is a field applied to many domains. When we discretize a problem by a graph, the properties of the graph help to study the given problem. Having a topology on the graph gives a richer structure to the graph and this have applications in the economy domain, the traffick flow study (Agnarsson *et al.*, 2007; Kandel *et al.*, 2007; Nogly *et al.*, 1996) and many other domains. Also, a graph can be characterized by some topological indices, see (Ali *et al.*, 2016; Cruz *et al.*, 2021; Gutman *et al.*, 2021; Naji *et al.*, 2018) and references therein.

Since the publication of the paper (Jafarian Amiri *et al.*, 2013), other researchers defined some topologies on graphs, as example we can cite (Abdu *et al.*, 2018; Hamza *et al.*, 2013; Kilicman *et al.*, 2018; Sasikala *et al.*, 2019; Shokry, 2015). In (Jafarian Amiri *et al.*, 2013), the authors defined the graphic topology τ_G on a locally finite (i.e. any vertex has a finite order) undirected graph $G = (V, E)$ with no isolated vertices by the subbasis:

$$S_G = \{A_x \mid x \in V\}, \tag{1}$$

where

$$A_x = \{z \in V \mid xz \in E\}. \tag{2}$$

One of the most interesting properties of (V, τ_G) was being an Alexandroff space, that is any intersection of open sets is an open set. This is equivalent to the topology has a unique minimal basis. The Alexandroff spaces were introduced by P. Alexandroff in 1937 in (Alexandroff, 1937) under the name Diskrete Räume spaces. We can find some results about these spaces and their importance and applications in (Herman, 1990; Kronheimer, 1992; Li *et al.*, 2019; McCord, 1966; Stong, 2015; Speer, 2007).

A topological space (V, T) is called graphic space if there exists a graph G such that $T = \tau_G$. In (Jafarian Amiri *et al.*, 2013), the authors posed two open problems: when an Alexandroff space can be graphic? When the graphic topology can be connected?
In (Zomam *et al.*, 2021), a partial answer to the first question was given. In this paper, we define a topology $\mathcal{Z}_G$ on the vertex set of an underacted graph $G = (V, E)$ such that $\mathcal{Z}_G$ is smaller than τ_G, when G is locally finite without isolated vertices, that is $\mathcal{Z}_G \subset \tau_G$. Also, we solve the two open problems of (

Jafarian Amiri *et al.*, 2013) for the $\mathcal{Z}$-graphic topology $\mathcal{Z}_G$.

The outlines of this paper are the following: Section 2 deals with some basic definitions and notations. In section 3, we define $\mathcal{Z}_G$ for an undirected graph $G = (V, E)$ and we prove that it is a topology on V, smaller than τ_G when τ_G exists. We investigate the trace topology of $\mathcal{Z}_G$ on subgraphs of G. In section 4, we prove the equivalence between the connectivity of the graph G and the $\mathcal{Z}$-graphic topology $\mathcal{Z}_G$. And we show that $\mathcal{Z}_G$ is an Alexandroff topology. Finally, in section 5 we prove that being $\mathcal{Z}$-graphic is a topology property and two isomorphic graphs have homeomorphic and symmetric $\mathcal{Z}$-graphic topologies.

2. Preliminaries

In this section, we give some general definitions and properties of a topological space. For more details, we can refer to (Arenas, 1937; Dugundji, 1966; Li *et al.*, 2019; Stong, 2015).
Recall that a topological space (X, T) is a non empty set X with a set T of subsets of X (i.e $T \subset \mathcal{P}(V)$) satisfying:

(i) $\emptyset$ and X are in T.

(ii) If A and B are two subsets of X and $A, B \in T$, then $A \cap B \in T$.

(iii) For any family $\{A_i\}_{i\in I} \subset T$, I a set, we have $\cup_{i\in I} A_i \in T$.

An element A of T will be called an open set of the space (X, T).

Example 1 *Let $X = \{a, b, c\}$, then*

$$T = \{\emptyset, \{a\}, \{b\}, \{a, c\}, \{a, b\}, X\}$$

is a topology for X.

In general, the intersection of open sets is not an open set in a topological space (X, T).

Definition 2.1 *(Alexandroff, 1937) A topological space is called an Alexandroff space if any intersection of open sets is an open set. Also, we say that the topology T is an Alexandroff topology of X.*

The space introduced in Example 1 is an Alexandroff space. In fact, any finite topological space is an Alexandroff space. Later, we will give an example of a non Alexandroff space.

Definition 2.2 *Let (X, T) be a topological space and let $\mathcal{B} \subset T$. $\mathcal{B}$ is called a basis of the topology T if for all $x \in X$, for all O_x an open set containing x, there exists an element $B \in \mathcal{B}$ such that $x \in B \subset O_x$. We say that the topology is generated by the basis $\mathcal{B}$.*

Example 2 *$\mathcal{B} = \{(a, b), -\infty < a < b < +\infty\}$ is a basis for the usual topology T on R.*

Now, if we consider the open sets

$$\Big(-\frac{1}{n}, \frac{1}{n}\Big), \;\; n > 0,$$

we have

$$\bigcap_{n>0} \Big(-\frac{1}{n}, \frac{1}{n}\Big) = \{0\},$$

and so, (R, T) is not an Alexandroff space.
A basis m is called minimal basis for a topology T if for all $\mathcal{B}$ a basis of T, we have $m \subset \mathcal{B}$.

Example 3 *For the topology given in the Example 1, $m = \{\{a\}, \{b\}, \{a, c\}\}$ is a minimal basis.*

Proposition 2.1 *Let (X, T) be an Alexandroff space. Then, T has a minimal basis.*

Proof. Let $x \in X$. The intersection of all open sets containing x is an open set. We set U_x such open set. Consider $\mathcal{U} = \{U_x, \;\; x \in X\}$. We have $\mathcal{U} \subset T$ and, if $x \in X$ and O_x an open set containing x, then $x \in U_x \subset O_x$. Hence, $\mathcal{U}$ is a basis for T.
Now, let B be a basis for the topology T. Since U_x is an open set containing x, there exists $B \in B$ such that $x \in B \subset U_x$ and so $B = U_x$. Hence, $U_x \in B$ and so, $\mathcal{U} \subset B$.

3. $\mathcal{Z}$-graphic topology and some properties

In the sequel, we suppose that all graphs are simple and undirected.
Let $G = (V, E)$ be a graph. In this part, we define a subset $\mathcal{Z}_G$ of the power set $\mathcal{P}(V)$ of V and we prove that $\mathcal{Z}_G$ is a topology on the vertex set V. We call the topology $\mathcal{Z}_G$ the $\mathcal{Z}$-graphic topology of the graph G. We compare the $\mathcal{Z}$-graphic topology and the graphic topology on a graph G. Finally, we study the $\mathcal{Z}$-graphic topologies on subgraphs.

Definition 3.1 *Let $G = (V, E)$ be a graph and $A \subset V$. $A \in \mathcal{Z}_G$ if and if for any vertex $x \in A$, if there exists a path joining x to a vertex y in G then $y \in A$.*

Notation. When two vertices x and y are adjacent, we write $x \sim y$ and when they are joined by a path P, we denote $x \sim_P y$. In particular, $x \sim y$ means $x \sim_{x,y} y$ $(P = x, y)$.

Theorem 3.1 *For any graph $G = (V, E)$, $\mathcal{Z}_G$ is a topology on the vertex set V.*

Proof. (i) By definition, $\emptyset$ and V are in $\mathcal{Z}_G$.
(ii) Let A_1 and A_2 two elements in $\mathcal{Z}_G$. Suppose that $x \in A_1 \cap A_2$ and let $y \in V$ such that x joined by a path P to y: $x \sim_P y$.
We get $x \in A_1$ and $x \sim_P y$, so $y \in A_1$ since $A_1 \in \mathcal{Z}_G$.
In a similar way $y \in A_2$ and then $y \in A_1 \cap A_2$. Therefore $A_1 \cap A_2 \in \mathcal{Z}_G$.
(iii) Let $\{A_i\}_{i \in I}$ a countable infinite family of elements in $\mathcal{Z}_G$. Let $x \in \cup_{i \in I} A_i$ and suppose $y \in V$ such that $x \sim_P y$.
Since $x \in \cup_{i \in I} A_i$, there exists $i_0 \in I$ such that $x \in A_{i_0}$. From the fact that $A_{i_0} \in \mathcal{Z}_G$, we get $y \in A_{i_0}$. Therefore, $y \in \cup_{i \in I} A_i$ and then the Theorem 3.1 follows.

Theorem 3.2 *Let $G = (V, E)$ be a graph. If G is locally finite without isolated vertices, then $\mathcal{Z}_G \subset \tau_G$.*

Proof. Let $A \in \mathcal{Z}_G$. Then, $A = \cup_{x \in A} A_x$, where A_x, given by Equation 2. Indeed, If $x \in A$ and $y \in A_x$, then $x \sim_{x,y} y$. Since $A \in \mathcal{Z}_G$, the vertex $y \in A$. That is $A_x \subset A$ and then $\cup_{x \in A} A_x \subset A$.
Conversely, Let $y \in A$. Since G is without isolated vertices, there exists $x \in V$ such that $x \sim y$. So, $y \in A_x$. Also, we have: $A \in \mathcal{Z}_G$, $y \in A$ and $y \sim x$. Therefore, $x \in A$ and $y \in A_x$. Hence $y \in \cup_{x \in A} A_x$ and then $A \subset \cup_{x \in A} A_x$.

Now, since $A = \cup_{x \in A} A_x$, by definition of τ_G we have $A \in \tau_G$.

In the next example, we show that the two topologies $\mathcal{Z}_G$ and τ_G are different.

Example 4

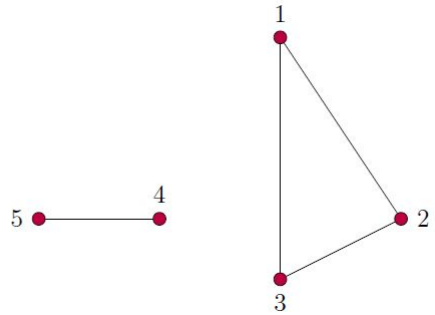


Fig. 1. *Graph with $\mathcal{Z}_G \neq \tau_G$*

In this example, $\mathcal{Z}_G = \{\emptyset, \{4, 5\}, \{1, 2, 3\}, V\}$ and τ_G is the discrete topology.

Recall that a subgraph of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$and $E' \subset E$. On the set V' we can define the $\mathcal{Z}$-graphic topology $\mathcal{Z}_H$ and we have also the topology induced by $\mathcal{Z}_G$, denoted $\mathcal{Z}_{G,H}$.

Theorem 3.3 *Let $G = (V, E)$ be a graph and $H = (V', E')$ be a subgraph of G. Then, $\mathcal{Z}_H = \mathcal{Z}_{G,H}$.*

Proof. Let $A \in \mathcal{Z}_{G,H}$. Then there exist $O \in \mathcal{Z}_G$ such that $A = O \cap V'$. Suppose that $x \in A$ and $y \in V'$ satisfying $x \sim_P y$ for some path P in H. We get $x \in O$, $y \in G$ and $x \sim_P y$ with P in G. Hence, $y \in O$ and so $y \in O \cap V'$, that is, $y \in A$. So, $A \in \mathcal{Z}_H$.
Conversely, suppose that $A \in \mathcal{Z}_H$ and $A \neq \emptyset$. As in the proof of Theorem 3.2, we prove that $A = \cup_{x\in A}(A_x \cap V')$. Therefore $A = (\cup_{x\in A}A_x) \cap V'$. But $\cup_{x\in A}A_x$ is not necessary in $\mathcal{Z}_G$ as we will see in the Example 2 below. Let us consider C_x the connected component of G containing x. Since $A \in \mathcal{Z}_H$, then $A = \cup_{x\in A}(C_x \cap V')$. Or C_x is an open set of $(V, \mathcal{Z}_G)$ and $A = (\cup_{x\in A}C_x) \cap V'$, it follows that $A \in \mathcal{Z}_{G,H}$.

Example 5 *Consider the following graph G.*

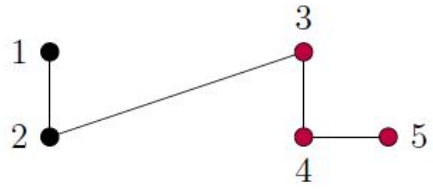


Fig. 2. *$\mathcal{Z}$-graphic topology and subgraph*

Let $H = (V', E')$ with $V' = \{1, 2\}$ and $E' = \{(1, 2)\}$. For $A = V' = \{1, 2\}$, in the graph G, we have $\cup_{x\in A}A_x = \{1, 2, 3\}$ and $\mathcal{Z}_G = \{\emptyset, \{1, 2, 3, 4, 5\}\}$.

4. $\mathcal{Z}$-graphic topology and connectedness

In this section, we will prove the equivalence between the connectivity of a graph G and the connectivity of its Z-graphic topology. Recall that the empty set is called a trivial open set in a topological space V and an open set is called proper if it is not equal to V.

Definition 4.1 *Let V be a topological space. V is called connected if it cannot be written as the union of two proper disjoint open sets. If $\mathcal{T}$ is the topology of V, we say that the topology $\mathcal{T}$ is connected.*

Example 3. Consider $V = \{1, 2, 3\}$, $\tau_1 = \{\emptyset, \{1\}, \{1, 2\}, \{1, 3\}, V\}$
and $\tau_2 = \{\emptyset, \{1\}, \{2, 3\}, V\}$. It is clear that τ_1 is connected but the topology τ_2 is not connected.

Definition 4.2 *Let $G = (V, E)$ be a graph. G is called connected if any two vertices can be joined by a path, that is, there exists a path in G from one to the other vertex.*

When a graph is not connected, we can define its connected components.

Definition 4.3 *(Agnarsson et al., 2007; Diestel, 2005) Let $G = (V, E)$ be a graph. Let $H_1 = (V_1, E_1), H_2 = (V_2, E_2), \cdots$ be connected subgraphs of G such that*

(i) $V = \cup_i V_i$;

(ii) $E = \cup_i E_i$;

(iii) $V_i \cap V_j = \emptyset$, *for all* $i \neq j$;

(iv) $E_i \cap E_j = \emptyset$, *for all* $i \neq j$.

Then, each subgraph H_j *is called connected component of the graph* G.

Remark 4.1 *When a graph* G *is connected, it has one connected component and if it is finite, it has a finite connected components.*

We have the following results with an immediate proof for the first theorem, so we omit it.

Theorem 4.1 *Let* $G = (V, E)$ *be a graph. The following properties hold.*

(1) The space $(V, \mathcal{Z}_G)$ *is compact if, and only if,* G *is a finite.*

(2) The topology $\mathcal{Z}_G$ *is discrete if, and only if,* G *is null graph (i.e* $E = \emptyset$*).*

Theorem 4.2 *Let* $G = (V, E)$ *be a graph. The graph* G *is connected if, and only if,* $\mathcal{Z}_G$ *is a connected topology on* V.

Proof. Suppose that the graph G is connected, that is any two points are joined by a path. From the Definition 3.1, the only open sets for $(V, \mathcal{Z}_G)$ are the empty set and the set V itself. And so, the topological space $(V, \mathcal{Z}_G)$ is connected.
Conversely, we suppose that $(V, \mathcal{Z}_G)$ is a connected topological space and we shall prove that the graph G is connected.
We argue by contradiction. Suppose that the graph G is not connected and so it has more than one connected components $H_1 = (V_1, E_1), H_2 = (V_2, E_2), \cdots$
Denote $W = \cup_{i \geq 2} V_i$. Since H_i is connected, then V_i is in $\mathcal{Z}_G$, for all i. Then, W is a proper open set satisfying $V = V_1 \cup W$ and $V_1 \cap W = \emptyset$. This makes contradiction with the fact that $(V, \mathcal{Z}_G)$ is a connected topological space. Our assumption is false, and so the graph G is connected.

Recall that a topological space is called Alexandroff space if any intersection of open sets is also open. We end this section by proving that the topology $\mathcal{Z}_G$ is an Alexandroff topology, for any graph G.

Theorem 4.3 *Consider a graph* $G = (V, E)$. *Then,* $\mathcal{Z}_G$ *is an Alexandroff topology.*

Proof. Suppose that $H_1 = (V_1, E_1), H_2 = (V_2, E_2), \cdots$ are the connected components of the graph G. From the Definition 3.1, we have A is an open set of $(V, \mathcal{Z}_G)$ if and only if $A = V_i$, for some i or $A = \emptyset$. So, any intersection of open sets is an open set by the characterisation of the connected components given in the Definition 4.3.

5. Isomorphic graphs and $\mathcal{Z}$-graphic topologies

Definition 5.1 *Let* $(X_1, \mathcal{T}_1)$ *and* $(X_2, \mathcal{T}_2)$ *be two topological spaces. A function*
$\psi : X_1 \to X_2$ *is called continuous if for all* $A \in \mathcal{T}_2$, $\psi^{-1}(A) \in \mathcal{T}_1$.
When the function ψ *is bijective and,* ψ *and* ψ^{-1} *are continuous, we say that the spaces are homeomorphic and we write* $X_1 \sim_h X_2$.

Definition 5.2 *Let* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *be two simple graphs. We say that* G_1 *and* G_2 *are isomorphic and we denote* $G_1 \cong G_2$ *if there exists a bijective map* $\phi : V_1 \to V_2$ *such that the function*
$\widetilde{\phi} : E_1 \longrightarrow E_2$
$(x, y) \mapsto (\phi(x), \phi(y))$ *is also bijective.*

Remark 5.1 *Let* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *be two isomorphic graphs and the isomorphism is* $\phi : V_1 \to V_2$. *It follows that if* $P = x_1 x_2 \cdots x_n$ *is a path joining* x_1 *and* x_n *in* G_1, *then* $P' = \phi(x_1)\phi(x_2)\cdots\phi(x_n)$ *is a path joining* $\phi(x_1)$ *and* $\phi(x_n)$ *in* G_2.
Conversely, if Q *is a path joining* v_1 *and* v_2 *in* G_2, *then we have a path* Q' *joining* $\phi^{-1}(v_1)$ *and* $\phi^{-1}(v_2)$ *in* G_1.

Theorem 5.1 *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two isomorphic graphs. Then the spaces $(V_1, \mathcal{Z}_{G_1})$ and $(V_2, \mathcal{Z}_{G_2})$ are homeomorphic.*

Proof. Let $\phi : V_1 \to V_2$ the bijective map inducing the isomorphism of the two graphs G_1 and G_2. We are going to prove that ϕ and ϕ^{-1} are continuous.
First, let $O \in \mathcal{Z}_{G_2}$ such that $\phi^{-1}(O) \neq \emptyset$. Suppose that $x \in \phi^{-1}(O)$ and $y \in V_1$ such that $x \sim_P y$, that is x and y are joined by a path in G_1. By the Remark 5.1, $\phi(x)$ and $\phi(y)$ are joined by a path in G_2. So, $\phi(y) \in O$ and hence $y \in \phi^{-1}(O)$. Then, $\phi^{-1}(O) \in \mathcal{Z}_{G_1}$.
Conversely, let $O \in \mathcal{Z}_{G_1}$. If $O = \emptyset$, then $\phi(O) = \emptyset \in \mathcal{Z}_{G_2}$.
If $O \neq \emptyset$, suppose that $x \in \phi(O)$ and $x \sim_Q y$ in G_2 (Q is a path in G_2). We have $x = \phi(x_1)$ for some $x_1 \in O$ and $y = \phi(y_1)$ for some $y_1 \in G_1$. From the Remark 5.1, x_1 and y_1 are joined by a path in G_1. Since, O is an open set of V_1, then $y_1 \in O$ and so $y = \phi(y_1) \in \phi(O)$. Therefore $\phi(O) \in \mathcal{Z}_{G_2}$.

In general, the converse of the Theorem 5.1 is not true.
Consider C_4 and K_4, their $\mathcal{Z}$-graphic topologies are homeomorphic but the two graphs are not isomorphic.
in the paper (Hamza *et al.*, 2013), the authors define a symmetry between two topologies. Next, we prove that if two graphs are isomorphic, then their $\mathcal{Z}$-graphic topologies are symmetric.

Definition 5.3 *(Hamza et al., 2013) Let $(X_1, \mathcal{T}_1)$ and $(X_2, \mathcal{T}_2)$ be two topological spaces. We say that these two spaces are symmetric and we write $X_1 \sim_s X_2$ (or $\mathcal{T}_1 \sim_s \mathcal{T}_2$) if $|\mathcal{T}_1| = |\mathcal{T}_2|$ and for all $A \in \mathcal{T}_1$ there exists an open set $B \in \mathcal{T}_2$ such that $|A| = |B|$ and conversely for all $B \in \mathcal{T}_2$ there exists an open set $A \in \mathcal{T}_1$ such that $|A| = |B|$.*

Theorem 5.2 *Let $G_i = (V_i, E_i)$, $i = 1, 2$, be two graphs. If $G_1 \cong G_2$ then $\mathcal{Z}_{G_1} \sim_s \mathcal{Z}_{G_2}$.*

Proof. From the proof of the Theorem 4.1, we get a bijective function, still denoted ϕ, $\phi : \mathcal{Z}_{G_1} \to \mathcal{Z}_{G_2}$, defined by $\phi(O) = \{\phi(x);\ \ x \in O\}$. So, $|\mathcal{Z}_{G_1}| = |\mathcal{Z}_{G_2}|$. Since $\phi : V_1 \to V_2$ is bijective, for all $A \in \mathcal{Z}_{G_1}$, the set $B = \phi(A) \in \mathcal{Z}_{G_2}$ and $|A| = |B|$.
Conversely, for all $B \in \mathcal{Z}_{G_2}$, the set $A = \phi^{-1}(B) \in \mathcal{Z}_{G_1}$ and $|A| = |B|$. The Theorem 5.2 follows.

The converse of the Theorem 5.2 is false, since the $\mathcal{Z}$-graphic topologies of C_4 and K_4 are symmetric but the two graphs are not isomorphic.

Definition 5.4 *Let $(V, \mathcal{T})$ be a topological space. $(V, \mathcal{T})$ is said $\mathcal{Z}$-graphic space if there exists a graph $G = (V, E)$ such that $\mathcal{T} = \mathcal{Z}_G$. We say also, $\mathcal{T}$ is a $\mathcal{Z}$-graphic topology.*

Being $\mathcal{Z}$-graphic is a topological property, that is, invariant under homeomorphisms.

Theorem 5.3 *Let $(V, \mathcal{T})$ and $(V', \mathcal{T}')$ be homeomorphic spaces. Suppose that $(V, \mathcal{T})$ is a $\mathcal{Z}$-graphic, then $(V', \mathcal{T}')$ is also a $\mathcal{Z}$-graphic space.*

Proof. Suppose that $\psi : V' \to V$ is a homeomorphism and $G = (V, E)$ is a graph such that $\mathcal{T} = \mathcal{Z}_G$. Consider

$$E' = \{(x', y') \in V' \times V' \mid (\psi(x'), \psi(y')) \in E\}. \tag{3}$$

We claim that $\mathcal{T}' = \mathcal{Z}_{G'}$, where $G' = (V', E')$. Indeed, let $A \in \mathcal{Z}_{G'}$. First, we want to prove that $\psi(A) \in \mathcal{Z}_G$. Let $x \in \psi(A)$ and $y \in V$ such that $x \sim_P y$ for some path P in G. We set $P = x_1, x_2, \cdots, x_n$ with $x_1 = x$ and $x_n = y$. So, since ψ is bijective, we have $x_i = \psi(x'_i)$ for $i = 1, \cdots, n$ and also $x'_1 \in A$.
Therefore, from the Equation 3, we have a path $P' = x'_1, x'_2, \cdots, x'_n$ in G' joining x'_1 and x'_n. But $x'_1 \in A$ and $A \in \mathcal{Z}_{G'}$. From the definition of the $\mathcal{Z}$-graphic topology, we get $x'_n \in A$ and so $y = x_n = \psi(x'_n)$ is in $\psi(A)$.
Then, $\psi(A) \in \mathcal{Z}_G$. That is, $\psi(A) \in \mathcal{T}$. Hence $A = \psi^{-1}\Big(\psi(A)\Big) \in \mathcal{T}'$.

Conversely, let $A \in \mathcal{T}'$. In order to prove that $A \in \mathcal{Z}_{G'}$, let $x' \in A$ and $y' \in V'$ such that $x' \sim_{P'} y'$ for some path P' in G'. Denote $P' = x'_1, x'_2, \cdots, x'_n$, where $x'_1 = x'$ and $x'_n = y'$.
$P = \psi(x'_1), \psi(x'_2), \cdots, \psi(x'_n)$ is a path in G joining $\psi(x')$ and $\psi(y')$.
Now, since $A \in \mathcal{T}'$ and ψ is a homeomorphism, $\psi(A) \in \mathcal{T}$. Hence, $\psi(A) \in \mathcal{Z}_G$ and so $\psi(y') \in \psi(A)$. Since, ψ is bijective, $y' \in A$. Therefore, $A \in \mathcal{Z}_{G'}$. So the Theorem 5.3 follows.

Now, we give a necessary and sufficient conditions for a topological space to be $\mathcal{Z}$-graphic (The corresponding problem 1 in (Jafarian Amiri *et al.*, 2013)).

Theorem 5.4 *Consider an Alexandroff topological space $(X, \mathcal{T})$ and denote $S(z)$ the smallest open set containing z, for $z \in X$. $(X, \mathcal{T})$ is $\mathcal{Z}$-graphic if, and only if, for all $z_1, z_2 \in X$, $S(z_1) = S(z_2)$ or $S(z_1) \cap S(z_2) = \emptyset$.*

Proof. First, suppose that $(X, \mathcal{T})$ is a $\mathcal{Z}$-graphic space. Let $G = (X, E)$ be a graph such that $\mathcal{T} = \mathcal{Z}_G$. In this case $S(z)$ is the vertex set of the connected component of G containing x. So, for all $z_1, z_2 \in X$, $S(z_1) = S(z_2)$ or $S(z_1) \cap S(z_2) = \emptyset$, from the Definition 4.3.
Next, suppose $(X, \mathcal{T})$ is a topological space such that $S(z_1) = S(z_2)$ or $S(z_1) \cap S(z_2) = \emptyset$, for all $z_1, z_2 \in X$. Denote

$$E = \{(x, y) \in X \times X \mid S(x) = S(y)\}. \tag{4}$$

Consider the graph $G = (X, E)$, we are going to prove that $\mathcal{T} = \mathcal{Z}_G$. let $A \in \mathcal{T}$. Suppose that $x \in A$ and $y \in X$ such that $x \sim_P y$, where P is a path in G. Since $x \in A$ and A an open set, we have $S(x) \subset A$. Since $x \sim_P y$ and from the definition of the edge set (4), we get $S(x) = S(y)$ and hence $y \in S(y) \subset A$. Therefore $A \in \mathcal{Z}_G$.

Conclusion

Let $G = (V, E)$ an undirected graph. The graphic topology τ_G is a topology defined on V. When the graph G is connected, the topological space (V, τ_G) is not necessarily connected. In this paper, we introduce the $\mathcal{Z}$-graphic topology $\mathcal{Z}_G$ on V which satisfies $G = (V, E)$ is a connected graph if and only if $(V, \mathcal{Z}_G)$ is a connected topological space.
Also, we have proved that two isomorphic graphs have homeomorphic and symmetric $\mathcal{Z}$-graphic topologies. As future work, we can think about graphic topology and $\mathcal{Z}$-graphic topology for directed graphs.

References

Abdu, K.A., & **Kilicman, A. (2018)**. Bitopological spaces on undirected graphs. J. Math. Computer Sci., 18, 232-241.

Agnarsson, G., & **Greeniaw, R. (2007)**. Graph Theory: Modeling Application, and Algorithms. Person Education Inc., (2007).

Alexandroff, P. (1937). Diskrete Räume. Mat. Sb. (N.S.) 2 , 501-518.

Ali, A., Z. Raza, Z., & **Bhatti, A.A. (2016)**. On the augmented Zagreb index. Kuwait J. Sci. 43 (2), 123-138.

Arenas, F.G. (1999). Alexandroff spaces. Acta Math. Univ. Comenian. 68 (1), 17-25.

Cruz, R., Gutman, I., & **Rada, J. (2021)**. Sombor index of chemical graphs. Applied Mathematics and Computation 399, 126018; doi.org/10.1016/j.amc.2021.126018

Diestel, R. (2005). Graph Theory. 3rd edition, Graduate Texts in Mathematics, 173, Springer-Verlag, Berlin.

Dugundji, J. (1966). Topology. Allyn and Bacon, Inc., Boston.

Gutman, I., & **Kulli, V.R. (2021)**. Nirmala energy. Open J. Discret. Appl. Math., 4(2), 11-16; doi:10.30538/psrp-odam2021.0055

Hamza, A.M., & **Al-khafaji, S.N. (2013)**. Construction A Topology On Graphs. Journal of Al-Qadisiyah for computer science and mathematics Vol.5 No.2, 39-46.

Herman, G.T. (1990). On topology as applied to image analysis. Comput. Vision, Graphics Image Process 52, 409-415.

Jafarian Amiri, S.M., Jafarzadeh, A., & **Khatibzadeh, H. (2013)**. An Alexandroff topology on graphs. Bulletin of the Iranian Mathematical Society Vol. 39 (4), 647-662.

Kandel, A., Bunke, H., & **Last, M. (2007)**. Applied graph theory in computer vision and pattern recognition. Studied in Computational Intelligence, 52, Springer- Verlag, Heidelberg-Berlin.

Kilicman, A., & **Abdulkalek, K. (2018)**. Topological spaces associated with simple graphs. Journal of Mathematical Analysis, Vol. 9 No. 4, 44-52.

Kronheimer, E.H. (1992). The topology of digital images. Top. and its Appl. 46, 279-303.

Li, Y., Li, J., Feng, J., & **Wang, H. (2019)**. Minimal bases and minimal sub-bases for topological spaces. Filomat 33 (7), 1957-1965; doi.org/10.2298/FIL1907957L

McCord, M.C. (1966). Singular homology and homotopy groups of finite topological spaces. Duke Math. Jour, 33, 465-474.

Naji, A.M., & **Soner, N.D. (2018)**. The First Leap Zagreb Index of Some Graph Operations. International Journal of Applied Graph Theory, Vol.2, No.1, 07 - 18.

Nogly, D., & **Schladt, M. (1996)**. Digital topology on graphs. Comput. Vis. Image. Und. 63, 394-396.

Sasikala, D., & **Divya, A. (2019)**. An Alexandroff Bitopological Space on Undirected Graphs. IJSRR 8(2), pp. 3720-3728; doi.org/10.12988/ams.2015.5154

Shokry, M. (2015). Generating Topology on Graphs by Operations on Graphs. Applied Mathematical Sciences, Vol. 9 no. 57, 2843 - 2857; doi.org/10.12988/ams.2015.5154

Speer, T. (2007). A short study of Alexandroff spaces. At http://arxiv.org/abs/0708.2136.

Stong, R.E. (2015). Finite topological spaces. Trans. A.M.S. 123, 325-340.

Zomam, H.O., Othman, H.A., & **Dammak, M. (2021)**. Alexandroff spaces and graphic topology. Advances in Mathematics: Scientific Journal 10 (2021), no.5, pp. 2653-2662; doi.org/10.37418/amsj.10.5.28

Submitted: 03/12/2021
Revised: 20/03/2022
Accepted: 10/04/2022
DOI: 10.48129/kjs.17541

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-5)

An improved robust variance inflation factor: Reducing the negative effects of good leverage points

Osman U. Ekiz
Dept. of Statistics, Gazi University, Turkey
Corresponding author: ufukekiz@gazi.edu.tr

Abstract

In multiple linear regression analysis, the variance inflation factor is a well-known collinearity measure. It is defined as the function of the coefficient of determination between the explanatory variables, and it is based on the maximum likelihood estimator of the regression coefficients. Nevertheless, in addition to outliers, leverage observations can have significant impact on the coefficient of determination, and thereby the variance inflation factor. This study presents an improved robust variance inflation factor estimator that is not affected by these observations. Simulation studies and a real data analysis indicate that the modified robust variance inflation factor estimator performs better than the traditional one.

Keywords: Collinearity-inducing leverage; collinearity-masking leverage; linear regression; outlier; robust statistics

1. Introduction

The multiple linear regression model is used to make inferences about a response variable using explanatory variables, and it is defined as $Y = X\beta + \epsilon$. The maximum likelihood (*ML*) estimator of β, which is known as the best linear unbiased estimator, is expressed as

$$\hat{\beta}_{ML} = \left(X'X\right)^{-1} X'Y,$$

(Graybill, 1961). In the presence of collinearity problem, the well-know ridge regression estimators are proposed (Hoerl & Kennard, 1970). There are many studies in the literature that focus on ridge regression (Dorugade, 2014). Moreover, studies have suggested the use of robust and ridge-type robust estimators if there are outliers, or both collinearity and outliers, in the regression data (Aftab & Chand, 2018; Alshqaq, 2021; Maronna, 2011; Silvapulle, 1991). The presence of both outliers and one or more leverage observations in the data may have an impact on the severity of collinearity. Here, these collinearity-influencing leverage observations are categorized into two groups according to how they affect collinearity. The first group consists of *collinearity-masking leverage* observations. These observations may lead to the misconception that there is no collinearity in the data. For the second group of observations, called *collinearity-inducing leverage* observations, the outcome is just the opposite. They may lead to a misinterpretation of collinearity in the data.

The variance inflation factor ($VIF_{ML} = 1/\left(1 - R^2_{ML}\right)$) is a measure used to make inferences about collinearity. If its value is larger than 10, there is severe collinearity in the data (Gujrati, 2004). R^2_{ML} is the largest coefficient of determination between X_j, $j = 1, ...k$, and the rest of the explanatory variables. If extreme observations are present in the data, these points would impact $\hat{\beta}_{ML}$ and $\bar{y}$, which means the resulting residuals $(y_i - \hat{y})$ might be larger than they are in reality. This leads to the employment of robust determination coefficient to diagnose collinearity by using

$$R^2_r = 1 - \frac{\sum_{i=1}^{n} w_i \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} w_i \left(y_i - \bar{y}_w\right)^2},$$

where r denotes a robust estimator and $\bar{y}_w = \sum_{i=1}^{n} w_i y_i / \sum_{i=1}^{n} w_i$. The weights, w_i, and predictions, $\hat{y}_i$, are produced by applying a robust regression estimator (Renaud & Victoria-Feser, 2010). However, this estimator performs well in parameter estimations only in the presence of outliers in the *X* or *Y* direction. The calculated value of the robust VIF $\left(VIF_r = 1/(1 - R_r^2)\right)$ based on R_r^2 with *collinearity-inducing leverage* observations, also called good leverage points, leads to the perception that collinearity exists. Note that, here, R_r^2 denotes the largest robust coefficient of determination established by a robust regression estimator between X_j and the remaining explanatory variables. Since *collinearity-inducing leverage* observations have an impact on this estimator, it is important to build an R_r^2 that is strong despite the presence of these points.

This study aims to improve the R_r^2 and VIF_r, which are referred to as the new R_r^2 $\left(newR_r^2\right)$ and new VIF_r $(newVIF_r)$ based on the $newR_r^2$. The severity of collinearity is determined more accurately with the $newVIF_r$, which is also not impacted by *collinearity-inducing leverage* observations. This makes it easier to determine the best estimator for the regression analysis. In Section 2, robust estimators are mentioned to construct new underlined estimators. The suggested approach is introduced in Section 3. The results, using a real data set, are presented in Section 4. Furthermore, this section provides simulation details that allow for comparisons of the estimators utilized. These findings demonstrate that the $newVIF_r$ based on the $newR_r^2$ provides better results compared to the VIF_r. The paper ends with conclusion in Section 5.

2. Robust *LMS*, *LTS*, and *S* estimators

There are various robust estimators for estimating the parameters in multiple linear regression models. In this study, the most common robust estimators the least median of squares $(\hat{\beta}_{LMS})$, least trimmed square $(\hat{\beta}_{LTS})$ (Rousseeuw & Leroy, 1987), and S $(\hat{\beta}_S)$ (Rousseeuw & Yohai, 1984) are employed to determine the performance of the improved estimator $newVIF_r$.

These estimators are calculated from

$$\hat{\beta}_\ell = \left(X' W_{\ell-1} X\right)^{-1} X' W_{\ell-1} Y,$$

where $W_{\ell-1}$ defines the diagonal weight matrix with elements $w\left(r_i\right)$ and the r_i denotes the residuals, $i = 1, ..., n$ (Rousseeuw & Leroy, 1987). Note that for $\hat{\beta}_{LMS}$ and $\hat{\beta}_{LTS}$, $w_i = 1$ when observation $i \in t$th sub-sample. Otherwise, $w_i = 0$. The weights for the S estimator should be established in each iteration by employing Tukey's bi-weight function (Maronna *et al.*, 2006; Rousseeuw & Yohai, 1984).

3. An improved robust *VIF*

The R_r^2 is not affected by the presence of *collinearity-masking leverage* observations. However, it does not yield good results when there are leverage observations that induce collinearity because it is robust only against outliers. In addition, leverage observations that are considered to be good and regular in the direction of $X_{(-j)}$ (the design matrix X excluding the *j*th explanatory variable) can induce collinearity. Thus, a VIF_r that is dependent on R_r^2 would be adversely affected by these observations as well. In order to overcome this negative effect, it is recommended that the *collinearity-inducing leverage* observations be removed from the $X_{(-j)}$ direction before the R_r^2 is calculated. For this purpose, the VIF_r is improved and called the new VIF_r $(newVIF_r)$ (Ekiz, 2021). The detailed description of the algorithm is as follows:

- For each $X_{(-j)}$ compute the robust estimators $\hat{\tau}\left(X_{(-j)}\right)$ and $\hat{\Sigma}^{-1}_{X_{(-j)}}$ of the location and scale parameters, respectively. In this study minimum covariance determinant (*MCD*) estimators are employed (Rousseeuw & Driessen, 1999).

- Compute Mahalanobis distances, MD_i^2 based on $\hat{\tau}\left(X_{(-j)}\right)$ and $\hat{\Sigma}^{-1}_{X_{(-j)}}$ (Maronna *et al.*, 2006).

- If $MD_i^2 > \chi^2_{k-1,1-\alpha}$, x_i is determined to be an *collinearity-inducing leverage* (outlier) observation. Additionally, this point is referred to as good leverage when regressing X_j on $X_{(-j)}$. $\chi^2_{k-1,1-\alpha}$ is the upper-α quantile of the chi-square distribution. At the end of this step, a total of m observations are identified as *collinearity-inducing leverage*.

- Considering that there are *collinearity-inducing leverage* points during the application of the regression of X_j on $X_{(-j)}$, subtract m observations from the data. Both R_r^2 and VIF_r are then computed by constructing the regression analysis with a clean $n-m$ observation.
- Report the estimates from $n-m$ observations as $newR_r^2$ and $newVIF_r$.

When the computed $newVIF_r$ is larger than 10, there is severe collinearity in the data.

4. Application

In this section, the improved measure, $newVIF_r$, is compared with the VIF_r by applying *Body fat* data, (Kutner *et al.*, 2004), which consists of *collinearity-masking leverage* observations. There are three explanatory variables, each of which has 20 observations: Tricep skin thickness (X_1), thigh circumference (X_2), and midarm circumference (X_3).

Let $newVIF_r$ $(r = LMS, LTS, S)$ be the new robust measure, and let VIF_{ML} denote the VIF computed using the *ML* estimator. The values of VIF_r based on LMS, LTS, and S estimators are calculated as 250.2497, 688.5522, and 792.8248, respectively. The values of $newVIF_r$ based on the same estimators are calculated as 825.7449, 790.7602, and 793.9697, respectively. Here, $\alpha = 0.05$. All of these values are much higher than VIF_{ML} which is 36.4631. This is the evidence of the presence of more severe collinearity. Hence, in the case of *collinearity-masking leverage* in the data, the use of VIF_r and $newVIF_r$ estimates will be useful to diagnose the severity of collinearity for the appropriate regression model.

The $newVIF_r$ would not be affected from the *collinearity-inducing leverage* observations existing in the data, in contrast to VIF_r. To illustrate this point of view a detailed simulation study is carried out in Section 4.1. The results both in the application and the simulation study are obtained by using Matlab.

4.1 Simulation study

In this simulation, the datasets are generated so that they are contaminated with leverage observations that effect collinearity. An evaluation of the performance of the VIF_r and $newVIF_r$ estimators with contaminated data is conducted by comparing their Monte Carlo (*MC*) means with the *uVIF* computed from the uncontaminated portion of the data. When the *MC* mean of the estimator is close to the *uVIF*, it can be said that the estimator is not affected by contaminated data (Ekiz, 2021). Note that $uVIF = 1/\left(1 - C_{X_j,X_{(-j)}} C_{X_{(-j)},X_{(-j)}} C'_{X_j,X_{(-j)}}\right)$, where C denotes the correlation matrix of the distribution of the uncontaminated part of the data (Mardia *et al.*, 1979). The datasets are simulated from the contaminated normal distribution, where the number of explanatory variables is set to 3 ($k = 3$). The joint probability distribution of (X_1, X_2, X_3) is defined as $F = (1-\lambda)\, G + \lambda H$, where $G \sim N_k(\mu, \Sigma)$, $H \sim N_k(\theta, \Sigma)$, and $\Sigma = C$. The mixture parameter, $\lambda \in [0, 1]$, provides $\lambda \ll 1$ (Maronna *et al.*, 2006). Additionally, $\mu_X = (\mu_{X_1}, \mu_{X_2}, \mu_{X_3})$ and $\theta_X = (\theta_{X_1}, \theta_{X_2}, \theta_{X_3})$ are used as the location parameters of G and H, respectively. To simulate an n sized dataset consisting of only high-leverage points (masking or inducing) with a proportion of λ, the leverage observations are generated from $N_k(\theta, \Sigma)$ and the non-leverage observations are generated from $N_k(\mu, \Sigma)$. In this way, the set of design parameters $\mu_{X_1}, \mu_{X_2}, \mu_{X_3}, \theta_{X_1}, \theta_{X_2}, \theta_{X_3}$ can be utilized to manipulate the level and type of contamination.

Using a covariance matrix Σ, with ones on the diagonal, the dataset includes *collinearity-masking leverage*. The remaining elements of this matrix are selected as values close to one, providing strong collinearity between the explanatory variables. In the simulations, a λ proportion of high-leverage observations, taken from $H \sim N_k(\theta, \Sigma)$, where $\theta = (5, 7, 7)$, are integrated into the dataset as well. The VIF_{ML_G} and VIF_{ML_F} should be calculated from the observations that are produced from the distributions *G* and *H*, respectively. It can be seen that VIF_{ML_F} is much smaller than VIF_{ML_G}, even for small values of λ. Therefore, a small number of high-leverage observations may mask a strong collinearity that depends on the rest of the data. To create a dataset with *collinearity-inducing leverage*, the elements of Σ are chosen to be very close to zero. Thus, the value of the corresponding *uVIF* is small, indicating that there is no correlation between the explanatory variables. When the λ ratio of the *collinearity-inducing*

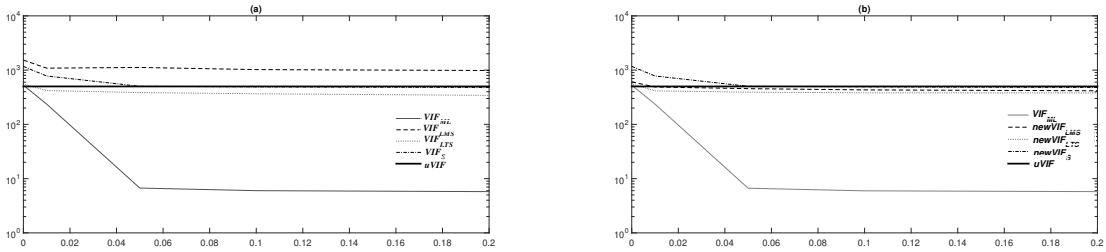


Fig. 1. Contaminated data with *collinearity-masking leverage*. The value of $uVIF$ is set at 501.3193

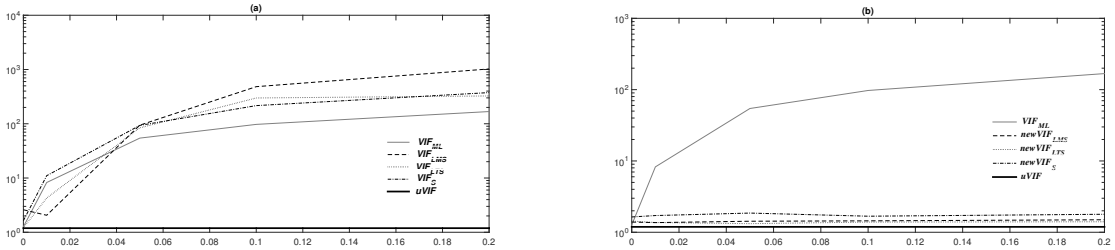


Fig. 2. Contaminated data with *collinearity-inducing leverage*. The value of $uVIF$ is set at 1.2121.

leverage observations generated from the H distribution, with the $\theta = (35, 32, 37)$, is integrated into the data, the calculated VIF_{ML_F} is much higher than the calculated VIF_{ML_G} without the *collinearity-inducing leverage* observations. This result indicates that a small number of *collinearity-inducing leverage* observations may increase the severity of collinearity.

The simulation procedure is based on 10000 iterations for all combinations of $n = 100$ and $\lambda = 0, 0.01, 0.05, 0.10, 0.20$. The ***MC*** estimations for the VIF_r and $newVIF_r$ values obtained in cases where the data is contaminated by *collinearity-masking* and *-inducing leverage* observations are given in the vertical axes of the graphs in Figure 1 and 2. In these graphs, the horizontal axes show the λ. ***MC*** estimates near $uVIF = E\left(VIF_{ML_G}\right)$ are considered to be good performance estimates. Note that ***E*** shows the expected value, and VIF_{ML_G} is the measure of the VIF obtained from the data produced by the ***G*** distribution, based on the ***ML*** estimator.

In the case of *collinearity-masking leverage*, the outcomes of both VIF_S and $newVIF_S$ seem to be good (see Figure 1(a) and (b), respectively). Moreover, as shown in Figure 1 and 2, in contrast to the other estimators, the $newVIF_S$ estimator outperforms in both cases, and its calculated values approach $uVIF$.

In the presence of *collinearity-inducing leverage* observations, it can be seen that the VIF_r yields very large results than the $uVIF$. This leads to the misconception of as if there is collinearity, as shown in Figure 2(a). However, according the plots in Figure 2(b) the $newVIF_r$ provides very reasonable results. When $n = 50$, $\lambda = 0.10$, and using the data simulated with *collinearity-inducing leverage* observations, the MC means of VIF_S and $newVIF_S$ are calculated as 350 and 1.80 . Thus, the bias of $newVIF_S$ from $uVIF = 1.2121$ is negligible compared to the value of VIF_S.

5. Conclusion

Before starting a regression analysis, it is important to investigate whether there are outliers and/or collinearity problems in the data. It is recommended that ridge, robust, and ridge-type robust estimators be used for problems with collinearity, outliers, and both collinearity and outliers, respectively (Silvapulle, 1991). Hence, accurately determining the severity of collinearity plays an important role in identifying the correct estimator to apply. When the leverage observations (outliers) in the direction of jth explanatory variable mask collinearity (*collinearity-masking leverage*), the results of VIF_r demonstrate that there is more severe collinearity in the data, compared to results based on VIF_{ML}. At the same time, similar results are observed from the proposed $newVIF_r$.

However, if the data contains *collinearity-inducing leverage* observations, the VIF_r is unable to recognize that there is actually no collinearity in the data. The VIF_r provides large numerical results, as if collinearity exists. In contrast, the values of the $newVIF_r$ estimator, improved in this study, are small in this situation. Furthermore, when *collinearity-masking* or *-inducing leverage* observations are present in the data, the $newVIF_S$ out-performs the other estimators. For this reason, this measure could be used to diagnose collinearity before deciding which estimator to use for parameter estimates.

References

Aftab, N., & **Chand, S. (2018)**. A simulation-based evidence on the improved performance of a new modified leverage adjusted heteroskedastic consistent covariance matrix estimator in the linear regression model. Kuwait Journal of Science, 45(3).

Alshqaq, S. S. (2021). On the least trimmed squares estimators for *JS* circular regression model. Kuwait Journal of Science, 48(3), 1-13.

Dorugade, A. V. (2014). On comparison of some ridge parameters in ridge regression. Sri Lankan Journal of Applied Statistics, 15(1), 31–45.

Ekiz, O. U. (2021). İyi kaldıraç noktalarından etkilenmeyen sa ̆glam varyans artış faktörü [A variance inflation factor which is robust against leverage points]. II. International Applied Statistics Conference. Proceedings book of the UYIK-2021, 117. Tokat, Turkey.

Graybill, F. A. (1961). Introduction to Linear Statistical Models. New York, USA: McGraw-Hill.

Gujrati, D. N. (2004). Basic Econometrics. New Delhi, IND: Tata McGraw-Hill.

Hoerl, A. E., & **Kennard, R. W. (1970)**. Ridge regression: Biased Estimation for nonorthogonal problems. Technometrics, 12(1), 55–67.

Kutner, M., Nachtsheim, C., & **Neter, J. (2004)**. Applied Linear Regression Models. New York, USA: McGraw-Hill.

Mardia, K. V., Kent, J. T., & **Bibby, J. (1979)**. Multivariate Analysis, New York, USA: Academic Press.

Maronna, R. A., Martin, D. R., & **Yohai, V. J. (2006)**. Robust Statistics: Theory and Methods, New York, USA: Wiley.

Maronna, R. A. (2011). Robust ridge regression for high-dimensional data. Technometrics, 53(1), 44–53.

Renaud, O., & **Victoria-Feser, M. P. (2010)**. A robust coefficient of determination for regression. Journal of Statistical Planning and Inference, 140(7), 1852–1862.

Rousseeuw, P., & **Yohai, V. (1984)**. Robust Regression by Means of S-Estimator. In J. Franke, W. Härdle, & D. Martin (Eds.), Robust and Nonlinear Time Series Analysis (pp. 256–272), New York, USA: Springer.

Rousseeuw, P. J., & **Leroy, A. M. (1987)**. Robust Regression and Outlier Detection. New York, USA: John Wiley & Sons.

Rousseeuw, P. J., & **Driessen, K. V. (1999)**. A fast algorithm for the minimum covariance determinant estimator. Technometrics, 41(3), 212–223.

Silvapulle, M. J. (1991). Robust ridge regression based on an M estimator. Australian Journal of Statistics, 33(3), 319–333.

Submitted: 02/08/2021
Revised: 15/11/2021
Accepted: 16/11/2021
DOI: 10.48129/kjs.15533

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-17)

Comparison of fast regression algorithms in large datasets

Sengul Cangur[1]*, Handan Ankarali[2]
[1] *Dept. of Biostatistics and Medical Informatics, Duzce University, Turkey*
[2] *Dept. of Biostatistics and Medical Informatics, Istanbul Medeniyet University, Turkey*
**Corresponding author: sengulcangur@duzce.edu.tr*

Abstract

The aim is to compare the performances of fast regression methods, namely dimensional reduction of correlation matrix (DRCM), nonparametric dimensional reduction of correlation matrix (N-DRCM), variance inflation factor (VIF) regression, and robust VIF (R-VIF) regression in the presence of multicollinearity and outliers problems. In all simulation-scenarios, all the target variables were chosen for final models using four methods. The DRCM and N-DRCM are the methods that reach the final model in the shortest time, respectively. The time to reach the final model using R-VIF regression was approximately twice shorter than that of VIF regression. In each method, as the number of variables and the level of outliers increased, the time taken to reach the final model increased. When the level of multicollinearity and the number of variables ($p > 500$) increased, the times to reach the final models using DRCM in datasets with outliers were slightly shorter than the those of N-DRCM. The largest numbers of noise variables were selected to the model using DRCM and N-DRCM, but the least number of them were selected to the model using the R-VIF regression. The RMSE values obtained using DRCM, N-DRCM and VIF regression were similar in each scenario. As a result of the real dataset, the final model selected using R-VIF regression had the highest R^2. It also had the lowest RMSE value among those obtained with other approaches excluding VIF regression. As such, the R-VIF regression method demonstrated a better performance than the others in all datasets.

Keywords: Dimensional reduction; large data; robust; variance inflation factor

1. Introduction

In many fields, large data are studied, where the number of variables and observations is quite high. Through the development of modern technology, recording and storing information has become significantly easier. However, many researchers still experience issues in relation to accessing suitable information using datasets. Common issues include associated time-limit, theoretical, and costs among others. Researchers currently seek new approaches or algorithms that will allow them to access information quickly with minimal errors and few features. As such, algorithms that are easy to implement, can select the most suitable features for predictive statistical complex models, find solutions to frequently run into problems in modeling researches and application, and reach the final model quickly are being investigated. The most efficient approaches are becoming increasingly popular.

A review of current literature suggests the following algorithms are the ones most frequently used in relation to huge datasets especially high-dimensional datasets: least absolute shrinkage selection operator (LASSO) (Tibshirani, 1996), adaptive LASSO (Zou, 2006), elastic net (Zou & Hastie, 2005), least angle regression (LARS) (Efron *et al.*, 2004), robust LARS (Khan *et al.*, 2007), Dantzig (Candes & Tao, 2007), iterative sure independent screening (ISIS) (Fan & Lv, 2008), generalized path-seeking algorithm (GPS) (Friedman , 2008), forward-backward greedy algorithm (FoBa) (Zhang, 2009), variance inflation factor (VIF) regression (Lin *et al.*, 2011), robust variance inflation factor (R-VIF) regression (Dupuis

& Victoria-Feser, 2013), dimensional reduction of correlation matrix (DRCM) (Midi & Uraibi, 2014), jack-knife robust LARS (JKR-LARS) (Shahriari *et al.*, 2014), and VIF regression screening algorithm (VIFRegS) (Uraibi, 2020). Fast algorithms that meet the needs of researchers working with large datasets are currently being developed. Researches include the recently developed VIF regression method that has been used in health research (Liu *et al.*, 2017; Cai *et al.*, 2018), the DRCM method that claims to be faster and simpler than the VIF regression estimator, and the R-VIF regression method that can overcome issues including multicollinearity, overfitting, and outliers.

As previously noted, the assumptions of fast regression algorithms are not always met in dataset, or although it is claimed that some algorithms can overcome prominent issues in the cases that the severities of the problems and the number of variables increase, few studies have investigated how fast algorithms perform. As such, this simulation examines whether fast regression algorithms such as VIF regression, DRCM, R-VIF regression and nonparametric DRCM (N-DRCM) perform as well as current research suggests, especially in relation to the dataset containing multicollinearity and outliers. In addition, N-DRCM, which is the nonparametric version of DRCM, is discussed in this study. Whether this method can compete with others as a fast estimator is examined through implementing a multiple scenario simulation.

2. Methods

2.1 Variance inflation factor (VIF) regression method

The VIF regression is an approach developed from the streamwise variable selection algorithm with the α- investing rule. The streamwise algorithm ensures that the method implemented is fast, while the α-investment control is to prevent model overfitting. This method was improved using the sparsity assumption ($k \ll p$) when k is the subset of p predictors, and can control marginal false discovery rate-mFDR (Zhou *et al.*, 2006; Foster & Stine, 2008). Lin *et al.* (2011) improved this method as stepwise regression remained unresolved in relation to the multicollinearity problem. The regression model $\mathbf{y} = \beta_0 + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \beta_{new}\mathbf{x}_{new} + \boldsymbol{\varepsilon}$ ($\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$ was tested to obtain the predictive regression model through forward selection. In this model, $\mathbf{y}$ is the dependent variable, $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are independent variables, $\beta_0, \ldots, \beta_k$ are regression coefficients, and $\boldsymbol{\varepsilon}$ is error. Here, $\mathbf{X} = [\mathbf{1}_n\ \mathbf{x}_1\ \ldots \mathbf{x}_k]$, $\widetilde{\mathbf{X}} = [\mathbf{X}\ \mathbf{x}_{new}]$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)^T$, and $\widetilde{\boldsymbol{\beta}} = (\beta_0, \ldots, \beta_k, \beta_{new})^T$. The algorithm of this method is shown in Algorithm 1.

Algorithm 1.

Input: data $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2, \ldots$ (centered);
Set: $\alpha_0 = 0.50$, and pay-out $\Delta\alpha = 0.05$, and subsample size m;
Initialize $S = \{0\}$; $\mathbf{r} = \mathbf{y} - \widehat{\mathbf{y}} = \mathbf{r} = \mathbf{y} - \mathbf{X}_S\left(\mathbf{X}_S^T\mathbf{X}_S\right)^{-1}\mathbf{X}_S^T\mathbf{y}$;
$\widehat{\sigma} = \mathrm{sd}(\mathbf{y}) = \|\mathbf{r}\| / \sqrt{(n - |S| - 1)}$; $j = 1$; $\alpha_1 = \alpha_0$; $f = 0$.
Sample $I = \{j_1, \ldots, j_m\} \in \{1, \ldots, n\}$. // the subsample ${}_I\mathbf{x}$ randomly selected from predictors $\mathbf{x}$
Compute $\widetilde{\gamma}_{new} = \langle \mathbf{r}, \mathbf{x}_{new} \rangle / \|\mathbf{x}_{new}\|$ and
${}_IR^2 = \mathbf{x}_{newI}^{\mathrm{T}}\mathbf{X}_S\left({}_I\mathbf{X}_S^{\mathrm{T}}{}_I\mathbf{X}_S\right)^{-1}{}_I\mathbf{X}_S^{\mathrm{T}}\mathbf{x}_{new} / \|\mathbf{x}_{new}\|^2$.
repeat
set threshold $\alpha_j = \alpha_j$ / (1 + j - f)
get $\hat{t}_j = \widetilde{\gamma}_{new} / \widehat{\sigma}\sqrt{(1 - {}_IR^2)}$ // compute corrected t-statistic
if $2\Phi\left(|t_j|\right) > 1 - \alpha_j$ // compare p-value to threshold **then**
$S = S \cup \{j\}$ // add feature to model
update $\mathbf{r} = \mathbf{y} - \widehat{\mathbf{y}}_S$, $\widehat{\sigma} = \mathrm{RMSE}_S$
$\alpha_{j+1} = \alpha_j + \Delta\alpha$
$f = j$
else

$\alpha_{j+1} = \alpha_j - \alpha_j / \left(1 - \alpha_j\right)$
end if
$j = j + 1$
until maximum CPU time or memory is reached.

α_0: the initial alpha-wealth according to α-investing rule, $\Delta\alpha$: if a hypothesis is rejected, the change of alpha-wealth value, **r**: residuals, S: the set of predictors, α_j: α value in the jth test, sd: standard deviation, f: the time at which the last hypothesis is rejected, I: subsample, Φ: the standard normal cumulative distribution, RMSE: root mean squared error, CPU: central processing unit

This method contains two components: evaluation and search. The evaluation step contains forward stagewise regression and evaluates variables using marginal correlations. The stagewise regression algorithm contains small step sizes and behaves similarly to l_1 algorithms such as Lasso and LARS. As such, it suffers from collinearities between the predictors. Lin *et al.* (2011) corrected this bias by selecting a small sample from the dataset to calculate the VIF of each variable. The resultant evaluation phase is fast and contains no significant loss of accuracy. In the search step, each variable is sequentially tested using the α-investing rule. This rule ensures that models do not overfit and can generate highly accurate results. VIF procedure can be combined with various algorithms such as stepwise regression, LARS, and FoBa. This algorithm is particularly useful when feature systems are created dynamically and the size of the candidate features collection is unknown or even infinite. It can also serve as an "online" algorithm for loading extremely large-scale data into RAM according to its properties (Lin *et al.*, 2011).

2.2 Robust VIF regression method

Robust VIF regression method is developed by Dupuis & Victoria-Feser (2013) as the classical VIF regression method can be adversely affected by outliers in the dataset. It contains all properties of the classic approach. Dupuis & Victoria-Feser (2013) used the robust weighted slope estimator and the fast robust t-statistic in this method. Therefore, this method is very robust against small model deviations. The R-VIF regression procedure, which is based on a streamwise variable selection algorithm and the α-investing rule, is shown in Algorithm 2.

Algorithm 2.

Input: data $\mathbf{y}$, $\mathbf{x}_1$, $\mathbf{x}_2$, . . . (standardized);
Set: initial wealth $\alpha_0 = 0.50$, and pay-out $\Delta\alpha$ =0.05, and subsample size m, and robustness constant c
Compute efficiency e_c^{-1} where e_c is as in

$$e_c = \left[\int_{-c}^{c}\left(5\left(\frac{r}{c}\right)^4 - 6\left(\frac{r}{c}\right)^2 + 1\right)d\Phi(r)\right]^2 / \int_{-c}^{c} r^2\left(\left(\frac{r}{c}\right)^2 - 1\right)^4 d\Phi(r)$$

Get all marginal weights w_{ij} by fitting p marginal models $y = \beta_{01} + \beta_{1x_1} + \varepsilon_1, \ldots, y = \beta_{0k} + \beta_k x_k + \varepsilon_k$ using $\sum_{i=1}^{n} w_i(r_i;c) r_i x_i = 0$ and $w_i(r_i;c) = min\left\{1;\ \frac{c}{|r_i|}\right\}$ (c=1.345)
Initialize $j = 1, S = \{0\}$, $\mathbf{X}_S = 1$, $\mathbf{X}_S^w = \text{diag}\left(\sqrt{w_{iS}^0}\right)\mathbf{X}_S$ and $\mathbf{y}^w = \text{diag}\left(\sqrt{w_{iS}^0}\right)\mathbf{y}$ where w_{iS}^0 is computed using

$$w_i(r_i;c) = \begin{cases} \left(\left(\frac{r_i}{c}\right)^2 - 1\right)^2 & if\ |r_i| \le c, \\ 0 & if\ |r_i| > c, \end{cases}$$

where $\mathbf{r}^0 = \left(\mathbf{y} - 1\widehat{\beta}^0\right)/\widehat{\sigma}^0$ using $\mathbf{X}_0^w = \mathbf{X}_0^{w2} = 1$, $\widehat{\beta}^0 = \left[\left(\mathbf{X}_0^w\right)^{\mathrm{T}}\mathbf{X}_0^w\right]^{-1}\left(\mathbf{X}_0^{w2}\right)^{\mathrm{T}}\mathbf{y}$,

$\widehat{\sigma}^0 = 1.483 med\left|\widetilde{\mathbf{r}}^0 - med(\widetilde{\mathbf{r}}^0)\right|$ and $\widetilde{\mathbf{r}}^0 = \mathbf{y} - 1\widehat{\beta}^0$.

repeat

set $\alpha_j = \alpha_j/(1+j-f)$

compute $\mathbf{r}_S^w = \mathbf{y}^w - \mathbf{X}_S^w\left(\mathbf{X}_S^{wT}\mathbf{X}_S^w\right)^{-1}\mathbf{X}_S^{wT}\mathbf{y}^w$ //start *Fast Robust Evaluation Procedure*

$\widehat{\gamma}_j^w = \left(\mathbf{z}_j^{wT}\mathbf{z}_j^w\right)^{-1}\mathbf{z}_j^{wT}\mathbf{r}_S^w$ and $\widehat{\sigma} = MAD\left(\mathbf{r}_S^w - \mathbf{z}_j^w\left(\mathbf{z}_j^{wT}\mathbf{z}_j^w\right)^{-1}\mathbf{z}_j^{wT}\mathbf{r}_S^w\right)$

where $\mathbf{z}_j^w = \mathrm{diag}\left(\sqrt{w_{ij}}\right)\mathbf{z}_j$

sample $I = \{i_1,\ldots,i_m\} \in \{1,\ldots,n\}$ // the subsample ${}_I\mathbf{x}$ randomly selected from predictors

get $R_{jS}^{w2} = {}_I\mathbf{z}_j^{wT}\,{}_I\mathbf{H}_S^w\,{}_I\mathbf{z}_j^w\left({}_I\mathbf{z}_j^{wT}\,{}_I\mathbf{z}_j^w\right)^{-1}$ // a robust R^2 coefficient

where ${}_I\mathbf{H}_S^w = {}_I\mathbf{X}_S^w\left({}_I\mathbf{X}_S^{wT}\,{}_I\mathbf{X}_S^w\right)^{-1}\,{}_I\mathbf{X}_S^{wT}$, and find $\rho^w = 1 - R_{jS}^{w2}$

get $T_w = (\rho^w)^{-1/2}\widehat{\gamma}_j^w / \sqrt{\widehat{\sigma}^2\left(\sum_i z_{ij}^{w2}\right)^{-1} e_c^{-1}}$ from *Fast Robust Evaluation Procedure*

//compute the approximate robust t-statistic

if $2(1-\Phi(T_w)) < \alpha_j$ **then**

$S = S \cup \{j\}$, $\mathbf{X}_S = \left[\mathbf{1} \quad \mathbf{x}_j\right]$, $\mathbf{X}_S^w = diag\left(\sqrt{w_{iS}^0}\right)\mathbf{X}_S$, and $\mathbf{y}^w = diag\left(\sqrt{w_{iS}^0}\right)\mathbf{y}$,

where w_{iS}^0 is computed using $w_i(r_i;c) = \begin{cases} \left(\left(\frac{r_i}{c}\right)^2 - 1\right)^2 & if\ |r_i| \le c, \\ 0 & if\ |r_i| > c, \end{cases}$

where $\mathbf{r}^0 = \left(\mathbf{y} - \mathbf{X}_S\widehat{\beta}^0\right)/\widehat{\sigma}^0$ using $\mathbf{X}_0^w = \left[1 \quad \sqrt{w_{ij}}\ x_{ij}\right]$, $\mathbf{X}_0^{w2} = \left[1 \quad w_{ij}x_{ij}\right]$, i=1,…,n,

$\widehat{\beta}^0 = \left[\left(\mathbf{X}_0^w\right)^{\mathrm{T}}\mathbf{X}_0^w\right]^{-1}\left(\mathbf{X}_0^{w2}\right)^{\mathrm{T}}\mathbf{y}$,

where $\widehat{\sigma}^0 = 1.483 med\left|\widetilde{\mathbf{r}}^0 - med(\widetilde{\mathbf{r}}^0)\right|$ and $\widetilde{\mathbf{r}}^0 = \mathbf{y} - \mathbf{X}_S\widehat{\beta}^0$

$\alpha_{j+1} = \alpha_j + \Delta\alpha$

$f = j$

else $\alpha_{j+1} = \alpha_j - \alpha_j/\left(1-\alpha_j\right)$

end if

$j = j+1$

until all p covariates have been considered.

α_0: the initial alpha-wealth according to α-investing rule, $\Delta\alpha$: if a hypothesis is rejected, the change of alpha-wealth value, r and $\mathbf{r}$: residuals, S: the set of predictors, α_j: α value in the jth test, c = 4.685, w_i: Tukey's biweight weights, r_i: standardized residuals, Φ: the standard normal cumulative distribution, med: median, MAD: median absolute deviation, diag: diagonal, R_{jS}^{w2}: a robust R^2 coefficient proposed by Renaud and Victoria-Feser (2010)

2.3 Dimensional reduction of correlation matrix (DRCM) method

The DRCM method was suggested by Midi & Uraibi (2014). This method can reduce the time for selecting only the variables which provide important information to the response variable. The procedure consists of two steps: in the first step, DRCM tries to reduce the dimension of correlation matrix by including only those variables that have absolute correlations greater than a threshold value, in the potential model. In the second step, the p-values for the parameter estimates of potential model were computed using multiple linear regression method. The final regression model only includes those variables that are significant. The algorithm of this method, which is based on the regression method, is

shown in Algorithm 3.

Algorithm 3.

Input: data $\mathbf{y}$, $\mathbf{x}_1$, $\mathbf{x}_2$, . . . (standardized);
Initialize $S_1 = \{0\}$, $S = \{0\}$, $j = 1$,
$Cos(\theta_{\mathbf{x},\mathbf{y}}) = \frac{<\mathbf{x},\ \mathbf{y}>}{\|\mathbf{x}\|.\|\mathbf{y}\|} = \frac{Cov(\mathbf{x},\ \mathbf{y})}{\sqrt{Var(\mathbf{x})Var(\mathbf{y})}} = Corr(\mathbf{x}, \mathbf{y})$,
$\widehat{\beta}=\left[\mathbf{X}^{\mathrm{T}}\mathbf{X}\right]^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$, $\frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{y}=\frac{1}{n}\widehat{\beta}=R_{\mathbf{x}y}$, $\mathbf{r}=\mathbf{y}-\overline{\mathbf{y}}=\mathbf{r}=\mathbf{y}-\mathbf{X}\left[\mathbf{X}^{\mathrm{T}}\mathbf{X}\right]^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$,
$Cov(\widehat{\beta}) = \sigma^2\left[\mathbf{X}^{\mathrm{T}}\mathbf{X}\right]^{-1}$, $\widehat{\sigma}^2 = \mathbf{y}^{\mathrm{T}}\mathbf{y}-\widehat{\beta}^{\mathbf{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y}$=MSE. // from the linear regression model $\mathbf{y} = \mathbf{x}\beta + \varepsilon$
Compute $Cos(\theta_{\mathbf{x},\mathbf{y}}) = \frac{<\mathbf{x},\ \mathbf{y}>}{\|\mathbf{x}\|.\|\mathbf{y}\|} = \frac{Cov(\mathbf{x},\ \mathbf{y})}{\sqrt{Var(\mathbf{x})Var(\mathbf{y})}} = Corr(\mathbf{x}, \mathbf{y})$ // First step
$\widehat{\beta}=\left[\mathbf{X}^{\mathrm{T}}\mathbf{X}\right]^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$, $\frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{y}=\frac{1}{n}\widehat{\beta}=R_{xy}$ // R_{xy} is the correlation between $\mathbf{x}$ and $\mathbf{y}$
// The value of $\left|R_{xy}\right|$ is between 0 and 1.
where $\mathbf{X}^{\mathrm{T}}\mathbf{X}=\mathbf{I}$
set threshold $M = \frac{\sum_{j=1}^{p}\left|R_{xy}\right|}{p}$ // Pearson correlation matrix $R_{\mathbf{x}y}$; the number of all candidate covariates p
if $\left|\widehat{\beta}\right| = \left|R_{\mathbf{x}y}\right| \geq M$
compare *Corr(***x**,**y***)* -values to threshold
// The dimension of the correlation matrix is reduced
then
$S_1 = S_1 \cup \{j\}$ // add candidate feature for model
end if
$j = j + 1$
until all p covariates have been considered.
// Second step
set $\alpha_j = \alpha_j$ //(1+*j*-*f*), $S_1 = \{0\}$, $f = j$
get $\hat{t} = \widehat{\beta}_j/(\widehat{\sigma}^2(\left[\mathbf{X}^{\mathrm{T}}\mathbf{X}\right]^{-1})$ // compute t-statistic
if $2\left(1-\Phi\left(\hat{t}\right)\right) < \alpha_j$ // compare p-value
then
$S = S \cup \{j\}$ // add feature from S_1 to model
else
$\alpha_{j+1} = \alpha_j - \alpha_j/\left(1-\alpha_j\right)$
end if
$j = j + 1$
until all covariates in S_1 have been considered.

S_1: the set of candidate predictors in first step, S: the set of predictors, $\left|R_{\mathbf{x}y}\right|$: the absolute values of correlation matrix, Φ: the standard normal cumulative distribution, f: the time at which the last hypothesis is rejected, α_j: α value in the jth test

2.4 Nonparametric DRCM (N-DRCM) method

The N-DRCM method is a nonparametric version of the DRCM method. The procedure consists of two steps. In the first step, Spearman correlation matrix is used to determine monotonic relationship between variables. These variables can be continuous or at least one of them can be ordinal. N-DRCM tries to reduce the dimension of correlation matrix by including only those variables that have absolute correlations greater than a threshold value, in the potential model. In the second step, the p-values for

the parameter estimates of potential model are computed by robust regression method using iteratively reweighted least squares (IRLS). The final regression model only includes those variables that are significant. This algorithm is shown in Algorithm 4.

Algorithm 4.

Input: data $\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2, \ldots$(standardized);
Initialize $S_1 = \{0\}, S = \{0\}, j = 1,$

$$_{SP}Cos\left(\theta_{(^r\mathbf{x}^r\mathbf{y})}\right) = \frac{Cov(^r\mathbf{x},^r\mathbf{y})}{\sqrt{Var(^r\mathbf{x})Var(^r\mathbf{y})}} = {}_{SP}Corr(^r\mathbf{x},^r\mathbf{y})$$

// Spearman correlation matrix for the ranked data $^r\mathbf{y}$, $^r\mathbf{x}_1$, $^r\mathbf{x}_2$,...

$$^r\widehat{\beta} = \left[^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{X}\right]^{-1}{}^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{y}\ ,\ \frac{1}{n}{}^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{y} = \frac{1}{n}{}^r\widehat{\beta}^{\mathrm{T}} = R_{(^r\mathbf{x},^r\mathbf{y})},$$

$$^r\mathbf{r} = {}^r\mathbf{y} - {}^r\overline{\mathbf{y}} = \mathbf{r} = {}^r\mathbf{y} - {}^r\mathbf{X}\left[^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{X}\right]^{-1}{}^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{y},$$

$$Cov(^r\widehat{\beta}) = {}^r\sigma^2\left[^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{X}\right]^{-1},\ ^r\widehat{\sigma}^2 = {}^r\mathbf{y}^{\mathrm{T}}{}^r\mathbf{y} - {}^r\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \text{MSE}.$$

// from the linear regression model $\mathbf{y} = \mathbf{x}\beta + \varepsilon$

Compute $_{SP}Corr(^r\mathbf{x},^r\mathbf{y}) \frac{Cov(^r\mathbf{x},^r\mathbf{y})}{\sqrt{Var(^r\mathbf{x})Var(^r\mathbf{y})}} = 1 - \frac{6\sum d_i^2}{\sqrt{Var(^r\mathbf{x})Var(^r\mathbf{y}))}}$ // First step

// Spearman correlation matrix

where $\sqrt{Var(^r\mathbf{x})\,Var(^r\mathbf{y})} = \begin{cases} n(n^2-1) & if \text{ all } n \text{ ranks are distinct integers,} \\ (n^2-1)/12 & if \text{ all ranks are distinct,} \end{cases}$

$$^r\widehat{\beta} = \left[^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{X}\right]^{-1}{}^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{y}\ ,\ \frac{1}{n}{}^r\mathbf{X}^{\mathrm{T}}{}^r\mathbf{y} = \frac{1}{n}{}^r\widehat{\beta}^{\mathrm{T}} = R_{(^r\mathbf{x},^r\mathbf{y})}$$

$_{SP}R_{xy}$ is the Spearman correlation between the ranked $\mathbf{x}$ and $\mathbf{y}$

// The value of $\left|{}_{SP}R_{(^r\mathbf{x},^r\mathbf{y})}\right|$ is between 0 and 1.

where $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{I}$

set threshold $M = \frac{\sum_{j=1}^{p}\left|{}_{SP}R_{(^r\mathbf{x},^r\mathbf{y})}\right|}{p}$ // Spearman correlation matrix $_{SP}R_{(^r\mathbf{x},^r\mathbf{y})}$;

// The number of all candidate covariates p

if $\left|{}_{SP}R_{(^r\mathbf{x},^r\mathbf{y})}\right| \geq M$

compare $_{SP}Corr(^r\mathbf{x},^r\mathbf{y})$ -values to threshold

// The dimension of the correlation matrix is reduced

then

$S_1 = S_1 \cup \{j\}$ // add candidate feature for model

end if

$j = j + 1$

until all p covariates have been considered.

// Second step

Compute $\underset{\beta}{min}\ \sum_{j=1}^{n}\rho\left(\frac{y_j - x_j^{\mathrm{T}}\beta}{\sigma}\right)$ // minimize β's using the standardized data from the linear model $\mathbf{y} = \mathbf{x}\beta + \varepsilon$

$\sum_{j=1}^{n} x_{ij}\psi\left(\frac{y_j - x_j^{\mathrm{T}}\beta}{\sigma}\right) = 0$ for all i=0,1,2,...,p // solution using nonlinear optimization method – Iteratively reweighted least squares (IRLS)

where $\psi = \rho^{\mathrm{T}}$, $x_{i0} = 1$, $\sigma = \widehat{\sigma}^0 = 1.483 med\left|\left(y_j - x_j\widehat{\beta}_0\right) - med(y_j - x_j\widehat{\beta}_0)\right|$,

$$\beta_{t+1} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{w}_t\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{w}_t$$

where $w_{jt} = \begin{cases} \frac{\psi\left[\left(y_j - x^{\mathrm{T}}\beta_{jt}\right)/\sigma_t\right]}{\left(y_j - x^{\mathrm{T}}\beta_{jt}\right)/\sigma_t} & if\ y_j \neq x^{\mathrm{T}}\beta_{jt} \\ 1 & if\ y_j = x^{\mathrm{T}}\beta_{jt} \end{cases}$

$w(u) = min\begin{cases} 1 & if\ |u| < 0 \\ \frac{c}{|u|} & if\ |u| \geq 0 \end{cases}$ // Huber' method (c=1.345)

$Cov\left(\widehat{\beta}\right) = \sigma^2 \frac{\sum_{i=1}^{n} \psi^2\left[\left(y_i - x_j^{\mathrm{T}}\beta\right)/\sigma\right]}{\left\{\sum_{i=1}^{n} \psi^{\mathrm{T}}\left[\left(y_i - x_j^{\mathrm{T}}\beta\right)/\sigma\right]\right\}^2}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}, Var\left(\widehat{\beta}\right) = \widehat{\sigma}^2\left(\mathbf{X}^{\mathrm{T}}\mathbf{w}_t\mathbf{X}\right)^{-1}$
set $\alpha_j = \alpha_j$ /(1+*j*-*f*), $S_1 = \{0\}$, f $f = j$
get $\hat{t}_w = \left(\mathbf{X}^{\mathrm{T}}\mathbf{w}_t\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{w}_t\mathbf{y} \,/\, \sqrt{\widehat{\sigma}^2(\mathbf{X}^{\mathrm{T}}\mathbf{w}_t\mathbf{X})^{-1}}$ // compute the robust *t*-statistic
if $2\left(1 - \Phi\left(\hat{t}_w\right)\right) < \alpha_j$ // compare *p*-value
then
$S = S \cup \{j\}$ // add feature from S_1 to model
else
$\alpha_{j+1} = \alpha_j - \alpha_j/\left(1 - \alpha_j\right)$
end if
$j = j + 1$
until all covariates in S_1 have been considered.

S_1: the set of candidate predictors in first step, S: the set of predictors, d_i: difference in paired ranks, $\mathbf{w}_t$: diagonal matrix of weights, $\rho(.)$: likelihood function for a suitable choice of the distribution of the residuals, Φ: the standard normal cumulative distribution, f: the time at which the last hypothesis is rejected, α_j: α value in the *j*th test, ψ: influence function

2.5 Simulation study

This simulation study has been designed in a similar way to studies conducted by Rahman & Khan (2010) and Dupuis & Victoria-Feser (2013). A linear model was established as

$$y = x_1 + x_2 + \ldots + x_k + \sigma\varepsilon_j \tag{1}$$

where $x_1, x_2, \ldots, x_k$ are multivariate normal variables with $E\left(x_j\right) = 0$, $Var\left(x_j\right) = 1$, and $corr\left(x_j, x_i\right) = \theta$ $(i \neq j,\ \ i, j = 1, \ldots, k)$. θ is chosen to produce a range of theoretical $R^2 = \left(\mathrm{Var}\,(\mathrm{y}) - \sigma^2\right)/\mathrm{Var}\,(\mathrm{y})$ values for (1) and σ to give t values for target covariates of about 5-6 under normality. $x_1, x_2, \ldots, x_k$ represent k target covariates. ε is an independent standard normal variable. A set of p predictors was generated as follows:

$$x_{k+1} = x_1 + \delta e_{k+1}$$
$$x_{k+2} = x_1 + \delta e_{k+2}$$
$$\vdots$$
$$x_{3k} = x_k + \delta e_{3k} \tag{2}$$

Variables $x_{k+1}, x_{k+2}, \ldots, x_{3k}$ were noise covariates that correlated with target covariates. Variables $x_{3k+1}, , \ldots, x_p$ were the noise covariates that did not correlate with the target covariates ($x_j = e_j$, $j = 3k + 1,\ 3k + 2, \ldots, p$). $e_{k+1}, \ldots, e_p$ were independent standard normal variables. In each scenario, the number of target covariates was set as five. The constant $\delta = 3.18$ was selected so that $corr(x_1, x_{k+1}) = corr(x_1, x_{k+2}) = \ldots = corr(x_k, x_{3k}) = 0.3$. The estimated final model was given in equation 3.

$$\mathbf{y} = \beta_0 + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \beta_{new}\mathbf{x}_{new} + \varepsilon \tag{3}$$

The datasets consisted of "normal (no contamination)" and "outliers (with 5% and 10%)" to examine the effect outliers had on datasets. The datasets were generated using $\varepsilon \sim \mathrm{N}(0,\ 1)$ for normal data, $\varepsilon \sim 95\%\mathrm{N}(0,\ 1) + 5\%\mathrm{N}(30,\ 1)$ for the dataset with 5% outliers and $\varepsilon \sim 90\%\mathrm{N}(0,\ 1) + 10\%\mathrm{N}(30,\ 1)$ for the dataset with 10% outliers. To examine the effect of multicollinearity in datasets, correlations

among target regressors were specified as $\theta_1 = 0.1$ $\left(R^2 = 0.20\right)$ and $\theta_2 = 0.85$ $\left(R^2 = 0.80\right)$ so that $corr\left(x_j, x_i\right) = \theta, \quad (i \neq j \quad i,j = 1,\ldots,k)$. A total of 36 scenarios were created through combining different data types, including the uncontaminated dataset and the datasets with 5% and 10% outliers, with 50, 100, 250, 500, 750, and 1,000 independent variables. The sample size was 5,000 and the number of repetition was 100. A total of 14,400 models were examined. The initial-wealth and pay-out were respectively selected 0.5 and 0.05 for VIF and R-VIF regression methods. In each condition, the root mean square error (RMSE) values calculated through the four methods were recorded. This simulation was executed using the MATLAB/Simulink R2015a program (toolboxes: statistics and machine learning, curve fitting, optimization, and global optimization) by a computer with Intel(R) Core(TM) i7-6500U CPU @ 2.50 GHz, 2592 Mhz, two cores, and four logical processors.

2.6 Real data

Crime dataset taken from UCI Machine Learning Respiratory (Redmond, 2009) was used to compare the performances of DRCM, N-DRCM, VIF regression and R-VIF regression methods. This dataset consists socio-economic data from the 1990 US Census, law enforcement data from the 1990 US Law Enforcement Management and Administrative Statistics (LEMAS) survey, and crime data from the 1995 Federal Bureau of Investigation' Uniform Crime Reporting (FBI UCR). Crime dataset includes $n = 1994$ observations, the violent crime per capita variable ($\mathbf{y}$), and 122 predictors ($\mathbf{x}$) that have a possible relationship with crime in order to estimate ($\mathbf{y}$). The RMSE, R^2 and estimation values (beta, standard error, t-statistic, and p-value) of the final models selected using each method were calculated.

3. Results

3.1 Simulation

In case presences of multicollinearity and outliers, while the number of candidate covariates that can be included in the model increased, the values (average time, average numbers of covariates with different relationships) that show the performances of DRCM, N-DRCM, VIF regression and R-VIF regression methods are demonstrated in Table 1, Table 2, and Table 3, respectively.

In all scenarios, all the target independent variables were selected to the final models by four methods. Respectively, the DRCM and N-DRCM methods reached the final model in the shortest time.The plots of average times taken to reach the final models for fast regression methods in datasets with outliers for each theta value were given Fig. 1, respectively. When the number of variables was 250 or less in datasets with 5% and 10% outliers, the times taken to reach the final models for both DRCM and N-DRCM were similar. However, when theta value was 0.10 and the number of variables was 500 or more, the times taken to reach the final models in datasets with 10% outliers were significantly longer than the those of DRCM and N-DRCM in datasets with 5% outliers. Moreover, when theta value was 0.85 and the number of variables was 500 or more, the times to reach the final models using DRCM in datasets with outliers were slightly shorter than those of N-DRCM. When the number of variables was over 750 in both datasets with outliers, the times to reach the final models decreased in line with increasing theta values for both DCRM and N-DCRM. The decrement amount increased as the level of outliers increased. However, this was not observed in the R-VIF and VIF regression methods. The time to reach the final model using R-VIF regression was approximately two times shorter than that of VIF regression. The largest numbers of noise variables were selected to the final models using DRCM and N-DRCM methods.

The RMSE values obtained using DRCM, N-DRCM and VIF regression were similar in each scenario. The RMSE values calculated by each method were higher in the datasets with outliers compared to uncontaminated datasets. In addition, the RMSE values tended to decrease when the number of variables increased. This conclusion applies to the RMSE values obtained using R-VIF regression, except for when the number of variables in datasets with 10% outliers was 500 or above. When the number of variables in datasets with 10% outliers was 500 or above, the RMSE values obtained using R-VIF regression were lower than the values obtained in uncontaminated datasets.

When the number of variables was 500 or above, the approximate ratios of total noise variables chosen for the final models using DRCM and N-DRCM methods were found to be 2.1% in uncontaminated datasets, 2.3% in datasets with 5% outliers, and 4.1% in datasets with 10% outliers.In addition, when the number of variables was 500 or above, the approximate ratios of total noise variables chosen for the final model using R-VIF and VIF regression were 1.8‰in datasets with outliers and 1.7‰in uncontaminated datasets. In addition, in all datasets, when the number of variables was over 750, no noise variables were chosen for the final model by R-VIF regression. In all scenarios, the R-VIF regression method omitted noise covariates that did not correlate with the target variables in the final model. The time taken for each method to reach the final model was longer in datasets with outliers than in uncontaminated datasets. This became more evident as the number of variables increased. In addition, in dataset with 10% outliers, the time each method took to reach the final model was slightly higher than the time taken in dataset with 5% outliers. This became more evident when the number of variables was 500 or more.

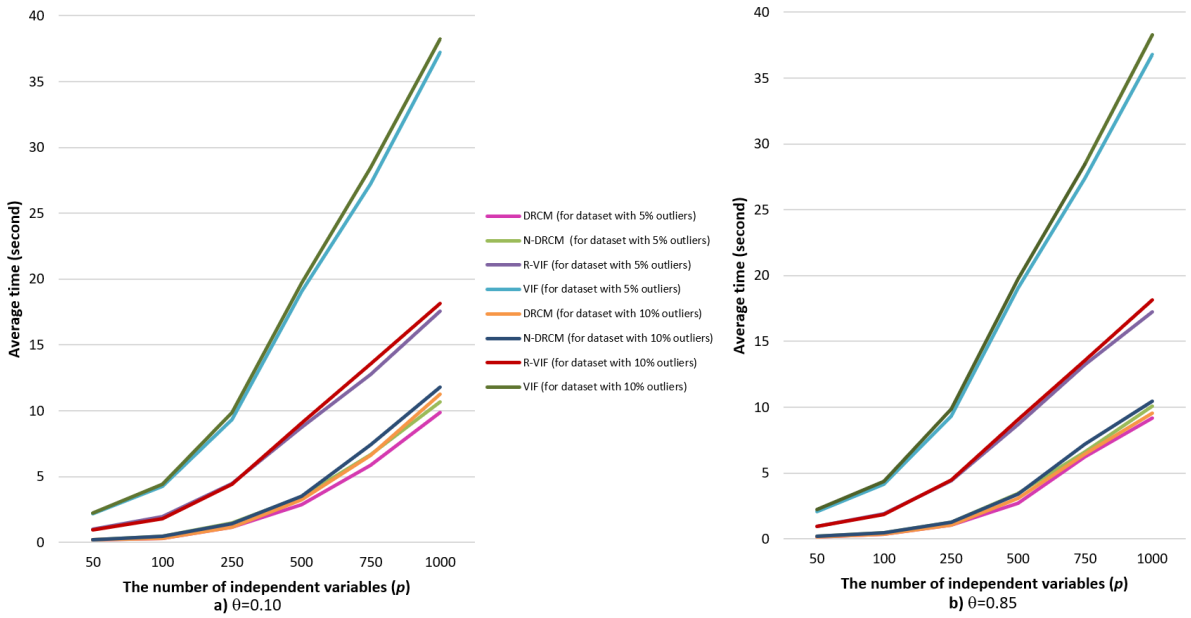


Fig. 1. The plots of average times taken to reach the final models for fast regression methods in the datasets with outliers for **a)** $\theta = 0.10$ and **b)** $\theta = 0.85$.

In all datasets, when the theta value was 0.85 and the number of variables was over 750, 19.8% of noise covariates that correlated with target variables was involved in the final models obtained by DRCM and N-DRCM. A further 9% were included in the final model when using VIF regression method. Also the numbers of total noise covariates selected to final models by both DRCM and N-DRCM methods increased slightly with increasing of multicollinearity level when the number of variables was over 100. It was determined that the numbers of total noise covariates selected to final models by the R-VIF and VIF regression methods decreased when the numbers of variables increased in both the datasets with outliers. The numbers of total noise covariates selected to finals model by both R-VIF and VIF regression methods had not changed considerably with increasing of multicollinearity level. Additionally, the numbers of total noise covariates selected to final models by the R-VIF and VIF regression methods were absent in uncontaminated dataset.

3.2 Real data

This large dataset with sample size (n=1994) and number of predictors ($p = 122$) was firstly examined in terms of multicollinearity and outliers. The VIF values of 88% of the variables were greater than 10, and their collinearity tolerance values were very close to zero. Condition index values of all dimensions except the twenty two dimensions were above 15. Moreover, the most of the variables were skew

Table 1. The performances of fast regression methods in uncontaminated dataset.

n=5000		**No Contamination**							
		R^2**=0.20,** (θ**=0.10**)				R^2**=0.80,** (θ**=0.85**)			
p	**Results**	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***
50	Avg.Time	0.118	0.140	0.569	1.304	0.106	0.130	0.570	1.305
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0	0	0	0	0	0
	C ($p_C = 35$) (%)	0.06	0.06	0	0	0.06	0.06	0	0
	D ($p_D = 45$) (%)	0.04	0.04	0	0	0.04	0.04	0	0
	RMSE	0.921	0.921	0.923	0.924	0.923	0.923	0.923	0.924
100	Avg.Time	0.182	0.250	1.092	2.480	0.159	0.224	1.119	2.487
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0	0	0	0	0	0
	C ($p_C = 85$) (%)	0.86	0.85	0	0	1.29	1.16	0	0
	D ($p_D = 95$) (%)	0.77	0.76	0	0	1.15	1.04	0	0
	RMSE	0.921	0.921	0.919	0.920	0.920	0.920	0.910	0.920
250	Avg.Time	0.592	0.705	2.702	5.817	0.604	0.704	2.693	5.823
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	9.9	0	0	0	0	0	0
	C ($p_C = 235$) (%)	1.28	1.26	0	0	1.59	1.49	0	0
	D ($p_D = 245$) (%)	1.22	1.61	0	0	1.53	1.43	0	0
	RMSE	0.907	0.906	0.904	0.904	0.907	0.906	0.904	0.904
500	Avg.Time	1.903	2.145	5.335	11.416	1.888	1.999	5.425	11.460
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.1	0.1	0	0.1	0.1	10	0	0.1
	C ($p_C = 485$) (%)	1.86	1.65	0	0	2.47	2.27	0	0
	D ($p_D = 495$) (%)	1.82	1.62	0	0.002	2.42	2.43	0	0.002
	RMSE	0.908	0.908	0.907	0.908	0.905	0.905	0.904	0.904
750	Avg.Time	4.190	4.482	8.000	17.004	4.386	4.811	8.842	17.944
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0	0	0	0	0	0
	C ($p_C = 735$) (%)	1.77	1.64	0	0	2.17	2.14	0	0
	D ($p_D = 745$) (%)	1.75	1.62	0	0	2.14	2.11	0	0
	RMSE	0.905	0.904	0.903	0.904	0.905	0.905	0.904	0.904
1000	Avg.Time	6.365	6.796	10.757	22.835	6.032	6.740	11.697	23.916
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	9.9	9.9	0	9.9	19.8	19.8	0	9.9
	C ($p_C = 985$) (%)	2.47	1.94	0	0	2.14	2.03	0	0
	D ($p_D = 995$) (%)	2.54	2.02	0	0.10	2.32	2.21	0	0.10
	RMSE	0.893	0.892	0.891	0.893	0.892	0.892	0.891	0.893

p: The number of predictors, Avg: Average, A: Average number of target covariates, B: Average number of noise covariates that correlated with target covariates, C: Average number of noise covariates that did not correlate with target covariates, D: Average number of total noise covariates, RMSEA: Root mean square error, VIF: Variance inflation factor, R-VIF: Robust VIF, DRCM: Dimensional reduction of correlation matrix, N-DRCM: Nonparametric DRCM

Table 2. The performances of fast regression methods in dataset with 5% outliers.

n=5000		**5% outliers**							
		R^2=0.20, (θ=0.10)				**R^2=0.80, (θ=0.85)**			
p	**Results**	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***
50	Avg.Time	0.186	0.213	1.003	2.186	0.153	0.210	0.965	2.094
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.2	0.2	0.1	0.1	0	0	0	0
	C ($p_C = 35$) (%)	0	0	0	0	0	0	0	0
	D ($p_D = 45$) (%)	0.04	0.04	0.02	0.02	0	0	0	0
	RMSE	0.968	0.968	0.970	0.970	0.969	0.969	0.970	0.970
100	Avg.Time	0.315	0.486	1.983	4.258	0.330	0.462	1.882	4.149
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0.1	0.1	0	0	0.1	0.1
	C ($p_C = 85$) (%)	1.41	1.43	0	0	1.62	1.55	0	0
	D ($p_D = 95$) (%)	1.26	1.28	0.01	0.01	1.45	1.39	0.01	0.01
	RMSE	0.967	0.967	0.966	0.966	0.967	0.967	0.966	0.966
250	Avg.Time	1.157	1.504	4.476	9.312	1.043	1.257	4.430	9.319
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0.1	0.1	0	0	0.1	0.1
	C ($p_C = 235$) (%)	1.74	1.74	0	0	2.08	2.04	0	0
	D ($p_D = 245$) (%)	1.67	1.67	0.004	0.004	2.00	1.96	0.004	0.004
	RMSE	0.953	0.954	0.953	0.953	0.951	0.950	0.952	0.953
500	Avg.Time	2.871	3.457	8.734	19.079	2.708	3.442	8.713	19.029
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.1	0.2	0.1	0.1	10	10.2	0.1	0.1
	C ($p_C = 485$) (%)	2.08	2.06	0	0	2.47	2.27	0	0
	D ($p_D = 495$)(%)	2.04	2.02	0.002	0.002	2.62	2.43	0.002	0.002
	RMSE	0.952	0.951	0.949	0.950	0.952	0.951	0.949	0.949
750	Avg.Time	5.835	6.683	12.745	27.248	6.214	6.585	13.242	27.412
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$)(%)	0.1	0	0.2	0.1	0	0.1	0.2	0.1
	C ($p_C = 735$) (%)	1.9	1.9	0	0	2.44	2.38	0	0
	D ($p_D = 745$) (%)	1.88	1.87	0.003	0.001	2.41	2.35	0.003	0.001
	RMSE	0.950	0.950	0.948	0.949	0.950	0.950	0.948	0.949
1000	Avg.Time	9.870	10.681	17.585	37.220	9.177	10.089	17.214	36.788
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	19.8	19.8	0	9.9	19.8	19.8	0	9.9
	C ($p_C = 985$) (%)	2.53	1.82	0	0	2.13	2.13	0	0
	D ($p_D = 995$) (%)	2.71	2.00	0	0.10	2.31	2.31	0	0.10
	RMSE	0.937	0.936	0.934	0.938	0.937	0.936	0.934	0.938

p: The number of predictors, Avg: Average, A: Average number of target covariates, B: Average number of noise covariates that correlated with target covariates, C: Average number of noise covariates that did not correlate with target covariates, D: Average number of total noise covariates, RMSEA: Root mean square error, VIF: Variance inflation factor, R-VIF: Robust VIF, DRCM: Dimensional reduction of correlation matrix, N-DRCM: Nonparametric DRCM

Table 3. performances of fast regression methods in dataset with 10% outliers.

n=5000		**10% outliers**							
		R^2=0.20, (θ=0.10)				**R^2=0.80, (θ=0.85)**			
p	**Results**	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***	***DRCM***	***N-DRCM***	***R-VIF***	***VIF***
50	Avg.Time	0.202	0.219	0.945	2.232	0.156	0.200	0.939	2.213
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.2	0.2	0.1	0.1	0	0	0	0
	C ($p_C = 35$) (%)	0	0	0	0	0	0	0	0
	D ($p_D = 45$) (%)	0.04	0.04	0.02	0.02	0	0	0	0
	RMSE	1.019	1.020	1.019	1.020	1.019	1.019	1.019	1.020
100	Avg.Time	0.327	0.473	1.824	4.395	0.341	0.447	1.860	4.383
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0.1	0.1	0	0	0.1	0.1
	C ($p_C = 85$) (%)	2.02	2.00	0	0	2.99	2.94	0	0
	D ($p_D = 95$) (%)	1.81	1.79	0.01	0.01	2.67	2.63	0.01	0.01
	RMSE	1.016	1.016	1.015	1.016	1.016	1.016	1.015	1.016
250	Avg.Time	1.152	1.438	4.408	9.881	1.030	1.251	4.455	9.853
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0	0	0.1	0.1	0	0	0.1	0.1
	C ($p_C = 235$) (%)	3.10	3.07	0	0	3.25	3.25	0	0
	D ($p_D = 245$) (%)	2.98	2.95	0.004	0.004	3.12	3.11	0.004	0.004
	RMSE	1.003	1.005	1.001	1.002	1.002	1.002	1.001	1.001
500	Avg.Time	3.217	3.525	9.058	19.706	3.063	3.380	9.058	19.723
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.2	0.2	0.1	0.1	10	10	0.1	0.1
	C ($p_C = 485$) (%)	3.47	3.20	0	0	3.84	3.26	0	0
	D ($p_D = 495$)(%)	3.40	3.14	0.002	0.002	3.96	3.40	0.002	0.002
	RMSE	0.997	0.996	0.728	0.997	0.998	0.997	0.728	0.997
750	Avg.Time	6.625	7.392	13.568	28.448	6.463	7.189	13.571	28.481
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	0.2	0.1	0.2	0.2	10	9.9	0.2	0.2
	C ($p_C = 735$)(%)	3.91	3.88	0	0	4.17	4.16	0	0
	D ($p_D = 745$) (%)	3.86	3.83	0.003	0.003	4.25	4.03	0.003	0.003
	RMSE	0.996	0.996	0.725	0.998	0.996	0.996	0.725	0.998
1000	Avg.Time	11.236	11.799	18.142	38.231	9.528	10.459	18.138	38.265
	A ($p_A = 5$) (%)	100	100	100	100	100	100	100	100
	B ($p_B = 10$) (%)	9.9	19.8	0	9.9	19.8	19.8	0	9.9
	C ($p_C = 985$) (%)	4.54	4.42	0	0	5.20	5.05	0	0
	D ($p_D = 995$) (%)	4.59	4.57	0	0.1	5.35	5.20	0	0.1
	RMSE	0.984	0.986	0.718	0.986	0.986	0.986	0.718	0.985

p: The number of predictors, Avg: Average, A: Average number of target covariates, B: Average number of noise covariates that correlated with target covariates, C: Average number of noise covariates that did not correlate with target covariates, D: Average number of total noise covariates, RMSEA: Root mean square error, VIF: Variance inflation factor, R-VIF: Robust VIF, DRCM: Dimensional reduction of correlation matrix, N-DRCM: Nonparametric DRCM

distributed and contained outliers. The estimation values of the final models (without constant) selected using each method for "crime data" are shown Table 4.

The racepctblack (percentage of population that is African American), PctIlleg (percentage of kids born to never married), PctPersDenseHous (percent of persons in dense housing (more than 1 person per room)), NumStreet (number of homeless people counted in the street) variables were selected for the final models by all four methods. The number of predictors selected for the final models ranged from 14 to 16. Approximately the numbers of predictors selected by all methods to their final models were similar. In descending order, these methods were N-DRCM, R-VIF regression, VIF regression, and DRCM. The highest R^2 value was obtained by the R-VIF regression method, followed by the N-DRCM method. The R^2 values obtained by VIF regression and DRCM methods were similar and considerably lower than the values obtained by the other two methods. While the RMSE value obtained with the VIF regression method was the lowest, this method was followed by the R-VIF regression, N-DRCM, and DRCM methods, respectively. Overall, R-VIF regression performed better because its final model had the highest R^2 among those obtained with other methods, and the lowest RMSE value among those obtained with others excepting VIF regression.

4. Discussion and conclusion

When large datasets contain multicollinearity and outliers, the use of fast regression algorithms has become mandatory to address the lack of traditional methods and the loss of information that occurs when using traditional methods. In the literature review, it was noted that a limited number of researches about fast regression methods are being conducted. Lin *etal.* (2021) compared stepwise regression, LASSO, FoBa, GPS methods to test the performance of the VIF regression method they developed. They found the performance of VIF regression to be better than other algorithms in terms of computation speed, out-of-sample, out-of-sample error, mFDR control, etc. Dupuis & Victoria-Feser (2013) and Seo (2018) suggested using the R-VIF regression in place of classical VIF regression to obtain faster estimations when working with large datasets that contain outliers. In addition, Midi & Uraibi (2014) compared DRCM, VIF regression and Adaptive Lasso methods, and they obtained that the performance of DRCM method was more efficient than the others. Shahriari (2014) examined the performances of LARS, R-LARS, R-VIF and JKR-LARS methods in datasets with outliers and/or leverage points. Shahriari (2014) found that JKR-LARS performed similarly to R-LARS and R-VIF in datasets with outliers while outperforming R-LARS in datasets with high leverage points. However, according to her study, R-VIF failed to robustly sequence predictor variables in datasets with high leverage points. Uraibi (2020) investigated that VIFRegSd2, VIFRegSd3, and ISIS method in ultrahigh dimensional feature space when presence of collinearity structure. Uraibi (2020) found that VIFRegSd2 and VIFRegSd3 methods outperform ISIS, additionally VIFRegSd2 method can be used in practice for ultrahigh feature space and small sample size.

In this study, the performances of DRCM, N-DRCM, VIF regression, and R-VIF regression in relation to large datasets with varying levels of multicollinearity and outliers were examined in different scenarios. This study proposed that the N-DRCM method could be used as a fast regression estimator. As the number of variables and the level of outliers increased, the time taken to reach the final model by each method increased. When the number of variables was 500 or above and the level of outliers in the dataset increased, the times taken to reach the final models by DRCM and N-DRCM methods increased. When the level of multicollinearity and the number of variables ($p > 500$) increased, the times to reach the final models using DRCM in datasets with outliers were slightly shorter than the those of N-DRCM. However, in all scenarios, DRCM and N-DRCM were found to be the fastest methods to reach the final models. When the number of variables was over 750 in uncontaminated datasets, the times taken to reach the final models using DRCM and N-DRCM methods decreased with increasing of multicollinearity level. Moreover the numbers of total noise covariates selected to final models by both DRCM and N-DRCM methods increased slightly with increasing of multicollinearity level when the number of variables was over 100. It was observed that the numbers of total noise covariates and the numbers of total noise covariates that did not correlate with the target variables selected for the final models by the DRCM and N-DRCM methods were higher than those achieved via R-VIF and VIF

Table 4. The estimation values of final models selected using each methods (n=1994, p=122).

Methods	Variables	Beta	SE	t-statistic	p-value	R^2	RMSE
VIF R.	racepctblack	0.177	0.024	7.329	<0.001	0.640	0.140
	pctUrban	0.054	0.008	6.871	<0.001		
	pctWInvInc	-0.263	0.024	-10.763	<0.001		
	MalePctNevMarr	-0.104	0.023	-4.523	<0.001		
	PctWorkMom	-0.117	0.020	-5.962	<0.001		
	PctIlleg	0.345	0.034	10.289	<0.001		
	PersPerOccupHous	-0.356	0.036	-9.860	<0.001		
	PctPersDenseHous	0.281	0.036	7.756	<0.001		
	PctHousLess3BR	-0.139	0.034	-4.073	<0.001		
	MedNumBR	-0.053	0.018	-3.000	0.003		
	PctVacantBoarded	0.066	0.018	3.596	<0.001		
	MedOwnCostPctIncNoMtg	-0.061	0.018	-3.318	0.001		
	NumStreet	0.242	0.036	6.767	<0.001		
	LemasSwornFT	-0.275	0.074	-3.715	<0.001		
	PolicOperBudg	0.204	0.076	2.685	0.007		
R-VIF R.	racepctblack	0.220	0.021	10.681	<0.001	0.904	0.439
	agePct12t29	-0.185	0.044	-4.226	<0.001		
	agePct16t24	0.147	0.041	3.575	<0.001		
	numbUrban	-0.129	0.027	-4.859	<0.001		
	pctUrban	0.064	0.013	5.112	<0.001		
	pctWWage	-0.065	0.030	-2.175	0.030		
	pctWRetire	-0.033	0.015	-2.295	0.022		
	OtherPerCap	1.119	0.010	108.563	<0.001		
	PctEmploy	0.082	0.029	2.845	0.005		
	MalePctDivorce	0.070	0.020	3.485	0.001		
	PctKids2Par	-0.211	0.042	-4.979	<0.001		
	PctWorkMom	-0.053	0.013	-3.983	<0.001		
	PctIlleg	0.214	0.030	7.230	<0.001		
	PctPersDenseHous	0.220	0.015	14.621	<0.001		
	HousVacant	0.186	0.025	7.541	<0.001		
	NumStreet	0.111	0.014	8.090	<0.001		

R: Regression, SE: Standard error, RMSE: Residual mean square estimation, VIF: Variance inflation factor, R-VIF: Robust VIF, racepctblack: percentage of population that is African American, pctUrban: percentage of people living in areas classified as urban, pctWInvInc: percentage of households with investment, MalePctNevMarr: percentage of males who have never married, PctWorkMom: percentage of moms of kids under 18 in labor force, PctIlleg: percentage of kids born to never married, PersPerOccupHous: mean persons per household, PctPersDenseHous: percent of persons in dense housing (more than 1 person per room), PctHousLess3BR: percent of housing units with less than 3 bedrooms, MedNumBR: median number of bedrooms, PctVacantBoarded: percent of vacant housing that is boarded up, MedOwnCostPctIncNoMtg: median owners cost as a percentage of household income, NumStreet: number of homeless people counted in the Street, LemasSwornFT: number of sworn full time police officers, PolicOperBudg: police operating budget, agePct12t29: percentage of population that is 12-29 in age, agePct16t24: percentage of population that is 16-24 in age, numbUrban: number of people living in areas classified as urban, pctWWage: percentage of households with wage or salary income in 1989, pctWRetire: percentage of households with retirement income in 1989, OtherPerCap: per capita income for people with 'other' heritage, PctEmploy: percentage of people 16 and over who are employed, MalePctDivorce: percentage of males who are divorced, PctKids2Par: percentage of kids in family housing with two parents, HousVacant: number of vacant households

Table 4.(continue). The estimation values of final models selected using each methods (n=1994, p=122).

Methods	Variables	Beta	SE	t-statistic	p-value	R^2	RMSE
DRCM	racepctblack	0.873	0.103	8.490	<0.001	0.649	0.595
	agePct12t29	-1.768	0.115	-15.442	<0.001		
	pctUrban	0.141	0.036	3.904	<0.001		
	pctWPubAsst	0.324	0.113	2.875	0.004		
	PctLess9thGrade	-0.991	0.211	-4.689	<0.001		
	PctNotHSGrad	0.665	0.239	2.780	0.006		
	MalePctNevMarr	0.762	0.151	5.032	<0.001		
	PctIlleg	1.307	0.150	8.733	<0.001		
	PctPersDenseHous	0.956	0.095	10.063	<0.001		
	PctHousLess3BR	0.444	0.097	4.560	<0.001		
	HousVacant	0.681	0.119	5.711	<0.001		
	PctHousNoPhone	0.649	0.102	6.390	<0.001		
	MedOwnCostPctIncNoMtg	-0.533	0.074	-7.216	<0.001		
	NumStreet	0.559	0.170	3.298	0.001		
N-DRCM	racepctblack	0.232	0.022	10.574	<0.001	0.748	0.451
	agePct12t29	-0.135	0.025	-5.347	<0.001		
	Pct65up	-0.069	0.022	-3.081	0.002		
	pctWPubAsst	0.110	0.020	5.611	<0.001		
	PctLess9thGrade	-0.206	0.038	-5.441	<0.001		
	PctNotHSGrad	0.275	0.048	5.741	<0.001		
	PctOccupManu	-0.054	0.020	-2.707	0.007		
	MalePctNevMarr	0.052	0.024	2.179	0.030		
	PersPerFam	-0.136	0.020	-6.843	<0.001		
	PctIlleg	0.296	0.029	10.239	<0.001		
	PctNotSpeakEnglWell	-0.101	0.029	-3.445	0.001		
	PctPersDenseHous	0.376	0.030	12.487	<0.001		
	HousVacant	0.153	0.019	8.220	<0.001		
	NumStreet	0.076	0.014	5.535	<0.001		
	LemasSwornFT	-0.039	0.012	-3.268	0.001		
	LandArea	-0.034	0.016	-2.164	0.031		

R: Regression, SE: Standard error, RMSE: Residual mean square estimation, DRCM: Dimensional reduction of correlation matrix, N-DRCM: Nonparametric DRCM, racepctblack: percentage of population that is African American, agePct12t29: percentage of population that is 12-29 in age, pctUrban: percentage of people living in areas classified as urban, pctWPubAsst: percentage of households with public assistance income in 1989, PctLess9thGrade: percentage of people 25 and over with less than a 9th grade education, PctNotHSGrad: percentage of people 25 and over that are not high school graduates, PctHousNoPhone: percent of occupied housing units without phone, MedOwnCostPctIncNoMtg: median owners cost as a percentage of household income, MalePctNevMarr: percentage of males who have never married, PctIlleg: percentage of kids born to never married, PctPersDenseHous: percent of persons in dense housing, PctHousLess3BR: percent of housing units with less than 3 bedrooms, HousVacant: number of vacant households, PctHousNoPhone: percent of occupied housing units without phone, MedOwnCostPctIncNoMtg: median owners cost as a percentage of household income - for owners without a mortgage, NumStreet: number of homeless people counted in the street, agePct65up: percentage of population that is 65 and over in age, PctOccupManu: percentage of people 16 and over who are employed in manufacturing, PersPerFam: mean number of people per family, PctNotSpeakEnglWell: percent of people who do not speak English well, LandArea: land area in square miles, LemasSwornFT: number of sworn full time police officers

regression methods. As a result of the real dataset, the final model selected using R-VIF regression had the highest R^2. This model also had the lowest RMSE value among those obtained with other methods excluding VIF regression. Consequently, it was decided that the R-VIF regression method performed best in contaminated and uncontaminated datasets.

Due to recent technological advances, the authors of this study suggest to use fast regression methods instead of conventional methods. The R-VIF regression method is particularly recommended as a fast regression estimator in the datasets containing multicollinearity and outliers.

References

Cai, L., Huang, T., Su, J., Zhang, X., Chen, W., Zhang, F., He, L., & **Chou, K.C. (2018)**. Implications of newly identified brain eQTL genes and their interactors in schizophrenia. Molecular Therapy-Nucleic Acids, 12, 433-442.

Candes, E., & **Tao, T. (2007)**. The Dantzig selector: statistical estimation when p is much larger than n. The Annals of Statistics, 35(6), 2313-2351.

Dupuis, D.J., & **Victoria-Feser, M.P. (2013)**. Robust VIF regression with application to variable selection in large data sets. The Annals of Statistics, 7(1), 319-341.

Efron, B., Hastie, T., Johnstone, I., & **Tibshirani, R. (2004)**. Least angle regression. The Annals of Statistics, 32(2), 407-499.

Fan, J., & **Lv, J. (2008)**. Sure independence screening for ultra-high-dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(5), 849-911.

Friedman, J.H. (2008). Fast sparse regression and classification. Technical report. Stanford University, California. Available from: https://jerryfriedman.su.domains/ftp/GPSpaper.pdf.

Foster, D.P., & **Stine, R.A. (2008)**. α-investing: a procedure for sequential control of expected false discoveries. Journal of the Royal Statistical Society Series B, 70(2), 429-444.

Khan, J.A., Van Aelst, S., & **Zamar, R.H. (2007)**. Robust linear model selection based on least angle regression. Journal of the American Statistical Association, 102(480), 1289-1299.

Lin, D., Foster, D.P., & **Ungar, L.H. (2011)**. VIF regression: a fast regression algorithm for large data. Journal of the American Statistical Association, 106(493), 232-247.

Liu, C., Zhang, Y.-H., Deng, Q., Li, Y., Huang, T., Zhou, S., & **Cai, Y.-D. (2017)**. Cancer-related triplets of mRNA-lncRNA-miRNA revealed by integrative network in uterine corpus endometrial carcinoma. BioMed Research International, 2017, Article ID 3859582, 7 pages. http://dx.doi.org/10.1155/2017/3859582.

Midi, H., & **Uraibi, H.S. (2014)**. The dimensional reduction of correlation matrix for linear regression model selection. In *Mathematical and Computational Methods in Science and Engineering. Proceedings of the 16th International Conference on Mathematical and Computational Methods in Science and Engineering* (pp. 166-169). MACMESE '**14**. Kuala Lumpur, Malaysia: WSEAS Press.

Rahman, M.S., & **Khan, J.A. (2010)**. Robust stepwise algorithms for linear regression: a comparative study. Dhaka University Journal of Science, 58(2), 291-295.

Redmond, M. (2009). Communities and crime data set. UCI Machine Learning Repository. Available from: https://archive.ics.uci.edu/ml/datasets/communities+and+crime. Accessed: December, 2021.

Seo, H.S. (2018). Fast robust variable selection using VIF regression in large datasets. The Korean Journal of Applied Statistics, 31(4), 463-473.

Shahriari, S. (2014). Variable selection in linear regression models with large number of predictors. Ph.D. thesis, Universidade do Minho Escola de Ciências, Guimarães, Portugal.

Shahriari S., Faria S., Gonçalves A.M., & **Van Aelst S. (2014)**. Outlier detection and robust variable selection for least angle regression.In *Computational Science and Its Applications–ICCSA 2014. Lecture Notes in Computer Science*, vol 8581. Springer, **Cham**. pp.512-522.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B, 58(1), 267-288.

Uraibi, H.S. (2020). VIF-regression screening ultrahigh dimensional feature space. Journal of Modern Applied Statistical Methods, 19(1), eP2916.

Zhang, T. (2009). Adaptive forward–backward greedy algorithm for sparse learning with linear models. Advances in Neural Information Processing Systems, 21, 1921-1928.

Zhou, J., Foster, D.P., Stine, R.A., & **Ungar, L.H. (2006)**. Streamwise feature selection. Journal of Machine Learning Research, 7, 1861-1885.

Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 101(476), 1418-1429.

Zou, H., & **Hastie, T. (2005)**. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B, 67(2), 301-320.

Submitted: 10/09/2021
Revised: 15/03/2022
Accepted: 22/03/2022
DOI: 10.48129/kjs.16159

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-12)

A hybrid approach based on k-nearest neighbors and decision tree for software fault prediction

Manpreet, Jitender Kumar Chhabra
Dept. of Computer Engineering
National Institute of Technology, Kurukshetra-136119 INDIA
**Corresponding author: manibhangu92@gmail.com*

Abstract

Software testing is a very important part of the software development life cycle to develop reliable and bug-free software but it consumes a lot of resources like development time, cost, and effort. Researchers have developed many techniques to get prior knowledge of fault-prone modules so that testing time and cost can be reduced. In this research article, a hybrid approach of distance-based pruned classification and regression tree (CART) and k- nearest neighbors is proposed to improve the performance of software fault prediction. The proposed technique is tested on eleven medium to large scale software fault prediction datasets and performance is compared with decision tree classifier, SVM and its three variations, random forest, KNN, and classification and regression t ree. Four performance metrics are used for comparison purposes that are accuracy, precision, recall, and f1-score. Results show that our proposed approach gives better performance for accuracy, precision, and f1-score performance metrics. The second experiment shows a significant amount of running time improvement over the standard k-nearest neighbor algorithm.

Keywords: Decision Tree; k- nearest neighbors; machine learning; pruning; software fault prediction.

1. Introduction

The 21st century has seen an unprecedented growth of automation and software with more emphasis on security, interactive graphical user interface, and faster development with more features (Singh *et al.*, 2016). But all this necessitates reliable and bug-free software which can be achieved by effective software testing and maintenance.

Software testing is an essential part of the software development life cycle and is used to identify fault-prone and complex modules so that faults can be removed from fault-prone modules and refactoring of complex modules can be done during the software development process. But software testing and maintenance phase consume almost fifty percent of software development resources such as time, effort, and cost (Aziz *et al.*, 2019). It is necessary to reduce the testing time and cost to develop reliable software within a limited budget and resources. If a somehow testing team manages to get prior knowledge about fault-prone and complex modules that need more attention, then the team can directly focus on those modules and a significant amount of testing time and effort can be reduced. Hence early identification of the faulty modules has caught the attention of researchers.

Many approaches have been developed in the recent past to detect and predict fault-prone modules. Machine learning is one of these approaches which have gained popularity in the past few years in this area. Decision trees (C4.5 and CART) (Quinlan *et al.*, 1986), support vector machine and its variations (Noble *et al.*, 2006), multi-layer perception (Gardner *et al.*, 1998), k-nearest neighbors (Kozma *et al.*, 2008), and random forest (Biau *et al.*, 2016) are some standard machine learning approaches that are the most commonly used for software fault prediction(Beygelzimer *et al.*, 2008). But standard machine

learning approaches usually give an average performance in most cases (Cheng *et al.*, 2014). Researchers are developing new and hybrid approaches by combining existing approaches to get better performance.

In this research article, we propose a hybrid approach by combining distance-based pre-pruned classification and regression tree with weighted k-nearest neighbors. The next section explains previous works and research gaps. The third section explains the working of our proposed approach. The fourth section discusses the experimentation and performance evaluation of our proposed approach.

2. Related works

Studies related to software fault prediction area are summarized in this section. Saravanan et al. (Saravanan *et al.*, 2021) proposed an African buffalo optimizer based convolution neural network for fast training in the software fault prediction field. Kassaymeh et al. (Kassaymeh *et al.*, 2021) used a salp swarm optimizer for neural network training instead of backpropagation. Singh et al. (Singh *et al.*, 2021) proposed a new node splitting method for decision tree generation. Haouari et al. (Haouari *et al.*, 2020) presented an application of AIRS for inter-release software fault prediction. Yucalar et al. (Yucalar *et al.*, 2020) compared different ensemble learning approaches like voting, bagging, and boosting in the software fault prediction field. Alsghaier et al. (Alsghaier *et al.*, 2020) in 2020 applied genetic algorithm, PSO algorithm, and GA-PSO integrated algorithm to train support vector machine on twelve software fault prediction datasets and results show that GA-PSO integrated approach gives the best results. Abuassba et al. (Abuassba *et al.*, 2022) in 2022 developed a general plateform for ensembles in classification context. proposed framwork is applied on twelve datasets to prove the diversity and efficiency of ensemble learning approaches. Khan et al. (Khan *et al.*, 2016) in 2016 explained various machine learning approaches in their survey. Rajkumar et al. (Rajkumar *et al.*, 2015) in 2015 applied various machine learning approaches for thyroid problem diagnosis.

The decision tree was initially developed by Quinlan in 1986 (Quinlan *et al.*, 1986). The initial version of the decision tree is called ID3 and it can handle only categorical attributes. C4.5 is an extended version of ID3 that can handle continuous attributes also was developed by Ross Quinlan (Quinlan *et al.*, 1986). Both of these decision tree generation algorithms use information as a node splitting criterion. Ruggieri et al. (Ruggieri *et al.*, 2002) in 2002 developed an efficient C4.5 classifier based on the quicksort and counting sort algorithms to efficiently calculate information gain of continuous attributes. Safavian et al. (Safavian *et al.*, 1991) explained different types of decision tree classifiers and their building methods in detail in their survey. k-NN is a non-parametric classifier initially developed by Evelyn Fix and Joseph Hodges in 1951 (Fix *et al.*, 1989). In this classifier value of k is fixed and for the prediction of the class label of the testing sample, it checks the labels of k-nearest neighbors and assigns a label to the testing sample based on the majority labels of nearest neighbors.Zhang et al. (Zhang *et al.*, 2007) in 2007 developed a lazy learning approach called ML-KNN based on a standard KNN algorithm for a multi-label classification problem like test classification. Cheng et al. (Zhang *et al.*, 2014) in 2014 developed a new k nearest neighbors algorithm based on sparse learning with data-driven k values and neglecting the correlation of samples.

After studying previous literature, we develop a hybrid approach based on k-nearest neighbors and decision tree. The main contributions of the projected work are listed below:

1. A new decision tree pruning approach called distance-based pruning is proposed to prune decision tree nodes. Detailed steps of the distance-based decision tree pruning approach are explained in section 3.2.

2. Standard k nearest neighbor algorithm has $\mathcal{O}(n)$ running cost which is reduced to $\mathcal{O}(\log n) + c$ in our proposed approach. KNNs are added at leaf nodes of decision tree in training phase to reduce the running cost.

3. The CART decision tree is generated using distance based pruning approach and instead of storing class labels, k nearest neighbors are stored on leaf nodes of the decision tree.

4. The concept of weights is introduced based on the sigmoid function in the prediction phase to make

standard k nearest neighbors more effective. Weights are inversely proportional to the distance of nearest neighbors from the point under consideration.

3. Proposed approach

This section of the research article explains the notations used to formulate our proposed approach, working of proposed approach and running time cost of our proposed approach.

3.1 Notations

A variable $X \in \mathbb{R}^{n*m}$ represents training data. Where n represents the total number of training samples and m represents the total number of independent attributes (dimensions of the dataset). Symbol $K - Matrix$ is used to describe k-nearest neighbors matrix where each element of $K - Matrix$ is represented by symbol e_{ij}. Variable i means i^{th} row, and j represents the j^{th} column of the matrix. $Tolerance$ is a global parameter that contains a value between 0 and 1. P_i in m dimensional space represents each training sample.

3.2 Working of approach

This article proposes a hybrid approach based on distance-based pre-pruned classification and regression tree and weighted k-nearest neighbors. The proposed approach in this article considers all training samples as m dimensional points, where m is number of independent attributes in the dataset. A KNN-matrix $(K - Matrix)$ is generated in which element $e_{ij} = 0$ if j^{th} training sample is not the nearest neighbour of i^{th} training sample and element $e_{ij} = 1$ if j^{th} training sample is considered as the nearest neighbor of i^{th} training sample. After calculation of KNN-matrix, maximum distance $Max_{distance}$ among all points is calculated based on Euclidean distance formula and a constant parameter Tolerance is introduced to control the decision tree generation. At each decision tree node, the

Algorithm 1 Pseudo code of proposed approach (WK-Tree)

Input: Training Samples X, Training Classes Y
Output: Binary Classification Confusion Matrix
// X training samples
// Y labels attached to training samples

Training phase of proposed approach

Step 1: All training samples are considered as points in m dimensional space and the largest distance among all points is computed using Euclidean distance shown in equation (4).
Step 2: KNN's of all training samples are calculated using equation (3) and the matrix is created as shown in section 3.
Step 3: The CART decision tree is created with distance-based pre-pruning.
// pruning strategy is explained in detail in section 4.2.
Step 4: KNN's of all samples are stored in leaf nodes instead of storing class labels.
// KNN's are stored without repetition
// all duplicate KNN's are removed

Testing phase of proposed approach

Step 1: The first step is to reach the leaf node of the decision tree; the testing sample under consideration belongs.
Step 2: Label of the point under consideration is assigned based on weighted labels of nearest neighbors of the leaf node.
//smaller distance has more weight than the larger distance
Step 3: Confusion matrix is created based on predicted values by WK-Tree and actual values
Step 4: Accuracy, Recall (sensitivity), Precision, and F1-Score is calculated based on Confusion Matrix

$Max_{distance} * Tolerance$ condition is checked, if the node's training samples satisfy this condition then that node is considered as a leaf node and instead of storing class labels, nearest neighbors based on

KNN-matrix are stored at that leaf nodes. Detailed steps and pseudo code of the proposed approach are given in algorithm 1.

Figure 1 represents the flow chart of our proposed approach.

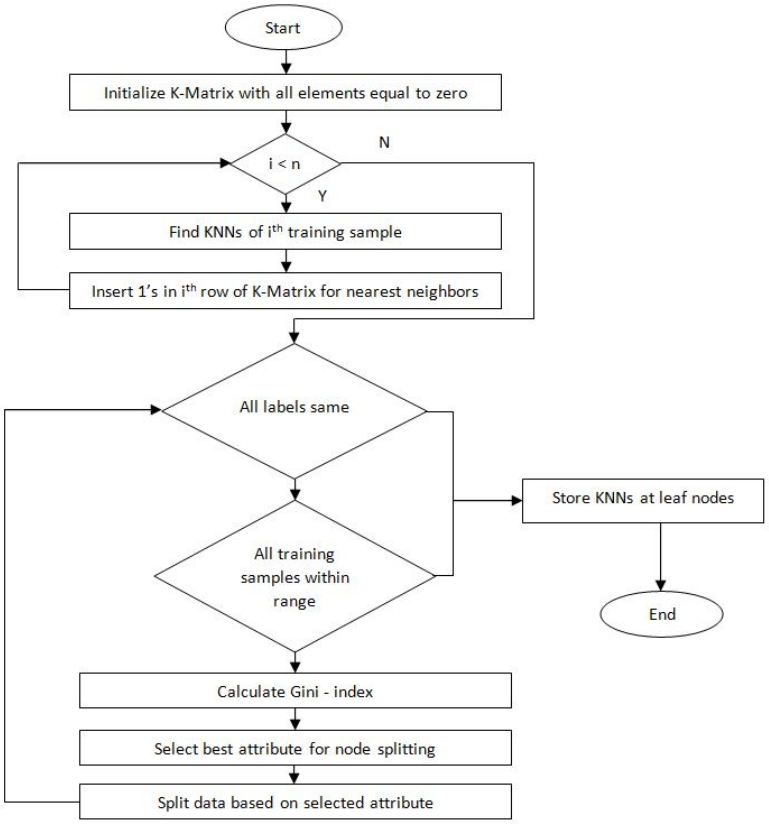


Fig. 1. Flow chart of the proposed approach

3.3 WK-Tree generation

This section of the research article explains the WK-Tree generation process in detail. Steps to develop WK-Tree are explained as below:

1. **Calculation of largest distance:** In the first step of proposed the approach, all training samples are considered as points in m dimensional space and the largest distance among all points is computed as shown in Figure 2.

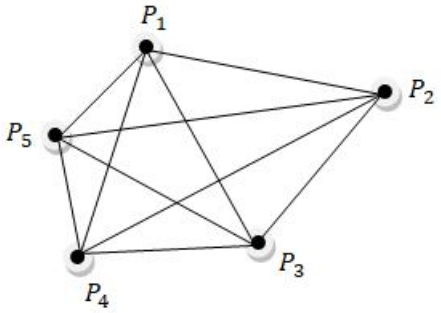


Fig. 2. Grid of training samples in m dimensional space

In figure 2 distance between point P_2 and point, P_5 is the largest among all points. It can be considered for decision tree pruning. Distance computed among all training samples in m dimensional

space is Euclidean distance and can be calculated based on equation (1)(Danielsson, 1980).

$$d(P_i, P_j) = \max_{1 \leq i,j \leq n} \sqrt{\sum_{k=1}^{m} \left(P_i^{(k)} - P_j^{(k)} \right)^2} \tag{1}$$

Where m is the total number of dimensions/features and n is the total number of training samples. k is variable to iterate over the number of dimensions and i & j are variable to iterate over the number of training samples.

2. **KNN matrix generation:** A $n * n$ matrix of k-nearest neighbors is generated for all training samples. Here n is the total number of training samples. $sqrt(n) + c$ function is selected to find the nearest neighbors and an example of $5 * 5$ KNN matrix is shown in equation (2) with all diagonal elements equal to 1. If element e_{ij} of K- matrix is 1 then point j is considered as the nearest neighbor of point i on the other hand if element e_{ij} of KNN matrix is 0 then point j is not considered as the nearest neighbor of point i. The final matrix will be a binary square matrix with all diagonal elements as 1. All diagonal elements are 1 because the point under consideration is always considered as the nearest neighbor of it and stored in the leaf node of the decision tree.

$$K - matrix = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{2}$$

After performing pruning of decision tree node instead of storing class label, nearest neighbors of training samples of the pruned node are stored based on the KNN matrix.

3. **Decision tree creation:** A decision tree is created based on the Gini index node splitting method with distance-based pre pruning. There are two types of decision tree node pruning methods pre-pruning and post-pruning. Pre-pruning based on the maximum depth of each leaf node from the root of the tree is not an effective approach so we have done distance-based pre-pruning which is explained in section 3.4 in detail. In distance-based pruning, a constant parameter $Tolerance$ is initialized between 0 and 1, and $Max_{distance}$ is multiplied with Tolerance to prune decision tree nodes.

4. **Labelling of testing samples:** In the labeling phase first classification and regression tree is traversed to reach the leaf node and then the weighted k-nearest neighbor approach is applied to assign the final label of the testing sample. Weights are assigned to each nearest neighbor based on equation (3) from the testing sample.

$$w_i = \left(1.0 - \frac{1}{1 + e^{-d_{ij}}} \right) * L_i \tag{3}$$

Where w_i is the weight assigned to nearest neighbor i and value of $L_i = -1$ for non-fault prone classes and $L_i = +1$ for fault-prone classes.

3.4 Distance-based pruning

In distance-based pre-pruning of decision tree first, we find out the maximum distance among all points and then take the fraction of maximum distance to prune decision tree. Maximum distance is calculated to cover all points. Detailed steps of pre-pruning based on the fraction of maximum distance among all training samples are explained as under:

1. All training samples are considered as points in m dimensional space in this strategy. Here m is the number of independent attributes.

2. Parameter $Max_{distance}$ is calculated based on equation (4).

$$Max_{distance} = \max_{0 \leq i,j \leq k} (distance(x_i, y_j)) \tag{4}$$

   Where k is the total number of training samples in a particular node of the decision tree and $(distance(x)_i, y_j)$ is the distance between point x_i and point $(y)_j$.

3. Global parameter $Tolerance$ is adjusted between 1 and 0 manually based on the density of points. If the parameter value is adjusted to 1, it means the whole dataset is considered as nearest neighbors and the decision tree will be able to build only root node on the other hand if the parameter value is adjusted to 0 then the full decision tree will be built without any pruned node.

4. While building the decision tree, at each decision tree node $Max_{distance} * Tolerance$ is tested for all training samples at that node. If the condition is satisfied for all training samples then the decision tree node is pruned and marked as a leaf node.

3.5 Time complexity

Training time is a one-time investment, so we will discuss only the testing time complexity of our proposed approach. The Decision tree traverses from the root node to the leaf node in the prediction phase in $\mathcal{O}(\log n)$ time complexity. Instead of storing labels, k-nearest neighbors are stored at leaf nodes in our proposed approach so a little constant c is added to the actual testing time complexity of the decision tree. The total running time complexity of our proposed approach in this research article is $\mathcal{O}(\log n + c)$.

4. Results and analysis

This section of the research article explains about the model validation approach, performance measurement metrics, datasets used , and comparison of results of the proposed approach with other machine learning models.

4.1 Model validation

In this research article K-fold, cross-validation is used with the value of K set to ten. Dataset is divided into ten equal parts and then the training phase of the approach is applied on nine parts and tested on the remaining part. The process is repeated for each part of the K-fold dataset and the final results are the average of all ten-part results.

4.2 Performance measurement

Four performance metrics are used to evaluate the performance of proposed approach that are calculated based on equations (5), (6), (7), and (8) (Ferri *et al.*, 2009).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

4.3 Datasets used

In this research article, we have used 11 NASA MDP datasets. Datasets are freely available and downloaded from the PROMISE and OPENML repositories (Karim *et al.*, 2017),(Bischl *et al.*, 2017). Dataset names, number of attributes, number of instances, and fault percentage per dataset are given in Table 1.

Table 1. Datasets used for experimentation

Dataset Name	Language	Total attributes	Total instances	Fault percentage
CM1	C	22	498	9.83
KC1	C++	22	2109	15.45
KC2	C++	22	522	20.49
KC3	JAVA	40	458	9.38
MC1	C and C++	39	9466	0.71
MC2	C	40	161	32.29
MW1	C	38	403	7.69
PC1	C	22	1109	6.94
PC2	C	37	5589	0.41
PC3	C	38	1563	10.23
PC4	C	38	1458	12.20

PC2 is the largest dataset with 5589 instances and MC2 is the smallest dataset with only 161 instances. But MC2 has the highest fault rate 32.29% on the other hand PC2 has the lowest fault rate with only 0.41% fault-prone modules.

4.4 Results

The proposed approach in this article is developed and tested on a machine with corei5 processor and 8GB RAM. Anaconda3 is used to develop this approach and compare it with other machine learning models.

4.4.1 Accuracy

In this research article proposed approach is applied to eleven open-source NASA MDP datasets taken from the OPENML repository (Bischl *et al.*, 2017). Table 2 compares our proposed approach with decision tree variations (C4.5 and CART), random forest, k-nearest neighbors, and three variations of support vector machine in terms of accuracy performance metric. The experiment is done by setting a tolerance parameter equal to 0.05. The best results for each dataset are shown in bold letters in Table 1. Accuracy is calculated up to five decimal places.

Out of all eleven datasets, our proposed approach gives better results in the case of nine datasets, including large-scale datasets like KC1, MC1, PC2, PC3, and P4, which contain more than 1500 modules. In the case of MC1, WA-SVM and GA-SVM give similar results as our proposed approach, which is more than 99%. In the case of MC2, the k-nearest neighbor gives slightly better performance, and in the case of PC1, GAWA-SVM gives marginally better results. In both cases, our proposed approach provides the second-best performance. The last row of Table 2 shows the average accuracy comparison of all datasets. Our proposed approach gives better results than all other approaches used for comparison purposes in the case of average accuracy.

Table 2. Accuracy Comparison of the proposed approach with other machine learning approaches, the value of constant $Tolrance = 0.05$

Dataset	C4.5	GA-SVM	CART	RF	KNN	WA-SVM	GAWA-SVM	WK-Tree
CM1/22	0.78417	0.90351	0.78768	0.87311	0.9	0.90175	0.90505	**0.93333**
KC1/22	0.76352	0.85143	0.75134	0.68693	0.75941	0.84527	0.84621	**0.86729**
KC2/22	0.76403	0.79894	0.74316	0.82322	0.75271	0.81611	0.80061	**0.83439**
KC3/40	0.87568	0.90352	0.87501	0.89979	0.90646	0.90391	0.90357	**0.91304**
MC2/40	0.64007	0.67095	0.66471	0.70735	**0.71434**	0.65808	0.67022	0.71428
MW1/38	0.88087	0.91829	0.87535	0.91285	0.92285	0.92317	0.91829	**0.95041**
PC1/22	0.87139	0.93669	0.86678	0.92566	0.93	0.93403	**0.93676**	0.93093
PC2/37	0.99177	0.99588	0.99213	0.99606	0.99588	0.99588	0.99588	**0.99761**
PC3/38	0.8714	0.90144	0.86309	0.89447	0.89767	0.90018	0.90272	**0.91257**
PC4/38	0.88103	0.88206	0.87517	0.87648	0.87789	0.88337	0.87790	**0.88127**
MC1/39	0.99461	**0.99503**	0.9945	0.99408	0.71434	**0.99503**	0.99492	**0.99503**
Average	0.84714	0.88706	0.84444	0.87181	0.85195	0.88697	0.88655	**0.90274**

Figure 3 shows the comparison of the accuracy of our proposed approach with other machine learning approaches. The accuracy distribution of our proposed approach is shown by pink colored box plot, which clearly shows better performance of the proposed approach in this article than other machine learning approaches.

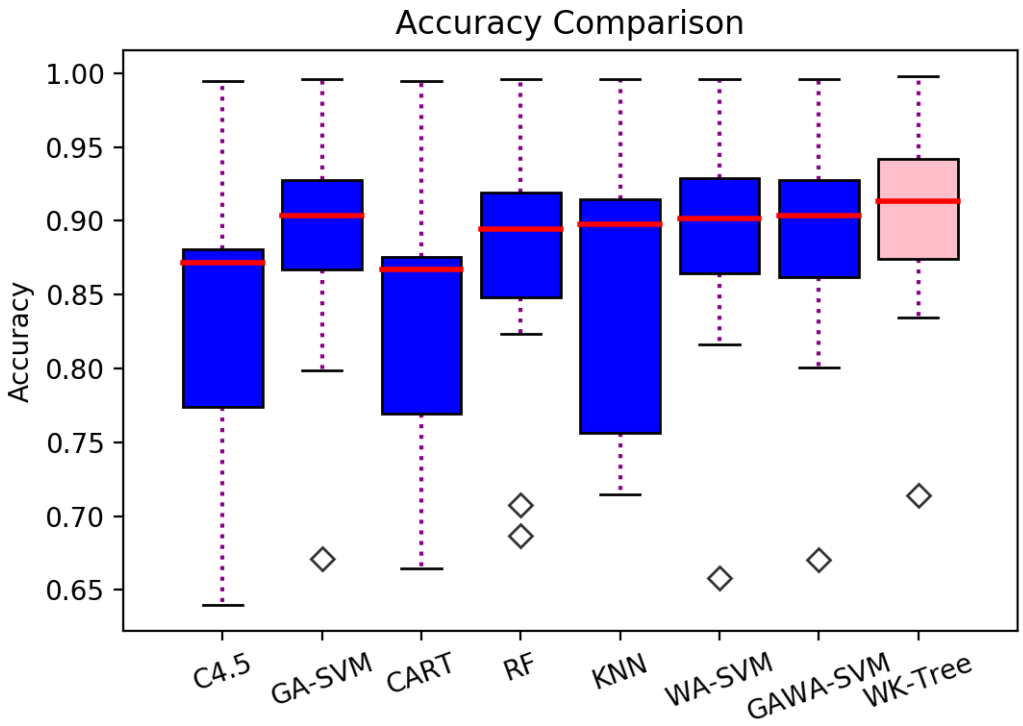


Fig. 3. Average accuracy comparison of all datasets in terms of box plots

4.4.2 Other performance metrics

Table 3 shows the average precision, recall, and f1-score of all techniques used for comparison on eleven datasets. The best performance value is shown in bold letters. For precision and f1-score performance metrics, proposed approach in this article gives better performance than all other approaches used for the comparison but in the case of recall GA-SVM and GAWA-SVM perform slightly better than our proposed approach. GAWA-SVM gives the best performance as shown in bold letters in Table 3.

Table 3. Average precision, recall, and f1-score of all techniques used for comparison

Technique	Precision	Recall	F1-Score
C4.5	0.90790	0.90950	0.89874
GA-SVM	0.89014	0.99470	0.93736
CART	0.90778	0.91206	0.89920
RF	0.88811	0.95419	0.91559
KNN	0.86958	0.96455	0.90392
WA-SVM	0.89398	0.98684	0.93670
GAWA-SVM	0.89022	**0.99223**	0.93665
WK-Tree	**0.91053**	0.98521	**0.94569**

Figure 4 shows the comparison of other metrics like precision, recall, and f1-score for all eight techniques used for comparison purposes. In this parallel bar graph, red-colored bar graphs show the precision value of different techniques, blue-colored bar graphs show recall value and green-colored bar graphs shows the f1-score comparison of all techniques used for comparison purpose.

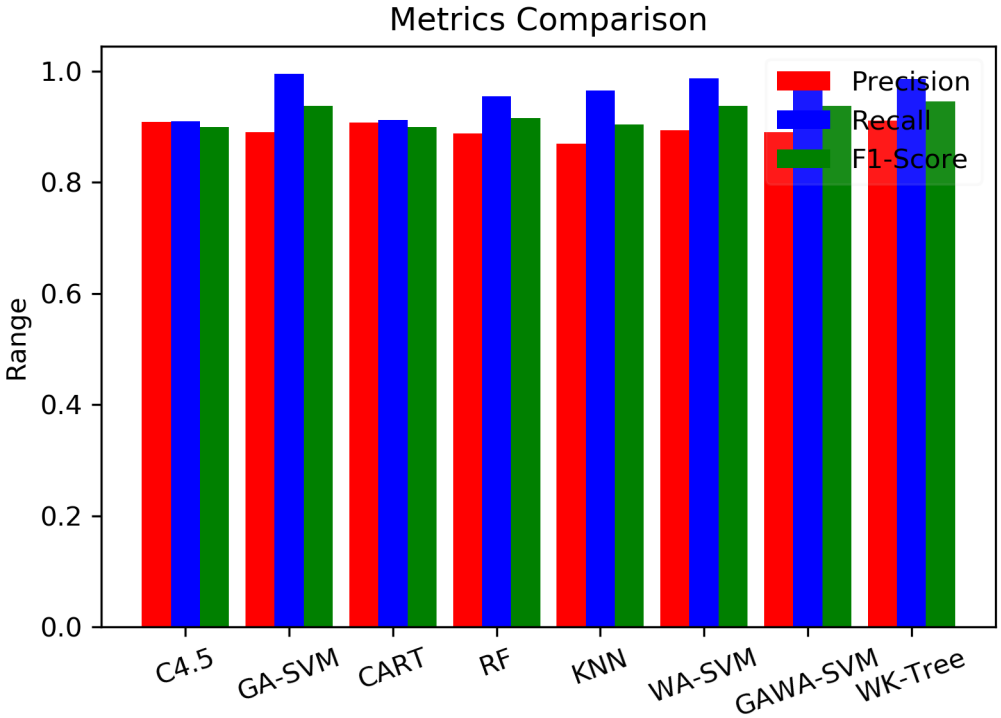


Fig. 4. Average precision, recall, and f1-score comparison of all datasets

4.4.3 Running time

The running time complexity of our proposed approach $\mathcal{O}(\log n)+c$. Figure 5 shows the comparison of running time in seconds of our proposed approach with k-nearest neighbors and weighted k-nearest neighbors.

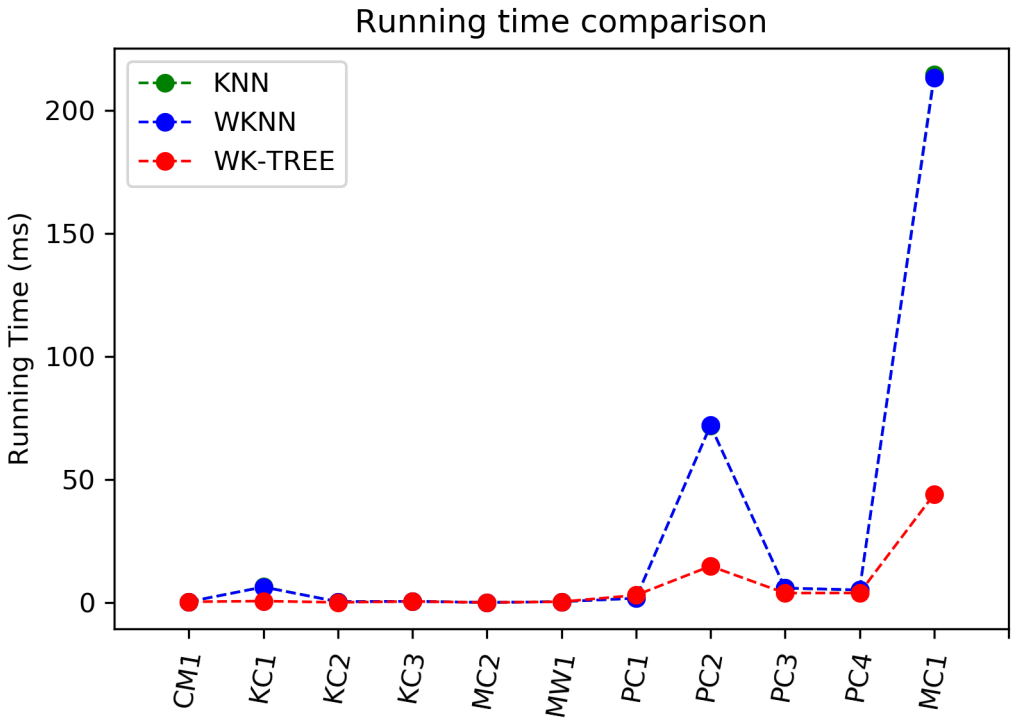


Fig. 5. Running time comparison of the proposed approach with KNN and WKNN

5. Conclusion

This article proposes a hybrid approach of pre-pruned classification and regression tree (CART) and k-nearest neighbors. The decision tree is pruned based on the distance among points in m dimensional space and leaf nodes of the decision tree store nearest neighbors of training samples on leaf nodes instead of storing class labels. The proposed approach is applied on eleven software fault prediction datasets and results are compared with eight machine learning models. Results show significant improvement in performance.

In future work, more hybrid approaches based on standard machine learning approaches can be developed to improve performance and make them work for real-life projects. Work can be done on running time complexity reduction to make the approach practical.

References

Abuassba, A. O., Dezheng, Z., Ali, H., Zhang, F., & Ali, K. (2022). Classification with ensembles and case study on functional magnetic resonance imaging. Digital Communications and Networks, 8(1), 80-86.

Alsghaier, H., & Akour, M. (2020). Software fault prediction using particle swarm algorithm with genetic algorithm and support vector machine classifier. Software: Practice and Experience, 50(4), 407-427.

Aziz, S. R., Khan, T., & Nadeem, A. (2019). Experimental validation of inheritance metrics' impact on software fault prediction. IEEE Access, 7, 85262-85275.

Beygelzimer, A., Langford, J., & Zadrozny, B. (2008). Machine learning techniques—reductions between prediction quality metrics. In Performance Modeling and Engineering (pp. 3-28). Springer, Boston, MA.

Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

Bischl, B., Casalicchio, G., Feurer, M., Hutter, F., Lang, M., Mantovani, R. G., ... & Vanschoren, J. (2017). Openml benchmarking suites. arXiv preprint arXiv:1708.03731.

Cheng, D., Zhang, S., Deng, Z., Zhu, Y., & Zong, M. (2014, December). kNN algorithm with data-driven k value. In International Conference on Advanced Data Mining and Applications (pp. 499-512). Springer, Cham.

Danielsson, P. E. (1980). Euclidean distance mapping. Computer Graphics and image processing, 14(3), 227-248.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. Pattern recognition letters, 30(1), 27-38.

Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique, 57(3), 238-247.

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

Haouari, A. T., Souici-Meslati, L., Atil, F., & Meslati, D. (2020). Empirical comparison and evaluation of Artificial Immune Systems in inter-release software fault prediction. Applied Soft Computing, 96, 106686.

Karim, S., Warnars, H. L. H. S., Gaol, F. L., Abdurachman, E., & Soewito, B. (2017, November). Software metrics for fault prediction using machine learning approaches: A literature review with PROMISE repository dataset. In 2017 IEEE international conference on cybernetics and computational intelligence (CyberneticsCom) (pp. 19-23). IEEE.

Kassaymeh, S., Abdullah, S., Al-Betar, M. A., & Alweshah, M. (2021). Salp swarm optimizer for modeling the software fault prediction problem. Journal of King Saud University-Computer and Information Sciences.

Khan, W., Daud, A., Nasir, J.A. and Amjad, T., 2016. A survey on the state-of-the-art machine learning models in the context of NLP. Kuwait journal of Science, 43(4).

Kozma, L. (2008). k Nearest Neighbors algorithm (kNN). Helsinki University of Technology, 32.

Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

Rajkumar, N. and Palanichamy, J., 2015. Optimized construction of various classification models for the diagnosis of thyroid problems in human beings. Kuwait Journal of Science, 42(2).

Ruggieri, S. (2002). Efficient C4. 5 [classification algorithm]. IEEE transactions on knowledge and data engineering, 14(2), 438-444.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

Saravanan, P., & Sangeetha, V. (2021). African buffalo optimized multinomial softmax regression based convolutional deep neural network for software fault prediction. Materials Today: Proceedings.

Singh, M., & Chhabra, J. K. (2021). EGIA: A new node splitting method for decision tree generation: Special application in software fault prediction. Materials Today: Proceedings.

Singh, P., Pal, N. R., Verma, S., & Vyas, O. P. (2016). Fuzzy rule-based approach for software fault prediction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(5), 826-837.

Yucalar, F., Ozcift, A., Borandag, E., & Kilinc, D. (2020). Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability. Engineering Science and Technology, an International Journal, 23(4), 938-950.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40(7), 2038-2048.

Zhang, S., Zong, M., Sun, K., Liu, Y., & Cheng, D. (2014, December). Efficient kNN algorithm based on graph sparse reconstruction. In International Conference on Advanced Data Mining and Applications (pp. 356-369). Springer, Cham.

Submitted: 19/01/2022
Revised: 17/03/2022
Accepted: 25/04/2022
DOI: 10.48129/kjs.18331

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-15)

An efficient gravitational search decision forest approach for fingerprint recognition

Mahesh Kumar*, Devender Kumar

Dept. of Computer Science and Engineering,
Baba Mastnath University, Asthal Bohar, Sector-29, Rohtak, India
**Corresponding author: malkanimahesh@gmail.com*

Abstract

Fingerprint based human identification is one of the authentic biometric recognition systems due to the permanence and uniqueness of the finger impressions. There is the extensive usage of fingerprint recognition in personalized electronic devices, security systems, banking, forensic labs, and especially in law enforcement agencies. Although the existing systems can recognize fingerprints, they lack in case of poor quality and latent fingerprints. The latent fingerprints are captured by law enforcement agencies during the crime scene to find the criminal. Consequently, it is essential to develop a novel system that can efficiently recognize both complete and latent fingerprints. The current work proposes an efficient Gravitational Search Decision Forest (GSDF) method, which is a combination of the gravitational search algorithm (GSA) and the random forest (RF) method. In the proposed GSDF approach, the mass agent of GSA determines the solution by constructing decision trees in accordance with the random forest hypothesis. The recognition of the fingerprints is accomplished by mass agents in the form of a final generated decision forest from the set of hypothesis space as the mass agents can create multiple hypotheses using random proportional rules. The experiments for fingerprint recognition are conducted for both the latent fingerprints (NIST SD27 dataset) and the complete fingerprints (FVC2004 dataset). The effectiveness of the proposed GSDF approach is analyzed by evaluating the results with machine learning classifiers (random forest, decision tree, back propagation neural networks, and k-nearest neighbor) as well. The comparative analysis of the proposed approach and incorporated machine learning classifiers indicates the outperformed performance of the proposed approach.

Keywords: Back propagation neural networks; decision tree; fingerprint recognition; gravitational search algorithm; k-nearest neighbor; latent fingerprints; machine learning; random forest

1. Introduction

There are numerous biometric systems for human identification, including iris recognition, face recognition, fingerprint recognition, etc. (Nadeem *et al.*, 2022). Among these methods, fingerprint recognition is the most widely adapted method in practice. The concept of fingerprint recognition can be represented in two aspects: verification and identification (Maltoni *et al.*, 2009). The verification aspect is the 1:1 comparison of a human's fingerprints with previously stored data. The identification aspect is the 1:N comparison to determine the identity of a human by comparing the unknown fingerprints with the available overall fingerprint databases. The verification aspect is used for complete fingerprint based biometric systems, and the identification aspect is used by law enforcement agencies to identify the suspect on the basis of acquired latent or complete fingerprints. The current work focuses on both aspects by experimenting with both complete and latent fingerprints. An illustration of latent and complete fingerprints is depicted in Figure 1. Latent fingerprints are poor quality distorted finger impressions, whereas complete fingerprints can be plain or rolled impressions. Plain finger impressions are made by pressing

Fig. 1. Types of Fingerprints.

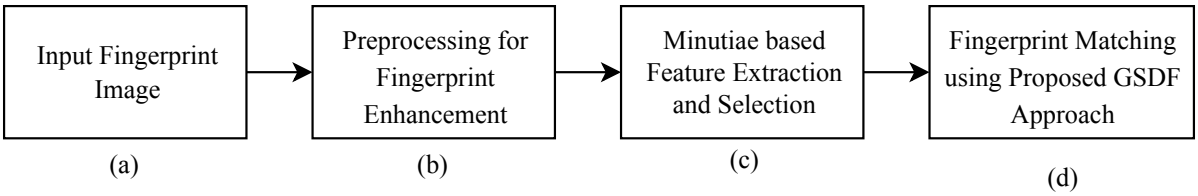


Fig. 2. Process of Fingerprint Recognition (a) Fingerprint Consideration, (b) Preprocessing, (c) Feature Extraction & Selection, and (d) Fingerprint Matching.

the finger on a surface, whereas rolled finger impressions are made by rolling the finger from one side of the fingernail to the other.

The process of fingerprint recognition is discussed by different modules: fingerprint consideration; preprocessing; feature extraction & selection; and fingerprint matching. A brief overview of the fingerprint recognition process is described in Figure 2. For the incorporated fingerprints, the preprocessing module enhances the poor quality and latent fingerprints by using the ridge dictionary and Gabor filter. Further, the minutiae-based features are extracted by using the crossing number concept. In the feature selection phase, the spurious minutiae are removed prior to beginning the fingerprint matching. The final module of fingerprint matching is performed using the proposed GSDF approach, which recognizes the fingerprints by constructing the decision forest with the help of mass agents. The amalgamation of the machine learning based RF algorithm with the GSA algorithm is owing to the stability of the GSA method, which is theoretically modeled using Newton's laws. In addition, the GSA's effective use in several fields of bioinformatics, digital image processing, robotics, and optimization (Kumar *et al.*, 2020) has prompted its use in the present work of fingerprint recognition. The main contributions of the work are summarized as follows.

- The proposal of an efficient GSDF approach by amalgamating GSA and RF algorithms for fingerprint recognition.
- The autonomous enhancement of latent and poor quality fingerprints using a combination of ridge dictionary and Gabor filter.
- The incorporation of a feature selection module to remove spurious minutiae extracted during the minutiae extraction phase. The removal of spurious minutiae enhances recognition accuracy.
- The testing of the proposed approach for both the latent fingerprints (NIST SD27 dataset) and the complete fingerprints (FVC2004 dataset).

The organization of the remaining portions of the paper is as follows: Section 2 presents the state-of-the-art work related to fingerprint recognition. Section 3 describes the preprocessing of the input fingerprint images. Section 4 depicts the minutiae based feature extraction and selection module for fingerprint matching. Section 5 explains the proposed GSDF approach, which is utilized for fingerprint

recognition. Section 6 evaluates the results for the experiments on the FVC2004 and NIST SD27 datasets. Also, the comparative analysis of the proposed approach with incorporated machine learning classifiers is conducted in Section 6. Finally, Section 7 illustrates the conclusion of the work along with future directions.

2. Related work

Fingerprint recognition systems should be autonomous and reliable. Inaccurate information might lead to mishaps, especially in the case of latent fingerprints acquired from the crime scene. In 2004, the FBI erroneously detained a man from Oregon in connection with the explosion investigation. This raised problems regarding the fundamentals of forensic science and technology (Newman, 2007). This incident has increased the focus of researchers on the development of efficient fingerprint recognition technologies. Here, the recent studies in the field of fingerprint recognition are discussed.

Guo *et al.* (2014) adapted the decision tree rule-based approach for fingerprint classification. The authors also incorporated the methods of balance arm flow and center to data flow for the recognition of indistinguishable fingerprints. Hsieh & Hu (2014) hybridized the support vector machine (SVM) with particle swarm optimization (PSO) for the classification of fingerprints. The hybridized approach was served with multi-objective optimization to handle the penalty errors of the SVM algorithm. Babatunde (2015) proposed minutiae-based matching for the fingerprints from the different data sources. The spatial and Euclidean relations among the minutiae were evaluated, and pattern matching was conducted from the singular core points. Murugan & Rose (2017) used the back propagation neural network for the recognition of plain and rolled fingerprint images. Lee *et al.* (2017) worked on the recognition of partial fingerprints using ridge shape features (RSF) and minutiae information. The authors designed this algorithm to improve the recognition of fingerprints on small scanning devices such as smart phones.

Cao & Jain (2018) focused on latent fingerprint matching using the convolutional neural network. The feature attributes of minutiae information and texture templates were adapted for the fingerprint feature representation. Castillo-Rosado & Hernández-Palancar (2019) used the distinctive ridge point method for latent fingerprint matching. Wong & Lai (2020) adapted the orientation field information along with the multi-tasking convolutional neural network for the restoration of corrupted fingerprints. Kumar & Garg (2020) introduced the hybrid approach of particle swarm optimization and cuckoo search for latent fingerprint recognition. Jindal & Singla (2021) used an ant colony optimization algorithm for matching the minutiae of latent fingerprints with original fingerprints. Deshpande *et al.* (2021) presented a ratio to minutiae triangles based method which is a rotation and scale invariant approach. The presented method was used for the identification of latent fingerprints. Pradeep & Ravi (2022) incorporated the artificial neural network (ANN) for fingerprint classification after extracting the features using Gabor filter. Singla *et al.* (2022) hybridized the features of pores and minutiae points for the identification of latent fingerprints. Existing studies indicate the usability of different techniques for latent and complete fingerprint recognition systems. This work addresses the following research gaps in some existing studies.

- The focus of the researchers is observed either on the complete fingerprints with some noise value or latent fingerprints with good quality images. There is a need to develop a system that can handle the complete as well as latent fingerprints of low quality images.
- The manual analysis of complex latent fingerprint structures is also challenging for matching with complete fingerprints. The present work autonomously performs the module.
- The recognition accuracy of the existing fingerprint recognition systems should be improved, especially the latent fingerprint recognition, as false values can lead to punishment for any benign person.
- During fingerprint extraction, there may be false minutiae extracted along with the actual minutiae. The removal of spurious minutiae information should also be incorporated as the post processing step to reduce the false positive and false negative rates. The present work also addresses this concern as the feature selection module.

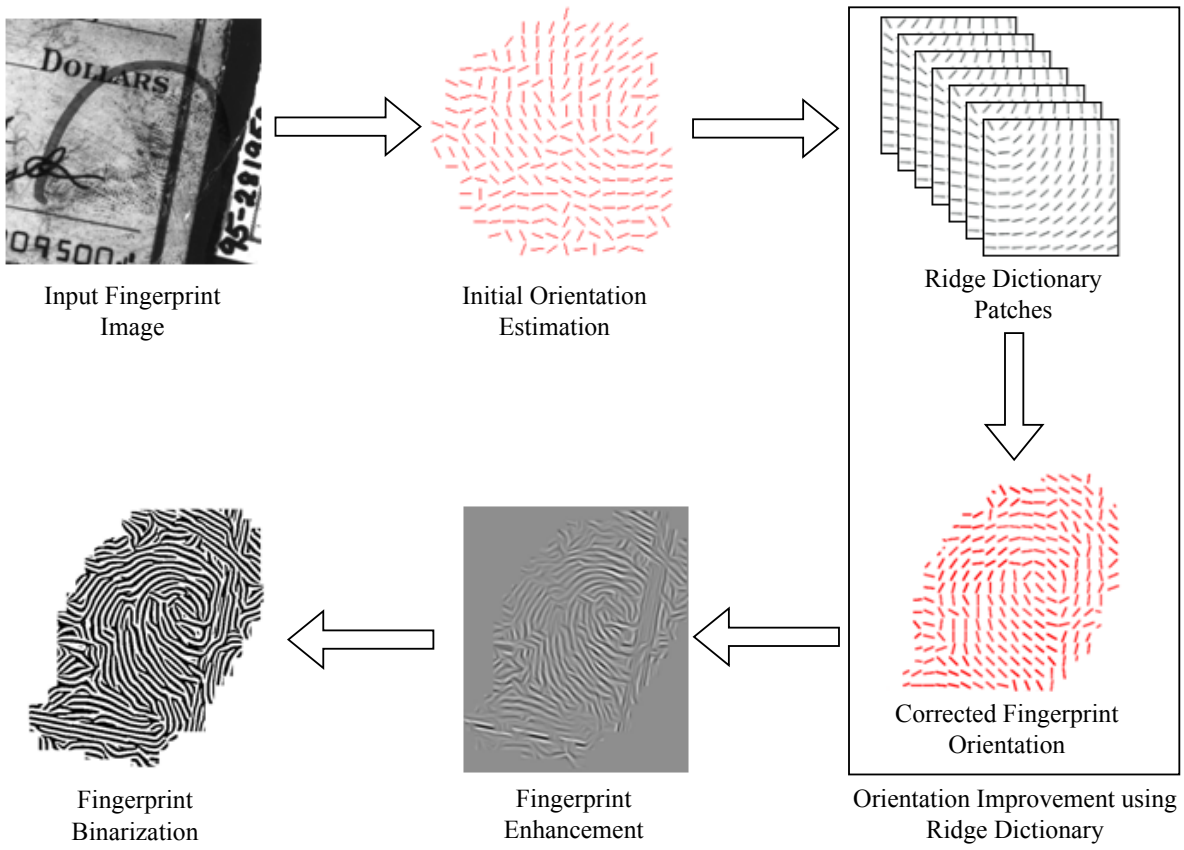


Fig. 3. Preprocessing of Fingerprints.

3. Fingerprint preprocessing

Fingerprint preprocessing is the essential module of the fingerprint recognition process as poor quality fingerprints cannot extract the minutiae features efficiently. Fingerprint preprocessing is composed of four essential steps: fingerprint orientation estimation; orientation improvement using a ridge dictionary; enhancement; and binarization. These steps are also depicted in Figure 3 by considering a latent fingerprint image from the NIST SD27 dataset.

3.1 Fingerprint orientation estimation

Initially, the fingerprint images are segmented and normalized to estimate the orientation of the fingerprints. Segmentation separates the foreground region from the background while preserving the fingerprint ridges and other features. The region of interest (finger impression) from the image is extracted using the variance method. For this process, the image $I(i,j)$ is splitted into 16×16 blocks and the variance $V(I)$ is evaluated for each block. The blocks with a variance value greater than the threshold are retained because the background regions possess a lower threshold value. The obtained finger impression is normalized to reduce the variations in the grey-level of fingerprints while retaining the valley and ridge information unaffected. The normalization $N(i,j)$ of the image at pixel-level is conducted by considering the desired mean and variance values of M_0 and V_0, respectively. The normalized image is processed for orientation estimation using the gradient vectors, which determine local orientation towards the ridge direction flow (Jindal & Singla, 2021).

The orientation image illustrates the invariant coordinates of the fingerprints and analyzes the local ridge information. For the gradient vector method, the normalized image $N(i,j)$ is divided into blocks of size 16×16. At each pixel (i,j) of each block, orientation $O(i,j)$ is estimated with least square estimation using Equation (1).

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{G_y(i,j)}{G_x(i,j)}\right) \tag{1}$$

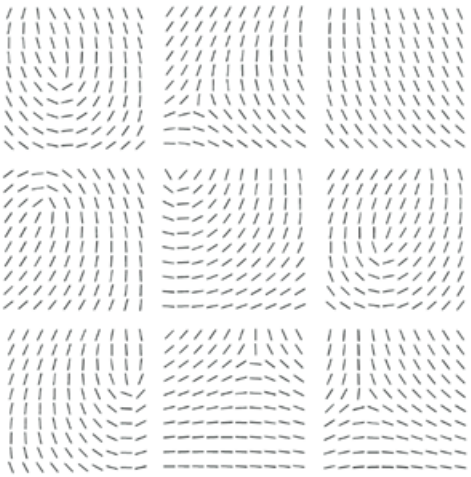


Fig. 4. Samples of Orientation Patches from the NIST SD4 dataset for Ridge Dictionary Construction.

Where, the gradient vector $G_x(i,j)$ and $G_y(i,j)$ are evaluated using the Sobel operator (Hong *et al.*, 1998) for the gradients $\partial_x(i,j)$ and $\partial_y(i,j)$ with respect to the x-axis and y-axis, respectively. The calculated orientation values are kept as matrices.

Due to the low quality of the input latent fingerprint, as seen in Figure 3, the estimated orientation field is noisy. Consequently, orienting is enhanced with the use of ridge dictionary. Further, the ridge dictionary is constructed and the orientation field is smoothed.

3.2 Orientation improvement using ridge dictionary

The ridge dictionary is constructed from the NIST SD4 dataset, which is composed of high-quality rolled fingerprints. The orientation patches, including ridge information, are retrieved from the fingerprints of this dataset with a block size of 16×16 pixels. Each orientation patch consists of 10×10 orientation elements. Figure 4 shows some of the high-quality orientation patches that were taken from the NIST SD4 dataset (Cao & Jain, 2015).

In the constructed ridge dictionary, only the unique patches with a quality index greater than the threshold are included, with no recurrence of ridge patterns. With the addition of the ridge dictionary, fingerprint orientation gets corrected to a great extent. Further, the fingerprint image with corrected ridge orientation is smoothed using a low-pass filter (Jain *et al.*, 2000) in which the image is initially converted to a continuous vector field as depicted by Equations (2)-(3).

$$\Phi_x(i,j) = \cos(2\theta(i,j)) \tag{2}$$

$$\Phi_y(i,j) = \sin(2\theta(i,j)) \tag{3}$$

Where, Φ_x and Φ_y are the vector field components with respect to the x and y axes respectively. As per the low-pass filter, the resulting vector field is determined in terms of Φ'_x and Φ'_y using Equations (4)-(5).

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_x(i-uw, j-vw) \tag{4}$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_y(i-uw, j-vw) \tag{5}$$

Where, $W(u,v)$ is the low-pass filter with a filter size of $w_\Phi \times w_\Phi$. Further, the final ridge orientation O' is estimated using Equation (6).

$$O'(i,j) = \frac{1}{2}\tan^{-1}\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)} \tag{6}$$

The corrected orientation image with the help of the ridge dictionary is illustrated in Figure 3.

3.3 Fingerprint enhancement

Enhancement is conducted to remove the undesired noise and preserve the corrected ridge and orientation information. The attributes of the Gabor filter, such as orientation-selective and frequency-selective, can efficiently remove the noise by preserving the ridge structure and orientation information. Moreover, it is efficient in both the frequency and spatial domains. This makes the Gabor filter a perfect fit for the enhancement process. The formulation of the Gabor filter (Hong *et al.*, 1998) in the spatial domain is described by Equations (7)-(9).

$$H(x, y; f, \phi) = \exp\left\{-\frac{1}{2}\left[\frac{x_\phi^2}{\delta_x^2} + \frac{y_\phi^2}{\delta_y^2}\right]\right\}\cos(2\pi f x_\phi) \tag{7}$$

$$x_\phi = x\cos\phi + y\sin\phi \tag{8}$$

$$y_\phi = -x\sin\phi + y\cos\phi \tag{9}$$

Where, f is the filter frequency, and ϕ is the orientation of the Gabor filter. δ_x and δ_y are the standard deviations with respect to axes x and y.

3.4 Binarization

The binarization process transforms the grey-level filtered image into a binary image. Here, the local adaptive binarization approach is adapted for transformation in which the mean intensity value is evaluated as a threshold value. The final image is obtained by assigning the value of 1 to the pixels whose values are higher than the threshold, and the assigning value 0 to the rest of the pixels.

4. Feature extraction and selection

For fingerprint recognition, minutiae-based features are extracted, which are specific to normal pixel, ridge bifurcation, and ridge endings. Here, the crossing number method is utilized to evaluate the minutiae-based features. It determines the feature types by analyzing the surrounding pixels of a pixel P within the 3×3 pixel window. The finding of crossing numbers 1, 2, 3, or greater than 3 reveals the minutiae features of ridge ending, a normal ridge pixel, and ridge bifurcation, respectively. Figure 5 depicts the assessment of features using the crossing number approach. These minutiae feature types are extracted for fingerprint matching.

Prior to considering the extracted minutiae features for fingerprint matching, these features are filtered to exclude any irrelevant minutiae. The feature selection procedure eliminates spurious minutiae such as dots, ladders, lakes, triangles, breaks, etc. Here, the ridge dictionary is utilized to identify the spurious minutiae. The feature selection is required since spurious minutiae might lead to erroneous fingerprint matching. The selected feature set is stored for fingerprint matching. A sample of minutiae features after the feature selection is illustrated in Figure 6.

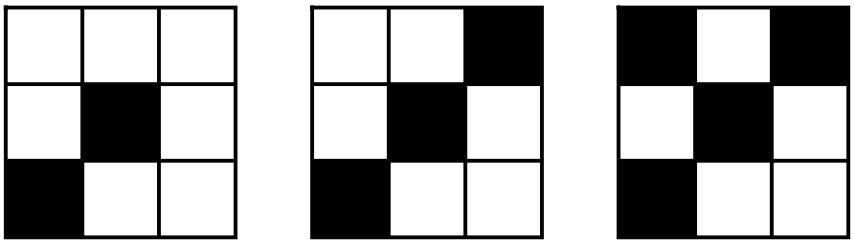


Fig. 5. Minutiae Feature Types using the Crossing Number Method.

5. Fingerprint matching using proposed GSDF approach

The fingerprint matching is conducted using the proposed GSDF approach, which is an amalgamation of GSA and RF algorithms. The GSA algorithm is a physics-inspired meta-heuristic algorithm that follows Newton's laws of gravity and motion for optimization (Jindal *et al.*, 2022). The RF algorithm is an ensemble of decision trees constructed with randomly selected characteristics (Manpreet & Chhabra, 2022). In the proposed GSDF approach, GSA's mass agents construct decision trees by following the hypothesis of a random forest algorithm in which random solutions are generated based on splitting rules and thresholds. The mass agents also determine the new random sub-space to handle the increasing decision trees and keep the tradeoff between exploration and exploitation. This amalgamation process conducts the fingerprint matching with better accuracy compared to the decision tree alone (Kozak, 2019). The process of GSDF approach initiated by considering N number of mass agents, with the initial position of i^{th} agent as X_i, the initial gravitational constant G_0, and a decision table with decision attributes d_a. The initial force acting (Rashedi *et al.*, 2009) on the agent i by the agent j in d-dimensions is determined by Equation (10). The mass agents describe the pixels of the fingerprint image and construct an overall decision forest to determine which fingerprint in the fingerprint database matches the input fingerprint.

$$F_{ij}^{d}(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{10}$$

Where, $G(t)$, M_{pi}, and M_{aj} are the gravitational constant, passive mass for agent i, and active mass for agent j respectively. The term ε is constant and R_{ij} is the Euclidean distance between mass agents. The addition of stochastic attributes to the GSDF upgrades the Force on mass agents as depicted by Equation (11).

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{11}$$

Where, $rand_j$ is a random number in the range [0, 1].

For the movement of the mass agents in nodes to construct the decision trees, the acceleration value is also evaluated by following Newton's law of motion. The formula to evaluate the acceleration $a_i^d(t)$ is depicted by Equation (12).

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{12}$$

Where, $M_{ii}(t)$ is the inertial mass of i^{th} agent.

As the inertial and gravitational masses are computed using the fitness function, which states that a higher mass value indicates a superior agent. For a better solution space with heavy masses, the inertial and gravitational masses are equalized. This updates the masses as described in Equations (13)-(14).

$$M_{ii} = M_{pi} = M_{ai} = M_i \tag{13}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{14}$$

Where, the value of $m_i(t)$ is evaluated (Equation (15)) by considering the best ($best(t)$) and worst ($worst(t)$) values for mass agents.

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{15}$$

Where, $fit_i(t)$ is the fitness function.

Each mass agent constructs the decision tree by adapting the random attributes of the RF algorithm. Further, the generated multiple decision trees are ensembled and a final decision is made for the fingerprint matching. There is a test on the attributes of each node of the decision tree as depicted by Equation (16).

$$test : O \rightarrow R_{test} \tag{16}$$

Where, the set of objects is defined by O and the possible tests are annotated with $R_{test} = \{r_1, r_2, \ldots, r_z\}$. Further, the applicability of the test for the attributes $a : O \rightarrow A$ is described by Equation (17).

$$test : A \rightarrow R_{test} \tag{17}$$

Here, the possible sub-trees $(T_1, T_2, \ldots, T_z)$ can be constructed by each node which tests $(r_1, r_2, \ldots, r_z)$ for the consideration of the assumption that T_i sub-trees are created by test r_i. This derives the hypothesis $h(x)$, as shown in Equation (18).

$$h(x) = \begin{cases} h_1(x), test(x) = r_1 \\ h_2(x), test(x) = r_2 \\ \vdots \\ h_z(x), test(x) = r_z \end{cases} \tag{18}$$

For the n number of nodes, the size of constructed decision tree is evaluated using Equation (19).

$$s(T) = \frac{1}{n} \tag{19}$$

In the GSDF approach, the heuristic function η_{A_i,V_j} for the attributes A_i and values V_j is calculated by following the Twoing splitting criteria to attain the best split of the tree. It also retains the maximum homogeneity of the nodes in the tree (Vives *et al.*, 2021). The formula for the evaluation of η_{A_i,V_j} is described by Equation (20).

$$\eta_{A_i,V_j} = \frac{P_l P_r}{4} \left[\sum_{d=1}^{D} \left| p\left(d|node_{l(A_i,V_j)}\right) - p\left(d|node_{r(A_i,V_j)}\right) \right| \right]^2 \tag{20}$$

Where, D is the maximum number of possible decision classes, P_l and P_r are the probabilities for the left and right nodes. $p\left(d|node_{l(A_i,V_j)}\right)$ and $p\left(d|node_{r(A_i,V_j)}\right)$ are the conditional probabilities for the left and right nodes, respectively.

The movement of the mass agents from one node to another makes it necessary to determine the updated position and velocity of agents. The changes in position and velocity values as per the GSA algorithm are determined by Equations (21)-(22).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{21}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{22}$$

The process of construction of the decision tree continues, and iterations are also increments. In the later iterations, the mass agents can be trapped in the local optimum due to the heaviness of the masses with the increasing iterations. This situation is handled by introducing the function of $Kbest$ which is a function of time. The $Kbest$ agents also possess the highest mass value, the best fitness, and it decreases linearly with time. At the end, there will be the applicability of force by one agent to others, and the change in force is described by Equation (23).

$$F_i^d(t) = \sum_{j \epsilon Kbest, j \neq i} rand_j F_{ij}^d(t) \tag{23}$$

To determine the final match for the fingerprints, the outcome of each decision tree is analyzed which will be further ensembled to determine the outcome of the decision forest by following the attributes of the RF algorithm. The fingerprint classification and recognition outcome by each decision tree $(T(S))$ with training sample (S) is determined by Equations (24)-(25).

$$\epsilon\left(T(S), Dst\right) = \sum_{(x,y)\epsilon U} Dst(x,y) . L\left(y, T(S)(x)\right) \tag{24}$$

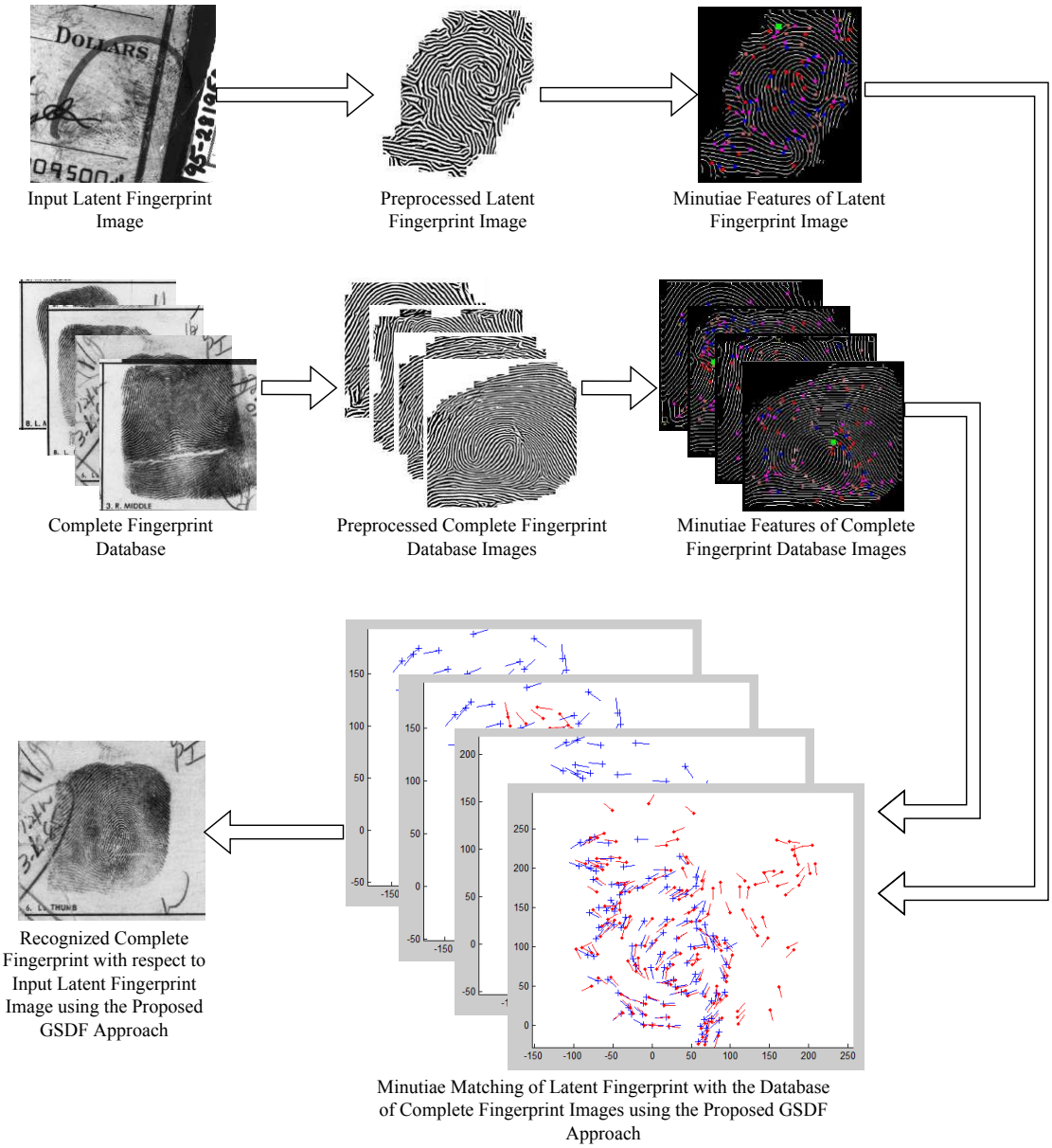


Fig. 6. Fingerprint Recognition using the Proposed GSDF Approach.

$$L\left(y, T\left(S\right)\left(x\right)\right) = \begin{cases} 1, & \text{if } y \neq T\left(S\right)\left(x\right) \\ 0, & \text{if } y = T\left(S\right)\left(x\right) \end{cases} \tag{25}$$

Where, the possible values of attributes are denoted by U and the distribution is denoted by Dst.

The overall results of the fingerprint classification are evaluated as a decision forest by combining the outcomes of the decision trees with the help of voting criteria. The availability of diversity in attributes of the GSDF approach makes the agents to choose different nodes for the construction of decision trees with different combinations, hence the ensemble decision forest. The final solution set is determined by the completion of maximum iterations and the evaluation of the solution by all the mass agents. The pseudo-code of the proposed GSDF approach for fingerprint recognition is described by Algorithm 1. The pictorial representation of the fingerprint recognition using the proposed GSDF approach is described in Figure 6.

Algorithm 1: Pseudo-Code of the Proposed GSDF Approach for Fingerprint Recognition

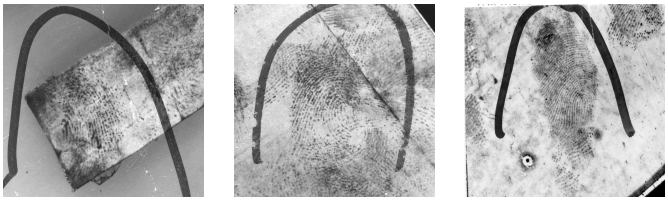
```
Initialize the parameters of the GSA and RF algorithms.
decision_forest=null;
iteration=1;
while iteration ≤ iteration_max do
    for (j = 1 to number_of_decision_trees) do
        best_decision_tree=null;
        fingerprint_classifier=choose_objects // Consider mass agents for the pixels of the
          fingerprint image data with equal probability.
        for (N = 1 to number_of_mass_agents) do
            Construct decision trees by considering subset of attributes at each node using
              attributes of GSDF approach.
            new_decision_tree=decision_tree_construction_using_GSDF_attributes.
            if (new_decision_tree_quality) > (best_decision_tree_quality) then
                best_decision_tree = new_decision_tree;
            end
        end
        Update position and velocity of mass agents.
        decision_forest.add (best_decision_tree);
    end
    iteration=iteration+1;
end
Outcome=decision_forest // with final classification and recognition of fingerprints.
```

6. Results and discussion

The fingerprint recognition results are evaluated for the latent fingerprint dataset of NIST SD27 and the complete fingerprint dataset of FVC2004. In the NIST SD27 dataset (Garris & McCabe, 2000), a total of 258 fingerprint images along with their rolled fingerprints are available. These latent fingerprints are available in three categories: ugly, bad, and good, with respective images of 85, 85, and 88. Further, the FVC2002 dataset is a composition of four sub-datasets of DB1, DB2, DB3, and DB4, collected using various sensors and technologies (Maio *et al.*, 2002). Each sub-dataset consists of 80 fingerprint images. The sample images of the NIST SD27 and FVC2004 datasets are illustrated in Figure 7.

The proposed GSDF approach determines the match of fingerprints by evaluating the similarity score of minutiae features. For the latent fingerprints (NIST SD27 dataset), the minimum threshold value of similarity score is considered to be 75% as the latent fingerprints are incomplete fingerprints. On the other hand, the similarity score is set to be 95% for the complete fingerprints (FVC2004 dataset). The attainment of the mentioned similarity score threshold signifies the accurate match of the fingerprints. The performance results are calculated in terms of precision, recall, f-measure, and recognition rate by using similarity score values. To analyze the effectiveness of the proposed approach, the results are also calculated for the machine learning algorithms of random forest (RF) (Manpreet & Chhabra, 2022), decision tree (DT) (Azad *et al.*, 2022), back propagation neural networks (BPNN) (Kiran *et al.*, 2021), and k-nearest neighbor (KNN) (Manpreet & Chhabra, 2022). Tables 1-3 present the performance evaluation results for the NIST SD27 dataset, and Tables 4-7 describe the performance evaluation results for the FVC2004 dataset.

The results depicted in Tables 1-3 for latent fingerprint (NIST SD27 dataset) recognition indicate that the proposed GSDF approach has matched the latent fingerprints with complete fingerprints efficiently. The proposed approach has attained the recognition rate of 87.06% for the ugly class, 91.76% for the bad class, and 98.86% for the good class of latent fingerprints. The incorporated machine learning algorithms have also matched the latent fingerprint with complete fingerprints, but performance is inferior to the proposed GSDF approach.



Ugly Bad Good

(a) NIST SD27 Dataset

DB1 DB2 DB3 DB4

(b) FVC2004 Dataset

Fig. 7. Sample Images of the (a) NIST SD27 Dataset, and (b) FVC2004 Dataset.

Table 1. Performance Evaluation Results for the Ugly Fingerprint Class of the NIST SD27 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	72.55	87.06	79.14	87.06
RF	61.11	77.65	68.39	77.65
DT	54.46	71.76	61.92	71.77
BPNN	47.46	65.88	55.17	65.88
KNN	43.90	63.53	51.92	63.53

Table 2. Performance Evaluation Results for the Bad Fingerprint Class of the NIST SD27 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	76.47	91.76	83.42	91.76
RF	64.81	82.35	72.54	82.35
DT	60.91	78.82	68.72	78.82
BPNN	56.36	72.94	63.59	72.94
KNN	58.93	77.65	67.01	77.65

Table 3. Performance Evaluation Results for the Good Fingerprint Class of the NIST SD27 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	83.65	98.86	90.63	98.86
RF	76.47	88.64	82.11	88.64
DT	70.48	84.09	76.68	84.09
BPNN	69.23	81.82	75	81.82
KNN	72.12	85.23	78.13	85.23

The results for the complete fingerprint recognition illustrated in Tables 4-7 also indicate that the proposed GSDF approach is more efficient than incorporated machine learning algorithms. For the FVC2004 dataset, the proposed approach has attained a recognition rate of 98.75% for the DB1 class,

Table 4. Performance Evaluation Results for the DB1 Class of the FVC2004 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	96.34	98.75	97.53	98.75
RF	89.16	92.5	90.80	92.5
DT	83.53	88.75	86.06	88.75
BPNN	82.14	86.25	84.15	86.25
KNN	85.71	90	87.81	90

Table 5. Performance Evaluation Results for the DB2 Class of the FVC2004 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	96.30	97.5	96.89	97.5
RF	90.12	91.25	90.68	91.25
DT	83.72	90	86.75	90
BPNN	80	85	82.42	85
KNN	83.33	87.5	85.37	87.5

Table 6. Performance Evaluation Results for the DB3 Class of the FVC2004 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	91.46	93.75	92.59	93.75
RF	83.33	87.5	85.37	87.5
DT	75.58	81.25	78.31	81.25
BPNN	77.91	83.75	80.72	83.75
KNN	72.41	78.75	75.45	78.75

Table 7. Performance Evaluation Results for the DB4 Class of the FVC2004 Dataset.

Method	Precision (%)	Recall (%)	F-Measure (%)	Recognition Rate (%)
GSDF (Proposed)	93.90	96.25	95.06	96.25
RF	85.71	90	87.81	90
DT	80.23	86.25	83.13	86.25
BPNN	74.71	81.25	77.84	81.25
KNN	79.07	85	81.93	85

97.5% for the DB2 class, 93.75% for the DB3 class, and 96.25% for the DB4 class of the dataset which are superior to machine learning algorithms.

Furthermore, the overall comparison of the proposed approach with machine learning algorithms is conducted by incorporating the parameter of recognition rate. For overall comparison, the mean of the results for all the categories of the NIST SD27 and FVC2004 datasets is calculated separately. The comparative analysis is described by Figure 8.

In the overall results, the proposed approach has attained the recognition rate of 92.56% for latent fingerprints and 96.56% for complete fingerprints. For NIST SD27 dataset (Figure 8a), the recognition rate of the proposed GSDF approach is 9.68% better than the RF algorithm, 14.33% better than the DT algorithm, 19.01% better than the BPNN algorithm, and 17.09% better than the KNN algorithm. For FVC2004 dataset (Figure 8b), the recognition rate of the proposed GSDF approach is 6.25% better than the RF algorithm, 10% better than the DT algorithm, 12.5% better than the BPNN algorithm, and 11.25% better than the KNN algorithm. These comparative analysis results clearly indicate that the proposed GSDF approach is efficient compared to incorporated machine learning algorithms for both the latent and complete fingerprints.

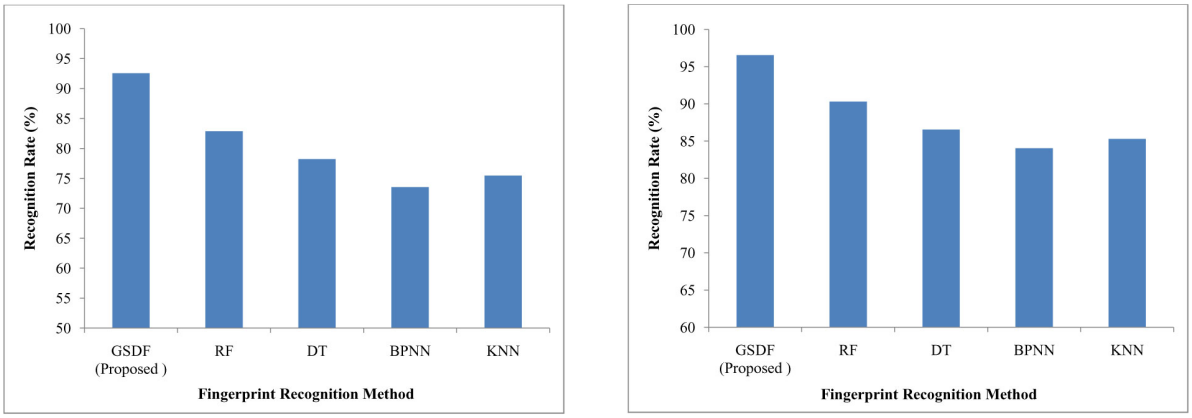


Fig. 8. Performance Comparison of the Proposed GSDF Approach with Machine Learning Algorithms for Experiments on (a) NIST SD27 Dataset (b) FVC2004 Dataset.

7. Conclusion

The paper has presented an automated fingerprint recognition system with the proposal of a novel GSDF approach. Initially, the fingerprints from the considered datasets (NIST SD27 and FVC2004) are preprocessed to enhance the poor quality images using a combined ridge dictionary and Gabor filter approach. Further, the minutiae-based features are extracted and spurious minutiae are filtered. The selected features feed into the proposed GSDF approach for fingerprint matching. The proposed approach efficiently determines the match of the fingerprints by constructing the decision trees using mass agents following the hypothesis of random forest. The final fingerprint match is determined by combining the outcomes of all the decision trees. The proposed approach has attained an average recognition rate of 92.56% for latent fingerprints (NIST SD27 dataset) and 96.56% for complete fingerprints (FVC2004 dataset), which are superior to incorporated machine learning algorithms of RF, DT, KNN, and BPNN.

Although the proposed approach has yielded efficient performance results for fingerprint recognition, the recognition rate for ugly latent fingerprints can be further optimized. It will also boost the overall performance of the proposed approach. In the future, we will combine the GSA method with a more effective classifier to improve the performance of ugly quality of the latent fingerprint.

References

Azad, M. & Moshkov, M. (2022). A bi-criteria optimization model for adjusting the decision tree parameters. *Kuwait Journal of Science*, **49**(2), 1-14.

Babatunde, I. G. (2015). Fingerprint matching using minutiae-singular points network. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, **8**(2), 375-388.

Cao, K. & Jain, A. (2015). Latent orientation field estimation via convolutional neural network. In *2015 International Conference on Biometrics (ICB)* (pp. 349-356). Phuket, Thailand: IEEE.

Cao, K. & Jain, A. K. (2018). Automated latent fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41**(4), 788-800.

Castillo-Rosado, K. & Hernández-Palancar, J. (2019). Latent fingerprint matching using distinctive ridge points. *Informatica*, **30**(3), 431-454.

Deshpande, U. U., Malemath, V. S., Patil, S. M. & Chaugule, S. (2021). Latent fingerprint identification system based on a local combination of minutiae feature points. *SN Computer Science*, **2**(3), 1-17.

Garris, M. D. & McCabe, R. M. (2000). Fingerprint minutiae from latent and matching tenprint images. *National Institute of Standards and Technology*, (1-36).

Guo, J. M., Liu, Y. F., Chang, J. Y. & Lee, J. D. (2014). Fingerprint classification based on decision tree from singular points and orientation field. *Expert Systems with Applications*, **41**(2), 752-764.

Hong, L., Wan, Y. & Jain, A. (1998). Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(8), 777-789.

Hsieh, C. T. & Hu, C. S. (2014). Fingerprint recognition by multi-objective optimization PSO hybrid with SVM. *Journal of Applied Research and Technology*, **12**(6), 1014-1024.

Jain, A.K., Prabhakar, S., Hong, L. & Pankanti, S. (2000). Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, **9**(5), 846-859.

Jindal, R. & Singla, S. (2021). Ant colony optimisation for latent fingerprint matching. *International Journal of Advanced Intelligence Paradigms*, **19**(2), 161-184.

Jindal, S., Sachdeva, M. & Kushwaha, A. K. S. (2022). Quantum behaved intelligent variant of gravitational search algorithm with deep neural networks for human activity recognition. *Kuwait Journal of Science*, 1-19. DOI: https://doi.org/10.48129/kjs.18531.

Kiran, P., Parameshachari, B. D., Yashwanth, J. & Bharath, K. N. (2021). Offline signature recognition using image processing techniques and back propagation neuron network system. *SN Computer Science*, **2**(3), 1-8.

Kozak, J. (2019). Ant colony decision forest approach. In *Decision Tree and Ensemble Learning Based on Ant Colony Optimization* (pp. 119-134). Cham: Springer.

Kumar, T. & Garg, R. S. (2020). The recognition of latent fingerprints using swarm intelligence based hybrid approach. *International Journal on Emerging Technologies*, **11**(5), 90-97.

Kumar, Y., Verma, S. K. & Sharma, S. (2020). Quantum-inspired binary gravitational search algorithm to recognize the facial expressions. *International Journal of Modern Physics C*, **31**(10), 2050138(1-24).

Lee, W., Cho, S., Choi, H. & Kim, J. (2017). Partial fingerprint matching using minutiae and ridge shape features for small fingerprint scanners. *Expert Systems with Applications*, **87**, 183-198.

Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L. & Jain, A. K. (2002). FVC2002: Second fingerprint verification competition. In *2002 International Conference on Pattern Recognition* (pp. 811-814). Quebec City, QC: IEEE.

Maltoni, D., Maio, D., Jain, A. K. & Prabhakar, S. (2009). Handbook of fingerprint recognition. Springer Science & Business Media, London.

Manpreet & Chhabra J. K. (2022). A hybrid approach based on k-nearest neighbors and decision tree for software fault prediction. *Kuwait Journal of Science*, 1-12. DOI: https://doi.org/10.48129/kjs.18331.

Murugan, A. & Rose, P. A. L. (2017). Fingerprint matching through back propagation neural network. *Indian Journal of Science and Technology*, **10**(29), 1-7.

Nadeem, A., Ashraf, M., Rizwan, K., Qadeer, N., AlZahrani, A., Mehmood, A. & Abbasi, Q. H. (2022). A Novel Integration of Face-Recognition Algorithms with a Soft Voting Scheme for Efficiently Tracking Missing Person in Challenging Large-Gathering Scenarios. *Sensors*, **22**, 1153(1-24).

Newman, D. (2007). The limitations of fingerprint identifications. *Criminal Justice*, **22**, 36.

Pradeep, N. R. & Ravi, J. (2022). An Efficient Machine Learning Approach for Fingerprint Authentication Using Artificial Neural Networks. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 852-859). Tirunelveli, India: IEEE.

Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, **179**(13), 2232-2248.

Singla, N., Kaur, M. & Sofat, S. (2022). Hybrid framework for identifying partial latent fingerprints using minutiae points and pores. *Multimedia Tools and Applications*, **81**, 19525–19542.

Vives, L., Tuteja, G. S., Manideep, A. S., Jindal, S., Sidhu, N., Jindal, R. & Bhatt, A. (2021). A novel hybrid approach of gravitational search algorithm and decision tree for twitter spammer detection. *International Journal of Modern Physics C*, 2250060(1-23).

Wong, W. J. & Lai, S. H. (2020). Multi-task CNN for restoring corrupted fingerprint images. *Pattern Recognition*, **101**, 107203(1-11).

Submitted: 24/05/2022
Revised: 16/07/2022
Accepted: 21/07/2022
DOI: 10.48129/kjs.20635

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-18)

Improved energy efficiency using meta-heuristic approach for energy harvesting enabled IoT network

Rekha*, Dr. Ritu Garg
Dept. of Computer Engineering, National Institute of Technology Kurukshetra, India
**Corresponding author: rekha_6170023@nitkkr.ac.in*

Abstract

Energy scarcity is a major problem for resource constrained Internet of Things (IoT) devices. Nowadays, Energy Harvesting (EH) has emerged as a promising solution to prolong the network lifetime using radio signals in wireless relay networks. In this article, we propose an optimization algorithm, based on meta-heuristic, to enhance the energy efficiency of amplify and forward relay IoT networks. Energy constraint relay exploits power-splitting based relay protocol to acquire energy from the source and transfer information to destination. We derive an expression for energy efficiency of the system using the throughput at destination and outage probability for performance evaluation. This investigation studies energy efficiency of the network against the various system parameters which are relay location, power-splitting factor, power transmitted, data rate, energy conversion efficiency and noise power and it enables us to find out which parameters need to be optimized. Further, an objective function is formulated to achieve the optimal solution for power transmitted by the source and an adaptive particle swarm optimization (OPA-APSO) algorithm is proposed to attain maximized energy efficiency. OPA-APSO differs from most existing approaches as it provides the best amount of energy harvested while optimizing the energy efficiency. Finally, simulation results demonstrate that OPA-APSO improves energy efficiency and throughput of the network significantly as compared to other existing techniques.

Keywords: Energy harvesting; internet of things; meta-heuristic; relaying protocol; wireless energy.

1. Introduction

In the past few years, a new trend Internet of Things (IoT) has evolved in the wireless communication area. IoT represents a 3A idea according to which any media can be connected anytime anywhere (Srivastava, 2006). IoT has become very popular in the information industry due to its applications in each and every aspect of life e.g. Figure 1.

To meet these numerous applications, billions of devices are required to be connected which are battery powered with limited life-time. Recharging and supplanting batteries can improve the device lifetime, but it can be costly and risky when devices are deployed in unfavorable conditions e.g., health, military applications, etc. To address this limited power battery problem in IoT, Energy Harvesting (EH) has become very popular in research areas and is a promising solution for power limited environments (Do *et al.*, 2017; Yan and Liu, 2017; Rekha and Garg, 2018).

EH enabled relay based IoT networks is very captivating in studies, as in (Lv *et al.*, 2018; Omoniwa *et al.*, 2018; Rauniyar *et al.*, 2019; Ashraf *et al.*, 2021). Transmitting simultaneous wireless information and power transfer (SWIPT) is not a new concept. Dual use of RF signals was first highlighted by (Varshney, 2008). To take advantage of SWIPT, (Zhou *et al.*, 2013) proposed two architectures, time-switch and power-split, for the relay nodes.

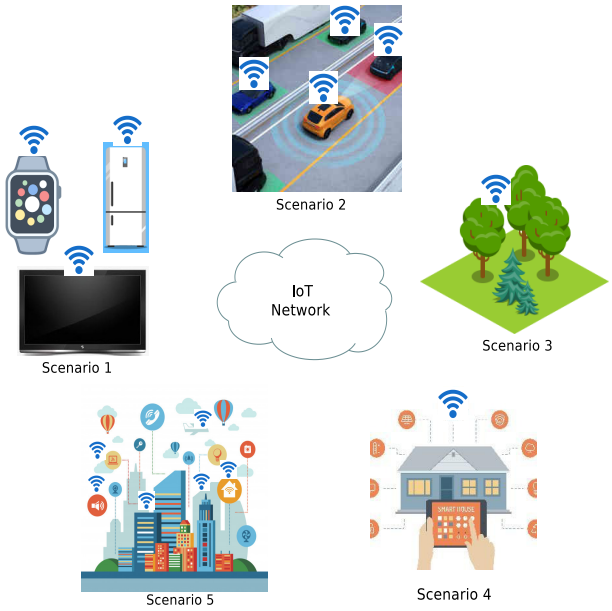


Fig. 1. IoT Scenario

(Chen *et al.*, 2014) studied the impact of power-splitting factor in dual-hop cooperative relaying system for the SWIPT scheme and evaluated the outage probability and ergodic capacity of the system. For the same relaying system, (Shah *et al.*, 2016) investigated the throughput of dual-hop cooperative relaying system by introducing a SWIPT scheme and analytical results described that at higher transmission rate (Shah *et al.*, 2016) outperformed (Chen *et al.*, 2014). Further, (Huang *et al.*, 2018) studied another network, in which both relay and direct branches can be used for transmission, but only a single branch is active at a time. In this, authors evaluated the performance of switch and stay technique using outage probability. In addition to this, (Yan *et al.*, 2018) introduced a framework for RF energy harvesting in relay based underlay cognitive networks. In this paper, prime focus was on energy harvesting using the SWIPT approach.

Further, the impact of energy harvested by the relay on outage probability and throughput was investigated in (Do, 2015). Authors proposed a scheme for an energy harvesting cooperative network and evaluated it using monte-carlo method. Later, authors introduced a dynamic allocation scheme exploiting PSR protocol for AF relaying network in (Do, 2019) and the monte-carlo method was used for analysis. Also, (Zou *et al.*, 2019) introduced PS based EH enabled optimal relay selection approaches in IoT network. (Nasir *et al.*, 2013) analyzed dual-hop AF system relay system (using both TSR and PSR) for optimal throughput using numerical analysis. Later, (Nasir *et al.*, 2014) examined throughput and ergodic capacity of EH enabled relay network by employing TSR and PSR protocols. Results showed PSR outperforms TSR protocols at a wide range of SNR, small relay distance etc.

Also, there are research works in literature which aim to optimize their objective to improve the performance of EH-enabled relaying networks. (Tang *et al.*, 2018) proposed an optimization algorithm to solve optimal power allocation problem for wireless acoustic relay sensor networks and analyzed the throughput of the system. (Rauniyar *et al.*, 2018) developed an algorithm to maximize sum-throughput using the Golden section search method and evaluated it in a PS based IoT relay system.

In addition to this, (Gurjar *et al.*, 2018), analyze the impact of SNR and target rate on throughput and energy efficiency of EH enabled IoT communication system. It can be inferred from the results that the energy efficiency depends on SNR value. Further, (Ji *et al.*, 2018) focused on energy efficiency of IoT network exploiting the PS relaying scheme. For this situation, the authors formulated an optimization problem to focus on energy management and solved this using the Lagrangian multiplier method. Also, (Lv *et al.*, 2018) introduced the iterative optimization algorithm employing Lagrange multipliers to maximize the energy efficiency of an IoT network.

As mentioned above, the majority of the existing studies mainly deal with outage probability and throughput of the system. The techniques in literature attain the optimal value of throughput/energy efficiency using numerical analysis or analytical analysis without considering the amount of energy harvested by the relay.

1.1 Contributions

Here, we propose a meta-heuristic algorithm for energy efficiency optimization in EH-enabled IoT networks to reduce time and mathematical formulation complexity. Algorithm optimizes the energy efficiency as well as gives the best value of the amount of energy harvested by relay for that particular value of energy efficiency. To the best of our knowledge, this is the first work to study energy efficiency of a system against various parameters and to propose a meta-heuristic based optimization scheme. Main contribution of this article is listed as below:

1. Considering the dual-hop AF relay network, we present the single expression for energy efficiency of the network in delay-limited transmission mode. For achievable energy efficiency, first we obtain the outage probability, and then we evaluate throughput at the destination.

2. To gain insights, we analyze the impact of various system parameters Power transmitted (P_s), energy conversion efficiency (η), power-splitting factor (ρ_h), Transmission Rate (R), relay location and noise variances on achievable energy efficiency.

3. Further, based on this analysis, we propose a meta-heuristic based OPA-APSO algorithm to optimize the energy efficiency of a system constrained to signal-to-noise ratio. In addition to optimized energy efficiency, the proposed algorithm provides the best value of the amount of energy harvested corresponding to the achieved energy efficiency.

4. Results demonstrate significant improvement in throughput and energy efficiency compared to existing approaches. Further, statistical analysis has been carried out to evaluate the performance of the proposed algorithm.

Nomenclature: Various types of symbols used throughout this article and their meanings are given in Table 1.

1.2 Organization

Organization of remaining paper is as follows. Section 2, gives the description network model with its assumptions and information processing and energy harvesting process in detail. This Section also presents mathematical expressions for system's throughput and energy efficiency. Following this, the optimization problem is formulated in Section 3. To solve this formulated problem, Section 4 explains OPA-APSO algorithm in detail. Section 5 demonstrates obtained results and a comparison with existing approaches. Finally, we summarize the paper in Section 6.

Table 1. Nomenclature

Parameters	Meaning	Parameters	Meaning
P_s	Power transmitted by source	n_r^a	additive white Gaussian noise (AWGN) at relay node
T	Time Block	n_r^c	additive conversion noise at relay
η	RF to power conversion efficiency	n_d^a	additive white Gaussian noise at destination
s_i	Transmitted signal	n_d^c	additive conversion noise at destination
ρ_h	Power splitting factor	P	Power received by relay
d_{sr}	Distance between source and relay	$\mathbb{E}_h$	Energy harvested by relay
d_{rd}	Distance between relay and destination	P_{out}	Outage Probability
m	Path loss exponent	$\mathbb{SNR}_d$	Signal-to-noise ratio at destination
R	transmission Rate	EE	Energy Efficiency of system
h and g	Channel gain between source and relay and between relay and destination	γ_{thr}	minimum value of SNR at destination node

Note: In the article symbol $\mathbb{S}$ represents the signal. Symbols r/s/d in subscript represent whether signal is at relay, source or destination. Symbols rec/tra in superscript of $\mathbb{S}$ represent whether signal is transmitted or received.

2. Network model and description

In this article, we consider a scenario as presented in Figure 2 for AF IoT relay network. This system consists of 3 nodes, featuring a single antenna for each node. In this dual-hop communication system, source (S) transmits wireless information to destination (D) via relay node. There is no direct communication between S and D, and communication takes place through R only. Relay node is power restrained. First, R acquires energy using a signal received from S and then uses the energy to amplify and forward the data to D. Channel gain coefficients from S to R and R to D are denoted by h and g respectively. For this system, quasi-static block fading channels are assumed which are independent and remain fixed over one time block.

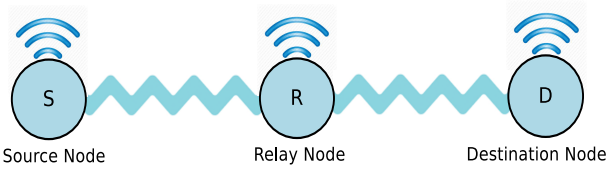


Fig. 2. System Model

2.1 Energy harvesting and information processing in PSR based IoT network

Figure 3 depicts the transmission block diagram according to PSR(Nasir *et al.*, 2014) for wireless information and energy transmission. In this protocol, the total time block (T) is partitioned into two slots. During the first slot, T/2, communication takes place by transferring data from source to relay. In the second slot, the relay node communicates with the destination. Relay harvests the energy along with information processing in the first time slot. Relay harvests the energy by using a fraction (ρ_h) of power received (P), i.e ($\rho_h P$), and the rest of received power, i.e ($(1-\rho_h)P$), is for data processing.

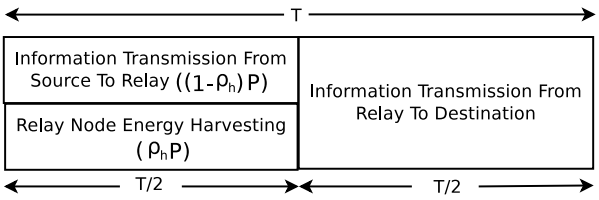


Fig. 3. PSR protocol illustration

Relay scavenges the energy first in the energy harvesting phase which is consumed in the transmission phase. Energy harvested by relay depends on both power received by relay and time duration of harvesting phase. Relay scavenges the energy for $T/2$ time period. So, harvested energy by relay ($\mathbb{E}_h$) is

$$\mathbb{E}_h = \frac{\eta \rho_h P_s |h|^2}{d_{sr}^m} T/2, \tag{1}$$

where, η is between 0 and 1 and its value depends on the circuitry (Shaikh and Zeadally, 2016). Signal received by relay is not the same as transmitted by source. Hence, after adding the noise signal n_r^a by the receiver at relay, signal received at relay $\mathbb{S}_r^{rec}$ is

$$\mathbb{S}_r^{rec} = \frac{\sqrt{P_s(1-\rho_h)} h s_i}{\sqrt{d_{sr}^m}} + (1-\rho_h) n_r^a, \tag{2}$$

where s_i is signal transmitted with unit power, h$\sim$ $\mathbb{CN}$(0,1) is channel gain between source and relay.

Relay processes $\mathbb{S}_r^{rec}$ by converting it from RF to baseband. During the conversion, additive noise n_r^c is added to the signal due to conversion. So, $\hat{\mathbb{S}}_r^{rec}$, signal obtained after the down conversion at relay node, is given as

$$\hat{\mathbb{S}}_r^{rec} = \frac{\sqrt{P_s(1-\rho_h)}hs_i}{\sqrt{d_{sr}^m}} + (1-\rho_h)n_r^a + n_r^c, \tag{3}$$

Before retransmitting the received signal, it is amplified at the relay node. Hence, relay transmits information $\mathbb{S}_r^{tra}$ which is as follows

$$\mathbb{S}_r^{tra} = \frac{\sqrt{\mathbb{P}_r}\hat{\mathbb{S}}_r^{rec}}{\sqrt{\frac{(1-\rho_h)P_s|h|^2}{d_{sr}^m} + (1-\rho_h)(\sigma_r^a)^2 + (\sigma_r^c)^2}}, \tag{4}$$

where, $\mathbb{P}_r$ is transmitted power to destination by the relay. $\mathbb{P}_r$ can also be calculated as

$$\mathbb{P}_r = \frac{\mathbb{E}_h}{T/2} = \frac{\eta P_s|h|^2\rho_h}{d_{sr}^m}, \tag{5}$$

T/2 is the total duration during which communication takes place between relay and destination. Denominator in eq.(4) represents the power constraint factor at the relay node. By replacing the variance of n_r^a and n_r^c with $n_r \triangleq \sqrt{(1-\rho_h)}n_r^a + n_r^c$, combined variance $\sigma_r^2 \triangleq (1-\rho_h)(\sigma_r^a)^2 + (\sigma_r^c)^2$, eq.(4) can be expressed as

$$\mathbb{S}_r^{tra} = \frac{\sqrt{\mathbb{P}_r}\hat{\mathbb{S}}_r^{rec}}{\sqrt{\frac{(1-\rho_h)P_s|h|^2}{d_{sr}^m} + (\sigma_r)^2}}. \tag{6}$$

Destination node receives signal $\mathbb{S}_d^{rec}$ which can be given as

$$\mathbb{S}_d^{rec} = \frac{g\mathbb{S}_r^{tra}}{\sqrt{d_{rd}^m}} + n_d^a + n_d^c, \tag{7}$$

Using eq.(3),(5) and (6), signal received at destination in eq.(7) can be simplified as

$$\mathbb{S}_d^{rec} = \underbrace{\frac{\sqrt{\eta\rho_h(1-\rho_h)}P_s g h^2 s_i}{\sqrt{d_{rd}^m d_{sr}^m}\sqrt{(1-\rho_h)P_s|h|^2 + d_{sr}^m(\sigma_r)^2}}}_{\text{Signal Part}} + \underbrace{\frac{\sqrt{\eta\rho_h P_s}ghn_r}{\sqrt{d_{rd}^m}\sqrt{(1-\rho_h)P_s|h|^2 + d_{sr}^m(\sigma_r)^2}} + n_d}_{\text{Noise Part}}, \tag{8}$$

where $n_d \triangleq n_d^a + n_d^c$ is combined AWGNs at destination. $\mathbb{S}_d^{rec}$ in eq.(8) consists of two parts, i.e., signal part and noise part. Hence, the signal-to-noise ratio ($\mathbb{SNR}_d$), i.e. $\frac{\mathbb{E}\{\text{Signal Part}^2\}}{\mathbb{E}\{\text{Noise Part}^2\}}$ at node D can be expressed as eq.(9).

$$\mathbb{SNR}_d = \frac{\eta\rho_h(1-\rho_h)P_s^2 g^2 h^4}{\eta\rho_h P_s g^2 h^2 d_{sr}^m(\sigma_r)^2 + P_s|h|^2 d_{sr}^m d_{rd}^m(1-\rho_h)(\sigma_d)^2 + (d_{sr}^m)^2 d_{rd}^m(\sigma_r)^2(\sigma_d)^2} \tag{9}$$

Throughput: This article considers delay limited transmission mode where throughput of the system is analyzed by calculating outage probability (P_{out}) for a particular data rate (R bits/sec/Hz) and $R \triangleq \log_2(1+\gamma_{thr})$, where γ_{thr} is threshold SNR, i.e $\gamma_{thr} = 2^R - 1$, for which destination can correctly detect the data. The P_{out} can be determined as

$$\mathsf{P}_{out} = \mathsf{Pr}(\mathbb{SNR}_d < \gamma_{thr}) \tag{10}$$

The outage probability of destination for the protocol is given by the following proposition(Nasir *et al.*, 2013).

Proposition 1: For PSR protocol, outage probability at destination D can be determined as

$$\mathsf{P}_{out} = 1 - \frac{1}{M_h}\int_{k=\frac{z}{y}}^{\infty} e^{-\left(\frac{k}{M_h}+\frac{wk+x}{(yk^2-zk)M_g}\right)}dk \tag{11}$$

$$\mathsf{P}_{out} \approx 1 - e^{-\frac{z}{yM_h}}\beta\mathcal{K}_1(\beta) \tag{12}$$

For convenience, we have defined

$$w = P_s d_{sr}^m d_{rd}^m \sigma_d^2 (1-\rho_h)\gamma_{thr},$$
$$x = d_{sr}^{2m} d_{rd}^m \sigma_r^2 \sigma_d^2 \gamma_{thr},$$
$$y = \eta\rho_h(1-\rho_h)P_s^2,$$
$$z = \eta\rho_h P_s d_{sr}^m \sigma_r^2 \gamma_{thr},$$
$$\beta = \sqrt{\frac{4w}{yM_hM_g}},$$

here, M_h and M_g represent means for the exponential random variables $|h|^2$ and $|g|^2$ respectively. And $\mathcal{K}_1(.)$ denotes first order modified Bessel function of the second kind(Gradshteyn and Ryzhik, 2014). Detailed derivation of this proposition is given in (Nasir *et al.*, 2013)[1]. Here, effective communication time is $T/2$, hence throughput at destination is give as:

$$THR = \frac{(1-\mathsf{P}_{out})RT}{2T} = \frac{R(1-\mathsf{P}_{out})}{2} \tag{13}$$

Energy Efficiency: Energy efficiency of a system is characterized as a ratio of spectrum efficiency of a system over the whole power consumption of an IoT network. Here, total power expenditure is represented as $aP_s + b$ as in (Ji *et al.*, 2018). $a > 1$ and $b > 0$ are factors considering power conversion efficiency and the hardware circuits in the power consumption model. Thus, using eq.(13), we present energy efficiency at node D here, which can be determined as given below

$$EE = \frac{THR}{aP_s + b} = \frac{R(1-\mathsf{P}_{out})}{2(aP_s+b)} \tag{14}$$

3. Problem formulation

To enhance the energy efficiency of the system, this section deals with the first step of optimization i.e. optimization problem formulation to attain the optimal value of P_s. Here, we formulate our objective function to maximize the energy efficiency of the system subjected to constraint to minimum SNR at destination as follows:

$$\begin{aligned} &\underset{P_s}{Max}\ EE\left(P_s\right), \\ &\text{s.t. } \mathbb{SNR}_d \geq \gamma_{thr} \end{aligned} \tag{15}$$

Here, $EE\left(P_s\right)$ represents energy efficiency as a function of power transmitted by source. Further, formulated objective function can be given as by inserting eq.(14) into eq.(15):

$$\begin{aligned} &\underset{P_s}{Max}\frac{(1-\mathsf{P}_{out})R}{2(aP_s+b)} \\ &\text{s.t.}\frac{\eta\rho_h(1-\rho_h)P_s^2 g^2 h^4}{\eta\rho_h P_s g^2 h^2 d_{sr}^m(\sigma_r)^2 + P_s|h|^2 d_{sr}^m d_{rd}^m(1-\rho_h)(\sigma_d)^2 + (d_{sr}^m)^2 d_{rd}^m(\sigma_r)^2(\sigma_d)^2} \geq \gamma_{thr} \end{aligned} \tag{16}$$

The optimization problem represented by eq.(16) is a non-linear constraint problem. Also, complex computational terms involved in computing outage probability need to be solved iteratively with low implementation and time complexity. Therefore, we proposed an OPA-APSO algorithm to attain the optimal solution.

[1]Detailed proof is provided in (Nasir *et al.*, 2013) and omitted here due to the space limitation.

4. Proposed algorithm

This section introduces a novel optimization algorithm OPA-APSO to maximize the energy efficiency of a system. OPA-APSO optimizes system parameters to maximize the achievable energy efficiency. Further, the proposed algorithm has an extra characteristic that it keeps track of the amount of energy harvested while maximizing the energy efficiency. OPA-APSO uses a meta-heuristic approach to give the finest energy efficiency for the considered IoT network. To solve the intractable optimization problem, meta-heuristics techniques are very impressive in the research area (Mortazavi and Ahmadi, 2019; Rao *et al.*, 2020; Gupta *et al.*, 2021; Devi and Prabakaran, 2021). There is no doubt that this field will continue to develop in the near future in the studies (Dokeroglu *et al.*, 2019). Also, opposite to exact methods which require high computational time to find the optimal solution, meta-heuristic techniques attain near optimal solution rather quickly (Hussain *et al.*, 2019).

(Poli *et al.*, 2007) introduced a meta heuristic approach inspired by the social behaviour of birds and fishes known as "Particle Swarm Optimization (PSO)". Many optimization problems have been solved successfully using PSO. PSO has the ability to explore the global space and exploit local space. PSO is very robust and also converges to optima very fast. Also, PSO has been used in a large and various real life applications. So, we opt the PSO for energy efficiency optimization. To get better results, we make it adaptive by varying the inertia weight. Using the time-varying inertia weight, premature convergence and local optima is avoided.

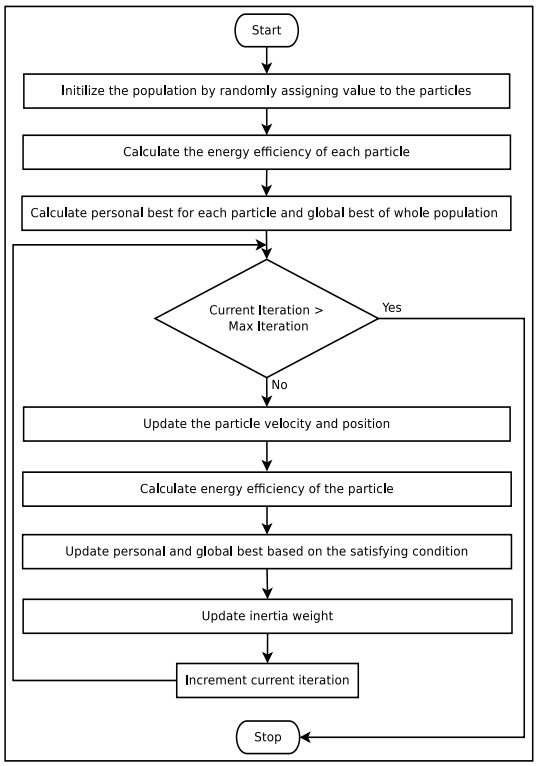


Fig. 4. Flowchart of OPA-APSO

Here, we present the Optimal Power Allocation algorithm using Adaptive PSO (OPA-APSO) to solve the optimization problem formulated in eq.(16). Flow chart of the proposed scheme is shown in Figure 4. Here, the algorithm is explained in detail.

Algorithm 1 is divided into two sections: Initialization and Updation. In the initialization section , all the algorithm parameters are initialized and in the updation section, values are updated to find the optimal result.

4.1 Initialization

Algorithm 1 : Optimal Power Allocation algorithm using Adaptive PSO
Procedure OPA-APSO

1: *Initialize the nPop, MaxIt, c1, c2, wMax, wMin, gBest=0*
2: **for** *i=1 to nPop* **do**
3: *Initialize the positions of particles x_i by assigning random values of power transmitted by source.*
4: *Initialize the velocity of particle v_i using random value*
5: *Evaluate the fitness EE(x_i) using Algorithm 2*
6: *Set the $pBest_i$ to the current position x_i*
7: **for** *i=1 to nPop* **do**
8: **if** $EE(pBest_i) > EE(gBest)$ **then** $gBest$=$pBest_i$
9: **if** $EE(pBest_i) = EE(gBest)$ **then**
10: **if** $EH(pBest_i) > EH(gBest)$ **then** $gBest$=$pBest_i$
11: **for** *it=1 to MaxIt* **do**
12: **for** *each particle x_i* **do**
13: *Update the velocity of particle using eq.(17)*
14: *Update new position using eq.(18)*
15: *Evaluate the Fitness EE(x_i) using Algorithm 2*
16: **if** *EE(x_i)*$> EE(pBest_i)$ **then** $pBest_i$=x_i
17: **if** *EE(x_i)*$= EE(pBest_i)$ **then**
18: **if** *EH(x_i)*$> EH(pBest_i)$ **then** $pBest$=x_i
19: **if** $EE(pBest_i) > EE(gBest)$ **then** $gBest$=$pBest_i$
20: **if** $EE(pBest_i) = EE(gBest)$ **then**
21: **if** $EH(pBest_i) > EH(gBest)$ **then** $gBest$=$pBest_i$
22: *Update inertia w=wMax-it*((wMax-wMin)/MaxIt);*
23: **return** *gBest*

From steps 1 to 8, all the parameters are initialized. In step 1, various parameters are set which are:

a. nPop: Total number of population.

b. MaxIt: Total number of iterations.

c. Learning Parameters (c1,c2): c1 is a cognitive learning parameter and represents the particle's desirability moving towards its own success. c2 is a social learning parameter and represents the particle's desirability moving towards the neighbor's success.

d. Inertia weight (w): used to control variation of velocity in the succeeding iteration from the previous one. The value of w has an impact on exploration and exploitation. Higher value of w facilitates exploration, while smaller w is beneficial for local search.

In steps 3 and 4, population vector and velocity vectors are initialized. In the population vector, each particle is assigned position x_i randomly. Velocity vector is initialized by assigning a random velocity v_i to each particle. Then calculate the fitness of particles using step 5. Assign current particle position to personal best ($pBest$) for each particle in step 6, which is the best position of particle till now. From all the personal bests, find the global best position ($gBest$) in step 7 to step 10. If EE of the personal best is greater than the global best then set $gBest$ to $pBest$ in step 8. If EE of the $pBest$ and $gBest$ are the same then their energy harvested is checked in step 9. If the EH($pBest$) is greater than the EH($gBest$) then $gBest$ is reset to $pBest$ in step 10.

4.2 Updation

In this section, values are updated to find the optimal solution.

a Velocity Update: In step 13, the velocity of each particle is updated in each iteration using personal best position and global best. The velocity is updated using eq.(17) to move the particle towards global best and its own best (Chen and Yu, 2005).

$$v_{ij} = w * v_{ij} + \underbrace{c_1 * r_1 * (pBest_i - x_{ij})}_{\text{particle personal best}} + \underbrace{c_2 * r_2 * (gBest - x_{ij})}_{\text{global best}} \quad (17)$$

Here, x_{ij} and v_{ij} are the position and velocity of i^{th} particle in j^{th} iteration respectively. r_1 and r_2 are random values between 0 and 1.

b Position Update: Using the updated velocity in step 13, the position of each particle is updated so that the particle can move towards optimal value. In step 14, a new position for each particle is obtained using eq.(18) in each iteration. As the velocity is calculated using both personal best and global best factors, the same impact will be on particle position.

$$x_{ij} = x_{ij} + v_{ij} \quad (18)$$

c Personal Best Update: Step 15 calculate the fitness value for each particle and then based on new fitness, each particle's personal best is updated. If the new fitness value is higher than the $pBest$ of the particle the $pBest$ is updated to that position in steps 16-18.

d Global Best Update: Based on the previous steps, steps 19-21 update the global best to current best position. It yields the highest fitness value among all personal bests till that iteration along with the highest energy harvested for the same energy efficiency.

e Inertia Weight Update (w): Value of w affects the ability of exploitation and exploration. We need to avoid local minima and exploit the global space. Hence, to obtain exploration & exploitation trade-off, time adaptive w is used (Shi and Eberhart, 1998). The inertia weight is calculated as:

$$w = wMax - it * ((wMax - wMin)/MaxIt), \quad (19)$$

where, $wMax$ represents initial inertia weight and $wMin$ is the final value of inertia weight. it is the current iteration.

If the number of iteration exceeds $MaxIt$ then the algorithm stops by returning the $gBest$.

Algorithm 2 : Evaluate Fitness EE(x) and Energy Harvested EH

1: *Input all the parameters* P_s,a,b,g,h,$\eta, \rho_h, d_{sr}, d_{rd}, \sigma_d, \sigma_r$
2: *Calculate* $\mathbb{SNR}_d$ *at destination using eq.(9)*
3: *Calculate outage probability* P_{out} *using eq.(12)*
4: *Calculate throughput THR of system using eq.(13)*
5: *Calculate energy harvested EH by relay using eq.(1)*
6: *Calculate energy efficiency EE of system using eq.(14)*
7: **return** *EE and EH*

Algorithm 2 describes the evaluation of fitness function. Step 1 initializes the various parameters for the system model. Using all the parameters and eq.(9), signal-to-noise ratio at destination is calculated in step 2. Then using $\mathbb{SNR}_d$ and eq.(12), step 3 calculates the outage probability which is used in step 4 to obtain the throughput. Step 5 provides the energy harvested by the relay node. Finally, in step 6 energy efficiency of the system is evaluated which is our objective function.

4.3 Computational complexity

OPA-APSO aims to maximize the system energy efficiency with the best value of energy harvested by relay. In each iteration, OPA-APSO moves towards the convergence by finding the optimal value of power transmitted. Computational complexity is analyzed under the worst case scenario, i.e, convergence is obtained after completing every iteration.

We assume that the algorithm takes *m* population size and *n* number of iterations. Step 1 initializes all the parameters with O(1) time complexity. For loop (Steps 2 to 6) runs for *m* times to calculate the $pBest$ of each particle. So, the complexity of this loop is O(*m*).

The next $pBest$ of i^{th} particle (i=1,....., *m*) is achieved by some set of operations like addition, multiplication and comparison. Hence, predicting the $pBest$ of each particle is computed in *m* computation time. Therefore, the computation time of steps 7 to 10 is O(*m*), as all the operations have O(1) time complexity. Now, each operation from steps 13 to 21 is performed for each particle in each iteration, i.e, *m* times. So, the time complexity of steps 11 to 22 is equivalent to O(*m***n*).

Hence, overall complexity of proposed OPA-APSO is O(*m*)+O(*m*)+O(*m***n*) in dual-hop relay based IoT system which is equivalent to O(*m***n*).

5. Results and analysis

Analytical results using the expressions derived in the previous section are presented here. We have obtained results into two sets using MATLAB 2016. First set is carried out to investigate the effect of system parameters on energy efficiency. This set provides how energy efficiency varies with each parameter. And based on this analysis, the second set is used to optimize the energy efficiency keeping in consideration to give the best value of energy harvested.

We compare our approach with approaches (Ji *et al.*, 2018), (Nasir *et al.*, 2013) and (Do, 2019). (Ji *et al.*, 2018) consider a dual-hop relay network system exploiting PSR protocol and optimize the energy efficiency at the destination. Authors optimize the solution by using the optimal value of the power-splitting factor. (Nasir *et al.*, 2013) study the impact of various parameters on throughput of the system for both TSR and PSR protocols and optimize the throughput. They provide the analysis which protocol performs better in which situation. Further, (Do, 2019) optimizes the throughput of relay based model using PSR. They find the optimal value of the power-splitting factor to optimize the throughput of the system. Mentioned approaches use numerical methods to solve the optimization problem.

5.1 Impact of various system parameters

In our considered system, default values for the various parameters are adopted as P_s =1 Joules/sec, η =1, m =2.7 and R =3 bits/sec/Hz. d_{sr} and d_{rd} are normalized to 1. Antenna noise covariances (σ_a^2) and conversion noise covariances (σ_c^2) at both relay and destination are assumed equal for simplicity. The mean values M_h and M_g of channel gain parameters $|h|^2$ and $|g|^2$ are assigned unit values. These simulation settings are in line with work by (Nasir *et al.*, 2013). Power consumption parameters: *a* varies from 2 to 10 and *b* = 10,100 and 300 (Ji *et al.*, 2018).

From eq.(14), it can be seen that a system's energy efficiency depends on various parameters $P_s, R, \eta, \rho_h, d_{sr}, d_{rd}, \sigma_c^2, \sigma_a^2$, etc. So, we study the analysis of different parameters on the system's energy efficiency individually keeping all other parameters fixed.

Figure 5 plots energy efficiency of the system vs. power transmitted by the source node for various values of power consumption parameters. From Figure 5, it is obvious that the energy efficiency increases with the increase in P_s till it reaches an optimal value and then it starts decreasing for each curve. It is due to an increase in total network power consumption (aP_s+b) with the increase in transmitted power. Throughput increases as P_s increases. For lower value of power, increase in throughput is more considerable than the total power consumption. On the other hand, increase in total power consumption is more considerable than throughput at the higher values of power. So this results in first increasing the energy efficiency of the system upto optimal value then it starts decreasing. Total power expenditure is low for the lower values of a and b, but it increases with increase in a and b. For the lower values of

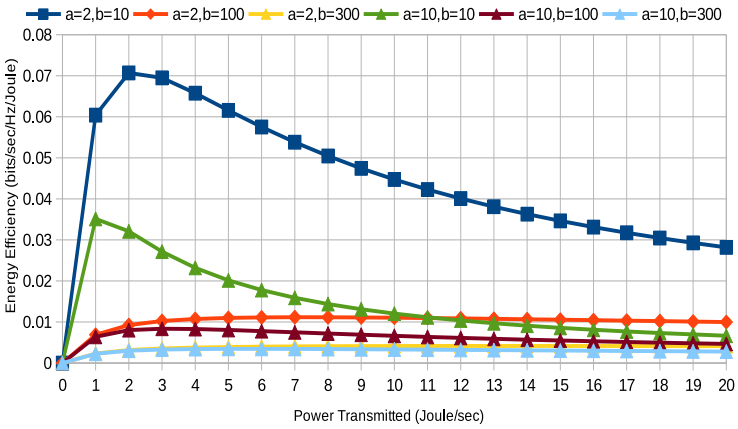


Fig. 5. Energy Efficiency vs. P_s

b, aP_s is considerable when P_s changes. It results in a significant change in energy efficiency with an increase in P_s for lower values of a and b. But due to the high value of b, change in aP_s is significantly low as compared to b with the increase of P_s. Hence, there is negligible change in energy efficiency.

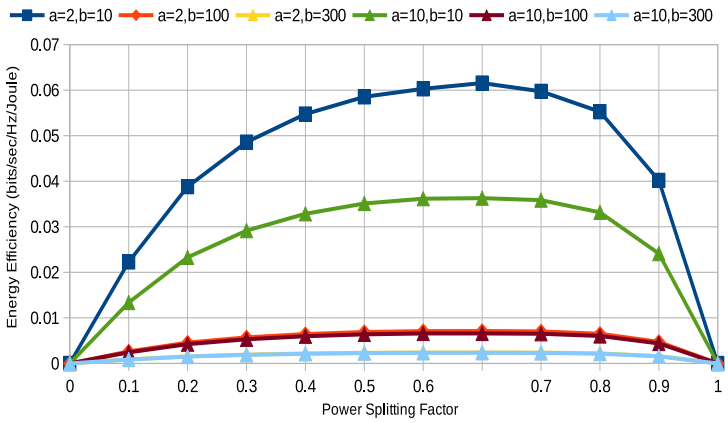


Fig. 6. Energy efficiency vs. power splitting factor (ρ_h)

Figure 6 shows achievable energy efficiency as a function of power splitting factor (ρ_h). We can see the energy efficiency of the system first increases upto some optimal point and then start decreasing as ρ_h approaches to 1 for various values of a and b as depicted in Figure 6. Reason is that for the smaller values of ρ_h relay harvests less power which yields lower energy efficiency of the system. On the contrary, for the values of ρ_h larger than the optimal value, the relay node has more power to harvest and less energy to process the information. Therefore, the relay node has low signal strength and it results in lower energy efficiency.

Further, the location of the relay (d_{sr}) between source and destination also affects the efficiency as shown in Figure 7. Here, d_{rd} is set to $d_{rd} = 2 - d_{sr}$ for all curves. As we can see from Figure 7, the system's energy efficiency decreases as d_{sr} increases. It is due to the reason that as d_{sr} increases both signal received and energy harvested by the relay decrease which results in lower energy efficiency.

Figure 8 plots the variation of energy efficiency with different values of R. Energy efficiency increases with increase in R upto optimal value and then starts decreasing as shown in Figure 8 for every curve. At lower data transmission rate energy efficiency increases with increase in data rate. Contrary to this, at higher values of R, the receiver is not able to decode a large amount of data correctly in a limited period. Therefore, there is an increase in outage probability ($\mathbb{P}_{out}$), which leads to decrease in energy efficiency.

Figure 9 plots the variation of energy efficiency with different values of energy conversion efficiency

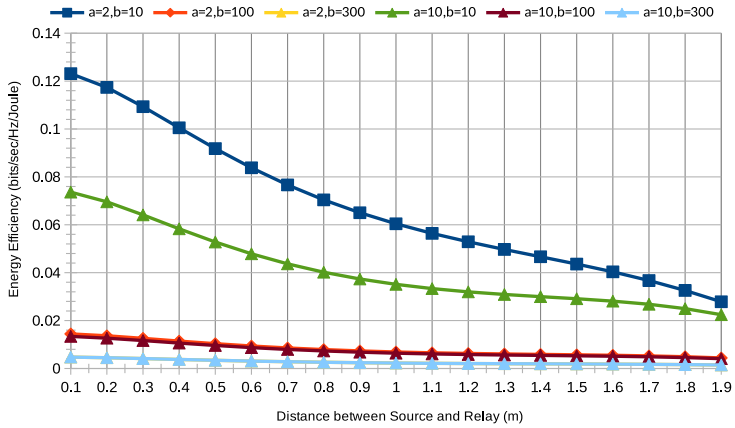


Fig. 7. Energy efficiency vs. distance between source and relay (d_{sr})

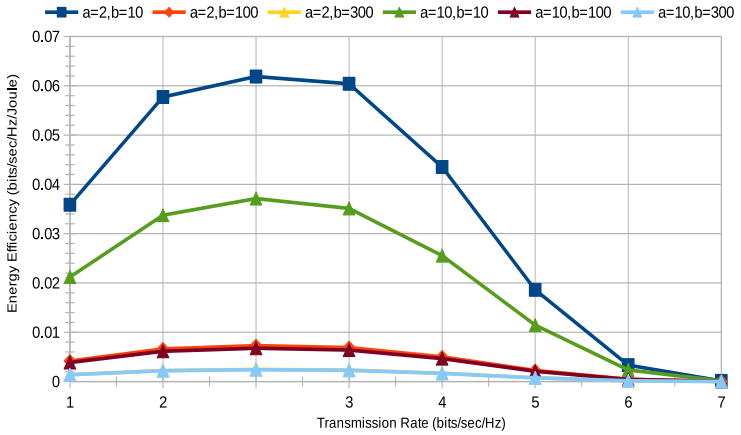


Fig. 8. Energy efficiency vs. data transmission rate (R)

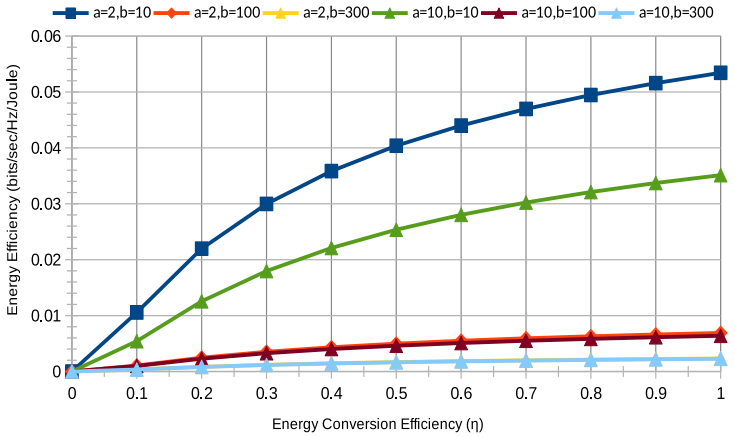


Fig. 9. Energy efficiency vs. energy conversion efficiency (η)

(η). Energy efficiency increases with increase in η.

Figure 10 depicts the effect of conversion noise variance (σ_c^2) on the energy efficiency of a system by keeping all other parameters fixed. From Figure 10, it can be observed that energy efficiency decreases with increase in σ_c^2. The increased conversion noise affects the throughput at destination which results in lowering the energy efficiency for various values of a and b. And the similar trend is followed in

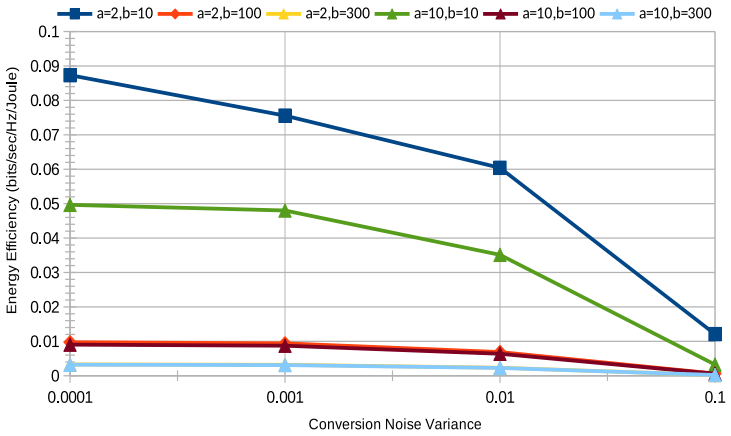


Fig. 10. Energy efficiency vs. conversion noise variance

Figure 11, which plots the variation of energy efficiency with different values of antenna noise variance (σ_a^2).

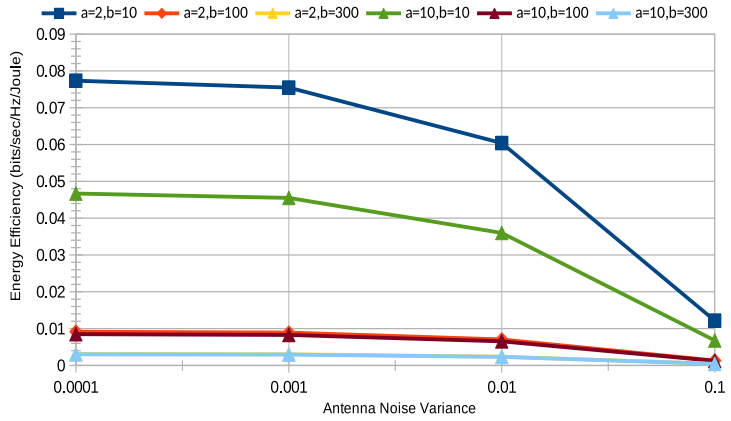


Fig. 11. Energy efficiency vs. antenna noise variance

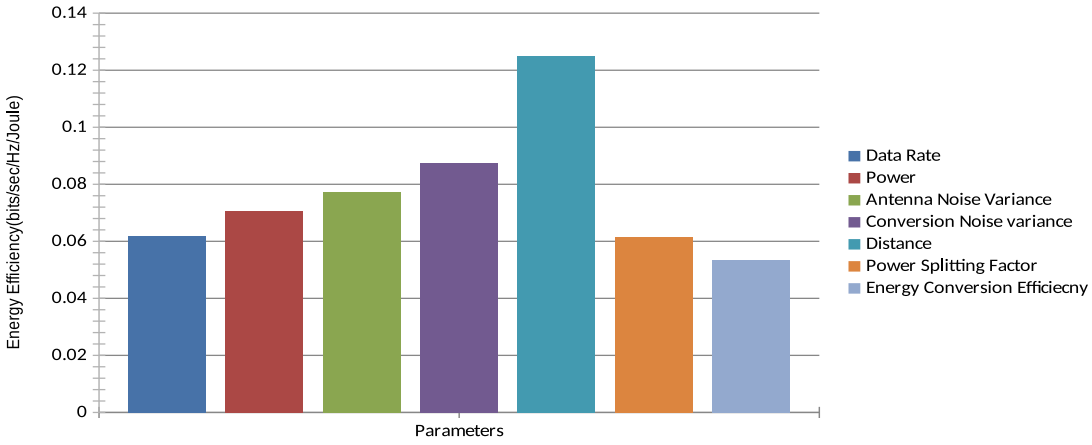


Fig. 12. Optimized Energy Efficiency for various parameters

5.2 Optimized energy efficiency using OPA-APSO

In the previous section, we analyzed how energy efficiency of system is affected by various system parameters. Energy efficiency varies linear fashion with η, d_{sr}, σ_c^2 and σ_a^2 while with P_s, ρ_h, R parameters varies in parabolic pattern. Based on this analysis, we employ the OPA-APSO to find the optimal values of power transmitted to optimize the energy efficiency. Based on this analysis, we employ the OPA-APSO to find the optimal values of system parameters to optimize the energy efficiency. OPA-APSO optimizes the energy efficiency against only one parameter at a time. We also optimize the R and ρ_h using OPA-APSO. Figure 12 represents the optimized energy efficiency of the system for the various parameters and the obtained optimal values of different system parameters $P_s, R, \eta, \rho_h, d_{sr}, \sigma_c^2, \sigma_a^2$ are 2.0468,2.6288,0.63799,1.2024E-07,0.0001,0.0001 and 1 respectively.

5.3 Statistical analysis

We run the OPA-APSO algorithm over 15 cycles and the simulation results are represented by the mean values. To evaluate the statistical performance of the proposed algorithm, we have used the standard deviation and coefficient of variance (CoV). Standard Deviation (SD) is a method used to measure the distribution of the data about the mean value. CoV% is calculated as:

$$CoV\% = \frac{SD}{Mean} * 100$$

Lower values of SD and CoV mean results provided by the algorithm are stable. Table 2 gives the values of mean, SD and CoV for various parameters.

Table 2. Statistical Analysis of Results

Parameter	Mean	SD	CoV%
Data Rate	0.061872	1.11E-05	0.018
Power	0.070682	1.24E-05	0.01756
Antenna Noise Variance	0.077338	1.51E-05	0.01954
Conversion Noise Variance	0.087339	1.48E-05	0.01689
Distance	0.125	2.23E-05	0.01782
Power Splitting Factor	0.061559	1.17E-05	0.01895
Energy Conversion Efficiency	0.053423	1.07E-05	0.01998

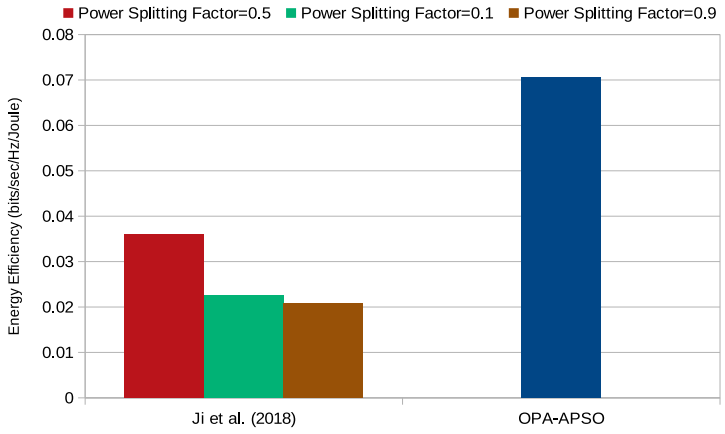


Fig. 13. Optimized Energy efficiency of OPA-APSO and (Ji *et al.*, 2018) at various ρ_h values

5.4 Comparison of OPA-APSO with existing approaches

To show the efficacy of the proposed approach, we compare OPA-APSO with already existing relaying techniques (Nasir *et al.*, 2013; Ji *et al.*, 2018; Do, 2019) for energy harvesting. Table 3 summarizes above discussed approaches w.r.t. to parameters, objective and method used to achieve optimal results.

Figure 13 shows the comparison of optimized energy efficiency between OPA-APSO and (Ji *et al.*, 2018). Optimized energy efficiency values of (Ji *et al.*, 2018) are shown for three different values of ρ_h 0.1, 0.5 and 0.9 as shown in Figure 13. OPA-APSO achieves 96% higher efficiency than (Ji *et al.*, 2018).

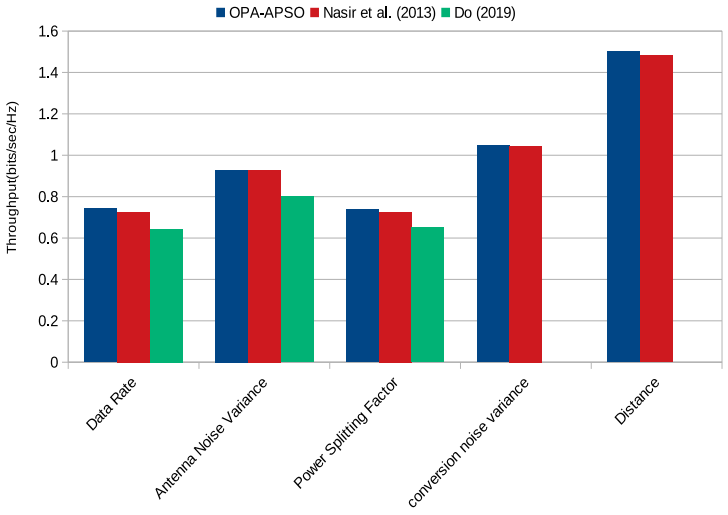


Fig. 14. Comparison of optimized throughput with existing approaches at various parameters

Figure 14 presents a comparison between throughput of the considered IoT system by using OPA-APSO and approaches used in (Do, 2019) and (Nasir *et al.*, 2013) and it is observed that OPA-APSO gives better results than these approaches for optimal value of power-splitting factor, data rate, antenna and conversion noise variance, and distance respectively. The results show that there is a considerable improvement in the throughput using the OPA-APSO algorithm to find out the optimal transmission power. Throughput is enhanced by 50% and 35% over approaches (Do, 2019) and (Nasir *et al.*, 2013) respectively.

Table 3. Comparison of proposed approach with existing approaches

Author	(Nasir *et al.*, 2013)	(Ji *et al.*, 2018)	(Do, 2019)	**OPA-APSO**
System	Dual-Hop	Dual-Hop	Dual-Hop	Dual-Hop
Type	Amplify-and-Forward	Amplify-and-Forward	Amplify-and-Forward	Amplify-and-Forward
Technique	Numerical Analysis	Lagrangian multiplier method	Monte Carlo Method	Adaptive PSO
Objective	Throughput	Energy Efficiency	Throughput	Energy Efficiency
Parameter	Power-Splitting Factor	Transmitted Power	Power-Splitting Factor	Transmitted Power
Throughput	0.724	-	0.65	**0.98955**
Energy Efficiency	No	0.036	No	**0.070682**
Considering Amount of Energy Harvested	No	No	No	**Yes**

6. Conclusions and future directions

In this article, we have studied the EH enabled cooperative communication network for IoT devices. Relay employs PSR to harvest the energy and process the information in the amplify-and-forward IoT network. Our main motive is to optimize the system's energy efficiency. For this, we present the expressions for the outage probability and energy efficiency for delay limited transmission mode under quasi-static block fading. Also, we investigate the impact of $P_s, R, \eta, \rho_h, d_{sr}, d_{rd}, \sigma_c^2, \sigma_a^2$ on energy efficiency individually. Numerical results reveal how these parameters affect energy efficiency and drive us to optimize the parameters to obtain the maximized energy efficiency. Further, we formulate the optimization problem for achievable energy efficiency at the destination, simultaneously considering the amount of energy harvested by the relay. In order to solve the optimization problem, we have proposed

a meta-heuristic based OPA-APSO algorithm to achieve the maximized energy efficiency. The proposed approach also gives the best value of the amount of harvested energy by the relay node for the achieved energy efficiency. Results show the efficacy of OPA-APSO over the existing schemes. Further, statistical analysis is performed which shows the stability of the algorithm. In the future, it would be interesting to optimize other important factors along with energy efficiency as multi-objective optimization problem.

References

Ashraf, N., Sheikh, S. A., Khan, S. A., Shayea, I. & **Jalal, M. (2021).** Simultaneous wireless information and power transfer with cooperative relaying for next-generation wireless networks: a review. IEEE Access.

Chen, G. C. & **Yu, J. S. (2005).** Particle swarm optimization algorithm. Information Control-Shenyang 34(3),p.318.

Chen, Z., Xia, B., Liu, H. & **Jiao, S. (2014).** Wireless information and power transfer in two-way amplify-and-forward relaying channels. IEEE Global Conference on Signal and Information Processing (GlobalSIP) (pp. 168–172).

Devi, R. P. & **Prabakaran, N. (2021).** Hybrid cuckoo search with salp swarm optimization for spectral and energy efficiency maximization in NOMA system. Wireless Personal Communications, 1–23.

Do, D. T. (2015). Time power switching based relaying protocol in energy harvesting mobile node: optimal throughput analysis. Mobile Information Systems 2015.

Do, D. T. (2019). Optimal energy harvesting strategy in relaying networks : dynamic allocation scheme and performance analysis. Wireless Personal Communications 108(2), 1097–1111.

Do, T. P., Song, I. & **Kim, Y. H. (2017).** Simultaneous wireless transfer of power and information in a decode-and-forward two-way relaying network. IEEE Transactions on Wireless Communications 16(3), 1579–1592.

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. & **Cosar, A. (2019).** A survey on new generation metaheuristic algorithms. Computers & Industrial Engineering 137, p.106040.

Gradshteyn, I. S. & **Ryzhik, I. M. (2014).** Table of integrals, series, and products, Academic press.

Gupta, P., Dimri, P. & **Saroha, G. P. (2021).** Performance evaluation of an adopted model based on big-bang big-crunch and artificial neural network for cloud applications. Kuwait Journal of Science 48(4).

Gurjar, D. S., Nguyen, H. H. & **Tuan, H. D. (2018).** Wireless information and power transfer for IoT applications in overlay cognitive radio networks. IEEE Internet of Things Journal 6(2),3257–3270.

Huang, H., Xia, J., Liu, X., Na, Z., Yang, Q. & **Chen, H. (2018).** Switch-and-stay combining for energy harvesting relaying systems. Physical Communication journal 28, 28–34.

Hussain, K., Salleh, M. N. M., Cheng, S. & **Shi, Y. (2019).** Metaheuristic research: a comprehensive survey. Artificial Intelligence Review 52(4), 2191–2233.

Ji, B., Song, K., Li, C., Zhu, W.-p. & **Yang, L. (2018).** Energy harvest and information transmission design in internet-of-things wireless communication systems. AEU-International Journal of Electronics and Communications 87, 124–127.

Liu, X. & **Ansari, N. (2017).** Green relay assisted d2d communications with dual batteries in heterogeneous cellular networks for IoT. IEEE Internet of Things Journal 4(5), 1707–1715.

Lv, T., Lin, Z., Huang, P. & **Zeng, J. (2018).** Optimization of the energy-efficient relay-based massive IoT network. IEEE Internet of Things Journal 5(4), 3043–3058.

Mortazavi, A. and Ahmadi, M. (2019). Applying an optimized low risk model for fast history matching in giant oil reservoir. Kuwait Journal of Science 46(1).

Nasir, A. A., Zhou, X., Durrani, S. & **Kennedy, R. A. (2013).** Relaying protocols for wireless energy harvesting and information processing. IEEE Transactions on Wireless Communications 12(7), 3622–3636.

Nasir, A. A., Zhou, X., Durrani, S. & **Kennedy, R. A. (2014).** Throughput and ergodic capacity of wireless energy harvesting based DF relaying network. In 2014 IEEE International Conference on Communications (ICC)(pp. 4066–4071).

Omoniwa, B., Hussain, R., Adil, M., Shakeel, A., Tahir, A. K., Hasan, Q. U. & **Malik, S. A. (2018).** An optimal relay scheme for outage minimization in fog-based internet-of-things (iot) networks. IEEE Internet of Things Journal 6(2), 3044–3054.

Poli, R., Kennedy, J. & **Blackwell, T. (2007).** Particle swarm optimization. Swarm intelligence 1(1), 33–57.

Rao, A. N., Naik, B. R. & **Devi, L. N. (2020).** On the relay node placement in WSNs for lifetime maximization through metaheuristics. Materials Today: Proceedings.

Rauniyar, A., Engelstad, P. E. & **Østerb, O. N. (2018).** RF energy harvesting and information transmission based on power splitting and noma for IoT relay systems. In 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), IEEE, (pp. 1–8).

Rauniyar, A., Engelstad, P. E. & **Østerbø. O. N. (2019).** Performance analysis of RF energy harvesting and information transmission based on noma with interfering signal for IoT relay systems. IEEE Sensors Journal 19(17), 7668–7682.

Rekha & **Garg, R. (2018).** Energy management in wireless sensor networks: A state of art. In International Conference on Intelligent Data Communication Technologies and Internet of Things, Springer (pp. 492–499).

Shah, S. T., Munir, D. & **Chung, M. Y. (2016).** Information processing and wireless energy harvesting in two-way amplify-and-forward relay networks. IEEE 83rd Vehicular Technology Conference (VTC Spring) (pp. 0–4).

Shaikh, F. K. & **Zeadally, S. (2016).** Energy harvesting in wireless sensor networks: A comprehensive review. Renewable and Sustainable Energy Reviews 55, 1041–1054.

Shi. Y. & **Eberhart. R. (1998).** A modified particle swarm optimizer. In 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360) (pp. 69–73).IEEE.

Srivastava, L. (2006). Pervasive, ambient, ubiquitous: the magic of radio. European Commission Conference "From RFID to the Internet of Things", Bruxelles, Belgium.

Tang, K., Shi, R. & **Dong, J. (2018).** Throughput analysis of cognitive wireless acoustic sensor networks with energy harvesting. Future Generation Computer Systems 86, 1218–1227.

Varshney, L. R. (2008). Transporting information and energy simultaneously. IEEE International Symposium on Information Theory-Proceedings (pp. 1612–1616).

Yan, J. & **Liu, Y. (2017).** A dynamic swipt approach for cooperative cognitive radio networks. IEEE Transactions on Vehicular Technology 66(12), 11122–11136.

Yan, Z., Chen, S., Zhang, X. & **Liu, H.L. (2018).** Outage performance analysis of wireless energy harvesting relay-assisted random underlay cognitive networks. IEEE Internet of Things Journal 5(4), 2691–2699.

Zhou, X., Zhang, R. & **Ho, C. K. (2013).** Wireless information and power transfer: architecture design and rate-energy tradeoff. IEEE Transactions on communications 61(11), 4754–4767.

Zou, Y., Zhu, J. & **Jiang, X. (2019).** Joint power splitting and relay selection in energy-harvesting communications for IoT networks. IEEE Internet of Things Journal 7(1), 584–597.

Submitted: 05/10/2021
Revised: 16/12/2021
Accepted: 20/12/2021
DOI: 10.48129/kjs.16583

QoS based congestion evasion clustering framework of wireless sensor networks

Soumyabrata Saha[1,*], Rituparna Chaki[2]

[1]*Dept. of Information Technology, JIS College of Engineering, West Bengal, India*

[2]*A. K. Choudhury School of Information Technology, University of Calcutta, West Bengal, India*

[*] *Corresponding author: som.brata@gmail.com*

Abstract

Congestion is a significant issue for event-based applications due to the continuous data collection and transmission by the sensors constituting the network. The congestion control technique monitors the process of adjusting the data and intends to manage the network traffic level to the threshold value. The information gathered from an intensive study is required to strengthen the knowledge base for devising a QoS based congestion evasion clustering framework of wireless sensor networks. In this scheme, the cluster heads are optimally determined and dispersed over the network. The data aggregation approach has been applied in a clustered network and set out a crucial paradigm for WSN routing. The proposal employs to mitigate congestion while messages are being forwarded via an alternate route to distribute the traffic and increase the throughput. This technique aims to balance the energy ingestion among the sensor nodes, reduce energy consumption, improve network lifetime, and achieve the quality of services. The result analysis revealed that the proposed scheme recommends 22.5% better throughput, 21% lesser end-to-end delay, 25.5% better delivery ratio, and efficiently relieves congestion while preserving the network's performance for attaining QoS in wireless sensor networks.

Keywords: Clustering; congestion control; data aggregation; quality of services; wireless sensor networks.

1. Introduction

Wireless sensor networks comprise profuse sensor nodes to create an ad hoc distributed data proliferation network that collects context information about the physical environment (Shahraki *et al.,* 2020). Routing (Zear *et al.,* 2021; Saha *et al.,* 2021) would not be an intricate calculation and can acclimate to dynamic topology changes, ensuring consistent energy indulgence across a network while also helping to accomplish the quality of services. The multipath routing strategy is extensively utilized in WSN to increase network performance by efficiently using the available network resources. Clustering (Ali *et al.,* 2020) is a network management technique for designing hierarchical structures that are both scalable and resilient. Hierarchical routing employs multi-hop

communication among the network nodes in a particular region and performs data aggregation to reduce the total delivered messages to the sink node to maintain energy consumption effectively.

Congestion (Pandey *et al.,* 2020) is one of the predominant snags due to the restricted resources for data processing, communication capacity, and energy supply. Sensor nodes near the sink node are more susceptible to node-level congestion where packet loss is encountered and affects the network's lifetime. Multiple sensor nodes attempt to access the transmission medium concurrently in link-level congestion. In order to achieve QoS, end-to-end congestion control adjusts the traffic rate of source and intermediary nodes. WSN applications have their specific QoS (Kaur *et al.,* 2019) requirements and are categorized as; network-specific QoS and application-specific QoS. Due to diverse traffic flows, changing network conditions, and the resource-constricted sensor nodes, accomplishing the quality-of-service requirements of several applications remnants a hard challenge for routing protocols. Several sensors in each location will acquire numerous redundant data due to the random distribution of network nodes. Route discovery in a flat network is made by flooding, where duplicate messages expand network load and necessitate additional bandwidth.

To solve the problem, we propose a QoS based congestion evasion clustering framework for sensor networks to enrich the network performance.

The followings are the main contributions of the proposed framework:

- We have introduced cluster formation mechanism, where dynamic cluster head selection process ensures even dissemination of energy among the sensor nodes to ensure that no nodes would run out of energy. The maximum number of cluster members is restrained during cluster formation to balance the energy consumption and create routing trees where cluster heads appear as the child node of the tree.
- We have proposed cluster member level and cluster head level data aggregation strategies to assure distinct data delivery to sink node.
- We have forged the node level congestion mitigation technique for priority and regular data where sensor nodes would be aware of the congestion level of the upstream or downstream neighbour nodes before forwarding the data packets.
- In this proposal, the message forwarding has been carried out via multipath routing, which is crucial for maintaining alternate routes, distributing traffic loads, and increasing throughput.

Extensive simulation shows that our proposed framework outperforms other existing protocols and achieves better network lifetime, energy efficiency, and accomplishes the quality of services.

The rest of the paper is delineated as follows. Section 2 attempts to introduce a holistic view of the state-of-the-art congestion control technique along with the hierarchical cluster-based routing. A comprehensive study of QoS mechanisms is offered here. In section 3, we have proposed a QoS based congestion evasion clustering framework of wireless sensor networks. The simulation in section 4 reveals that the proposed technique outperforms than other existing algorithms. This paper has been concluded in section 5.

2. Related Works

This section includes a comprehensive fine-grained survey on the distinct routing protocols of WSN. Several well-known clustering algorithms have been studied to recognize the pros and cons of those proposals for designing the novel hierarchical clustering routing.

The LEACH (Heinzelman *et al.,* 2000) protocol employs a cluster-based hierarchical architecture with random cluster head rotation to disperse the energy load across the sensor nodes but is inappropriate for large networks and cannot confirm load balancing. The data aggregation in EELEACH (Arumugam *et al.,* 2015) impedes a significant amount of energy while routing is implemented based on adequate data collection and optimum clustering. In CDAS (Devi *et al.,* 2020), latency and packet loss reduction lessen the overhead and end-to-end delay while improving energy utilization and network lifetime. In (Khediri *et al.,* 2020), intra-cluster communication employs single hop; in contrast, inter-cluster communication manages multi-hop communication mode and achieves energy utilization. Although the network lifetime is the most significant concern (Han *et al.,* 2020), offline parameter optimization has a high-level complexity, creates computational overhead, and does not concern multi-hop communication. EASS (Khan *et al.,* 2020) defines different states depending on the sensor node's internal elements and aligns them based on the contents of data packets and the incidence of produced traffic. In (Salim *et al.,* 2021), cluster heads are designated based on the continuing energy and distance between the cluster heads and confirms fault tolerance level. In (Behera *et al.,* 2021) presented an adaptive, resilient cluster head selection where the threshold value of CH election is adjusted based on enduring energy and the optimum number of clusters. Brainstorm optimization with levy distribution-based clustering was proposed in (Cho *et al.,* 2021), whereas data aggregation approaches for curtailing energy intemperance are not considered. In Q-DAEER (Yoo *et al.,* 2021), a data aggregation method is utilized to compute the optimum path to extend the network's lifespan while minimizing energy utilization. Priority would be calculated using the priority function in CPMEA (Ranga *et al.,*2016), and accordingly, actors would be chosen. The major objective is choosing the smallest number of actors or the smallest overlap between their respective positions. In (Adhikary *et al.,* 2021), the clustering scheme achieves load distribution and ensures energy efficient route discovery, but this proposal does not consider data aggregation mechanism. The preceding study shows that the choice of cluster heads is a crucial issue in hierarchical cluster routing. Incredibly, the construction of clusters and the rotation of the cluster head have a substantial effect on the entire network's performance.

To identify the congestion-related parameters, we have studied a variety of congestion control mechanisms to weigh the benefits and drawbacks of those proposals. In (Bhandari *et al.,* 2018), a multi-criteria decision-making method and different routing metrics are used to identify the optimum substitute parent node that is used to alleviate the congestion. In (Singh *et al.,* 2018), the proposal uses a multi-objective optimization strategy to limit the arrival rate depending on priority by allowing priority-based communication. The authors (Farsi *et al.,* 2019), proposed congestion-aware clustering routing to reduce end-to-end delay and extend the network's lifetime by selecting the primary and secondary cluster head. Authors (Srivastava *et al.,* 2019), devised an

algorithm that lowers the total end-to-end delay while increasing network endurance using the firefly optimization technique. The alternate hop selection method (Adil *et al.,* 2021) diverts sensor communication to the neighbors and regulates network traffic in a congested environment while also extending the network lifetime.

The aforesaid study identifies that the network performance has been affected due to the congestion. Congestion evasion methods should be implemented to regulate the network traffic when there is likely to be transitory congestion.

QoS mechanisms have been put through a thorough analysis that highlights the performance issues, which would help design the proposed proposal. In (Deepa *et al.,* 2020), an alternative path was dynamically selected, reducing transmission latency and communication overhead to save energy consumption and improve load balancing. The clustering technique (Faheem *et al.,* 2018) consolidates sensor nodes into a linked hierarchy for energy and traffic load distribution within the network that shrinks data route loops and network latency. Clustering, duty cycling, and collaborative communication combine in ECO-LEACH (Bahbahani *et al.,* 2018) to achieve improved energy efficiency and energy-neutral operation across several layers of the system architecture. EADCR (Panchal *et al.,* 2020) employs the residual energy, Euclidean distance, and cluster centroid as crucial factors in extending network lifespan. Efficient and secure path inference with the lowest latency and optimal bandwidth use are significant aspects of the proposed method (Alghamdi *et al.,* 2021) that improve network performance. The hybrid protocol (Sharma *et al.,* 2021) was devised for diverse networks and executed based on the multi-objective optimization approach for rate optimization and governing the data transfer rate from child to parent node. It's been revealed that uneven traffic load allocation among sensor nodes might lead to sensor node energy depletion quicker than expected. In QoS protocol, energy utilization should be distributed equally across the sensor nodes along the path to the sink node.

Table 1. Comparison of Routing Protocols

Advantage	Disadvantage
It is a low complexity algorithm that reduces control messages overhead	Uniform distribution of cluster heads are not offered
Performs better than LEACH	More complex, lacks integrity of data and scalability scope
Avoids unnecessary retransmissions, waiting	Starvation may occur for low priority data
Achieves uniform distribution in spatial domain of cluster head	For lifetime measurement authors measured life time of node only
Prolong the network lifetime and improve network throughput	Needs to enhance multi-hop inter-cluster communication
Successfully reduces data, extends network lifetime	Consist of many complex mechanism
Outperformed in terms of network lifetime, average residual energy, throughput	Higher complexity than LEACH
Outperformed in terms of energy efficiency, network lifetime, PDR, delay	Data aggregation technique is not offered
Involves security alongside malicious attacks as well as utilizes the bandwidth efficiently to improve QoS	The standard quality measurement parameters have not estimated

Advantage	Disadvantage	Protocol	Data Aggregation	Scalability	Power Usage	Network Lifetime	Multi-path	Delay
CoAR improves PRR, end-to-end delay, packet loss ratio, throughput, energy consumption	CoAR is described considering only a static network topology	LEACH (Heinzelman *et al.,*2000)	Yes	Low	High	Medium	No	Small
Achieves better performance in terms of packet loss, end-to-end delay, Queue Size, throughput, congestion level etc	Load balancing problem, security issues in WSNs have not addressed	EELEACH (Arumugam *et al.,*2015)	Yes	High	Low	High	No	Less than LEACH
It increases the network lifetime, does not suffer from data overflow, stability is achieved	In place of transmitting all data, transmits only changed data	CDAS (Devi *et al.,*2020)	Yes	Low	Low	High	No	Low
Achieve significant improvement in communication cost, computational cost, traffic congestion, throughput etc	It is not implemented in real IoT environment	OK-Means (Khediri *et al.,*2020)	No	Low	Low	High	Yes	Low
Achieves prominent data communication with reasonable energy conservation	This proposal is not deal with enhancing fault tolerance, security etc	CPMA (Han *et al.,* 2020)	No	High	Low	High	No	Low
Achieves better network performance	Node position and mobility would include	QDAEER (Yoo *et al.,*2021)	Yes	Low	Low	High	No	Low
Achieves the efficiency in terms of throughput and network lifetime metrics	The proposed approach assumes a static network	F-LEACH, (Behera *et al.,*2021)	Yes	High	Low	High	No	High
Provides better results in terms of a lifetime, residual energy, and coverage of the network	The standard quality measurements have not estimated	HMBCR (Cho *et al.,*2021)	No	Low	Low	High	No	Low
It is well suited to design WSN in real-world and real-time situations	The adaptive ability is not tested	EQRP (Alghamdi *et al.,*2021)	No	Low	Average	Average	No	Low

Protocol	Data Aggregation	Scalability	Power Usage	Network Lifetime	Multi-path	Delay
CoAR (Bhandari *et al.*,2018)	No	High	Low	High	Yes	Low
PSOGSA (Singh *et al.*,2018)	No	High	Low	High	Yes	Low
CCR (Farsi *et al.*,2019)	Yes	High	Medium	High	No	Low
DHSSRP (Adil *et al.*,2021)	No	Low	Low	High	Yes	Low
OQoSCMRP (Deepa *et al.*,2020)	No	Medium	Low	High	Yes	High
ECO-LEACH (Bahbahani *et al.*,2018)	No	Low	Low	High	No	Low
CMEEBZ (Adhikary *et al.*,2021)	No	Low	Low	High	Yes	Low
EADCR (Panchal *et al.*,2020)	Yes	Low	Low	High	Yes	Medium
QBEEP (Sharma *et al.*,2021)	Yes	Low	Low	High	Yes	Low

During the above study following limitations have been identified. It has been observed that most of the researchers have concentrated on the cluster head selection and cluster formation process, but very few proposals are associated to the restriction of the maximum number of cluster members has been discussed. Rather than concentrating on both cluster member and cluster head level aggregation, maximum authors concentrated on cluster head level aggregation. There has not been any precise proposal put out to alleviate the congestion for the priority data. It is unlikely that less attention is paid to reduce bottleneck conditions of the hierarchical cluster routing tree. In light of data aggregation and congestion mitigation, no specific solution has been noticed to attain the quality of services. To overcome the above concerns, we have proposed a novel QoS based congestion evasion clustering framework of WSN that optimizes energy management and achieves the quality of services.

3. Proposed framework: QoS based congestion evasion clustering framework of WSN

The previous section reveals a wide range of congestion control mechanisms and found that congestion significantly impacts the overall network performance of WSN. According to the findings of the study, cluster head selection and proper cluster formation have a considerable influence on network performance. Before sending data to the sink node, data aggregation is recommended to minimize the number of messages delivered to the node. Energy efficiency is often recognized as a significant design consideration to solve the inadequacies of the previously outlined approaches. It is a challenge to design a new framework that can fulfill all these objectives while still being as simple to implement as possible. We present a QoS based congestion evasion clustering framework of wireless sensor networks to optimize energy efficiency and improve network performance to achieve the quality of services.

The proposed framework consists of five modules. Module 3.1 introduces the cluster head selection process, whereas Module 3.2 depicts the cluster formation technique. Module 3.3 represents an aggregation technique. Module 3.4 discusses a method for congestion mitigation. Module 3.5 implements an alternate path creation technique to carry out the communication operation.

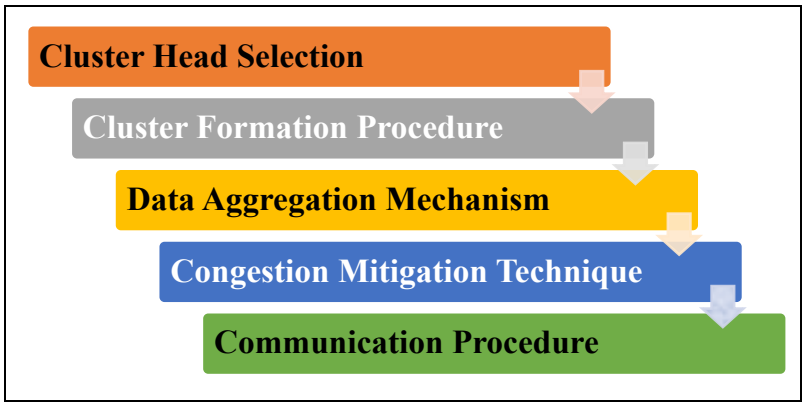


Fig. 1. System Flow of the Proposed Framework

Module 3.1: Cluster Head Selection

This module proposes a dynamic cluster head selection mechanism where node-specific information is deemed for cluster head selection. This process is initiated and monitored by sink node. The current cluster head will be substituted by the new cluster head when the energy level drops lower than the threshold value. A balanced energy distribution among the sensor nodes is confirmed by rotating the cluster head, guaranteeing that none of the nodes run out of power owing to their responsibilities. Each sensor node would find its maximum number of neighbors within a single hop distance.

The degree difference (Δns_i) for every node is:

$$\Delta ns_i = \sum_{s_i \in N(S)} (d_{s_i} - d_{s_j}),\ \left[\text{where } s_i \neq s_j \text{ and } \{ dist(s_i, s_j) \leq t_r \}\right] \quad (1)$$

For each network node, the average distance among the neighbors is:

$\Delta ads_i = \frac{1}{n}\left[\sum_{s_j \in N(S)} dist(s_i, s_j)\right]$, where n ≥ 1 and $dist(s_i, s_j)$ is the euclidean distance between node s_i and node s_j. (2)

The distance between the sensor node and the sink node is: $\Delta snds_i = dist(SN, s_i)$

The minimum distance with the sink node is: $[min|\Delta snds_i|\} = \min\{dis(SN, s_i)\} \forall\, s_i \in S = [\sum(SN - s_i)^2 |\ \forall\, s_i \in S\ ]$ (3)

The maximum distance with the sink node is: $[max|\Delta snds_i|\} = \max\{dis(SN, s_i)\} \forall\, s_i \in S = [\sum(SN - s_i)^2 |\ \forall\, s_i \in S\ ]$ (4)

After a specific time interval, compute the energy ratio of each node and update the ND_ENGY_TBL $\{s_i, eins_i, ers_i\ erts_i, t_n\}$ table. Depends on the initial energy and residual energy, the energy ratio $(erts_i)$ is calculated as:

$$erts_i = \left(\frac{ers_i}{eins_i}\right) \quad (5)$$

The tier id $(tids_i)$ of each node is calculated based on the energy ratio and the distance between the sensor node and sink node:

$$tids_i = \lceil\left(\frac{erts_i}{\Delta snds_i}\right)\rceil \quad (6)$$

Based on the initial energy, residual energy, distance between the sensor node and sink node, calculate the node priority (ps_i):

$$ps_i = \{a * \left(\frac{ers_i}{eins_i}\right) + b * \left(1 - \frac{\Delta snds_i - \min\ |\Delta snds_i|}{\max\ |\Delta snds_i| - \min\ |\Delta snds_i|}\right)\}, where\{[0 \le (a+b) \le 1] \quad (7)$$

Calculate node state (s_{ste}):

$s_{ste} = f\{ps_i, flg\}$, set flg=0.25, iff, s_i already executed as cluster head, otherwise set flg=0.75 (8)

Evaluate the cluster coefficient for each node by using the equation;

$cfs_i = \{\prod_{i=1}^{6}(x_i)^{cf_i}\}$, $where\ \sum_{i=1}^{6} cf_i = 1$, $[0 < cf_i < 1]$ and $[x_1 = ps_i, x_{2=}\ erts_i, x_3 = \Delta ads_i, x_4 = \Delta ns_i,\ x_5 = s_{ste},\ x_6 = tids_i]$ (9)

The node with the highest cluster coefficient value would select as cluster head.

Algorithm: Cluster Head Selection

Input: Node information

Output: Selection of cluster head

Begin

For each network node (s_i)

Repeat

Step 1: Identify the degree of connectivity (ds_i)

Step 2: Degree difference (Δns_i) is populated using equation (1)
Step 3: Compute the average distance (Δads_i) using equation (2)
Step 4: Calculate the minimum [min $|\Delta snds_i|$] and maximum distance [max $|\Delta snds_i|$] with the sink node using equation (3) and (4)
Step 5: Calculate energy ratio ($erts_i$) using equation (5)
Step 6: Calculate tier id (tid_{s_i}) using equation (6)
Step 7: Compute node priority (ps_i) using equation (7)
Step 8: Calculate node state (s_{ste}) using equation (8)
Step 9: Evaluate cluster coefficient (cfs_i) using equation (9)
Step 10: Find max$|cfs_i|$ and corresponding node select as cluster head (ch_i)
Step 11: **If** $er_{chi} < er_{th}$
Step 12: Then repeat Step1 to Step 10 to select a new cluster head
Step 13: **Else**
Step 14: Continue with the current cluster head
Step 15: **End if**
End

Module 3.2: Cluster Formation Procedure

In the first phase, cluster members are connected to the cluster head through MAX_HEAP technique, wherein the second phase, cluster heads connect to neighbor cluster heads through $dARY\_HEAP$ topology. We presume that the sink node acts as the root, where cluster heads act as the child node of the constructed tree. The load balancing mechanism can distribute the network nodes among different clusters by impeding the maximum cluster members in a cluster.

The communication cost is estimated as: $comm_{cost} = \frac{intrach_{dist}}{ChSN_{dist}}$, where, (10)
$intrach_{dist} = dist(ch_i, s_i)$ and $ChSN_{dist} = dist(SN, ch_i)$

The node rank is calculated as: $rnk_{si} = \frac{erch_i}{dist(ch_i,s_i)*ers_i}$ (11)

$$chjoin_{si} = \{\alpha_1 * erts_i + \alpha_2 * (1 - \frac{\sum_{i=1}^{2} \beta_i * p_i}{rnk_{si}}) + \alpha_3 * bfravs_i\} \quad (12)$$

ch_i broadcasts the $CH\_ADV\_MSG\{ch_i, msg_{id,} cfch_i, erch_i, ttl\}$ and receives the $CM\_RPLY\_MSG\ \{ch_i, s_i, chjoin_{si}, msg_{id}, ers_i,\ ttl\}$ from neighbour nodes and store $NH\_TBL\ [s_i, chjoin_{si}, msg_{id}, ttl]$ table. Based on $chjoin_{si}, MAX\_HEAP$ is constructed where ch_i act as the root of the corresponding cluster. By sending the $CLM\_CNF\_MSG\{ch_i, cfch_i, erch_i, cm_j, pcm_j,\ ttl\}$, ch_i confirms cluster membership to cm_j. Maximum number of nodes belong to cluster $\leq (2^{h+1} - 1)$, $[where\ h = level\ of\ ch_i]$

Case 1. If s_i receives only one message $CH\_ADV\_MSG\{ch_i, msg_{id}, cfch_i, erch_i, ttl\}$ from ch_i, then it would send the $CM\_RPLY\_MSG\ \{ch_i, s_i, chjoin_{si}, msg_{id}, ers_i,\ ttl\}$ to join in the corresponding ch_i.

Case 2. If s_i receives two or more CH_ADV_MSG $\{ch_i, msg_{id}, cfch_i, erch_i, ttl\}$ from the different ch_i, then based on the equation (12) it would send the $CM\_RPLY\_MSG\{ch_i, s_i, chjoin_{si}, msg_{id}, ers_i, ttl\}$ to the particular ch_i and would want to become a cluster member of the stated cluster.

In the second phase, we assume that sink acts as root node at level 0. ch_i adds itself as the child of the sink node and sets its level to 1 when it has its place within the transmission range of the sink node. The remaining cluster heads in the network use the same technique, and a tree formation is carried out.

Algorithm: Cluster Formation Procedure

Input: Cluster head details
Output: Cluster formation
Begin
For each network node, **do**
Step 1: ch_i Broadcast $CH\_ADV\_MSG\{\}$
Step 2: **If** ((isClusterHead) || (isExistingClusterMember)) received $CH\_ADV\_MSG\{\}$
Step 3: Then discards $CH\_ADV\_MSG\{\}$
Step 4: **End If**
Step 5: **If** (isSingleClusterHead sends $CH\_ADV\_MSG\{\}$) **then**
Step 6: s_i receives $CH\_ADV\_MSG\{\}$ from one ch_i
Step 7: s_i calculates $chjoin_{si}$ by using Equation (12)
Step 8: s_i reply $CM\_RPLY\_MSG\{\}$ to corresponding ch_i
Step 9: ch_i maintains NH_TBL[$s_i, chjoin_{si}, msg_{id}$, ttl]
Step 10: CM_HEAP ( )
Step 11: ch_i sends $CLR\_FRM\_MSG\{\}$ and confirms the membership to cm_j
Step 12: **Else**
If (isMultipleClusterHead send $CH\_ADV\_MSG\{\}$) **then**
Step 13: Repeat Step7 and send reply $CM\_RPLY\_MSG\{\}$ to ch_i having [max$|cfch_i|$]
Step 14: **End If**
Step 15: **End If**
Step 16: Level of SN←0
Step 17: $dARY\_Parent(i) = \left\lfloor \frac{i+d-2}{d} \right\rfloor$
Step 18: **For** each ch_i **do**
Step 19: **Repeat**
Step 20: Broadcast RT_MSG{}
Step 21: **For** i=1 to n do
Step 22: dARY_HEAP ( )
Step 23: **If** (($|ertch_i| > th_{er}$) && ($|ChSN_{dist}| \leq th_{dist}$)) **then**
Step 24: Reply with SNC_MSG{} to Parent Node SN
Step 25: $dARY\_Child(i,j) = [(i-1)d + j + 1]$
Step 26: **End If**
Step 27: **End For**
End

Algorithm: dARY_HEAP ( )

Step 1: MAX_HEAP (A)
Step 2: **For** i=length[A] downto 2 do
Step 3: swap(A[1] ↔ A[i])
Step 4: HeapSize[A] ← HeapSize[A]-1
Step 5: dARY_MAX_HEAP(A,1)
Step 6: **End For**
End

Algorithm: MAX_HEAP (A)

Step 1: HeapSize[A] ← length[A]
Step 2: **For** i=k down to 1 do, [where k=$\left\lfloor \frac{\text{length [A]}-2}{d} \right\rfloor$]
Step 3: dARY_MAX_HEAPIFY (A, i+1)
Step 4: **End For**
End

Algorithm: dARY_MAX_HEAPIFY (A, i)

Step 1: SN←i
Step 2: largest ← i+1
Step 3: **For** j= 1 to d do
Step 4: **If** (j<= HeapSize[A] && A[Child (i+1, j)] > A[i+1] **then**
Step 5: largest ← child (i+1, j)]
Step 6: **End If**
Step 7: **End For**
Step 8: **If** (largest! = i+1) **then**
Step 9: swap(A[i+1] ↔ A[largest])
Step 10: dARY_MAX_HEAPIFY (A, largest)
Step 11: **End If**
End

Module 3.3: Data Aggregation Mechanism

Due to the high-level node density in sensor networks, many sensor nodes sensed similar data, causing redundancy. Additional bandwidth is required for redundant data transmission that makes the network more volatile. This section introduces two-level data aggregation strategies, i.e., cluster member level and cluster head level aggregation, to forward the aggregate data to the sink node and achieve energy optimization while minimizing the number of transmissions. In query driven WSN, sensor nodes forward the aggregated data in reply to the query request of the sink node.

In order to calculate the performance of the aggregation function, aggregation ratio and packet size co-efficient (Cui *et al.,* 2014) have been considered: Aggregation ratio (w) is defined

as the ratio of the number of aggregated packets (n) and total packets generated (N), where w ∈ [0,1]. Let, s_i transmits the number of units of raw data φ(v), the number of unit-size packets forwarded denoted by δ(v) that is defined as; $\delta(v) = \lceil \frac{\varphi(v)}{w} \rceil$. Packet size co-efficient (λ) shows the change in packet size due to the aggregation function $\left[\lambda = \frac{d'_i}{d_i}\right]$, where d'_i is the size of the aggregated packet, and d_i is the size of the original packet. At $t_{i+\zeta}$ time instance sensor node collects $d_{i+\zeta}$ raw data and checks for the data similarity. According to the similarity index, the concerned cluster members would make packet forwarding decisions.

$$d_{sim}(d_i, d_{i+\zeta}) = \left[\frac{d_i \cap d_{i+\zeta}}{d_i \cup d_{i+\zeta}}\right] \quad (13)$$

In this proposal, the similarity threshold index (Δth_{indx}) is set to 0.5. If the data similarity is less than the threshold index, then sensor nodes send both data packets to the cluster head; otherwise, apply the aggregation technique on the collected data. In this framework, the aggregation cost is introduced during cluster head-level aggregation. i.e., $[aggr_{cost} = \left[\frac{w*\lambda}{rnk_{si}}\right] * d_{sim}]$. The aggregation level of each cluster head depends on the aggregation cost and energy ratio. i.e., $aggr_{level} = f(aggr_{cost}, ertch_i)$. A number of standard mathematical functions are taken into account in the development of this model.

Case 1. Sensor nodes collect the same data. The final aggregation value is: $\{d_{sm}(aggr) = \left(\frac{\alpha_{d1}}{2^{n-1}} + \frac{\alpha_{d2}}{2^{n-2}} + \frac{\alpha_{d3}}{2^{n-3}} \dots\dots + \frac{\alpha_{dk}}{2}\right)\}$ where $[\alpha_{d1} = \alpha_{d2} = \alpha_{d3} = \alpha_{dk}$ and 'n' is no of nodes.]

Case 2. Sensor nodes collect different data, i.e., The total amount of data gathered from all contributing sensors would be the final aggregate value.

$\{(d_{df}(aggr) = \left(\sum_{i=1}^{k} \beta_{di}\right)\}, where\ \beta_{d1} \neq \beta_{d2} \neq \beta_{d3} \neq \beta_{dk}]$

Case 3. Few sensor nodes collect the same data, and others collect different data, i.e., The final aggregation value is: $\{(d_{smdf}(aggr) = (\frac{¥_{d1}}{2^{q-1}} + \frac{¥_{d4}}{2^{q-2}} + \frac{¥_{d5}}{2^{q-3}} \dots\dots + \frac{¥_{dk-1}}{2}) + (¥_{d2} + ¥_{d3} + ¥_{dk})\}$

Case 4. The values collected by multiple sensor nodes for the same attribute; Maximum, Minimum, and Median value from the collected data is:

$\{(d_{mx}(aggr)\} = f(S_{1\dots}S_n) = max|S_i|, \text{where i} = 1..\text{n}$

$\{(d_{mn}(aggr))\} = f(S_{1\dots}S_n) = min|S_i|, \text{where i} = 1..\text{n}$

$\{(d_{mdn}(aggr))\} = \sum_{i=1}^{n} S_r, \text{ where r} = (\text{i}+1)/2$

Based on the query request from the sink node, sensor nodes forward the aggregated data packets to the cluster head. Depending on the aggregation level, ch_i applies aggregation mechanism on the received data from cm_j. Total data packets received by ch_i is $d(ch_i) = \sum_{j=1}^{n} d(cm_j)$. The total aggregated data received by the sink node is: $\sum_{i=1}^{m} d(ch_i) = \sum_{i=1}^{m} \sum_{j=1}^{n} d(cm_j)$

Algorithm: Data Aggregation Mechanism

Input: Collected data
Output: Aggregated data
Begin
For each network node **do**
Repeat
Each $\boldsymbol{t_{i+\zeta}}$ instance sensor node collects raw data
Step 1: cm_{ij} measures the data similarity $\{d_{sim}(d_{i,}d_{i+\zeta})\}$ using equation (13)
Step 2: **If** $\{d_{sim}(d_{i,}d_{i+\zeta})\} < \Delta\text{th}_{\text{indx}}$ **then**
Step 3: cm_{ij} sends $\{d_{sm}\,(aggr)\}$ data to ch_i
Step 4: **Else**
Step 5: ch_i broadcasts SN_Query_Msg {} to each cm_j
Step 6: Based on the query message, cm_j applies aggregation technique on the collected data and sends it to ch_i
Step 7: Case a: ${}_{a}cm_j$ sends $\{(d_{df}\,(aggr)\}$ to ch_i
Case b: cm_j sends $\{(d_{smdf}\,(aggr)\}$ to ch_i
Case c: cm_j sends $\{(d_{mx}\,(aggr)\}$ to ch_i
Case d: cm_j sends $\{(d_{mn}\,(aggr)\}$ to ch_i
Case e: cm_j sends $\{(d_{mdn}\,(aggr)\}$ to ch_i
Step 8: **End If**
Step 9: ch_i receives data from $cm_j, d(ch_i) = \sum_{j=1}^{n} d(cm_j)$
Step 10: **While** ($aggr_{level} \geq th_{level}$) **do**
Step 11: ch_i measures data similarity using Equation (14)
Step 12: **If** $\{d_{sim}(d_{i,}d_{i+\zeta})\} < \Delta\text{th}_{\text{indx}}$ **then**
Step 13: ch_i sends $\{d_{sm}\,(aggr)\}$ to next-hop neighbour
Step 14: **Else**
Step 15: ch_i repeats step 7 and forwards aggregated data to the next-hop neighbour
Step 16: **endif**
Step 17: **If** ($Next_{hop} == SN$) **then**
Step 18: ch_i sends aggregated data to SN
Step 19: Total aggregated data received by sink node is: $\sum_{i=1}^{m} d(ch_i) = \sum_{i=1}^{m}\sum_{j=1}^{n} d(cm_j)$
Step 20: **Else**
Step 21: ch_i sends aggregated data to the next upper-level neighbour cluster head
Step 22: Repeat from step 9 onwards
Step 23: **End If**
Step 24: **End While**
Step 25: **If** ($aggr_{level} < th_{level}$) **then**

Step 26: ch_i forwards the collected data to the same level neighbour cluster head, having [max|$erts_i$|].
Step 27: Repeat from step 9 onwards
Step 28: **End If**
END

Module 3.4: Congestion Mitigation Technique

We assume that during each slot σ_i, child nodes transferred data packets to their parent node. S_{LT} (S) represents the set of slots, $\alpha_\sigma(s_i)$ is the rate of data collection, $\beta_\sigma(s_i)$ denotes the rate of data reception, $\gamma_\sigma(s_i)$ signifies the rate of data forwarding during a slot $\{\sigma \in S_{LT}(S)\}$. In this framework, we have calculated the congestion scheduling ratio $(cgsrs_i)$ of node s_i. $\{cgsrs_i = \frac{cgpksrs_i}{cgshs_i}\}$, where congestion packet scheduling $(cgshs_i)$ is defined as the number of packets schedules per unit time to forward to the next hop. Congestion packet service rate $(cgpksrs_i)$ is the average rate at which packets have been forwarded to the next neighbour.
Let, D_{s_i} and U_{s_i} are the downstream and upstream neighbors of s_i. For $\forall j \in D_{S_i}$, $\forall k \in U_{S_i}$, (i, j) are downstream links of node s_i, while (k, i) are upstream links of s_i. Let $DSR_{S_iS_j}\{\forall s_i \in N, s_j \in D_{S_i}\}$ is the average downstream data rate from s_j to s_i and $USR_{S_kS_i}\{\forall s_i \in N, s_k \in U_{s_i}\}$ be the average upstream data rate from node s_i to s_k. To mitigate the congestion, s_i adjusts the packet receiving and packet forwarding rate.

$$cglvls_i = \{(cgsrs_i + \sum_{s_j \in D_{S_i}} DSR_{S_jS_i} - \sum_{s_k \in U_{S_i}} USR_{S_ik}), \forall s_i, \text{j, k,} \in N\} \quad (14)$$

Two different queues have been identified for storing the priority and regular data. Q_{Pmax} and Q_{Pmin} identifiers are used of priority data where as Q_{Rmax} and Q_{Rmin} used for regular data. When the queue length is less than the minimum threshold that ensures no congestion occurs, the congestion index is set to 0, and accordingly, the child node's transmission rate may be updated. The received data packets would be stored, i.e., $\{Q_L \leq Q_{Rmin}, Q_L \leq Q_{Pmin}, set\ Congs_{indx} = 0\}$.

When queue length is greater than a maximum threshold, significant congestion is recorded, and congestion index is assigned to 1, i.e., $\{Q_{Rmax} \leq Q_L,\ Q_{Pmax} \leq Q_L,\ set\ Congs_{indx} = 1\}$. The received data packets would be dropped, and the child node does not send the data packets to its parent node. For moderate congestion, the congestion index is set between 0 and 1 while the queue length is i.e., $\{Q_{Rmin} \leq Q_L \leq Q_{Rmax}, Q_{Pmin} \leq Q_L \leq Q_{Pmax}, set\ Congs_{indx} \in [0,1]\}$. Few packets of low priority will be discarded, while a few packets of high priority will be stored.

$$DP_i = \{\gamma_1 * ps_i + \gamma_2 * cglvls_i + \gamma_3 * hopcnt\}, \text{where } \sum_{i=1}^{3} \gamma_i = 1, [0 < \gamma_i < 1] \quad (15)$$

When DP_i exceeds a predetermined threshold, data is designated as a priority; otherwise, it is treated as regular. Received data will be put in the appropriate buffer queue based on the category and prevent to discard the data due to a lack of capacity. $PACK\{s_i, Q_L, Q_P, Q_{Pmax}, Q_R, Q_{Rmax}, Congs_{indx}, ttl\}$ would be sent to adjacent nodes when the buffer threshold value is updated. Neighbour nodes would decide for packet forwarding to the upstream node based on the $congs_{indx}$ and available buffer space**.**

Algorithm: Congestion Mitigation Technique

Input: Packet schedule rate, Packet service rate
Output: Minimize congestion
Begin
For each network node **do**
Repeat
Step 1: Calculates $cgsrs_i$
Step 2: s_i broadcast $\{cgsrs_i, buflvls_i\}$
Step 3: **If** $(cgsrs_i < cg_{th})$ **then**
Step 4: No congestion occurs
Step 5: **Else If** $(cgsrs_i > cg_{th})$ **then**
Step 6: $cgshs_i$ $greater\ than$ $cgpksrs_i$, and due to buffer overflow congestion occurs
Step 7: s_i informs to downstream child nodes
Step 8: Child nodes control the data transfer rate for (δ) time
Step 9: **Else If** $(cgsrs_i > 1)$ **then**
Step 10: $cgpksrs_i$ is greater than $cgshs_i$ and s_i adjusts the scheduling rate for (δ) time
Step 11: **End If**
Step 12: **End If**
Step 13: **End If**
Step 14: Calculate $cglvls_i$ using equation (14)
Step 15: Data categorization DP_i executed using equation (15)
Step 16: **If** $(DP_i > th)$ **then**
Step 17: Identify 'Priority' data or otherwise marked as 'Regular' data
Step 18: **End If**
Step 19: $Q_{Pmin} \leftarrow 0$ and $Q_{Rmin} \leftarrow [\lceil Q_L/2 \rceil + 1]$
Step 20: **While** $(!(Q_{Pmax} == [\lceil Q_L/2 \rceil - 1]) \,||\, (Q_{Rmax} == [Q_L - 1]))$ **do**
Step 21: **Repeat**
Step 22: Store the categorized data in the corresponding locations.
Step 23: **End While**
Step 24: **If**$(Q_{Pmax} == [\lceil Q_L/2 \rceil - 1] \,||\, Q_{Rmax} == [Q_L - 1])$ **then**
Step 25: Forward PACK {} to the neighbours
Step 26: Neighbour nodes explore the alternative path for data forwarding
Step 27: **Else**
Step 28: The data transfer process continues
Step 29: **End If**
END

Module 3.5: Communication Procedure

The proposed framework allows both intra-cluster and inter-cluster routing while consuming less energy. To prepare the traversing list, traversal strategies have been employed as; in-order, pre-order, post-order, level-order. The cluster head applies the TDMA technique to assign a transmission time slot to each member depending on the traversing list. According to the assigned slot, the member node forwards aggregated data packets at the beginning of the time slots. The cluster head receives aggregated data from cluster members, and the downstream cluster head

transmits the aggregated data packets to the upstream cluster head for delivery to the sink node via multipath routing. When the sensor node receives a PACK message from neighbour nodes, it does not send any data packets to its neighbours to avoid data loss. A new time slot would be allotted to the sensor node for data transmission to neighbours; otherwise, find the alternative neighbour cluster head through which data would be forwarded. As the sink has numerous child nodes and by using round-robin scheduling, data is transmitted to the sink through the different child nodes that minimize the bottleneck problem and manage the energy optimization.

Algorithm: Communication Procedure

Input: Network information
Output: Data transfer to sink node
Begin
For each network node **do**
Repeat
Step 1: Based on the traversing technique, formulate the traversing list $TL[\,]$
Step 2: Ch_i assigns transmission slot $TS[i]$ for each Cm_j
Step 3: Cm_jsends aggregated data to Ch_i
Step 4: Ch_i forwards aggregate data to $\{upstm(Ch_i)\}$
Step 5: **If** Ch_i receives PACK from upstream neighbour **Then**
Step 6: It doesn't send data packets to the corresponding Ch_i within $TS[i]$
Step 7: Allocate new $TS[i+1]$ slot for data transfer
Step 8: Select new Ch_i based on $[f\{(\max|cfch_i|), (!\,(chld\_upstm(Ch_i)))\}]$
Step 9: Forwards the data to the new next-hop neighbour Ch_i
Step 10: **End If**
Step 11: **If** multiple neighbour cluster heads have the same metric **then**
Step 12: Data packets would forward to the upstream node using round robin mechanism
Step 13: Repeat from step4 onwards unless the data is reached to sink node
Step 14: **End If**
END

Table 2. Data Dictionary

Parameter	Details		Parameter	Details
s_i	Sensor node		ch_i	Cluster ead
$eins_i$	Initial energy of s_i		cm_j	Cluster member
$eavgs_i$	Average energy of s_i		er_{chi}	Residual energy of cluster head
ps_i	Priority of Node s_i		rnk_{si}	Rank of the node s_i
SN	Sink node		t_n	Time instance
t_r	Transmission range		er_{th}	Threshold energy
s_{iloc}	Location of s_i		cf_i	Coefficient factor
msg_{id}	Message id		cfs_i	Cluster coefficient of node s_i
ttl	Time to leave		$comm_{cost}$	Communication cost
$clstr_{cmpct}$	Cluster compactness		$enrg_{cost}$	Energy cost
$intrach_{dist}$	Intra cluster distance		PACK	Positive acknowledgment

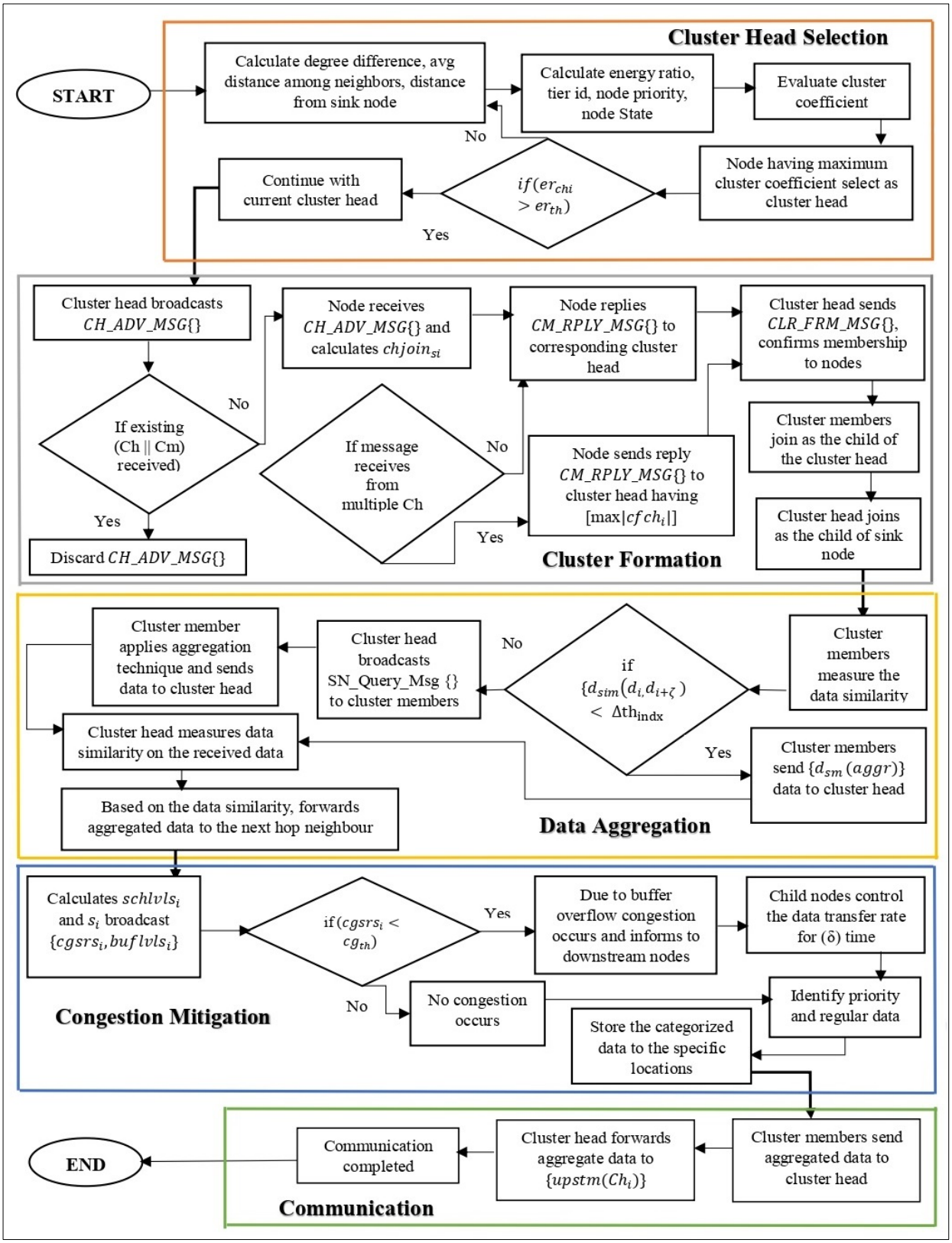


Fig. 2. Working Flow of the Proposed Framework

#Case Study: Example Network #

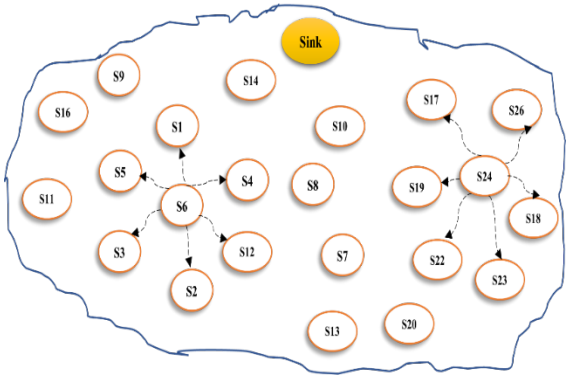


Fig. 3(a). Cluster Head Selection

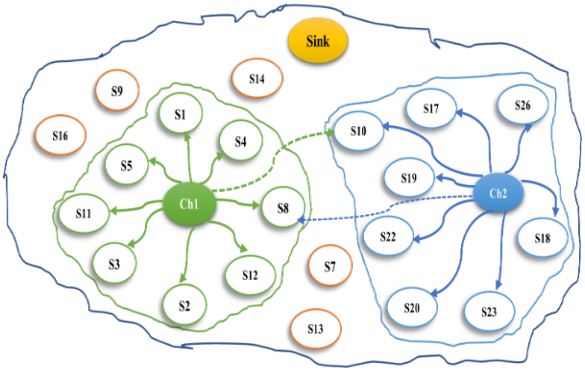


Fig. 3(b). Cluster Formation

- In Fig.3(a), Each participating node evaluates cluster coefficient. Sensor node (S6) having the maximum cluster coefficient and select as cluster head (Ch1).
- In Fig.3(b), Ch1 broadcasts $CH\_ADV\_MSG\{\}$ and neighbour nodes received the $CH\_ADV\_MSG\{\}$, calculate $chjoin_{si}$.
- In Fig.3(b), S1, S4, S8, S12, S2, S3, S11, S5 nodes reply $CM\_RPLY\_MSG\{\}$ to corresponding cluster head (Ch1)
- In Fig.3(b), Ch1 sends $CLR\_FRM\_MSG\{\}$ and confirms the membership to these nodes and they would act as the cluster member of the said cluster.
- In Fig.3(b), The same process is applicable for other cluster, where Ch2 acts as cluster head and S26, S18, S23, S20, S22, S19, S10, S17 nodes are selected as the cluster member of the said cluster.
- In Fig.3(b), S10 receives the $CH\_ADV\_MSG\{\}$ from Ch1 and truncates the message as it is already connected with Ch2. The similar process is applicable for S8 also, as this node is already the member of Ch1.

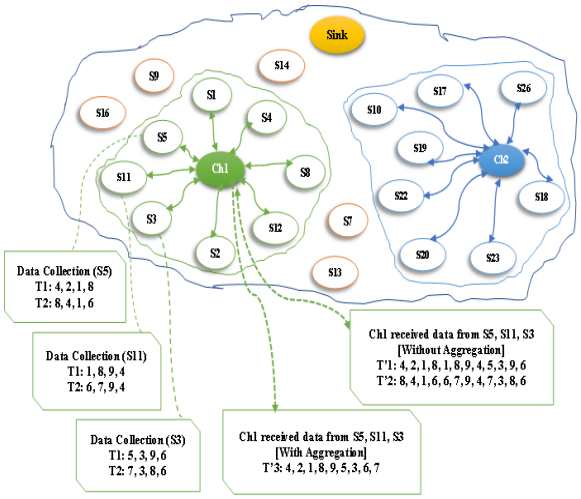


Fig. 3(c). Data Aggregation

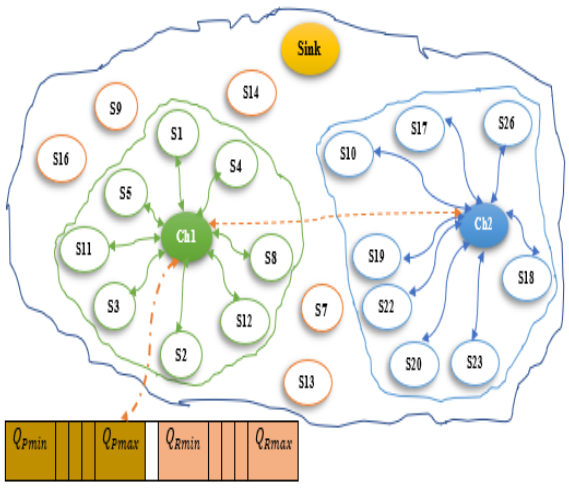
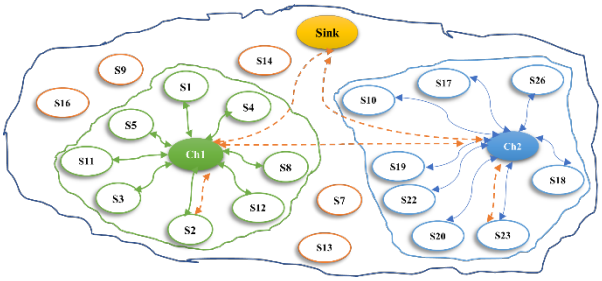


Fig. 3(d). Congestion Mitigation

- In Fig.3(c), [Without Aggregation Mechanism]: Cluster members (S3, S5, S11) collect data T1 and T2 time instance where few data are redundant, and others are distinct. The said cluster members send the collected raw data to Ch1. Cluster head received the redundant data along with distinct data from its cluster members.
- In Fig.3(c), [Considering Aggregation Mechanism]: Cluster members applied aggregation mechanism on the collected data and send the aggregated data to the cluster head. Ch1 applies aggregation mechanism on the received data from cluster members and forwards to next hop.
- In Fig.3(d), Q_{Pmax}, Q_{Pmin} are used of priority data and Q_{Rmax}, Q_{Rmin} are used for regular data. when $\{Q_L \leq Q_{Rmin}, Q_L \leq Q_{Pmin}\}$, it identifies that ensures no congestion occurs, $set\ Congs_{indx} = 0$. When $\{Q_{Rmax} \leq Q_L,\ Q_{Pmax} \leq Q_L\}$ the significant congestion is recorded, $set\ Congs_{indx} = 1$. For moderate congestion, $\{Q_{Rmin} \leq Q_L \leq Q_{Rmax}, Q_{Pmin} \leq Q_L \leq Q_{Pmax}, set\ Congs_{indx} \in [0,1]\}$.
- In Fig.3(d), Neighbour nodes would decide for packet forwarding to the upstream node based on the $congs_{indx}$ and available buffer space.



In Fig.3(e), The proposed framework allows intra-cluster and inter-cluster communication. The communication paths are: [S2→Ch1→Sink], [S23→Ch2→Ch1→Sink]

Fig. 3(e). Communication

4. Comparative performance analysis

The performance of our proposed framework is analyzed using MATLAB 2018a over a 64bit Windows 10 operating system. The simulation compares the performance to prominent WSN state-of-the-art routing protocols as; LEACH (Heinzelman *et al.,* 2000), EELEACH (Arumugam *et al.,* 2015), OQoSCMRP (Deepa *et al.,* 2020), CDAS (Devi *et al.,* 2020), DHSSRP (Adil *et al.,* 2021), CMEEBZ (Adhikary *et al.,* 2021)

Table 3: Simulation Parameters

Parameters	Value	Description
WSN Area	[(0,0)~(200,200)] m	Area of Deployment
Sensor Nodes	0~50	Number of Nodes
Network Topology	Random Deployment	Distribution of Nodes
Initial Energy	3 J	Each Node's Initial Energy
Sink Location	(50, 80)	Location of the Sink

The following QoS metrics as, the energy requirement of cluster formation, throughput, packet delivery ratio, end-to-end latency, network lifetime, etc., have been identified to measure the network performance of the proposed framework that helps to attain the QoS. Fig.4. reveals the relationship between the number of nodes engaged in cluster formation and the required energy. The proposed QC2EF technique has been found to consume less energy than the existing well-known routing algorithms as; LEACH, EELEACH, OQoSCMRP, CDAS.

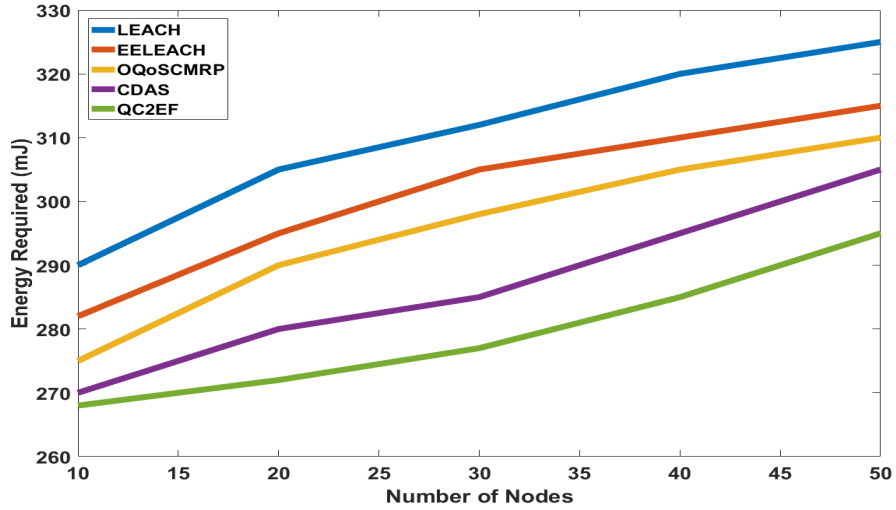


Fig. 4. Number of Nodes vs. Required Energy

Total data received in a certain period of time is used to calculate throughput. This is defined as; Throughput $= \sum_{i=0}^{n} P_s L_p$ where P_s is the total number of messages successfully received at the destination. A higher throughput would be achieved by multipath routing, which allows for greater P_s. Fig.5. shows that the proposed QC2EF produces 22.5% higher throughput than the existing routing protocols.

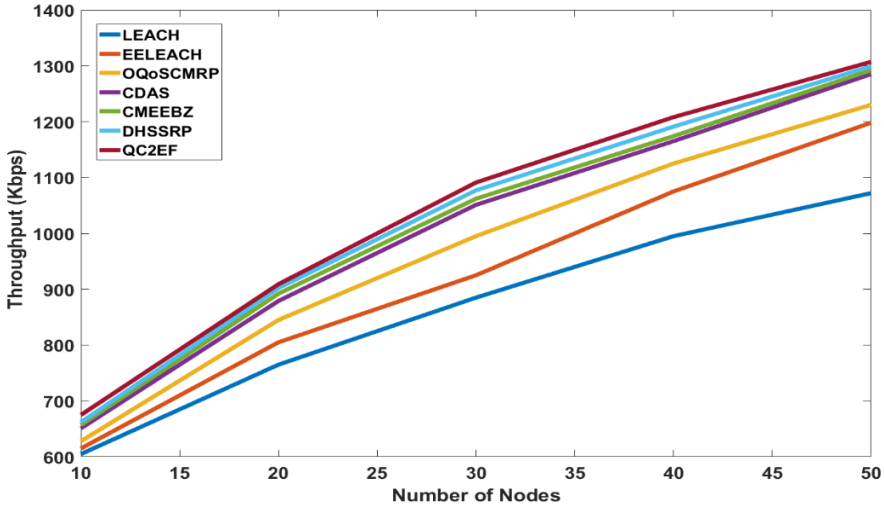


Fig. 5. Number of Nodes vs. Throughput

The packet delivery ratio is calculated as; $\left[PDR = \frac{\sum Number\_of\_Packet\_Received}{\sum Number\_of\_Packet\_Send}\right]$. In this proposal, congestion control and data aggregation mechanism are included to minimize unwanted data transfer in the network and help to enhance network performance. Fig.6. depicts that the PDR of the proposed QC2EF system offers 25.5% higher performance than the existing well-known selected routing protocols.

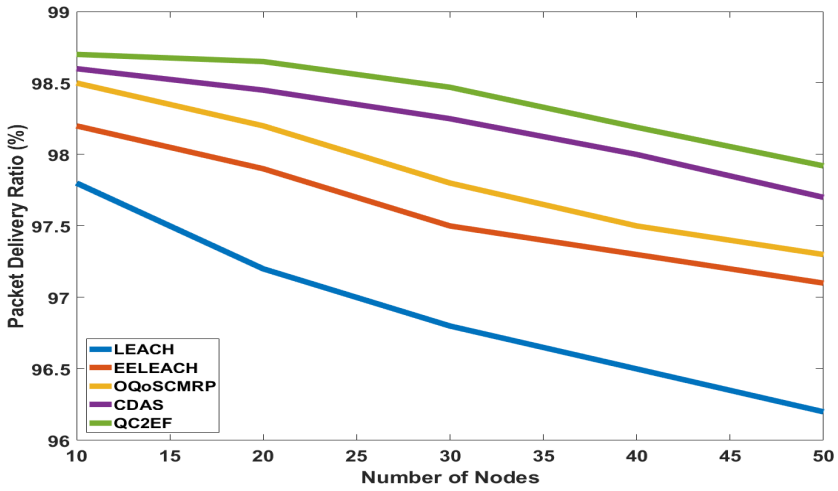


Fig. 6. Number of Nodes vs. Packet Delivery Ratio

The overall time takes for a data packet to deliver from the source node to sink node is known as the end-to-end delay and calculated as; $\left\{\text{End to end Delay} = \left[\frac{\sum arrival\ time - sending time}{\sum Number\ of\ connected\ Neighbours}\right]\right\}$. The proposed approach aggregates and forwards data more rapidly to the next neighbors with less routing load, resulting in a smaller delay and better QoS. Fig 7 compares the end-to-end delay of the proposed QC2EF protocol with other well-known protocols and finds that in all cases, the delay of the proposed mechanism is 21% lesser than of the other selected approaches.

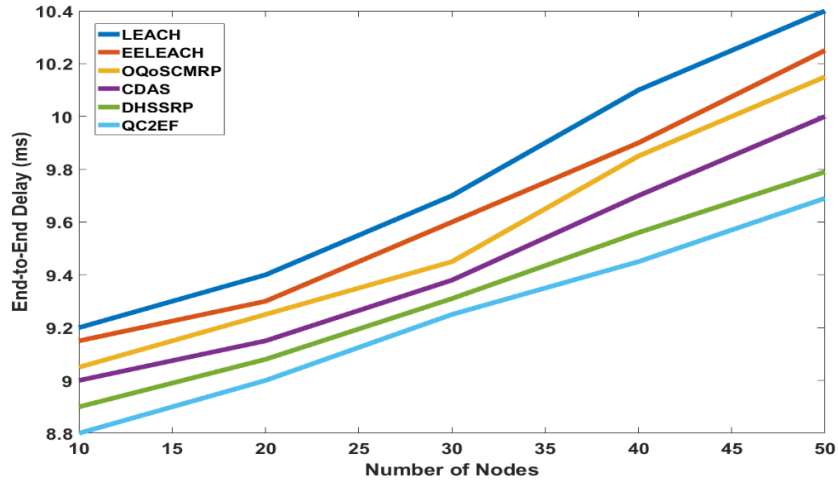


Fig. 7. Number of Nodes vs. End-to-End Delay

The network lifetime is the time it takes for all of its nodes to run out of energy. A number of important issues are considered while designing the proposal, such as dynamic cluster head selection and cluster formation, two-level data aggregation technique, congestion mitigation, and communication between network nodes using multipath routing. The proposed data aggregation methods reduce redundant data transfer while also consuming less energy. Congestion minimization strategy restricts the unsolicited data flowing over the network, all of which help to increase the network lifetime.

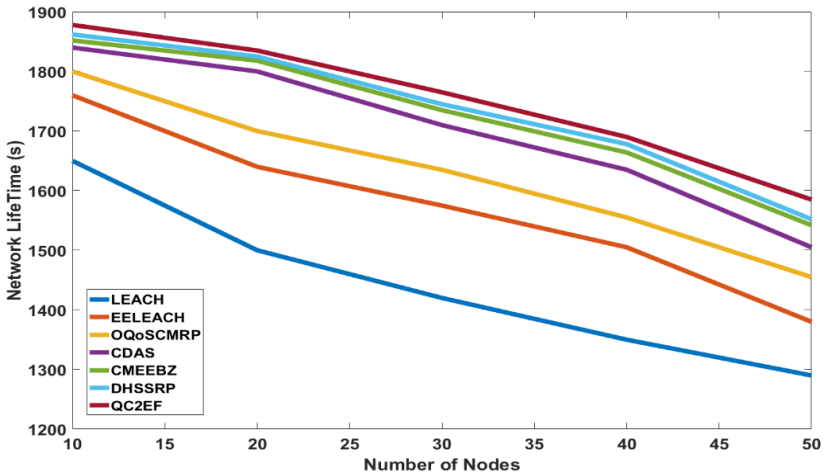


Fig. 8. Number of Nodes vs. Network Lifetime

The Fig.8. compares the network lifetime of the proposed QC2EF scheme with the others existing protocols and indicates that the proposed technique augments the network lifetime compared to others. As a result of packet drops and delays being reduced during communication, throughput has increased, helping to improve the lifetime of a network significantly. The above results identify that the proposed system outstrips better than the existing protocols and offers better throughput, less end-to-end delay, improved delivery ratio, energy efficiency, better network lifetime, and achieves the quality of services.

5. Conclusions

This comprehensive study of diverse clustering approaches and congestion control mechanisms reveals the pros and cons of the prevailing approaches. The empirical study to recognize several QoS metrics facilitates authors in assessing network performance and attaining the quality of services. The dynamic cluster head selection ensures an equitable energy load distribution among the sensor nodes and ensures that no sensor node would run out of energy earlier due to the additional responsibilities. Cluster members are connected to the cluster head through max heap topology. Cluster heads serve as child nodes of the sink node and are connected to neighbours through the dARY_HEAP topology. Two-level data aggregation techniques have been applied to curtail the redundant data flow that helps to minimize energy consumption. Prior to data transmission, the buffer occupancy level would sync with all relevant neighbours, ensuring that no data is lost due to congestion and optimal network performance is attained. The load balancing

mechanism provides the load distribution among the sensor nodes through multipath approaches. There is less possibility of a bottleneck forming since the sink node has an assorted number of children. Depending on the routing strategy, data can be routed to sink through any of the children. Alternative path construction is another crucial aspect for enabling real-time communication without introducing an additional delay.

The proposed framework has a greater throughput and better delivery ratio than the well-known existing techniques, as packet drops, and end-to-end delays are minimized during communication. Due to less energy consumption, the network has a more extended network lifetime and achieves the quality of services. The objective of the proposed QC2EF is attained. In future, this framework can be enhanced with a machine learning algorithm and would apply in the covid waste management systems in aspects of smart city.

References

Adhikary, D. R. D., Tripathy S., Mallick, D. K., Azad C. (2021), A Clustering Mechanism for Energy Efficiency in the Bottleneck Zone of Wireless Sensor Networks, Intelligent and Cloud Computing, Smart Innovation, Systems and Technologies, vol 194. Springer, Singapore. https://doi.org/10.1007/978-981-15-5971-6_76

Ali, H., Tariq, U. U., Hussain, M., Lu, L., Panneerselvam, J., Zhai, X. (2020), ARSH-FATI a Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks. IEEE Systems Journal, 1–12. doi:10.1109/jsyst.2020.2986811

Arumugam, G. S., Ponnuchamy, T. (2015), EELEACH: Development of Energy-Efficient LEACH Protocol for Data Gathering in WSN. EURASIP Journal on Wireless Communications and Networking, 2015(1). doi:10.1186/s13638-015-0306-5

Behera, T. M., Nanda, S., Mohapatra, S. K., Samal, U. C., Khan, M. S., Gandomi, A. H. (2021), Cluster Head Selection via Adaptive Threshold Design Aligned on Network Energy, in IEEE Sensors Journal, vol. 21, no. 6, pp. 8491-8500, 15 March15, 2021, doi: 10.1109/JSEN.2021.3051451.

Bhandari, K. S.; Hosen, A.S.M.S.; Cho, G.H. (2018), CoAR: Congestion-Aware Routing Protocol for Low Power and Lossy Networks for IoT Applications. Sensors 2018, 18, 3838. https://doi.org/10.3390/s18113838

Bahbahani, M. S., Alsusa, E. (2018), A Cooperative Clustering Protocol with Duty Cycling for Energy Harvesting Enabled Wireless Sensor Networks. IEEE Transactions on Wireless Communications, 17(1), 101–111. doi:10.1109/twc.2017.2762674

Cui, J., Valois, F. (2014), Data aggregation in wireless sensor networks: Compressing or forecasting? 2014 IEEE Wireless Communications and Networking Conference (WCNC). doi:10.1109/wcnc.2014.6952909

Deepa, O., Suguna, J. (2020), An Optimized QoS-based Clustering with Multipath Routing Protocol for Wireless Sensor Networks. Journal of King Saud University - Computer and Information Sciences. doi:10.1016/j.jksuci.2017.11.007

Devi, S., Ravi, T., Priya, S, B. (2020), Cluster Based Data Aggregation Scheme for Latency and Packet Loss Reduction in WSN. Computer Communications. doi:10.1016/j.comcom.2019.10.003

Faheem, M., G. Tuna, G., Gungor, V. C., (2018), QERP: Quality-of-Service (QoS) Aware Evolutionary Routing Protocol for Underwater Wireless Sensor Networks, in IEEE Systems Journal, vol. 12, no. 3, pp. 2066-2073, Sept. 2018, doi: 10.1109/JSYST.2017.2673759.

Han, Y., Li, G., Xu, R., Su, J., Li. J., Wen, G. (2020), Clustering the Wireless Sensor Networks: A Meta Heuristic Approach, in IEEE Access, vol. 8, pp. 214551-214564, 2020, doi: 10.1109/ACCESS.2020.3041118.

Heinzelman, W., Balakrishnan, H., Chandrakasan, A., (2000), Energy Efficient Communication Protocol for Wireless Microsensor Networks, in HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8, January 2000

Khan, M. N., Rahman, H. U., Khan, M. Z. (2020), An Energy Efficient Adaptive Scheduling Scheme (EASS) for Mesh Grid Wireless Sensor Networks. Journal of Parallel and Distributed Computing. doi:10.1016/j.jpdc.2020.08.007

Khediri, E. S., Fakhet, W., Moulahi, T., Khan, R., Thaljaoui, A., Kachouri, A. (2020), Improved Node Localization Using K-means Clustering for Wireless Sensor Networks. Computer Science Review, 37, 100284. doi:10.1016/j.cosrev.2020.100284

Kaur, T., Kumar, D. (2019), A Survey on QoS Mechanisms in WSN for Computational Intelligence Based Routing Protocols. Wireless Networks. doi:10.1007/s11276-019-01978-9

Muhammad Adil (2021), Congestion Free Opportunistic Multipath Routing Load Balancing Scheme for Internet of Things (IoT), Computer Networks, Volume 184, 2021, 107707, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2020.107707.

M. Farsi, M. Badawy, M. Moustafa, H. Arafat Ali, Y. Abdulazeem (2019), A Congestion Aware Clustering and Routing (CCR) Protocol for Mitigating Congestion in WSN, in IEEE Access, vol. 7, pp. 105402-105419, 2019, doi: 10.1109/ACCESS.2019.2932951.

Otaibi, S. Al., Rasheed, A. Al., Mansour, R. F., Yang, E., Joshi, G. P., and Cho, W., (2021), Hybridization of Metaheuristic Algorithm for Dynamic Cluster-Based Routing Protocol in Wireless Sensor Networks, in IEEE Access, vol. 9, pp. 83751-83761, 2021, doi: 10.1109/ACCESS.2021.3087602.

Panchal, A., Singh, R. K. (2020), EADCR: Energy Aware Distance Based Cluster Head Selection and Routing Protocol for Wireless Sensor Networks. Journal of Circuits, Systems and Computers, 2150063. doi:10.1142/s0218126621500638.

Pandey, D., Kushwaha, V., (2020), An Exploratory Study of Congestion Control Techniques in Wireless Sensor Networks. Computer Communications. doi:10.1016/j.comcom.2020.04.032

Ranga, V., Dave, M., Verma, A. K. (2016), Optimal Nodes Selection in Wireless Sensor and Actor Networks Based on Prioritized Mutual Exclusion Approach, Kuwait J. Sci. 43 (1) pp. 150-173, 2016.

Saha, S., Chaki, R. (2021), A Study on Energy Efficient Routing Protocols for Wireless Sensor Networks. In: Chaki R., Cortesi A., Saeed K., Chaki N. (eds) Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing, vol 1178. https://doi.org/10.1007/978-981-15-5747-7_9

Sharma, N., Singh, B.M., Singh. K., (2021), QoS Based Energy Efficient Protocols for Wireless Sensor Network, Sustainable Computing: Informatics and Systems, Volume 30,2021, 100425, ISSN 2210-5379, https://doi.org/10.1016/j.suscom.2020.100425.

Salim, E. K., Rehan, U. K., Nejah, N., Abdennaceur, K. (2021), Energy Efficient Adaptive Clustering Hierarchy Approach for Wireless Sensor Networks, International Journal of Electronics, 108:1, 67-86, DOI: 10.1080/00207217.2020.1756454

Shahraki, A., Taherkordi, A., Haugen, Ø., Eliassen, F. (2020), Clustering Objectives in Wireless Sensor Networks: A Survey and Research Direction Analysis. Computer Networks, 107376. doi:10.1016/j.comnet.2020.107376

Singh, K., Singh, K., Son, L. H., Aziz, A. (2018), Congestion Control in Wireless Sensor Networks by Hybrid Multi-objective Optimization Algorithm. Computer Networks, 138, 90–107. doi:10.1016/j.comnet.2018.03.023

Srivastava, V., Tripathi, S., Singh, K., Son, L. H. (2019), Energy Efficient Optimized Rate Based Congestion Control Routing in Wireless Sensor Network. Journal of Ambient Intelligence and Humanized Computing. doi:10.1007/s12652-019-01449-1

Turki Ali Alghamdi (2021), Enhanced QoS Routing Protocol Using Maximum Flow Technique, Computers & Electrical Engineering, Volume 89, 2021, 106950, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2020.106950

Yun, W. K., Yoo, S. J., (2021), Q-Learning-Based Data-Aggregation-Aware Energy-Efficient Routing Protocol for Wireless Sensor Networks, in IEEE Access, vol. 9, pp. 10737-10750, 2021, doi: 10.1109/ACCESS.2021.3051360.

Zear, A., Ranga, V. (2021), Distributed Partition Detection and Recovery Using UAV in Wireless Sensor and Actor Networks, Kuwait J.Sci., Vol.48, No.(4),October.2021,pp(1-16)

Submitted: 20/11/2021
Revised: 15/01/2022
Accepted: 15/01/2022
DOI: 10.48129/kjs.17331

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-20)

Quantum behaved intelligent variant of gravitational search algorithm with deep neural networks for human activity recognition

Sonika Jindal [1],* Monika Sachdeva [2], Alok K. S. Kushwaha [3]
[1,2,3] *Dept. of Computer Science and Engineering*
[1,2] *I. K. Gujral Punjab Technical University, Jalandhar, India*
[3] *Guru Ghasidas Vishwavidyalaya , Bilaspur, India*
**Corresponding author: sonikajindal@sbsstc.ac.in*

Abstract

Human activity recognition (HAR) encompasses the detection of daily routine activities to advance usability in detecting crime and preventing dangerous activities. The recognition of activities from videos and image sequences with higher exactitude is a major challenge due to system complexities. The efficient feature optimization approach can reduce system complexities by removing ineffective features, which also improves the activity recognition performance. This research work presents a novel quantum behaved intelligent gravitational search algorithm to optimize the features for human activity recognition. The proposed intelligent variant is termed as INQGSA, which optimizes the features by using the advantageous attributes of quantum computing (QC) and intelligent gravitational search algorithm (INGSA). In INQGSA, the intelligent factor avoids the trapping of mass agents in later iterations by using the information of the best and worst agents to update the position of agents. The addition of quantum computing based attributes (such as quantum bits, their superposition, and quantum gates, etc.) ensures a better diversity of discrete optimized features. To analyze the superiority of INQGSA, the feature optimization is also conducted with the gravitational search algorithm (GSA) and the quantum-inspired binary gravitational search algorithm (QBGSA). Finally, the optimized selected features are utilized by the deep neural networks (DNN) of ResNet-50V2 and ResNet-101V2 for the classification of activities. The activity recognition experiments are conducted on the UCF101 and HMDB51 datasets. The performance comparison of the proposed HAR system with state-of-the-art techniques signifies that the proposed system is superior and effective in detecting the different activities.

Keywords: Deep neural networks; feature optimization; gravitational search algorithm; human activity recognition; quantum computing

1. Introduction

The concept of video-based HAR has aroused the interest of industrialists and academicians in developing intelligent recognition systems. The effective recognition of activities can fulfil the future needs of building smart homes and intelligent monitoring systems. The data captured as RGB videos with the cameras is an effective means of recognizing the activities with great ease. The digitization of the world has increased the use of cameras in daily life with their already existing presence in public places such as airports, banks, hospitals, etc. Moreover, human beings themselves generate a massive amount of video content and upload it on social networking and other online websites. The primary motivation for adapting to video-based human activity recognition is the availability of a wide range of applications (Xu *et al.*, 2013; Serpush & Rezaei, 2020; Özyer *et al.*, 2021).

At the early stage of human activity recognition, the researchers focused on recognizing the simple kinematic activities from a video with a plain background. Recently, the focus of researchers has turned

towards the determination of activities in real-time and uncontrolled environments. Chen *et al.* (2012) used the extreme learning model for human activity recognition, which was a device displacement free recognition model. Shieh & Huang (2012) adapted a pattern recognition model to take care of aged people with video surveillance. Moreover, an autonomous falling detection algorithm was utilized to determine the falling activities. Khemchandani & Sharma (2016) proposed the robust least square twin support vector machine (RLSTW-SVM) model along with the feature descriptors of optic flow and silhouette. The work was effective in handling the heteroscedastic noise and incorporating the outlier effect. Kushwaha *et al.* (2017) used contour-based pose features from silhouettes as well as features based on the rotation invariant local binary approach.The activity classification was conducted using the multi-class support vector machine. Ijjina & Chalavadi (2017) utilized the deep convolutional neural networks with features through depth stream videos and RGB motion streams. The presented model could tolerate robust noise. Bouachir *et al.* (2018) used different machine learning and ensemble methods to determine the suicide attempt activities. The authors determined the SVM-RBF (SVM with radial basis kernel) method as superior among others. Kong *et al.* (2019) explored the three-stream convolutional neural network to determine the multi-view falling activities. The first two streams of the model adapted the Silhouettes and motion history images as the input, and the third stream considered the dynamic images. The method lacked effectiveness due to inefficient results for the lousy representation of video clips. Jaouedi *et al.* (2020) used the Gated Recurrent Neural Network for the recognition of human activities. The Kalman Filter and Gaussian Mixture Model were used to extract the features to recognize normal and sports activities. Verma *et al.* (2020) used RGB and skeleton information as the feature attributes to recognize human activities. The combined approach of convolutional and recurrent neural networks was adapted by the authors.

Although different methods are used in the discussed contributions for activity recognition, the usability of machine learning (Khan *et al.*, 2016) and deep learning (Al-Hmouz, 2020) techniques can be majorly noticed. Simple activities with fixed backgrounds can be easily recognized with higher recognition accuracy. The recognition of activities with diverse backgrounds, performed by different individuals, is a complex task. In addition, it is considerably more challenging to build automated systems with better precision. Computer vision has been used to make many automated systems, but the current systems cannot recognise very complicated human actions.

Most of the existing systems have adapted the autonomous approach to extract and select the features for activity recognition, which is less effective. The different types of activities captured in unconstrained scenarios need their relevant feature attributes to determine the type of activity. The present work has adapted distinct strategies for the different modules of the human activity recognition process. The proposed HAR system is described in four major modules: pre-processing, feature extraction, feature

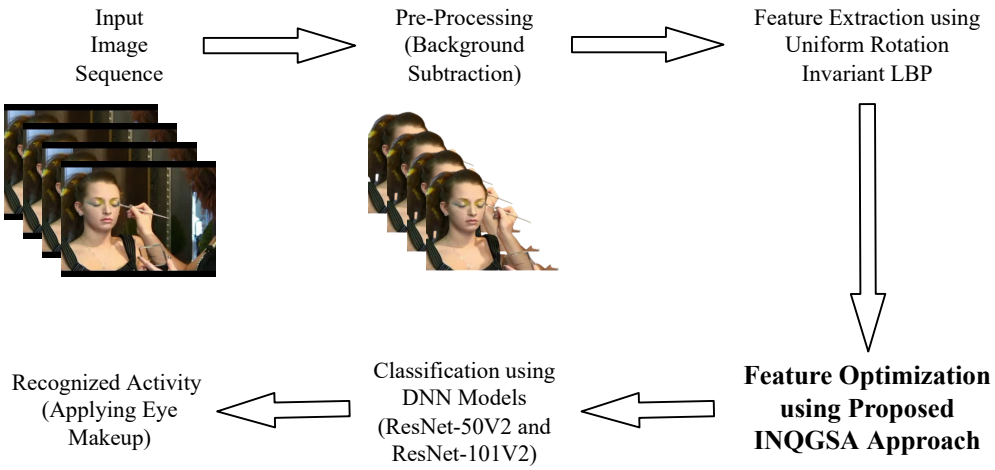


Fig. 1. Architecture of proposed HAR system.

selection, and classification. The architecture of the proposed HAR system is illustrated in Figure 1.

In the proposed HAR system, the process of recognising the activities begins with the pre-processing module that segments the background region from the extracted video frames. The processed frames (images) are evaluated to determine the features using the uniform rotation invariant LBP (Local Binary Pattern) technique. The extracted features need to be optimized to reduce the feature dimensionality and computation time by eliminating redundant and irrelevant features. Here, the INQGSA is proposed for the feature set optimization. The DNN models of ResNet-50V2 and ResNet-101V2 use the selected features to classify the activities. The performance of the proposed system is accessed for the UCF101 and HMDB51 datasets. The UCF101 dataset consists of 101 different activities, and the HMDB51 dataset is composed of 51 different activities. The focused section of the paper is the proposal of INQGSA for the feature optimization which selects the discrete feature set by adapting the attributes of an intelligent variant of GSA and the quantum computing concepts. In summary, the key contributions of the work are described as follows:

- The proposal of a novel INQGSA approach to optimize the features for the application of human activity recognition. The INQGSA approach avoids the trapping of mass agents in local optima by intelligently incorporating the advantageous attributes of QC and INGSA.
- The incorporation of advanced techniques of uniform rotation invariant LBP for multi-pose feature extraction and Deep Neural Networks (ResNet-50V2 and ResNet-101V2) for human activity recognition.
- The extensive experiments of the proposed HAR system for the video-based datasets of UCF101 and HMDB51.

The organization for the rest of the paper is described as follows. Section 2 presents the work related to feature selection and optimization techniques for human activity recognition. Section 3 illustrates the video data processing and feature extraction modules for activity recognition. Section 4 discusses the proposed INQGSA approach for the optimization of features. Section 5 exhibits the classification modules of the activities using DNN models. Section 6 describes the results and discussion of the experiments on the UCF101 and HMDB51 datasets. Section 7 concludes the paper with some future viewpoints.

2. Related work

The automation of the HAR from videos is an imperative research domain in pattern recognition as it is essential to meet the demand for a smart future in terms of automated video surveillance and smart homes. But the selection/optimization of features is the major concern in pattern recognition. During the feature extraction phase, the feature extractor can extract the different types of features for activity recognition. But the increasing feature vector can grow the dimensions of the Eigen vector, which increases the computational complexities and time consumption. Therefore, the selection and optimization of features is essential as it can determine higher recognition accuracy by consuming the least but appropriate features. The feature optimization phase contributes the relevant selected features to the HAR by removing the redundant and irrelevant features. The section describes the feature optimization based on relevant studies for video-based human activity recognition.

The feature optimization improves the HAR system performance compared to the usability of the entire feature set (Wang *et al.*, 2016). Siddiqi *et al.* (2014) presented the method of stepwise linear discriminant analysis for feature selection, which evaluates the localized features from video frames. The method was determined as efficient for the experiments on the single subject based dataset, but it lacks for the experiments on the real-time datasets having different subjects for different activities. Fang *et al.* (2014) used the inter-class distance method for feature selection and neural networks with a back propagation algorithm for activity recognition. The authors tested the results by incorporating six different feature sets and a recognition method that was evaluated as efficient compared to the Hidden Markov Model and Naive Bayes algorithm. Zheng (2015) adapted a hierarchical feature selection approach along with the classifiers of Naive Bayes and Least Squares Support Vector Machine for human

activity recognition. The authors defined the requirement to place the sensors at the correct place to determine the activity accurately. Mazaar *et al.* (2016) explored the ensemble learning model by incorporating the methods of random forest and gradient boosting for feature optimization. The classification of the activities is performed using support vector machine with a linear kernel. Baldominos *et al.* (2017) conducted the feature optimization at the dimension level and attribute level using the genetic algorithm. The authors presented four different feature selection methods by incorporating with and without feature sensibility for both the dimension and attribute levels. Wang *et al.* (2018) optimized the features using the correlation-based binary particle swarm optimization approach. In this approach, the k-nearest neighbor method was used as a fitness method to determine the performance of the optimized feature set. Siddiqui *et al.* (2018) presented a codebook-based feature selection approach that includes models of visual vocabulary learning, quantization of features based on learned visual vocabulary, and representation of images by using the frequency of visual words. In the final module, activity classification was conducted using the support vector machine algorithm.

Furthermore, Siddiqi *et al.* (2019) used a normalized mutual information-based feature selection technique for the optimization of features. The authors also used linear discriminant analysis to reduce the feature space for the extracted features by using the curvelet transform. The final classification of features was performed using a hidden Markov model. Sharif *et al.* (2019) explored strong correlation and the Euclidean distance method to select the optimal feature for activity recognition. Berlin & John (2020) used a particle swarm optimization approach with a multi-objective function to reduce the feature space by selecting an appropriate feature set. The activity recognition was conducted using a deep learning neural network model. Helmi *et al.* (2021) proposed a hybrid approach of Grey Wolf Optimizer (GWO) and Gradient-Based Optimizer (GBO) for feature optimization. The GWO method was used to optimize the performance of the GBO algorithm. Tian *et al.* (2021) presented a feature selection methodology by combining the wrapper and filter feature selection approach. In this method, the initial feature selection was conducted using a game-theory filter approach, and further reselection was performed using the wrapper approach of the binary firefly algorithm. Fan & Gao (2021) integrated the deep Q-network with bee swarm optimization for the feature optimization. The bee swarm optimization approach retains the exploration and exploitation balance in the feature space, and the deep Q-network uses the advantageous attributes of reinforcement learning to make the local search space more efficient. Bulbul *et al.* (2022) focused on enhancing the performance of 3D auto-correlation gradient features. The space-time auto-correlation of gradients descriptor was used to obtain the three vectors in the method. Siddiqi & Alsirhani (2022) employed the mutual information algorithm for feature selection. The method was the extension of the max-relevance and min-redundancy to select the more appropriate and relevant features for activity recognition. In the future, the authors indicated testing the presented method in a real-time scenario.

As per the existing studies, the feature optimization techniques significantly contribute to improving the system accuracy in HAR. However, the higher recognition accuracy requires the use of an appropriate technique that can select relevant features without redundancy and can reduce the computational complexities. In addition, the standard and individual optimization techniques are observed with lacking feature attributes that increase computational cost due to higher feature dimensionality (Helmi *et al.*, 2021). The improved and ensemble approaches are essential to increase the feature optimization ability in HAR. The current work proposes the INQGSA approach, which ensembles the attributes of the quantum computing concept with an intelligent gravitational search algorithm for feature optimization. To determine the superiority of the proposed INQGSA approach, the feature optimization is also performed using the standard GSA (Rashedi *et al.*, 2009) and QBGSA (Ibrahim *et al.*, 2012). The GSA and QBGSA use $Kbest$ agents to maintain the balance of exploration and exploitation, but the $Kbest$ is a reducing function, so its value decreases over time and iterations. This decreasing value leads to the trapping of agents at later iterations. The proposed INQGSA approach overcomes this drawback by using an intelligent variant of GSA in which the position of agents is updated intelligently by using the worst ($gWorst$) and best ($gBest$) information values of the agents (Mittal & Saraswat, 2019). The mass agents get attracted towards the $gBest$ information to attain the best position and start getting away from

the $gWorst$. This avoids the trapping of agents in local optima and optimizes the features effectively.

3. Video data processing and feature extraction

The section describes the pre-processing and feature extraction modules of the HAR process. These are the initial and essential modules for activity recognition.

3.1 Pre-processing

The pre-processing module segments the background region from the foreground of the image sequence (video frames). The image sequences for the proposed system are processed with a statistical model, which evaluates the variance to analyze the absolute variations and co-variance to determine the relative variations of the pixels (Singh *et al.*, 2019).

For an array of frames ($\eta_i = (\phi, \psi)$) with a starting value of SF and an ending value of EF, the variance (Var) is determined with Equation (1), and the co-variance ($Cov(\alpha, \beta)$) between the frames α and β is evaluated with Equation (2).

$$Var = \left(\frac{1}{EF} \sum_{i=0}^{EF-1} (\eta_i - \bar{\eta})^2 \right) \tag{1}$$

$$Cov(\alpha, \beta) = \left(\frac{1}{EF} \sum_{i=0}^{EF-1} \alpha_i \beta_i \right) - \left(\frac{1}{EF} \sum_{i=0}^{EF-1} \alpha_i \right) \left(\frac{1}{EF} \sum_{j=0}^{EF-1} \beta_j \right) \tag{2}$$

Where, $0 \leq i < EF$ and $\bar{\eta}$ is the mean of all the frames.

The variation in the intensity of the pixel compared to other pixels is evaluated based on the co-variance between frames. The variance and co-variance values for all the pixels are stored in the reference image $Ref(\phi, \psi)$. The objects are differentiated based on the reference image.

Further, the background model is updated to incorporate the change in intensity value and background of the different frames. Exceeding the threshold value of the counter ρ indicates the requirement to update the background model. The change in the background model is performed with Equation (3).

$$Ref_{new}(\phi, \psi) = (1 - \mu) \times frame_{\rho}(\phi, \psi) + \mu \times Ref(\phi, \psi) \tag{3}$$

Where, $Ref_{new}(\phi, \psi)$ denotes the updated model. The symbol μ describes the updating speed, and $frame_{\rho}(\phi, \psi)$ depicts the current frame of the video.

3.2 Feature extraction

The features are extracted using the uniform rotation invariant LBP (Local Binary Pattern) technique from the pre-processed image sequence. The incorporation of the uniform rotation invariant is conducted to handle the activities that possess multi-view poses. The image sequences are initially converted into grayscale images to extract the features. In an LBP operator, the features of an image $I(x, y)$ with g_c as the gray level of the central pixel and g_p as the gray level of its neighbor pixels can be extracted using Equation (4) (Pietikäinen *et al.*, 2011).

$$LBP_{P,D}(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \tag{4}$$

Where, P is the set of sample pixels in the circular neighborhood of the central pixel with radius D, $p = 0, 1, \ldots, (P-1)$, and 2^p is adapted to determine the size of histograms for the LBP operator. The values of $s(z)$ can be determined as described in Equation (5).

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{5}$$

The local circular neighbor pixels around the central pixel with a radius of D are described in Figure 2. Here, only the uniform patterns (U) of the LBP code are incorporated to retain the statistical

robustness. The uniform patterns hold the transition from 0 to 1, and the mapping of uniform LBP patterns produces $P(P-1)+3$ labels for the P sampling points.

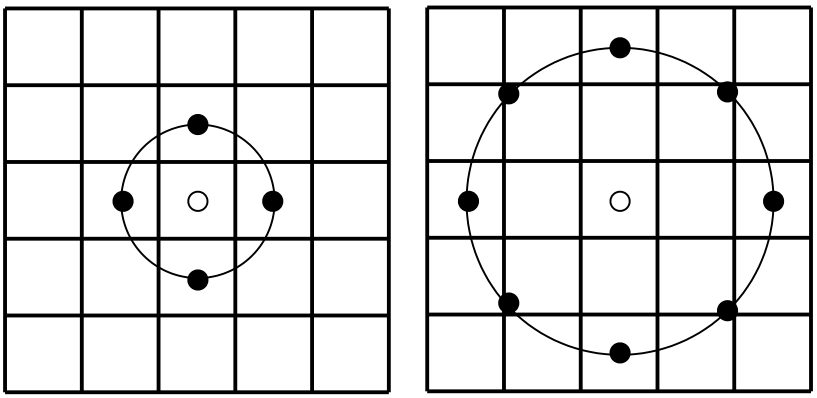


Fig. 2. Circular neighbors for central pixels in format (P,D): (a). (4,1), (b). (8,2).

With the rotation of the image $I(x,y)$, the LBP patterns are translated to another location for the rotation around their origin. The rotation of the patterns can be normalized with the rotation invariant mapping in which the LBP binary code is rotated to the minimum possible value, as depicted in Equation (6).

$$LBP^{ri}_{P,D} = \min_{i} ROR(LBP_{P,D}, i) \tag{6}$$

Where, $ROR(LBP_{P,D}, i)$ is the circular bit-wise rotation with i steps.

The features with the uniform rotation invariant LBP operator are extracted using Equations (7)- (8) that retain the robustness and higher stability (Singh *et al.*, 2019).

$$LBP^{riu2}_{P,D} = \begin{cases} \sum_{p=0}^{p-1} s\left(g_p - g_c\right), \text{ if } U(LBP_{P,D} \leq 2) \\ P+1, \text{ otherwise} \end{cases} \tag{7}$$

where,

$$U(LBP_{P,D}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{p-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{8}$$

The value of $s(z)$ is evaluated using Equation (5). The uniform operator $U(LBP_{P,D})$ is a rotation invariant operator with varying bits of 0 and 1 in circular symmetry.

4. Feature optimization using proposed INQGSA approach

The optimization of features is essential for the classifier to improve the performance of the system. The present work proposes the INQGSA approach to optimize the features for human activity recognition. The GSA is a population-based meta-heuristic algorithm inspired by the physics-based Newton′s laws of motion and gravity to optimize the solution set for high dimensional problems (Rashedi *et al.*, 2009). The proposal of the INQGSA approach is presented as the standard GSA (Rashedi *et al.*, 2009) and QBGSA (Ibrahim *et al.*, 2012) algorithms lack feature optimization. In the meta-heuristic algorithm, the balance of exploration and exploitation is essential for optimization. The GSA and QBGSA use the $Kbest$ agents to retain this balance, but the value of $Kbest$ decreases with the increasing iterations because $Kbest$ is a reducing function. This decreasing value leads to the trapping of agents at later iterations. The proposed INQGSA approach adapts the advantageous attributes of QC and intelligent variant of GSA to tackle the trapping of agents. In the proposed INQGSA approach, the position of agents is updated intelligently by using the worst ($gWorst$) best ($gBest$) information values of the agents (Mittal & Saraswat, 2019). The mass agents get attracted towards the $gBest$ information to attain

the best position and start getting away from the $gWorst$. This avoids the trapping of agents in local optima and optimizes the features effectively.

The proposed INQGSA algorithm begins by considering an n-dimensional system having N mass agents in which the position of the i^{th} agent can be defined by Equation (9).

$$X_i = (x_i^1, x_i^2, \ldots, x_i^d, \ldots, x_i^n);\ i = 1, 2, 3, \ldots, N \tag{9}$$

Where, x_i^d is the position of the i^{th} mass agent in d^{th} dimension.

The force acting by the considered i^{th} agent on the j^{th} agent is determined by Equation (10).

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \tag{10}$$

Where, ε is a constant and the masses are considered as active mass (M_{aj}) and passive mass (M_{pi}) for the j^{th} agent and i^{th} agent, respectively. In Equation (10), the distance R is incorporated instead of R^2 (in law of gravity) due to better performance with only R as per the existing studies (Rashedi *et al.*, 2009). Here, the Euclidean distance R_{ij} is determined by Equation (11).

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \tag{11}$$

Further, the addition of stochastic attributes changes the force evaluation with the total force acting on the agent i as depicted in Equation (12). By considering the total force, the acceleration evaluation is depicted in Equation (13).

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{12}$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{13}$$

Where, $rand_j$ is the random number that lies in [0, 1] and M_{ii} indicates the inertial mass.

Further, the movement of the particles is determined by evaluating the change in position, velocity, and masses by Equation (14)- (16).

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{14}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{15}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{16}$$

Where, $M_i = M_{ii} = M_{pi} = M_{ai}$ as the inertial and gravitational masses are assumed to be equal and calculated by the fitness function $fit_i(t)$. In Equation (16), $m_i(t)$ is evaluated using Equation (17).

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{17}$$

In the current research work, feature optimization is a minimization problem as it needs to minimize feature dimensionality. For the minimization problem, the values of $best(t)$ and $worst(t)$ are evaluated by Equations (18) and (19).

$$best(t) = min_{j\epsilon\{1,2,\ldots,N\}} fit_j(t) \tag{18}$$

$$worst(t) = max_{j\epsilon\{1,2,\ldots,N\}} fit_j(t) \tag{19}$$

In GSA, the mass agents can be trapped in later iterations, which can be avoided by introducing the intelligent variant of GSA. The INGSA incorporates the worst ($gWorst$) and best ($gBest$) information

values of the agents to update the position of each agent intelligently (Mittal & Saraswat, 2019). For the current feature optimization problem, which is a minimization problem, the values of $gBest$ and $gWorst$ are evaluated by Equations (20)- (21).

$$gBest(t) = x_e(t) \tag{20}$$

$$gWorst(t) = x_s(t) \tag{21}$$

Where, the notations e and s are concerned with the minimum and maximum fitness functions of the intelligent mass agents, which are evaluated by Equations (22)- (23).

$$fit_e(t) = \min\{fit_1, fit_2, fit_3, \ldots, fit_N\} \tag{22}$$

$$fit_s(t) = \max\{fit_1, fit_2, fit_3, \ldots, fit_N\} \tag{23}$$

The mass agents get attracted towards the $gBest$ information to attain the best position and start getting away from the $gWorst$. The update in the position of the mass agents as per INGSA is determined by Equation (24).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) + b(t) \times \frac{\left(gBest^d(t) - x_i^d(t)\right)}{\left|\omega \times gWorst^d(t) - x_i^d(t)\right|} \tag{24}$$

In Equation (24), the intelligent component is the third term. Here, $b(t)$ is a number that lies in [0,1] and is determined randomly. ω possesses a constant value of 0.7 and is incorporated to reduce the effect of $gWorst$ as it tries to mitigate the movement of mass agents towards $gBest$ (Mittal & Saraswat, 2019). As the mass agents move towards the $gBest$, the agents$'$ distance increases from $gWorst$ which helps to reduce the step size and avoid the trapping of agents in the local optima. Another scenario of greater distance from $gBest$ allows the agents to explore more.

Further, the concept of quantum computing is introduced with the INGSA. In quantum computing, the position and velocity of mass agent changes to quantum states with a probabilistic illustration (Ibrahim *et al.*, 2012). The Q-bit (quantum bit) is considered as the smallest unit and its state can be either 0 or 1 or their superposition, which can be analyzed for any complex numbers (C_1 and C_2) by Equation (25).

$$|\psi\rangle = C_1 |0\rangle + C_2 |1\rangle \tag{25}$$

The complex numbers C_1 and C_2 are the probability amplitudes for binary numbers 0 and 1, respectively, and they assures the normalization of states to unity by following Equation (26).

$$|C_1|^2 + |C_2|^2 = 1 \tag{26}$$

The states of the Q-bits are updated by using the quantum gates. Among the eminent quantum gates of the rotation gate, NOT gate, Hadamard gate, etc., this work incorporates the rotation gate due to its effective performance in the existing studies (Ibrahim *et al.*, 2012). The solution for the INQGSA-agents through the rotation gate is presented by Equation (27).

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{27}$$

Where, $\Delta\theta$ is the rotation angle for $i = 1, 2, \ldots, n$ that determines the position of the agents in terms of quantum state.

In the INQGSA approach, the movement of the quantum mass agents is determined by updating Equation (24) with the quantum movements, which is illustrated by Equation (28).

$$\theta_{ij}^d(t+1) = \theta_{ij}^d(t) + \Delta\theta_{ij}^d(t+1) + b(t) \times \frac{\left(gBest^d(t) - \theta_{ij}^d(t)\right)}{\left|\omega \times gWorst^d(t) - \theta_{ij}^d(t)\right|} \tag{28}$$

Algorithm 1: Pseudo Code of the INQGSA approach for Feature Optimization

Initialize the parameters of the QC concept and GSA algorithm such as as t_{max}, ϑ_0, ω, etc.
Determine the initial fitness value for the population of intelligent mass agents.
t=1; **while** $t < t_{max}$ **do**
  for $i = 1$ to N **do**
    Evaluate the ϑ and a values for the agents.
    Evaluated the $\theta_{ij}^d(t)$ and $\Delta\theta_{ij}^d(t)$ values for the agents
    Determine the information for the agents concerning the best and worst fitness information.
    Evaluate the fitness value.
    Update the position ($\theta_{ij}^d(t+1)$) and velocity ($\Delta\theta_{ij}^d(t+1)$) values for the agents using Equations (28) and (29).
  end
end
Store the optimal features determined by the best agents at optimal positions and best fitness value.

Where,

$$\Delta\theta_{ij}^d(t+1) = rand_i \times \Delta\theta_{ij}^d(t) + a_{ij}^d(t) \tag{29}$$

In Equation (29), $a_{ij}^d(t)$ is evaluated by putting the values of Equations (10)- (12) into Equation (13), which is further derived as per the INQGSA approach. In Equation (10), the value of ε is neglected as it is constant. The derived formulation for $a_{ij}^d(t)$ as per the INQGSA approach is presented by Equation (30).

$$a_{ij}^d(t) = \sum_{j=1, j\neq i} \left[rand_j \times \vartheta \times \gamma_i^k \times \left(\theta_{kj}^d(t) - \theta_{ik}^d(t) \right) \right] \tag{30}$$

Where, the symbol ϑ is $G(t)$ which decreases from ϑ_{max} to ϑ_{min} depending on the rotation angle. The ratio of the mass (M_{aj}) and distance (R_{ij}) are presented by a decision parameter (γ_i^k) which is evaluated by Equations (31)- (32) (Ibrahim *et al.*, 2012).

$$\gamma_i^k = \begin{cases} \lambda_i^k + 1, \text{if } fit(\theta_k^d(t)) = fit(\theta_i^d(t)) \\ \lambda_i^k, \text{ otherwise} \end{cases} \tag{31}$$

$$\lambda_i^k = \begin{cases} 1, \ M_k > M_i \text{ and } R_{ik} \leq \tau \\ 0, \text{ otherwise} \end{cases} \tag{32}$$

Where, τ represents the maximum number of different bits out of total bits in between the i^{th} and k^{th} agents that can put the active force on the i^{th} agent.

The optimized features are selected by the intelligent quantum mass agents upon the completion of their maximum iterations. At maximum iterations, the features selected by the best agents that possess optimized position, are retained. The pseudo-code of the feature optimization process using the INQGSA approach is illustrated in Algorithm 1.

5. Classification and recognition of activities

The classification of the activities is conducted with the deep residual networks (ResNet), which possess the deep neural network (DNN) architecture. DNN models are capable of mapping the features of layer data within deep networks. The network architecture of ResNet is a series of blocks connected to each other with parallel shortcut links for the output. The basic structure of the residual network block and its internal learning process are illustrated in Figure 3.

In Figure 3(a), incorporating the parameterized layer after the Addition module can reduce the ResNet's advantages, but incorporating the non-parameterized layer (ReLU) after the Addition has little impact on the ResNet (Kiliç *et al.*, 2020).The conventional CNN is not significant for in-depth learning

as the error increases (due to over-fitting) with the increase in the depth of layers. In ResNet, the residual values are formed after adding the blocks, which are fed to the succeeding layers in the model.

In Figure 3(b), x is incorporated as an input, and the output is obtained after the ReLU operation in the form of $H(x) = F(x) + x$. Here, the input (x) is passed from the weight layer (w), and the results are acquired in the form of $F(x)$. The final output is determined by adding the x input to $F(x)$.

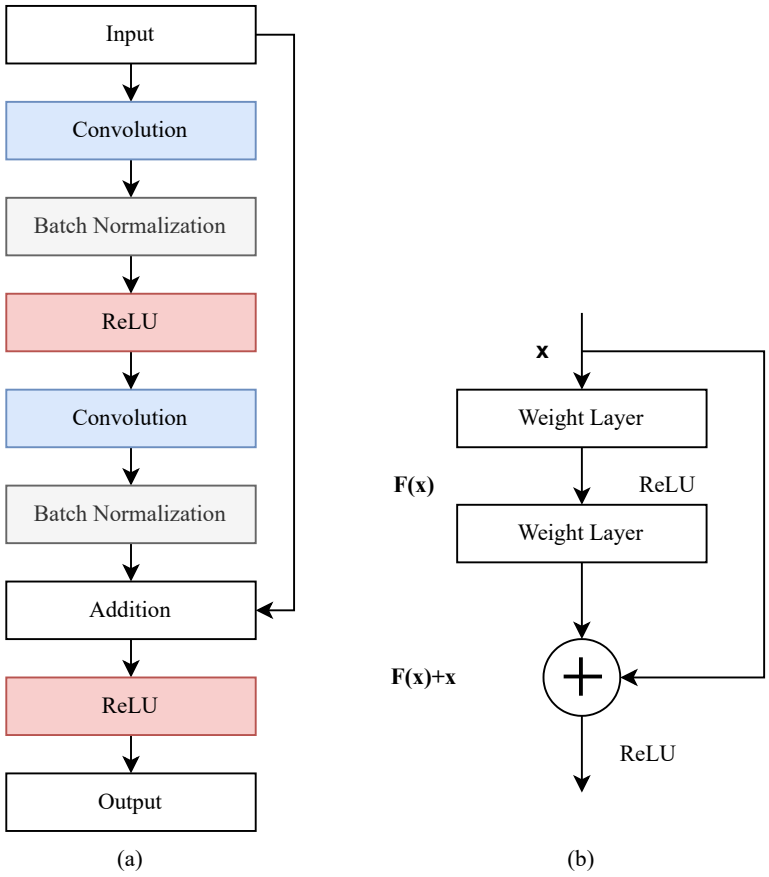


Fig. 3. (a) Basic ResNet Block (b) Internal Learning Process of Residual Block.

In this research work, the ResNet with 50 and 101 layers is adapted for the classification of activities. These networks are constructed using the architecture of 3-layer bottleneck blocks. There are 3.8×10^9 and 7.6×10^9 FLOPs in ResNet-50 and ResNet-101 respectively. The complexity of these networks is lower than VGG16/19, even after increasing the deep layers. The architectures of ResNet-50 and ResNet-101 are described in Table 1.

Here, version 2 (V2) of the ResNet is incorporated to direct the identity connections from input to output by removing the last non-linearity, which enhances the learning process and hence the classification of activities. In ResNet V2, the weight layers are pre-activated instead of post-activation. The present research work has incorporated the ResNet-50V2 and ResNet-101V2 for the human activities classification.

6. Experimental results and discussion

The results for the proposed HAR system are determined using evaluation measures of precision, recall, and f-measure for the experiments on the UCF101 and HMDB51 datasets. Furthermore, the recognition accuracy is also evaluated for the quantitative analysis of the proposed HAR system. The recognition accuracy is described in Equation (33).

$$\text{Recognition Accuracy} = \frac{\text{Correctly Classified Instances}}{\text{Total Number of Instances}} \times 100 \tag{33}$$

Table 1. Layer Architecture of the Residual Networks.

Layer	ResNet-50	ResNet-101	Output Size
Convolutional 1	7×7, 64, stride 2		112×112
Convolutional 2	3×3 max pooling, stride 2		56×56
	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	
Convolutional 3	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	28×28
Convolutional 4	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	14×14
Convolutional 5	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	7×7
	Average Pooling, 1000 Fully Connected Softmax		1×1

Table 2. Statistics of Datasets.

Parameter	UCF101	HMDB51
Actions	101	51
Resolution	320×240	320×240
Video Clips	13,320	6,766
Frame Rate	25 fps	30 fps
Min. Video Clip Length	1.06 sec	1 sec
Min. Video Clips Per Action	100	101

6.1 Datasets

The present work has utilized the UCF101 (Soomro *et al.*, 2012) and HMDB51 (Kuehne *et al.*, 2011) datasets, which are video-based datasets. The UCF101 dataset is composed of 101 realistic action videos gathered from YouTube. There are 13,320 videos of different actions available in this dataset, and the different activities are divided into five categories: sports, playing musical instruments, human-human interaction, body-motion, and human-object interaction. Whereas, the HMDB51 dataset is collected from the Prelinger archive, Google, and YouTube videos. The HMDB51 dataset embodies 6,766 video clips related to 51 action categories, which are majorly divided into five categories: general body movements, body movement for human interaction, body movement for object interaction, general facial actions, and facial actions with object manipulation. The statistics of both the datasets are illustrated in Table 2, and some sample frames indicating different activities are depicted in Figure 4.

6.2 Result evaluation

To perform the experiments for the proposed HAR system, both the datasets (UCF101 and HMDB51) are divided separately into the training and testing proportions of approximately 90:10. For both the datasets, 1,650 frames per activity are extracted. A total of 166,650 frames from the UCF101 dataset and 84,150 frames from the HMDB51 dataset are extracted. Among the total 1,650 frames per activity, 1,500 frames are utilized for training the residual networks and 150 frames are utilized for testing. The description of the training and testing settings is depicted in Table 3.

Before evaluating the testing results for the proposed HAR system, the data is validated by splitting the training data frames (151,500 frames of the UCF101 dataset and 76,500 frames of the HMDB51 dataset) into the ratio of 80:20. The 80% of the data (121,200 frames of UCF101 dataset and 61,200 frames of HMDB51 dataset) is utilized for the training and 20% of the data (30,300 frames of UCF101

Apply Lipstick | Cricket Shot | Horse Riding | Shaving Beard | Punch

(a) UCF101

Shoot Gun | Climb | Flic Flac | Clap | Hug

(b) HMDB51

Fig. 4. Sample Frames illustrating different Activities (a) UCF101 dataset (b) HMDB51 dataset.

Table 3. Training and Testing Setting.

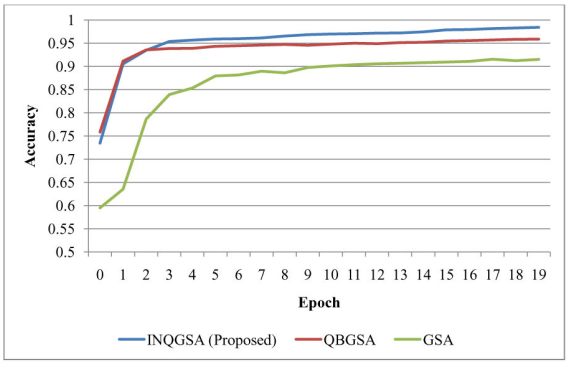
Parameter	Value
Input of spatial stream	Size of single frame = 3×224×224
Total number of frames	1,650 frames per activity
Number of frames (Training)	1,500 frames per activity
Batch Size	32
Number of Epochs	20
Initial learning rate	$5e^4$
Number of frames (Testing)	150 frames per activity

dataset and 15,300 frames of HMDB51 dataset) is used for the validation. Figures 5- 8 illustrate the accuracy and loss curves over the 20 epochs during the training and validation for both the UCF101 and HMDB51 datasets. In Figures 5- 8, the results are determined by incorporating the different feature optimization techniques (GSA, QBGSA, and proposed INQGSA) along with the DNN classifiers of ResNet-50V2 and ResNet-101V2.

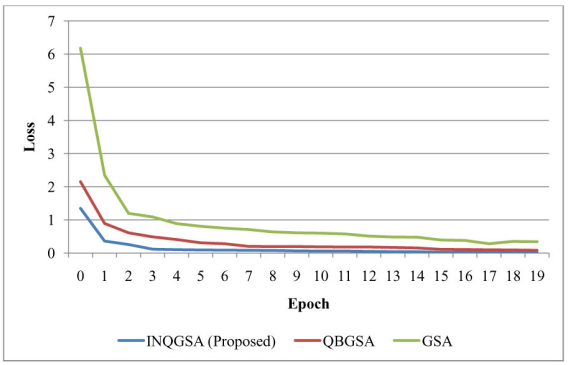
The graphs depicted in Figures 5- 8 indicate the higher oscillation of validation results in the case of GSA and QBGSA, which is due to the trapping of agents with the increase of epochs. On the other hand, the proposed INQGSA can be seen with the minor oscillations of result values. The training of the techniques can be found to be smooth compared to the validation results. The validation results for the UCF101 and HMDB51 datasets are illustrated in Tables 4 and 5, respectively. These results clearly indicate the higher accuracy and lower loss values of the proposed models. For the UCF101 dataset, the maximum validation accuracy values of 97.95% and 98.98% are attained by the proposed INQGSA+ResNet50V2 technique and the proposed INQGSA+ResNet101V2 technique, respectively. Furthermore, these values are 96.92% and 98.25% in the case of the HMDB51 dataset for the aforementioned techniques. These validation results are higher than other feature optimization techniques, which indicate the superiority of the proposed approach. It also indicates that the ResNet-101V2 attained superior performance to the ResNet-50V2.

Further, the testing results of the proposed INQGSA approach and other optimization techniques with ResNet-50V2 and ResNet-101V2 classifiers are determined in terms of precision, recall, f-measure score, and recognition accuracy. The classification results for the UCF101 and HMDB51 datasets are described in Tables 6 and 7.

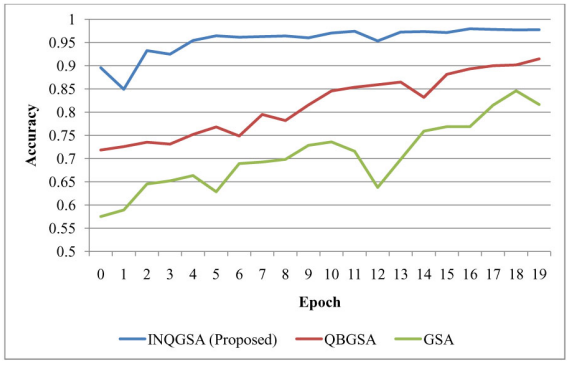
From the classification results depicted in Tables 6 and 7, it can be seen that the results values of INQGSA with both the DNN models (ResNet-50V2 and ResNet-101V2) are higher than the results evaluated with QBGSA and GSA. It indicates that the INQGSA can optimize the features more efficiently



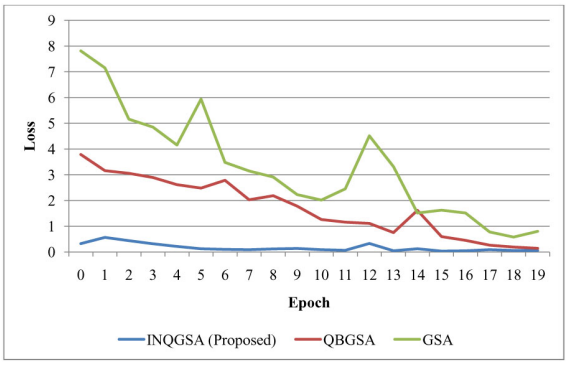
(a) Training Accuracy



(b) Training Loss

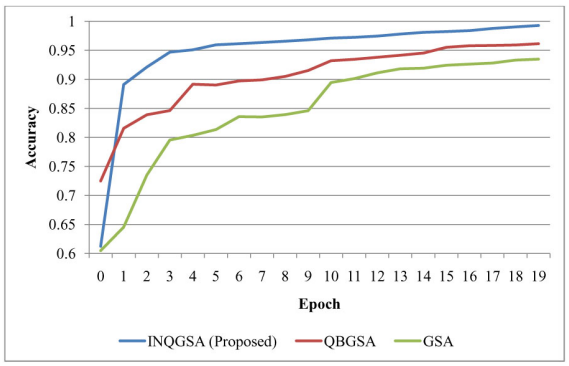


(c) Validation Accuracy

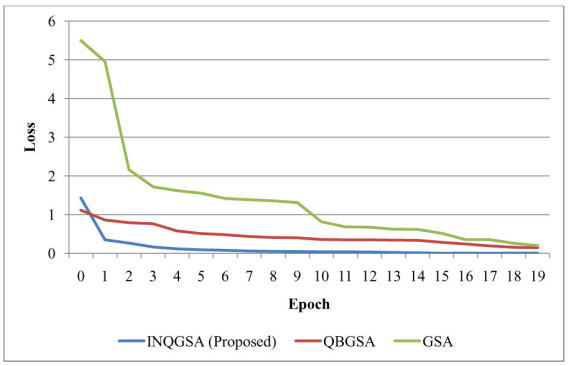


(d) Validation Loss

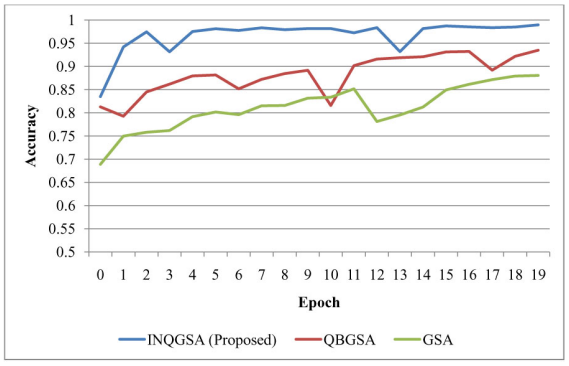
Fig. 5. Performance of ResNet-50V2 Classifier with different Feature Optimization Techniques for the UCF101 Dataset.



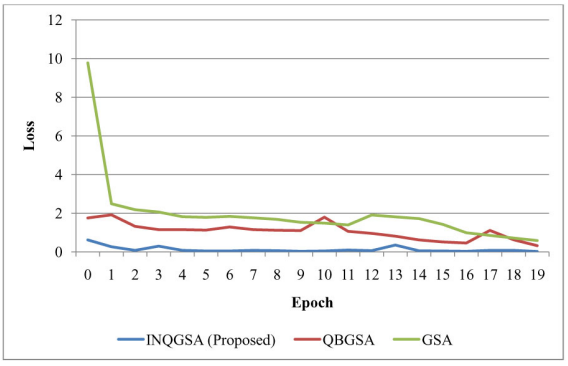
(a) Training Accuracy



(b) Training Loss

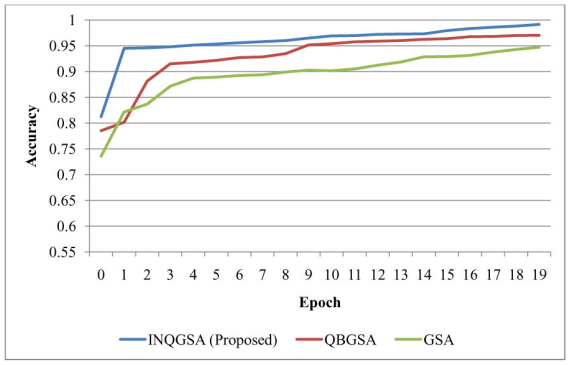


(c) Validation Accuracy

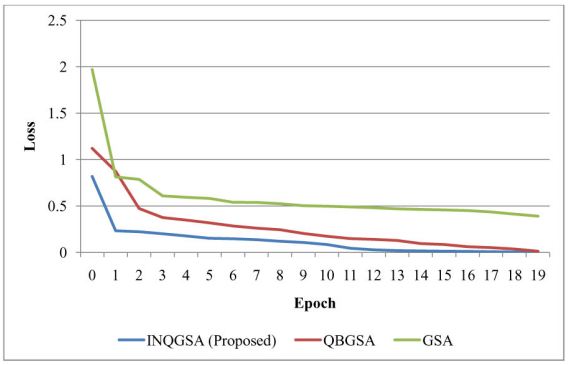


(d) Validation Loss

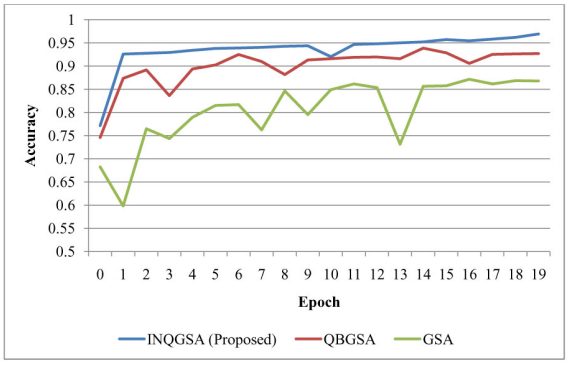
Fig. 6. Performance of ResNet-101V2 Classifier with different Feature Optimization Techniques for the UCF101 Dataset.



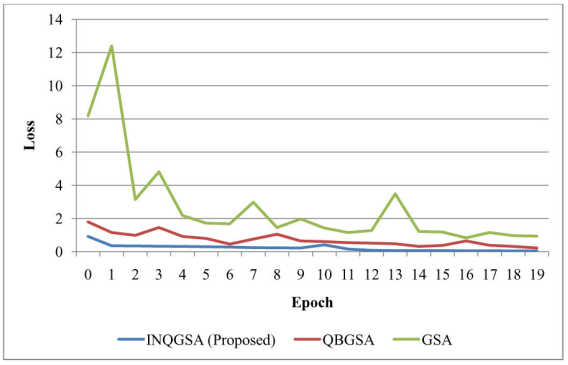
(a) Training Accuracy



(b) Training Loss

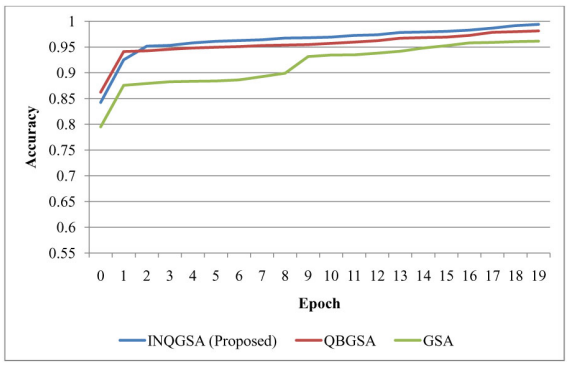


(c) Validation Accuracy

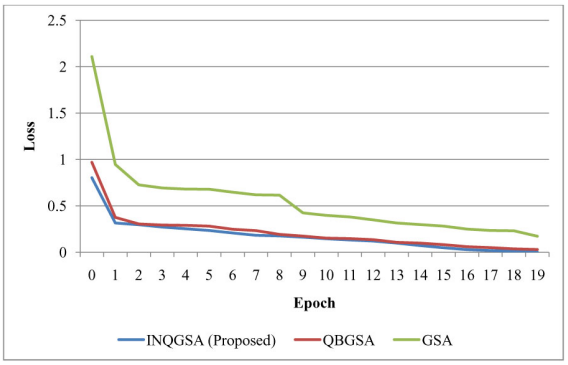


(d) Validation Loss

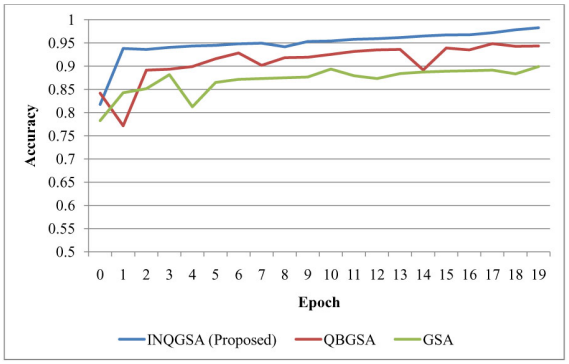
Fig. 7. Performance of ResNet-50V2 Classifier with different Feature Optimization Techniques for the HMDB51 Dataset.



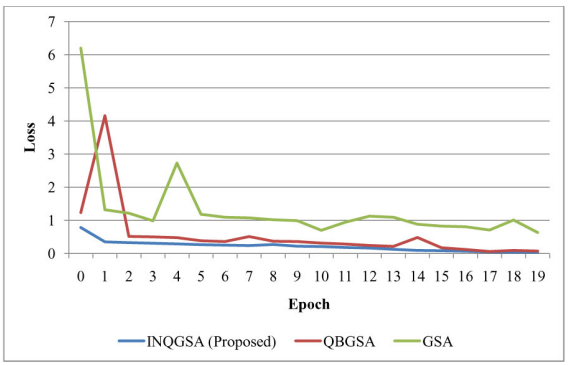
(a) Training Accuracy



(b) Training Loss



(c) Validation Accuracy



(d) Validation Loss

Fig. 8. Performance of ResNet-101V2 Classifier with different Feature Optimization Techniques for the HMDB51 Dataset.

Table 4. Validation Results for the UCF101 Dataset.

Technique	Max. Validation Accuracy	Min. Validation Loss
GSA+ResNet50V2	84.59%	0.579
GSA+ResNet101V2	88.06%	0.581
QBGSA+ResNet50V2	91.48%	0.1431
QBGSA+ResNet101V2	93.48%	0.318
Proposed INQGSA+ResNet50V2	97.95%	0.0327
Proposed INQGSA+ResNet101V2	98.98%	0.0243

Table 5. Validation Results for the HMDB51 Dataset.

Technique	Max. Validation Accuracy	Min. Validation Loss
GSA+ResNet50V2	86.85%	0.837
GSA+ResNet101V2	89.91%	0.631
QBGSA+ResNet50V2	93.85%	0.2247
QBGSA+ResNet101V2	94.82%	0.061
Proposed INQGSA+ResNet50V2	96.92%	0.0415
Proposed INQGSA+ResNet101V2	98.25%	0.0173

compared to the GSA and QBGSA. The maximum recognition accuracy values of 96.16% and 97.11% are attained by the proposed INQGSA+ResNet101V2 technique for the UCF101 and HMDB51 datasets, respectively. As the proposed techniques are superior to other optimization techniques, therefore only the proposed techniques are incorporated for further comparison with state-of-the-art techniques.

6.3 Comparative analysis

The proposed HAR system has incorporated the RGB frames for activity recognition from video datasets. Therefore, the comparative analysis of the proposed system is conducted with most of the RGB-based techniques for the experiments on the UCF101 and HMDB51 datasets. The comparative analysis of the proposed system with state-of-the-art techniques in terms of recognition accuracy is summarized in Table 8.

The proposed INQGSA approach outperformed with both the classifiers (ResNet-50V2 and ResNet-101V2) compared to the state-of-the-art techniques. For the UCF101 and HMDB51 datasets, the proposed INQGSA+ResNet101V2 technique has attained 0.78% and 1.27% higher accuracy values than the INQGSA+ResNet50V2 technique, respectively.

For the UCF101 dataset, the recognition accuracy of the proposed INQGSA+ResNet101V2 technique is 7.06% higher than MIFS (Multi-skIp Feature Stacking) (Lan *et al.*, 2015), 4.26% than Motion Map+MIFS (Sun *et al.*, 2018), 7.26% than MiCT-Net (Mixed Convolutional Tube Network) (Zhou *et al.*, 2018), 4.66% than CNN-OFF (Xu *et al.*, 2021), 3.27% than CNN (weighted product fusion) (Singh *et al.*, 2021), 4.12% than CNN (weighted average fusion) (Singh *et al.*, 2021), 4.6% than CNN (max fusion) (Singh *et al.*, 2021), 7.49% than CNN (sum fusion) (Singh *et al.*, 2021), 7.93% than CNN (spatio-

Table 6. Classification Results for the UCF101 Dataset.

Technique	Precision	Recall	F-Measure	Recognition Accuracy
GSA+ResNet50V2	86.08%	83.14%	84.58%	83.14%
GSA+ResNet101V2	87.89%	86.23%	87.05%	86.23%
QBGSA+ResNet50V2	90.44%	89.53%	89.98%	89.53%
QBGSA+ResNet101V2	94.15%	92.77%	93.45%	92.77%
Proposed INQGSA+ResNet50V2	96.17%	95.38%	95.77%	95.38%
Proposed INQGSA+ResNet101V2	96.91%	96.16%	96.53%	96.16%

Table 7. Classification Results for the HMDB51 Dataset.

Technique	Precision	Recall	F-Measure	Recognition Accuracy
GSA+ResNet50V2	88.12%	85.76%	86.92%	85.76%
GSA+ResNet101V2	90.47%	88.35%	89.40%	88.35%
QBGSA+ResNet50V2	93.48%	92.07%	92.77%	92.07%
QBGSA+ResNet101V2	94.13%	92.97%	93.55%	92.97%
Proposed INQGSA+ResNet50V2	97.09%	95.84%	96.46%	95.84%
Proposed INQGSA+ResNet101V2	98.37%	97.11%	97.74%	97.11%

Table 8. Comparison of the Proposed HAR System with State-of-the-art Techniques.

Technique	UCF101	HMDB51
MIFS (Lan *et al.*, 2015)	89.1%	65.1%
Motion Map+MIFS (Sun *et al.*, 2018)	91.9%	73.7%
MiCT-Net (Zhou *et al.*, 2018)	88.9%	63.8%
M-SVM (Sharif *et al.*, 2019)	-	92.6%
CNN-OFF (Xu *et al.*, 2021)	91.5%	67.9%
CNN (weighted product fusion) (Singh *et al.*, 2021)	92.89%	64.13%
CNN (weighted average fusion) (Singh *et al.*, 2021)	92.04%	63.87%
CNN (max fusion) (Singh *et al.*, 2021)	91.56%	62.79%
CNN (sum fusion) (Singh *et al.*, 2021)	88.67%	62.32%
CNN (spatio-temp) (Singh *et al.*, 2021)	88.23%	61.89%
CNN (spatial) (Singh *et al.*, 2021)	82.23%	57.20%
MSM-ResNets (Zong *et al.*, 2021)	93.5%	66.7%
PDaUM+DCNN (Khan *et al.*, 2021)	-	81.4%
Proposed INQGSA+ResNet50V2	95.38%	95.84%
Proposed INQGSA+ResNet101V2	96.16%	97.11%

temp) (Singh *et al.*, 2021), 13.93% than CNN (spatial) (Singh *et al.*, 2021), 2.66% than MSM-ResNets (Motion Saliency based multi-stream Multiplier ResNets) (Zong *et al.*, 2021).

For the HMDB51 dataset, the recognition accuracy of the proposed INQGSA+ResNet101V2 technique is 32.01% higher than MIFS (Lan *et al.*, 2015), 23.41% than Motion Map+MIFS (Sun *et al.*, 2018), 33.31% than MiCT-Net (Zhou *et al.*, 2018), 4.51% than M-SVM (Multi-class Support Vector Machine) (Sharif *et al.*, 2019), 29.21% than CNN-OFF (Xu *et al.*, 2021), 32.98% than CNN (weighted product fusion) (Singh *et al.*, 2021), 33.24% than CNN (weighted average fusion) (Singh *et al.*, 2021), 34.32% than CNN (max fusion) (Singh *et al.*, 2021), 34.79% than CNN (sum fusion) (Singh *et al.*, 2021), 35.22% than CNN (spatio-temp) (Singh *et al.*, 2021), 39.91% than CNN (spatial) (Singh *et al.*, 2021), 30.41% than MSM-ResNets (Zong *et al.*, 2021), and 15.71% than PDaUM (Poisson distribution along with Univariate Measures) + DCNN (Deep Convolutional Neural Network) (Khan *et al.*, 2021).

These comparisons indicate the superiority of the results for the proposed techniques over other techniques. Although the accuracy differences between the proposed techniques and other techniques are readily visible for both the datasets, a significant improvement in the results can be observed for the HMDB51 dataset. These results demonstrate that the proposed INQGSA technique considerably enhances the features that aid in the more accurate recognition of activities.

7. Conclusion

This paper proposed the INQGSA approach to optimize the features for human activity recognition. The proposed INQGSA approach intelligently updates the position of mass agents to avoid the trapping of agents in later iterations, which occurred in GSA and QBGSA. In this work, these intelligent attributes helps to improve the feature optimization for activity recognition. In the overall human activity recog-

nition system, a sequence of the latest techniques is incorporated for the different modules of activity recognition. The system incorporated the key techniques of uniform rotation invariant LBP for feature extraction, the proposed INQGSA approach for feature optimization, and deep neural network models (ResNet-50V2 and ResNet-101V2) for classification. The feature optimization technique reduces the complexity of the classifiers by feeding the selected features. The results of the proposed HAR system are evaluated for the UCF101 and HMDB51 datasets. For the UCF101 dataset, the proposed INQGSA+ResNet50V2 technique and the proposed INQGSA+ResNet101V2 techniques attained recognition accuracy of 95.38% and 96.16%, respectively. These values for the HMDB51 dataset are 95.84% and 97.11%, respectively. The comparative analysis of the proposed techniques with GSA and QBGSA based optimization techniques and state-of-the-art techniques indicates the outperformed performance of the proposed techniques.

In the future, the proposed INQGSA approach can be utilized for other applications such as network optimization, scheduling, robotic programs, etc. Moreover, the proposed HAR system can also be implemented in real time to determine abnormal activities in public places.

Acknowledgement

The authors gratefully acknowledge the Department of Computer Science and Engineering, I. K. Gujral Punjab Technical University, Jalandhar, India, for providing the opportunity to conduct this research.

References

Al-Hmouz, R. (2020). Deep learning autoencoder approach: Automatic recognition of artistic Arabic calligraphy types. *Kuwait Journal of Science*, **47**(3), 2-14.

Baldominos, A., Isasi, P. & Saez, Y. (2017). Feature selection for physical activity recognition using genetic algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2185-2192). Donostia, Spain: IEEE.

Berlin, S. J. & John, M. (2020). Particle swarm optimization with deep learning for human action recognition. *Multimedia Tools and Applications*, **79**(25), 17349-17371.

Bouachir, W., Gouiaa, R., Li, B. & Noumeir, R. (2018). Intelligent video surveillance for real-time detection of suicide attempts. *Pattern Recognition Letters*, **110**, 1-7.

Bulbul, M. F., Islam, S., Azme, Z., Pareek, P., Kabir, M. & Ali, H. (2022). Enhancing the performance of 3D auto-correlation gradient features in depth action classification. *International Journal of Multimedia Information Retrieval*, **11**, 61-76.

Chen, Y., Zhao, Z., Wang, S. & Chen, Z. (2012). Extreme learning machine-based device displacement free activity recognition model. *Soft Computing*, **16**(9), 1617-1625.

Fan, C. & Gao, F. (2021). Enhanced human activity recognition using wearable sensors via a hybrid feature selection method. *Sensors*, **21**(19), 6434(1-25).

Fang, H., He, L., Si, H., Liu, P. & Xie, X. (2014). Human activity recognition based on feature selection in smart home using back-propagation algorithm. *ISA Transactions*, **53**(5), 1629-1638.

Helmi, A. M., Al-Qaness, M. A., Dahou, A., Damaševičius, R., Krilavičius, T. & Elaziz, M. A. (2021). A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors. *Entropy*, **23**(8), 1065(1-20).

Ibrahim, A. A., Mohamed, A. & Shareef, H. (2012). A novel quantum-inspired binary gravitational search algorithm in obtaining optimal power quality monitor placement. *Journal of Applied Sciences*, **12**(9), 822-830.

Ijjina, E. P. & Chalavadi, K. M. (2017). Human action recognition in RGB-D videos using motion sequence information and deep learning. *Pattern Recognition*, **72**, 504-516.

Jaouedi, N., Boujnah, N. & Bouhlel, M. S. (2020). A new hybrid deep learning model for human action recognition. *Journal of King Saud University-Computer and Information Sciences*, **32**(4), 447-453.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942-1948). Perth, Australia: IEEE.

Kennedy, J. & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation* (pp. 4104-4108). Orlando, USA: IEEE.

Khan, M. A., Zhang, Y. D., Khan, S. A., Attique, M., Rehman, A. & Seo, S. (2021). A resource conscious human action recognition framework using 26-layered deep convolutional neural network. Multimedia Tools and Applications. *Multimedia Tools and Applications*, **80**(28), 35827-35849.

Khan, W., Daud, A., Nasir, J. A. & Amjad, T. (2016). A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait journal of Science*, **43**(4), 95-113.

Khemchandani, R. & Sharma, S. (2016). Robust least squares twin support vector machine for human activity recognition. *Applied Soft Computing*, **47**, 33-46.

Kiliç, Ş., Askerzade, İ. & Kaya, Y. (2020). Using ResNet Transfer Deep Learning Methods in Person Identification According to Physical Actions. *IEEE Access*, **8**, pp.220364-220373.

Kong, Y., Huang, J., Huang, S., Wei, Z. & Wang, S. (2019). Learning spatiotemporal representations for human fall detection in surveillance video. *Journal of Visual Communication and Image Representation*, **59**, 215-230.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T. & Serre, T. (2011). HMDB: a large video database for human motion recognition. In *2011 International conference on computer vision*, (pp. 2556-2563). Barcelona, Spain: IEEE.

Kushwaha, A.K.S., Srivastava, S. & Srivastava, R. (2017). Multi-view human activity recognition based on silhouette and uniform rotation invariant local binary patterns. *Multimedia Systems*, **23**(4), 451-467.

Lan, Z., Lin, M., Li, X., Hauptmann, A. G. & Raj, B. (2015). Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *IEEE conference on computer vision and pattern recognition*, (pp. 204-212). Boston, MA: IEEE.

Liu, L. C., Rustia, D. J. A. & Lin, T. T. (2021). Remote Surveillance Video Activity Recognition Using Spatiotemporal Convolutional Neural Networks for Greenhouse Workload Analysis. In *2021 ASABE Annual International Virtual Meeting*. American Society of Agricultural and Biological Engineers.

Mazaar, H., Emary, E. & Onsi, H. (2016). Ensemble based-feature selection on human activity recognition. In *10th International Conference on Informatics and Systems* (pp. 81-87). Giza, Egypt: ACM.

Mittal, H. & Saraswat, M. (2019). An automatic nuclei segmentation method using intelligent gravitational search algorithm based superpixel clustering. *Swarm and Evolutionary Computation*, **45**, 15-32.

Özyer, T., Ak, D. S. & Alhajj, R. (2021). Human action recognition approaches with video datasets—A survey. *Knowledge-Based Systems*, **222**, 106995 (1-36).

Pietikäinen, M., Hadid, A., Zhao, G. & Ahonen, T. (2011). Local binary patterns for still images. In *Computer vision using local binary patterns* (pp. 13-47). Springer, London.

Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, **179**(13), 2232-2248.

Serpush, F. & Rezaei, M. (2020). Complex human action recognition in live videos using hybrid fr-dl method. arXiv preprint arXiv:2007.02811.

Sharif, A., Khan, M. A., Javed, K., Gulfam, H., Iqbal, T., Saba, T., Ali, H. & Nisar, W. (2019). Intelligent human action recognition: A framework of optimal features selection based on Euclidean distance and strong correlation. *Journal of Control Engineering and Applied Informatics*, **21**(3), 3-11.

Shieh, W. Y. & Huang, J. C. (2012). Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system. *Medical Engineering & Physics*, **37**(7), 954-963.

Siddiqi, M. H., Ali, R., Rana, M., Hong, E. K., Kim, E. S. & Lee, S. (2014). Video-based human activity recognition using multilevel wavelet decomposition and stepwise linear discriminant analysis. *Sensors*, **14**(4), 6370-6392.

Siddiqi, M. H., Alruwaili, M. & Ali, A. (2019). A novel feature selection method for video-based human activity recognition systems. *IEEE Access*, **7**, 119593-119602.

Siddiqi, M. H. & Alsirhani, A. (2022). An Efficient Feature Selection Method for Video-Based Activity Recognition Systems. *Mathematical Problems in Engineering*, 5486004(1-13). DOI: 10.1155/2022/5486004.

Siddiqui, S., Khan, M. A., Bashir, K., Sharif, M., Azam, F. & Javed, M. Y. (2018). Human action recognition: a construction of codebook by discriminative features selection approach. *International Journal of Applied Pattern Recognition*, **5**(3), 206-228.

Singh, R., Khurana, R., Kushwaha, A. K. S. & Srivastava, R. (2021). A dual stream model for activity recognition: Exploiting residual-cnn with transfer learning. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, **9**(1), 28-38.

Singh, R., Kushwaha, A. K. S. & Srivastava, R. (2019). Multi-view recognition system for human activity based on multiple features for video surveillance system. *Multimedia Tools and Applications*, **78**(12), 17165-17196.

Soomro, K., Zamir, A. R. & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

Sun, Y., Wu, X., Yu, W. & Yu, F. (2018). Action recognition with motion map 3D network. *Neurocomputing*, **297**, 33-39.

Tian, Y., Zhang, J., Li, L. & Liu, Z. (2021). A Novel Sensor-Based Human Activity Recognition Method Based on Hybrid Feature Selection and Combinational Optimization. *IEEE Access*, **9**, 107235-107249.

Verma, P., Sah, A. & Srivastava, R. (2020). Deep learning-based multi-modal approach using RGB and skeleton sequences for human activity recognition. *Multimedia Systems*, **26**(6), 671-685.

Wang, A., Chen, G., Yang, J., Zhao, S. & Chang, C. Y. (2016). A comparative study on human activity recognition using inertial sensors in a smartphone. *IEEE Sensors Journal*, **16**(11), 4566-4578.

Wang, H., Ke, R., Li, J., An, Y., Wang, K. & Yu, L. (2018). A correlation-based binary particle swarm optimization method for feature selection in human activity recognition. *International Journal of Distributed Sensor Networks*, **14**(4), 1-17.

Xu, J., Song, R., Wei, H., Guo, J., Zhou, Y. & Huang, X. (2021). A fast human action recognition network based on spatio-temporal features. *Neurocomputing*, **441**, 350-358.

Xu, X., Tang, J., Zhang, X., Liu, X., Zhang, H. & Qiu, Y. (2013). Exploring techniques for vision based human activity recognition: Methods, systems, and evaluation. *Sensors*, **13**(2), 1635-1650.

Zheng, Y. (2015). Human activity recognition based on the hierarchical feature selection and classification framework. *Journal of Electrical and Computer Engineering*, 140820(1-9). DOI: 10.1155/2015/140820.

Zhou, Y., Sun, X., Zha, Z. J. & Zeng, W. (2018). Mict: Mixed 3d/2d convolutional tube for human action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 449-458). Salt Lake City, USA: IEEE.

Zong, M., Wang, R., Chen, X., Chen, Z. & Gong, Y. (2021). Motion saliency based multi-stream multiplier ResNets for action recognition. *Image and Vision Computing*, **107**, 104108 (1-8).

Submitted: 05/02/2022
Revised: 15/04/2022
Accepted: 25/04/2022
DOI: 10.48129/kjs.18531

Kuwait J.Sci., Vol.50, No.(2A),April.2023,pp(1-20)

Real time obstacle motion prediction using neural network based extended Kalman filter for robot path planning

Najva Hassan[1], Abdul Saleem[2]
[1] *Dept. of Electrical and Electronics Engineering, Government Engineering College, Thrissur,*
[2] *Dept. of Electrical and Electronics Engineering, Government Engineering College, Thrissur,*
Corresponding author: abdulsaleempk@gmail.com

Abstract

Navigation for mobile robots in dynamic environments necessitates estimating the path of dynamic obstacles, which is accomplished in this study using an enhanced kalman filter. The measured data, however, contains bias and noise. The SDAE, a deep learning-based neural network structure, delivers noise-free data that the Kalman filter uses to construct an optimal measurement noise covariance matrix. This matrix is then used by the Kalman filter to estimate an error-free obstacle path.The SDAE is trained using both the Adam and stochastic gradient learning algorithms. To ensure safe navigation, the robot's path is re-planned based on the estimated obstacle path. Numerical simulations using MATLAB demonstrate that the novel methodology is more relevant and superior to traditional Kalman and Particle filter approaches, and that it can be applied in a variety of navigational applications. In terms of computing time and robustness in closely spaced obstacles, simulation testing indicated that path planning using the proposed technique excels the hybrid A star, artificial potential field, and decision algorithms.

Keywords: Denoising autoencoder; dynamic path planning; Kalman filter; measurement noise covariance; motion prediction;

1. Introduction

As a result of recent robotics advancements, autonomous mobile robots are increasingly being employed in a wide range of applications, including military, hospitals, farm imaging, and surveillance. Mobile robots could operate in hazardous and unpredictably changing situations. Because the barriers are immovable in a static environment, path planning is rather simple, and offline path planning suffices. Path planning is a difficult problem in a dynamic environment with moving obstacles because the robot must re-plan its path to reach the destination without colliding.

To achieve intelligent navigation of mobile robots, sensor-actuator control methods are adopted. Most navigation approaches, including global navigation satellite systems and inertial navigation systems, use the Kalman filter (Wang S.L., 2013). A unique deep learning-based prediction method is developed in (Park, J.S., 2020) for generating collision-free trajectories for a robot working in an obscured environment near a human obstacle. In (Park, J.S., 2020), an occlusion-aware planner is employed to compute collision-free trajectories, resulting in improved human motion prediction accuracy. The Extended Kalman Filter and RGBD-SLAM are employed in order to solve landmark localization and build 2D and 3D maps of the environment (Khan, M.S.A., *et al.*, 2021). SLAM techniques are used on a two-wheeled mobile robot with an encoder to monitor feedback, and the robot is intelligently built to move autonomously in an indoor static environment. The authors of (Van Den Berg, *et al.*, 2005) employed road-maps for robot motion planning in dynamic scenarios. In this context, the local path planning has been developed using a depth-first search on an implicit grid. This method is applicable to any robot type in any configuration space, and the obstacle motion is unrestricted. Dynamic road maps, on the other

hand, demand additional processes for smoothing the path prior to execution, making path re-planning difficult. To handle the path planning problem and to have continuous re-planning of the path, (Volz A., *et al.*, 2019) presents a predictive route following controller. The ideal control actions for traveling along the intended path are computed here, and the path is regularly re-planned. Another approach is the one proposed in (Lin X., *et al.*, 2020) , which incorporates artificial potential field and decision tree concepts. The improved artificial potential field method addresses the problem of local minima and thus enables real-time path planning. However, the robot experienced vibrations under the influence of closely spaced obstacles. To avoid high speed obstacles, a viable two period velocity obstacle algorithm is proposed in (Liu Z., *et al.*, 2018). The first period predicts potential collisions within a limited time horizon, while the second period predicts collisions beyond that horizon. The robot's dynamic model and moving impediments have not been taken into account, resulting in lower prediction accuracy. The hybrid simulated annealing approach is utilized in (Saricicek I., *et al.*, 2022) to determine autonomous vehicle routes. An energy efficient routing and scheduling system is also offered in (Saricicek I., *et al.*, 2022) to reduce the total energy spent by the cars by taking both the traveled distance and the vehicle's weight into account. By merging vision-based estimation and control loops, in (Roggeman H., *et al.*, 2017) safe and autonomous navigation of mobile robots is achieved. To estimate the position of moving obstacles, a method based on stereo vision data is used. For powerful computation, GPU assistance is needed. In (Lin Y., *et al.* , 2017), a sampling-based path planning approach is designed for the safe operation of an unmanned aerial vehicle. The planning time is reduced using a simplified node connection. In (Zhu, Q., *et.al*, 2019), a path planner based on a recurrent fuzzy neural network (RFNN) is created to plan the trajectory and motion of mobile robots in order to accomplish a target. To improve nonlinear programming performance, RFNN integrates fuzzy logic inference and neural network learning characteristics. To improve the autonomy and intelligence of autonomous guided vehicles (AGVs) navigation control, (Ren, Z., *et.al*, 2021) presented a hybrid real-time optimum control strategy based on deep neural networks (DNNs). The motion planning problem of an AGV with static and dynamic obstacles is presented as a nonlinear optimum control problem (OCP) in (Ren, Z., *et.al*, 2021), and the optimal solution is obtained using a direct method incorporating a smooth transformation methodology. The Prognostics-aware Multi-Robot Route Planning (P-MRRP) algorithm is proposed in (Yayan, U., *et.al*, 2021) for improving the robot team's lifetime. In the P-MRRP algorithm, routes are first created using a route set generation algorithm, and then the most reliable route set is chosen by calculating PoRC based on the robot team's reliability, as well as the effect of load on the robots' path.

In (Elnagar A., 2001), the Kalman filter is utilized to forecast the future positions and orientations of moving obstacles in dynamic situations. Under the assumption that the prior position and orientation are known, the Kalman filter may efficiently anticipate obstacle positions. (Wei, H., *et al.*, 2021) proposed a method for estimating motion state based on region-level instance segmentation and the extended Kalman filter (EKF). To create optimum motion parameters, the EKF model takes into account ego-motion and integrates it along with optical flow and disparity. The Kalman filter's prediction, on the other hand, is dependent on the process noise covariance matrix R and the measurement noise covariance matrix Q. When the measurement noise covariance matrix is chosen arbitrarily, the filtering accuracy degrades. In (Mehra R, 1971), an iterative approach for obtaining unbiased and reliable estimations of Q and R has been developed. However, this iterative method can be used only for the case in which the form of Q is known and the number of unknown elements in Q is less than $n \times r$, where n is the dimension of the state vector and r is the dimension of the measurement vector. The measurement noise covariance is identified in (Diversi R. *et al.*, 2005) without any knowledge of the noise mean by considering linear discrete stochastic systems. An estimation of the measurement noise covariance is done in (Yuen K.V., *et al.*, 2013) using a probabilistic method. In (Yuen K.V., *et al.*, 2013), the Bayesian technique has been utilized to determine the optimal noise parameter estimation and associated estimation uncertainty. The noise covariance of a scalar system is estimated using the maximum likelihood approach by the authors of (Matisko P., *et al.*, 2010). However, in (Matisko P., *et al.*, 2010), they implemented a simple searching strategy that would be prohibitively expensive for larger systems. In (Shumway R.H., *et al.*, 2019), the measurement noise covariance matrix is computed using a gradient-based numerical optimization ap-

proach that can be applied to measurements taken at irregular intervals but demands a lot of computing power. The authors of (Valappil J., *et al.*, 2000) have developed a method for estimating the noise covariance matrix of an extended Kalman filter based on Monte Carlo simulations. Using a priori knowledge of the uncertainties, samples of the parameters are generated in (Valappil J., *et al.*, 2000) and provided a simplified approach for tuning the Kalman filter. An auto covariance least square method is proposed by the authors of (Odelson B.J., *et al.*, 2006) to estimate the Q and R of Kalman filter. A lagged auto covariance function between the measurements is defined in (Odelson B.J., *et al.*, 2006), which is used to develop a linear least squares formulation to estimate Q and R. A wavelet transform is proposed in (Park S., *et al.*, 2019) to estimate the time-varying measurement noise variance. The noise covariance matrix can be correctly predicted using the wavelet transform approach. The computation time, on the other hand, is longer. The authors of (Wu F., *et al.*, 2020) use temporal convolutional neural networks to accurately evaluate the measurement noise covariance matrix. The sensor data sequences are used to estimate the noise covariance via neural networks. Changes in the environment can be reflected using temporal convolutional neural networks. The approach proposed in (Wu F., *et al.*, 2020), on the other hand, has a high training cost and cannot be learned directly on the resource constrained integrated navigation platform. The enhanced Hough Transform (HT) algorithm and the Least Squares (LS) method are combined in (Gao, *et.al*, 2018) as an effective methodology for multi-objective recognition in 8-ball billiards vision system. In (Ariff, M.A.M.,, 2021), a time-series prediction technique based on the nonlinear auto-regressive exogenous neural network (NARX) algorithm is developed to forecast generator speed deviations after a system disturbance. Using the developed strategy, the author of (Ariff, M.A.M.,, 2021) is able to speed up the overall coherency detection procedure in a power system operation.
According to the literature review, the majority of dynamic path planning algorithms assume that the obstacle motion is known in advance (Xidias, 2021) or that it moves at a constant velocity (Lin X., *et al.*, 2020). In the vast majority of circumstances, however, assuming obstacle motion is impossible. Most path planning algorithms require more time to re-plan (Xidias, 2021), resulting in higher processing time (Lin Y., *et al.* , 2017) and a significant amount of computing labor (Roggeman H., *et al.*, 2017). The literature review also reveals that, dynamic path planning algorithms that use sensors for motion prediction may fail to generate a precise collision-free path due to erroneous obstacle path predictions caused by noisy data. In this study, we offer an approach for estimating the motion of obstacles in dynamic conditions, which aids the robot in avoiding obstacles, is applicable to varying velocity, and requires less computing time with higher prediction accuracy. The Kalman filter is an excellent option for predicting obstacle paths. For accurate prediction, however, knowledge of the noise error covariance matrices is essential. Furthermore, on-line processing of these matrices is often necessary for any time-varying nonlinear system, such as a mobile robot. In contrast to the use of approximation or random selection, this method employs the SDAE to determine measurement noise covariance. The following are the significant contributions of this work:

- This research develops an approach for determining obstacle motion in dynamic environments using multi-layer neural networks that is suitable to varying velocity and takes less computing time with improved prediction accuracy. The deep learning based neural network structure proposed in this work is highly reliable and robust against noise.
- Once trained, the developed stacked denoising autoencoder based extended Kalman filter is able to predict the obstacle state in the presence of both Gaussian and non- Gaussian noise.
- In terms of performance metrics such as integral squared error (ISE), mean absolute error (MAE), and integral absolute error (ISE), the developed SDAE methodology with Adam optimizer outperforms the conventional Kalman filter, Particle filter, and denoising autoencoder (DAE)based Kalman filter for both colored and Gaussian noise. As compared to (Sedighi S., *et al.*, 2019), (Ge S.S., *et al.*, 2002), and (Xidias, 2021), the developed methodology generates an optimal path in terms of processing time, path length, and obstacle avoidance.

The rest of this paper is organized as follows: Problem formulation is explained in section 2. Section

3 describes the proposed algorithm for motion prediction. Simulation results are given in section 4. Finally, section 5 presents the concluding remarks.

2. Problem formulation

The path planning problem is defined as finding a collision free path for an autonomous vehicle from a given start position to a goal position, satisfying a set of constraints. Assume that the mobile robot moves in a two dimensional (2D) space. The objective of robot the path planning is to find a path from a start

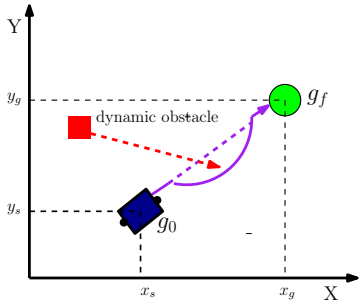


Fig. 1. Problem definition

position g_0 to a goal position g_f such that the robot avoids collision with obstacles. Let g represents the path which can be defined as

$$g = [g_0, g_1, g_2 \dots g_{n-2}, g_{n-1}, g_f] \tag{1}$$

where $g_1, g_2 \dots g_{n-1}$ are the via points.
To ensure that the path is collision free, there should be no static and dynamic obstacle in the robot's safety zone at any time i.e.,

$$\text{for } i = 1, 2, \dots, N,\ o_{pi}(t) \notin P(x(t)) \tag{2}$$

where N is the number of obstacles, $P(x(t))$ corresponds to the safety zone of the robot and o_{p_i} is the position of the obstacle. The state of the j^{th} obstacle is given by

$$o_j(t) = \begin{bmatrix} o_{p_j}(t) \\ o_{v_j}(t) \end{bmatrix} \tag{3}$$

where $o_{v_j}(t)$ is the velocity of the obstacle. Considering an obstacle with constant velocity, the relation between the position and velocity of the j^{th}obstacle using basic kinetic formula can be expressed as (Lin Y., *et al.* , 2017)

$$o_{p_j}(t) = o_{p_j}(t_0) + o_{v_j}(t_0) * (t - t_0) \tag{4}$$

The state space model of a robot can be represented as

$$\dot{r}(t) = f\left(r(t), u(t)\right) \tag{5}$$

where $r(t)$ is the state of the robot and $u(t)$ corresponds to the control vector. Besides the condition of collision free, the path should be shortest also which can be expressed mathematically as

$$g^{\star} = \text{argmin} \int_g dq \tag{6}$$

where dq is the differential of arc length of the path. In short, the problem can be defined as: Find a continuous path $g(x, y)$ from the start position $g_0(x_s, y_s)$ to the goal position $g_f(x_g, y_g)$ satisfying the constraints given by Equations (2), (4), and (5). These concepts are shown in Fig. 1

3. Proposed methodology

In a real world scenario, the robots are supposed to navigate in dynamic environments which consist of both static and dynamic obstacles. Obstacle motion prediction is a critical issue in dynamic path planning. While addressing the motion planning problem, uncertainty in the obstacle motion needs to be considered. The knowledge about obstacle motion information is very essential for the robots to complete their task effectively and safely. In most of the robot path planning algorithms, it is assumed that the obstacles move with a constant velocity or their positions are known to the robots. However, the data obtained using the sensors may not be precise and can be noisy. Hence, the goal of a successful robot navigation can be affected. The commonly adopted approach in navigation system for the obstacle path prediction is the use of extended Kalman filter. The prediction accuracy of the Kalman filter is greatly affected by the choice of measurement noise covariance matrix R. Filtering techniques and shallow neural networks such as denoising autoencoder (DAE) (Park S., *et al.*, 2019) for removing the noise have limited performance in the presence of noises other than Gaussian. In this work, a SDAE is proposed to obtain an optimum measurement covariance matrix which is used in an extended Kalman filter to estimate the states of the moving obstacle accurately. Adam and stochastic gradient descent (SGD) algorithm are used as the training algorithm to achieve maximum accuracy with reduced computation time.

3.1 Stacked Denoising Autoencoder Based Extended Kalman Filter

Kalman filter is a powerful tool for the state estimation of a system. It can provide a more accurate estimate even if the measurements are noisy. Kalman filter is capable of online real time processing and hence it can be used to estimate the position and velocity of moving obstacles in path planning problems. Kalman filter operates in two steps

- Prediction - Based on the past sensor data the next values are predicted.
- Updation - To obtain a value closer to the actual value, the predicted value is refined using the measured value.

The Kalman filter works well for the linear functions. However, obstacle motion paths can be nonlinear and so this work considers an extended Kalman filter for the obstacle path estimation. In the extended Kalman filter, the nonlinear equation is linearised using Jacobian matrix (Prevost C.G., *et al.*, 2007). Consider a moving robot car having the state

$$r_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{7}$$

where x_k, y_k, and θ_k corresponds to the x position, y position, and the orientation of the moving robot car respectively. The state space model of a robot car after linearisation is given by

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = A \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + B \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix} + v_{k-1} \tag{8}$$

where $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} \cos\theta_{k-1} * dk & 0 \\ \sin\theta_{k-1} * dk & 0 \\ 0 & dk \end{bmatrix}$, and $v_{k-1} = \begin{bmatrix} noise_{k-1} \\ noise_{k-1} \\ noise_{k-1} \end{bmatrix}$

The state at time step k is computed using the state space model, state estimate, and the control input vector at the previous time step (k-1)

$$\hat{r}_k = f(r_{k-1}, u_{k-1}) \tag{9}$$

The observation model is defined as

$$z_k = Hr_k + w_k \tag{10}$$

where w_k is the sensor noise and H matrix has the same number of rows as sensor measurements and the same number of columns as states. In a robot car model, the H matrix is defined as

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The updated state $\hat{r_k}'$ is calculated from

$$\hat{r}'_k = \hat{r}_k + K(z_k - H_k\hat{r_k}) \tag{11}$$

where K is the Kalman gain which is obtained using

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{12}$$

where R_k is the covariance of the sensor noise. Here P_k is the error covariance matrix and it is first predicted using

$$P_k = F_k P_{k-1} F_k^T + Q_k \tag{13}$$

where Q_k is the process noise covariance, F_k is equivalent to the A matrix in Equation (8) and then updated with

$$P'_k = P_k - KH_k P_k \tag{14}$$

From the above equations, it is clear that sensor noise covariance R and process noise covariance Q are important factors that determine the extended Kalman filter performance. For most of the cases, R is assumed to be constant or adjusted manually by trial and error approach. However, this may affect the performance of the extended Kalman filter and can result in an inaccurate estimation of the obstacle motion. A multi layer neural network based method is developed to estimate the obstacle state accurately. SDAE are used to denoise the sensor data. The measurement noise covariance matrix is calculated from the measured data and the noise free data obtained using the SDAE. The adaptively determined measurement noise covariance matrix is further used by the extended Kalman filter for predicting the obstacle state accurately. The training of the SDAEs is given in Algorithm 1 and the multi layer neural network based algorithm for estimating the measurement noise covariance R is described in Algorithm 2. The learning based estimation of noise covariance matrix R consists of three steps.

1. Train the neural network using a set of input-output data. A set of noise free data, S_{mi}, i=1, 2, 3, ..., n where n is the length of training data is collected which are considered as the target data of the neural network. Let T_i , be the data obtained by adding noises to S_{mi}. Both Gaussian noise and colored noise are considered in this work. Then T_i represents the input data to the neural network. The length of training data n is so chosen that the cost function C finally converges to zero. The trained DAE are stacked together such that maximum accuracy is achieved.

2. Apply the noisy measured real time data to the trained SDAE. Then the output of the neural network will be a noise free data S_{nf}.

3. Compute the measurement noise covariance matrix R using

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & \\ 0 & 0 & \Delta v^2 \end{bmatrix} \tag{15}$$

Where Δx is the difference between measured x-position and noise free x-position, Δy is defined as the difference between measured y position and noise free y position, and Δv is defined as the difference between measured velocity and noise free velocity.

Algorithm 1: Training of the SDAEs

Training;
Require
Target: Noise free data S_{mi}, $i = 1, 2, 3, \ldots, n$, n is the length of training data;
Input: Noise is added to the noise free data S_{mi} to obtain the input data;
α: Step size;
β_1, β_2 : Exponential decay rates for the moment estimates;
$C(\theta)$: Stochastic objective function with parameters θ;
θ_0: Initial parameter vector;
m_0: Initialize first moment vector;
v_0: Initialize second moment vector;
t: Initialize time step;
while θ_t *not converged* **do**
$t \leftarrow t + 1$;
$g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$ (Get gradients objective at timestep t);
$m_t \leftarrow \beta_1.m_{t-1} + (1 - \beta_1).g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2.v_{t-1} + (1 - \beta_2).g_t^2$ (Update biased second raw moment estimate);
$\hat{m}_t \leftarrow \frac{m_t}{(1-\beta^t)} g_t^2$ (Compute bias-corrected first moment estimate);
$\hat{v}_t \leftarrow \frac{v_t}{(1-\beta_t^2)}$(Compute bias-corrected second raw moment estimate);
$\theta_t \leftarrow \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$ (Update parameters);
end while;
return θ_t (Resulting parameters)
end

Algorithm 2: Online estimation of measurement noise covariance matrix R

Begin
Step 1: **Input**: Sensor data S_n
Step 2: Give the input to the trained SDAE "net$_\theta$"
for$(t = 0 : t_s)$
Step 3: Obtain the output

$$S_{nf} = net_\theta(S_n)$$

Step 4: Obtain

$$\Delta x = S_{nf}(x) - S_n(x)$$

$$\Delta y = S_{nf}(y) - S_n(y)$$

$$\Delta v = S_{nf}(v) - S_n(v)$$

Step 5: Calculate the measurement noise covariance using

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & 0 \\ 0 & 0 & \Delta v^2 \end{bmatrix}$$

Step 6: Return R
end

3.1.1 Stacked Denoising Autoencoders

Denoising autoencoders are neural networks which are the extension of autoencoders (Xing C., *et al.*, 2016). They are trained to obtain the original data from the corrupted version of it. A DAE consists of encoder-decoder and a set of hidden layers similar to that of a conventional autoencoder. But the input to the DAE is corrupted data and the decoder output is the noise free data. The working of the DAE is shown in Fig. 2. For training, a set of noise free measured data is obtained. Then the input signal $\hat{a}$ is

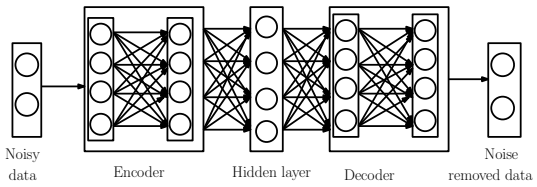


Fig. 2. Denoising autoencoder

obtained by adding noise to the noise free data, a. The noisy data $\hat{a}$ is mapped through the encoder to the hidden layer. The output of the neurons in the hidden layer is given by

$$h = f_e(W_{ih}\hat{a} + b_{ih}) \tag{16}$$

W_{ih} is the weight matrix connecting the input layer and hidden layer, f_e is the activation function of encoding layer, and b_{ih} is the bias in the hidden layer. The original data is reconstructed by the decoder through the hidden layer.

$$a_e = f_d(W_{ho}h + b_{ho}) \tag{17}$$

W_{ho} is the weight matrix connecting the output layer and hidden layer, f_d is the activation function of decoding layer, and b_{ho} is the bias in the output layer. The reconstruction error in a DAE is calculated as

$$C(a, a_e) = ||a - a_e||^2 \tag{18}$$

where a_e is the output. The cost function is minimized with respect to the DAE model weights

$$\theta = arg_\theta \; min \frac{1}{n} \sum_{i=1}^{n} C(a^{(i)}, a_e^{(i)}) \tag{19}$$

where θ corresponds to (W, b) and C is the cost function.

The DAEs are robust and provides better results when trained properly. However, its capabilities are limited and often do not perform well for data with large noise. Thus a SDAE is used in this paper. SDAEs are built by stacking DAE and have more than one hidden layer (Vincent P., *et al.*, 2010). It consists of two encoding layers and two decoding layers. The output of the first encoding layer is given as the input data to the second encoding layer. In this work, a data set of 5000 samples are used to train the SDAE. The additive white gaussian noise and the colored noise are added to the data set which gives the input data for training purpose. The developed SDAE consists of two hidden layers with 20 neurons in each layer. Initially, the first DAE is trained and the weights w, bias b and features h are obtained. These features h are provided as the input to the next encoding layer. Layer wise training of DAE is performed and are stacked together. Adam and stochastic gradient descent algorithms are used as the optimization algorithms for learning. The gradient estimate is computed by using a loss function in the stochastic gradient descent algorithm. The learning rate determines the magnitude of the parameter updation. Choosing of the learning rate is a non trivial task in stochastic descent algorithm. The advantages of both adaptive gradient and RMSprop algorithms are combined in an Adam optimizer. The adam algorithm updates the gradient (m_t) and squared gradient (v_t) , with the hyper-parameters β_1, β_2controlling the exponential decay rates of these moving averages. The moving averages are estimates of the gradient's first moment

(the mean) and second raw moment (Soydaner, D.,, 2020). The pseudo code ofthe Adam algorithm is explained in Algorithm 1. It works efficiently for problems with noisy and sparse gradients. The SDAE based extended Kalman filter is used to estimate the path of moving obstacle, which is explained in Algorithm 3.

Algorithm 3: Proposed SDAE based extended Kalman filter for obstacle motion prediction

Begin

Step 1: Input trained SDAE net_θ, noisy data S_n.

Step 2: Obtain noise free data S_{nf}.

$$S_{nf} = net_\theta(S_n)$$

Step 3: Calculate the measurement noise covariance R .

$$R = \begin{bmatrix} \Delta x^2 & 0 & 0 \\ 0 & \Delta y^2 & 0 \\ 0 & 0 & \Delta v^2 \end{bmatrix}$$

Step 4: Adjust the Kalman gain K.

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

using updated R.

Step 5: Estimate the moving obstacle state

$$\hat{r}'_k = \hat{r}_k + K(z_k - H_k \hat{r_k})$$

end

3.2 Path planning in dynamic environments

In real time applications, the environment that a robot has to navigate can be static or dynamic. If the environment is dynamic then the robot should be able to predict the obstacle motion so as to successfully avoid a possible collision with the obstacle. The schematic diagram of the proposed method for path planning in a dynamic environment is shown in Fig. 3. The developed method is divided into two phases. Initially, the path is planned considering the static obstacles. In the second phase, the obstacle motion is predicted and the robot path is re-planned so that the collision is avoided.

3.2.1 Initial path generation

Initially, an offline path planning is done assuming that the environment is static. Let the start and goal position be g_0 and g_f respectively. In this approach, we are assuming that the current position of the moving obstacles is known to us. Let the configuration space be C_{space}. It consists of a collision free space C_{fs} and a space with obstacles C_{obs}. Randomly choose a set of configurations P and check collision at each selected n closest neighbor points. Thus the shortest path is calculated initially using the algorithm proposed in (Chen J., *et al.*, 2019) within a time period t.

3.2.2 Obstacle motion prediction and path re-planning

In this work, Algorithm 3 is used to predict the obstacle motion. The obstacle path is predicted for the given time horizon t which is the time required to calculate the initial path. Now check if an intersection of the initially planned robot path and the estimated obstacle path exists or not. If an intersection of the two paths occurs then the robot path is re-planned. The new path is now the current robot path and the process of checking obstacle path and robot path is continued and re-planning is done when both paths intersect until the goal position is reached.

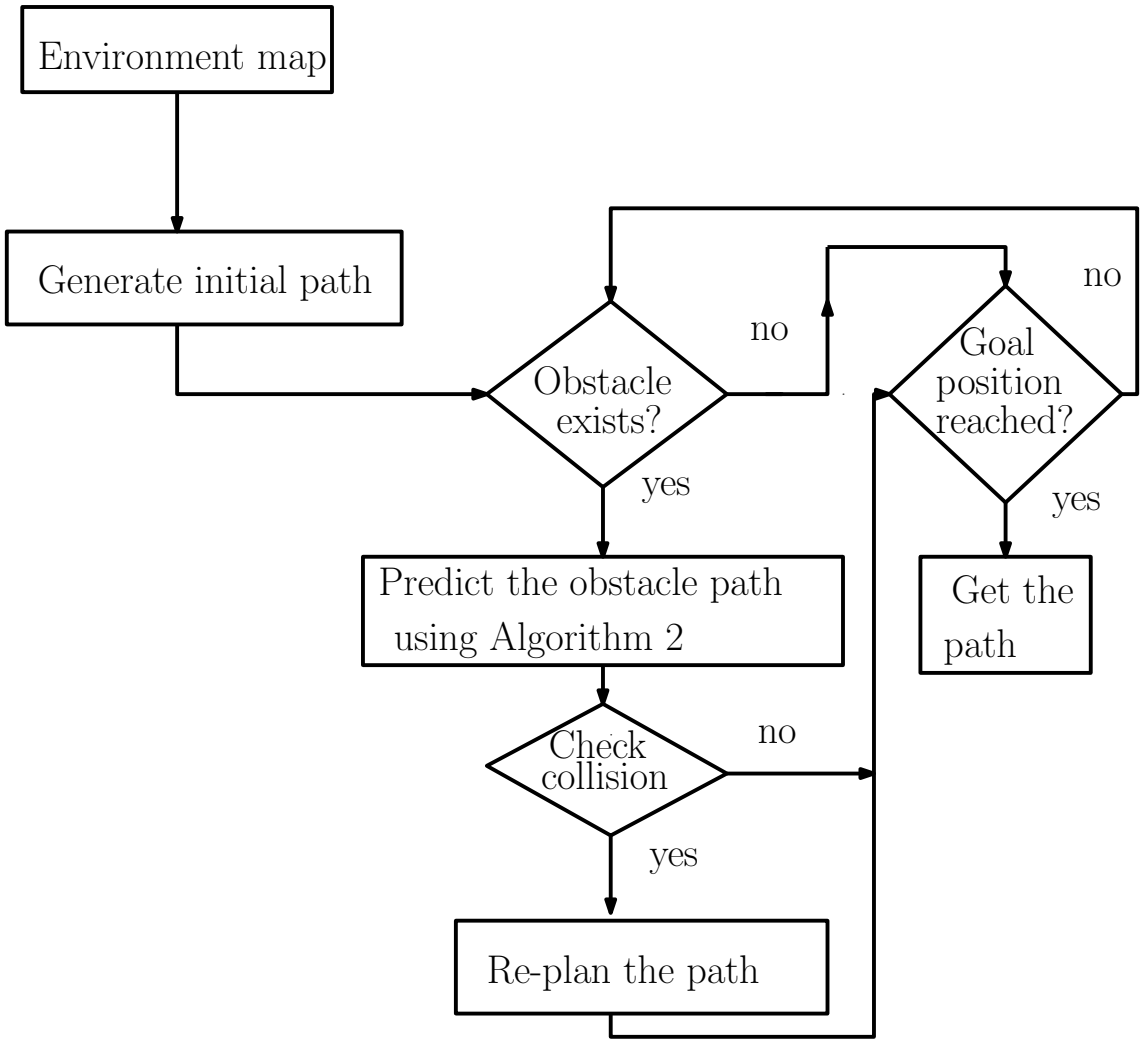


Fig. 3. Schematic diagram of proposed method for path planning in dynamic environments

4. Results and discussions

In this section the effectiveness of the developed algorithm for predicting the obstacle motion is validated using various simulations. A comparative assessment of the prediction algorithm is also performed by comparing with conventional Kalman filter, Particle filter and denoising autoencoder based Kalman filter. In order to assess the efficacy of the proposed method, various performance metrics such as IAE, ISE and MAE in the obstacle path prediction are analyzed.

$$ISE = \int_0^t e(t)^2 dt \tag{20}$$

The accumulated error is denoted by the integral of absolute error and is obtained by

$$IAE = \int_0^t |e(t)| dt \tag{21}$$

where e(t) is the difference between the obstacle's actual and estimated path. The performance of the algorithm is tested and validated for both static and dynamic obstacles. The performance of the proposed algorithm is evaluated using MATLAB simulated environments by comparing it with path planning algorithms (Sedighi S., *et al.*, 2019),(Ge S.S., *et al.*, 2002), and (Xidias, 2021).

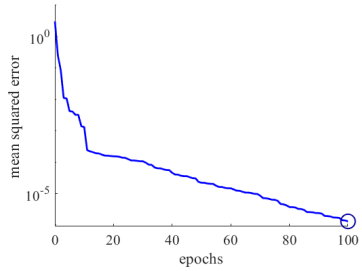


Fig. 4. Performance plot of of neural network

4.1 Neural Network training

The objective of neural network training is to generate SDAEs which gives a noise free data from a noisy data. MATLAB 2020a is used in this work to implement the SDAE. The pioneer-1 mobile robot data set is used for training the neural network. This noise free data set consists of sensor readings of pioneer-1 mobile robot which are the targets or desired outputs of neural network. The input to the neural network during the training is obtained by adding noises to the pioneer 1 data. We have considered both colored and white noises. The deep neural network structure used here consists of two hidden layers. The weights and bias are tuned using both Adam and stochastic gradient descent algorithms. The sigmoid function is used as the activation function for all the layers. Once the neural network is trained, the SDAEs will provide a noise free data if a noisy data is given as input to it. The parameters for training the SDAEs are given in Table 1. The performance plot which is the variation of the training record error values against the number of training epochs is shown in Fig. 4. At the end of the training phase, mean squared error reaches a value of order 10^-5. The small value of the mean squared error implies that the desired outputs and the neural networks outputs for the training set have become very close to each other.

Table 1. Parameters for training stacked denoising autoencoder

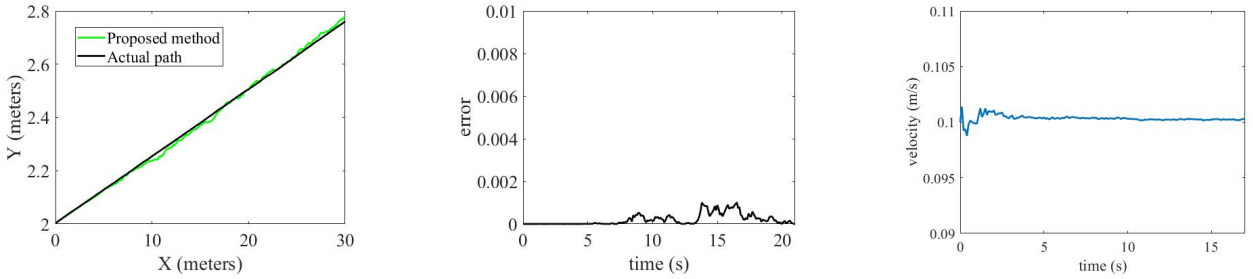
Parameters	**Value**
Learning rate	0.02
Number of epochs	100
Number of training data sequences in each iteration	100
Learning algorithm	Adam

The trained SDAEs are used to find the measurement noise covariance of the extended Kalman filter for estimating the obstacle path. The proposed algorithm is implemented on i7 core, 32gb laptop. The performance of the proposed SDAE based extended Kalman filter for estimating the obstacle path is discussed subsequently.

4.2 Performance of the stacked denoising autoencoder based extended Kalman filter

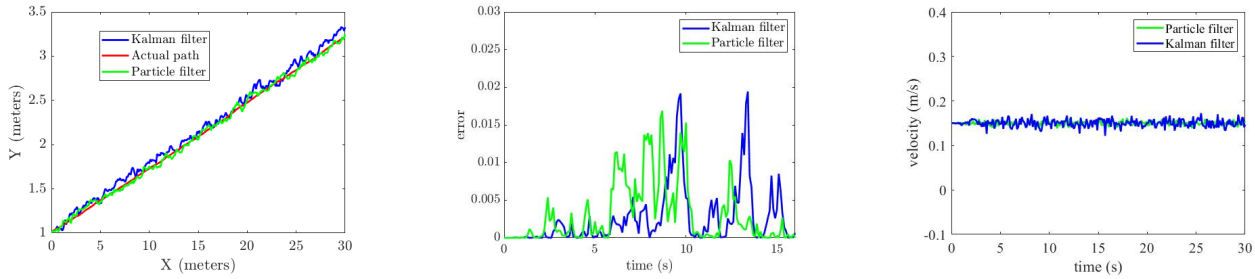
In this work, the role of the extended Kalman filter is to estimate the obstacle path. The accuracy of prediction using extended Kalman filter is dependent on the Kalman gain which further depends on the measurement noise covariance matrix. The SDAEs are trained using Algorithm 1 and are used to estimate the measurement noise covariance matrix using Algorithm 2 described in section 3. Initially, an obstacle moving with a constant velocity is considered.

The initial position of the moving obstacle is measured and is given as input to the trained SDAEs. Then the output of SDAEs gives noise free measured data. Now the measurement noise covariance



(a) Obstacle path **(b)** Estimation error **(c)** Velocity estimation

Fig. 5. Performance of the SDAE based Kalman filter



(a) Obstacle path **(b)** Estimation error **(c)** Velocity estimation

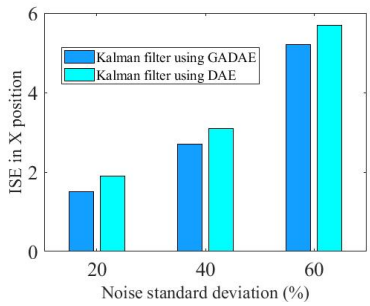
Fig. 6. Performance of the conventional Kalman filter and Particle filter

matrix can be found using Equation (15), which is computed as $R = \begin{bmatrix} 0.012 & 0 & 0 \\ 0 & 0.015 & 0 \\ 0 & 0 & 0.023 \end{bmatrix}$
The Kalman gain is calculated by substituting the estimated measurement covariance matrix in Equation (12). The obstacle path is estimated using Equations (9)-(14) repeatedly. The estimated obstacle path is shown in Fig. 5a. The actual path of the obstacle is calculated theoretically by using the basic kinetic formula given by Equation (4) and it is plotted in the same figure. From 5a, it is clear that the estimated obstacle path using the proposed algorithm follows the actual path of the obstacle. The error in the estimated path which is computed as

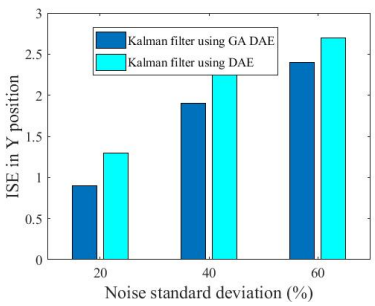
$$\text{error} = \sqrt{(\text{actual path} - \text{estimated path})^2}$$

is plotted in Fig. 5b. The maximum error in estimation is of the order of 10^-3 which is negligible and converges to zero. The velocity profile of the moving obstacle estimated using the SDAE based extended Kalman filter is shown in Fig. 5c. The estimated velocity of the moving obstacle remains constant with time and follows the actual velocity.

To evaluate the performance of proposed method, it is compared with conventional Kalman filter and Particle filter (Berntorp K., *et al.*, 2016). Fig. 6a shows the actual and estimated paths of an obstacle. It is obvious from this figure that the estimated path deviates from the actual path for both Kalman and Particle filters. Fig. 6b shows the error in the estimated path which is more than the error obtained while using the SDAE based Kalman filter and is not negligible. The estimation error is not negligible for both Kalman and Particle filters. The velocity of the moving obstacle estimated is given in Fig. 6c. The estimated velocity does not remain constant and produced oscillations. Comparing Figs. 5 and 6, it can be illustrated that the SDAE based Kalman filter outperforms the conventional Kalman filter and Particle filter by predicting the obstacle path and velocity more precisely. Table 2 summarizes a comparison of the performance of the developed prediction algorithm with that of the traditional Kalman filter, the particle filter, and the Kalman filter using DAE. As demonstrated in the table, the proposed method clearly outperforms existing methods [conventional Kalman filter, Particle filter, and Kalman filter using DAE]



(a) Integral squared error in prediction of X position



(b) Integral squared error in prediction of Y position

Fig. 7. Performance of the SDAE based extended Kalman filter (effect of noise)

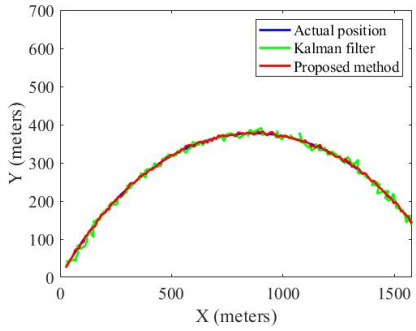
in terms of ISE, IAE, and MAE. Since, the proposed SDAE based extended Kalman filter can predict an error free obstacle path, it can be used in applications like welding and drawing robots where a precise and error free estimated obstacle path is required. Initially, the weights of the SDAEs are randomly chosen. The encoder performance will not be satisfactory if the measured data consists of large noise. The weights can be optimized using Genetic algorithm and thereby the performance of the SDAE can be improved. Gaussian noises of different standard deviation such as 20%, 40% and 60% are added to the measured data. The measurement noise covariance is computed using the SDAE (i) with randomly chosen initial weights and (ii) with Genetic algorithm optimized weights. The computed measurement covariance matrix in both cases is used to predict the obstacle position. The integral squared error in the estimated x and y position in each case is shown in Fig. 7. The Kalman filter using SDAE with optimized weights has better performance as compared to the Kalman filter using SDAE with randomly chosen weights.

Table 2. Comparison of obstacle path prediction algorithms (linear motion)

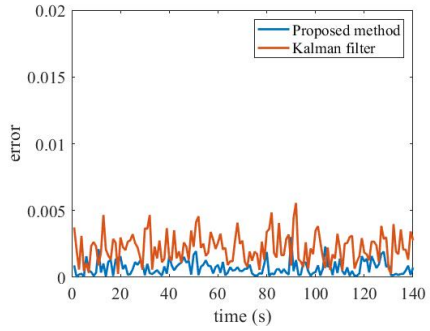
Algorithm	ISE	IAE	MAE
Proposed method	0.421	0.212	0.023
Conventional Kalman filter	4.543	2.276	0.562
Particle filter	3.213	1.562	0.287
Kalman filter using DAE	0.496	0.295	0.031

4.2.1 Obstacle with nonlinear path

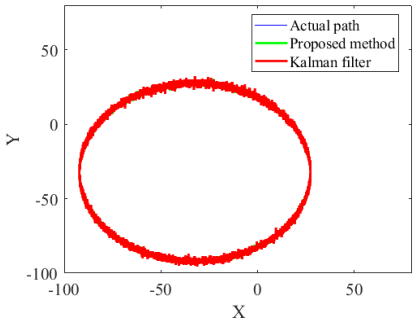
Let the obstacle be a mobile robot car with state space model given by Equation (8), which moves along a nonlinear path. To evaluate the robustness of the developed algorithm the colored noise is added to the raw data. Pink noise, Brownian noise, and Azure noise are generated with inverse frequency power $\alpha = 1$, $\alpha = 2$, and $\alpha = -1$ respectively. The noisy measured data are given as inputs to the trained SDAEs which give noise free data as outputs. The measurement noise covariance matrix is determined using Algorithm 2 and is computed as $R = \begin{bmatrix} 0.21 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$.
The measurement noise covariance matrix calculated is used for the computation of Kalman gain. The non linear path of obstacle is predicted using the SDAE based extended Kalman filter. Fig. 8a shows the estimated obstacle path using conventional extended Kalman filter and SDAE based extended Kalman filter. It is observed from this figure that the SDAE based extended Kalman filter is capable of estimating



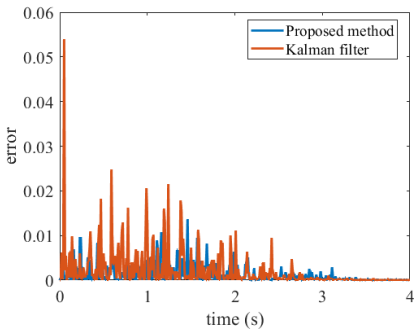
(a) Obstacle path estimated



(b) Error in estimated obstacle path



(c) Obstacle path estimated (circular path)

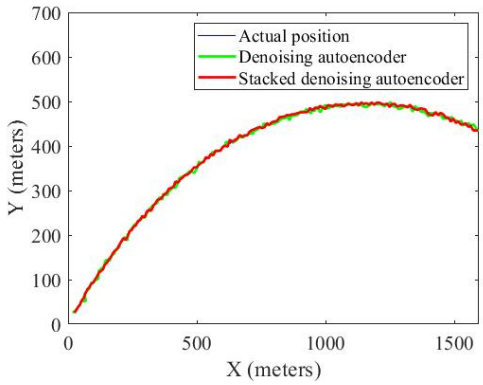


(d) Error in estimated obstacle path (circular path)

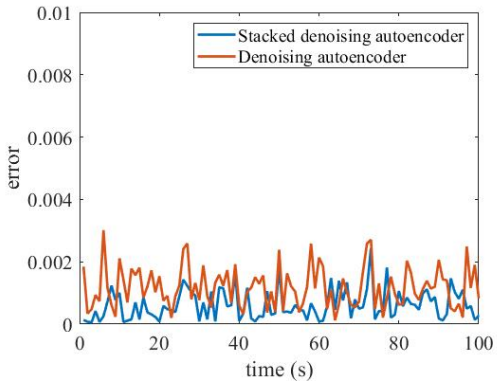
Fig. 8. Comparison of the SDAE based Kalman filter and conventional Kalman filter (nonlinear motion)

the nonlinear path more accurately as compared to the conventional extended Kalman filter. This observation is clearer from Fig. 8b which shows the estimated errors for the both methods. The estimated error is negligible for the proposed method. In Fig. 8c, the circular path predicted using both the traditional Kalman filter and the SDAE based Kalman filter is illustrated. The suggested technique has a higher estimation accuracy, as shown in Fig. 8d. Even though the error converges to zero in both cases, the conventional Kalman filter's maximum estimation error is substantial.

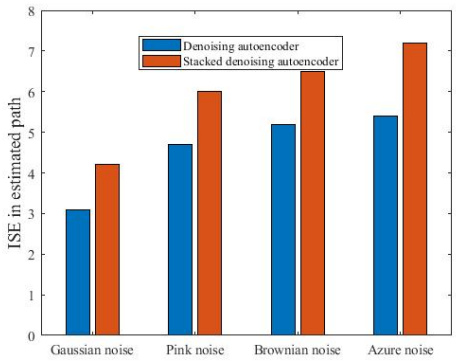
Neural network model with single layer fails to understand the training data set properly and produce results with error. More layers are added to extract more features from the data set. Thus, to produce an accurate output denoising autoencoder with stacked hidden layers are used. When SDAE and DAE are employed for determining the measurement noise covariance matrix R of the Kalman filter, the estimated nonlinear path and accompanying errors are shown in Figs. 9a and 9b, respectively. These figures demonstrate that the SDAE-based method produces the least amount of inaccuracy. To further understand the effectiveness of the proposed SDAE method, the integral squared error for both methods with Gaussian and the three colored noises are shown in Fig. 9c. In the presence of colored noise SDAE has better performance as compared to shallow neural network denoising autoencoder. The choosing of learning rate is one of the challenge in the stochastic gradient descent algorithm. Large learning rate results in the dwindling at minimum and small learning rate causes slow convergence. To increase the robustness of the stochastic gradient algorithm, Adam optimizer is used. The obstacle path is estimated using Kalman filter whose measurement noise covariance matrix are determined using SDAEs trained using both (i)Adam and (ii) stochastic gradient descent algorithms. During training both Gaussian noise and colored noise are considered. The performance of the proposed method with Adam and stochastic gradient descent learning algorithm is also analyzed which is shown in Fig. 10. A comparison of the performance of proposed method with existing algorithms in predicting the non linear motion of the obstacle is given in Table 3. The Adam optimizer has a better performance as compared to the stochastic gradient descent algorithm for both the colored and the Gaussian noises.



(a) Obstacle path



(b) Estimation error



(c) Performance comparison

Fig. 9. Comparison of stacked denoising autoencoder and denoising autoencoder

4.3 Performance of the proposed prediction algorithm in simulated environments

The performance of the proposed motion prediction algorithm is quantitatively tested in MATLAB simulated environments. In the simulation scenario 1, a dynamic environment with three moving obstacles shown in Fig. 11 is considered. Let the start position of the robot be (0,0) and the goal position be (12,10). Initially, the path is planned offline considering that the obstacles are static. The moving obstacles are detected using ultrasonic sensor. Once the dynamic obstacles are detected, the obstacle path has to be estimated to ensure collision free navigation. The obstacle path is predicted using the Kalman filter where the Kalman gain is calculated using Equation (12) for which the measurement noise covariance matrix is to be determined. The measurement noise covariance matrix is computed using Equation (15) and is obtained as $R = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}$.
The estimated obstacle path is compared with the robot path planned initially. From Fig. 11, it is clear that the initially planned path collides with the obstacle path so the path is to be re-planned. Thus, an optimal and collision free path is obtained. The uncertainty in prediction of the obstacle path using both the Kalman filter and the SDAE based Kalman filter is depicted in Fig. 12. The uncertainty in obstacle path prediction is large for the conventional Kalman filter which will affect the robot navigation in applications that require precise path.

4.3.1 Comparison of the performance of the proposed algorithm

To evaluate the efficacy of the proposed path planning algorithm using SDAE based extended Kalman filter, the proposed method is compared with that of (i) hybrid A star (ii) artificial potential field (iii) dynamic path planning using decision algorithm. The path length, computation time, and the ability to obtain collision free path in closely spaced obstacles are considered here for evaluation. The computation

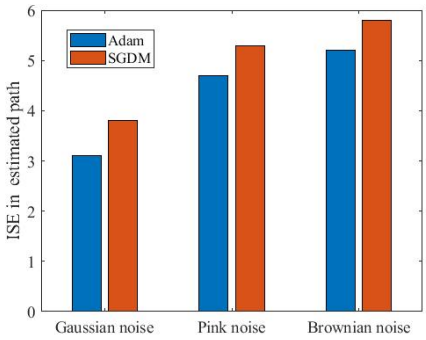


Fig. 10. Comparison of Adam and SGDM

Table 3. Comparison of obstacle path prediction algorithms (non-linear motion)

Algorithm	ISE	IAE	MAE
Proposed method (Adam optimizer)	0.534	0.158	0.021
Proposed method (Stochastic method)	0.942	0.382	0.043
Conventional Kalman filter	3.573	1.416	0.328
Kalman filter using DAE	1.32	0.4382	0.064

time is obtained using MATLAB 2020a. A MATLAB simulation environment is considered with both static and dynamic obstacles (scenario 2). The initial position of the robot is (8,0) and the goal position is (10,10). The proposed path planning algorithm estimates the obstacle path using SDAEs based extended Kalman filter whereas in the hybrid A star method, the obstacle motion is assumed to follow a constant velocity. The robot path planned using the proposed algorithm is shown in Fig. 13a. The initial planned path collides with the obstacle path and is re-planned. The path obtained using hybrid A star algorithm is given in Fig. 13b. The hybrid A star algorithm calculate the cost function at each node and finds the optimal path. Comparing Figs. 13a and 13b, it can be elucidated that the proposed path planning algorithm is able to find the shortest and optimal path from the initial position to final position and thus, the proposed algorithm outperforms the hybrid A star path planning algorithm. The path achieved by the potential field algorithm in the dynamic environment is shown in Fig. 13c. The dynamic obstacle is having a random motion and is shown in Fig. 13c. The potential field algorithm fails to achieve a collision free path when the obstacles are closely packed. The proposed algorithm finds the shortest and collision free path from the start position to the goal position when compared to hybrid A star and

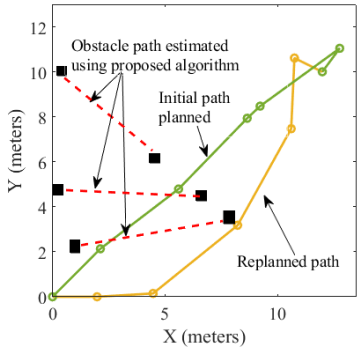


Fig. 11. Path planning (scenario 1)

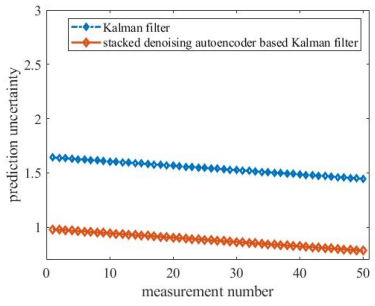
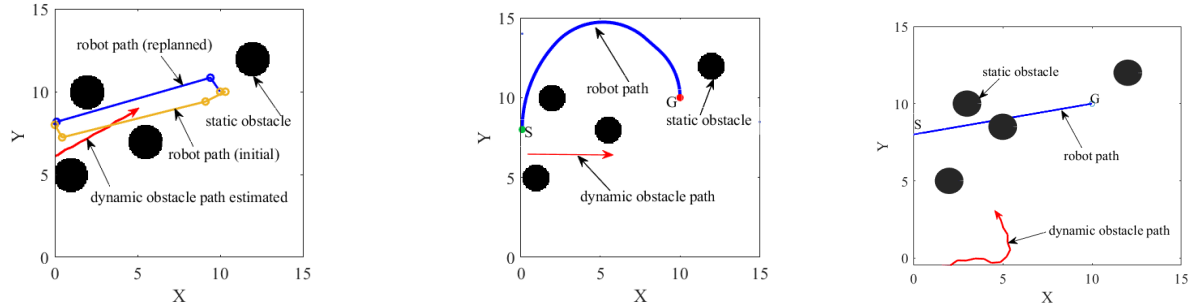
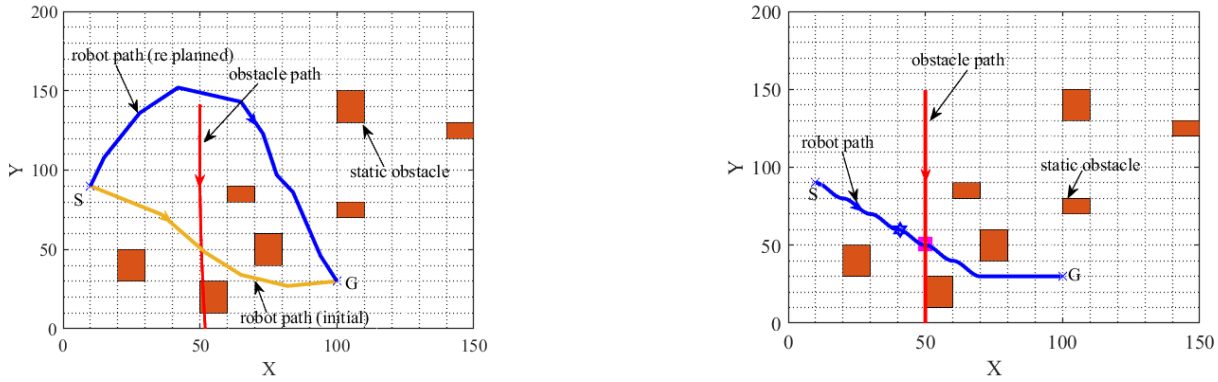


Fig. 12. Uncertainty in prediction



(a) Proposed algorithm **(b)** Hybrid A star **(c)** Artificial potential field

Fig. 13. Comparison of performance of proposed algorithm (scenario 2)



(a) Path planning using proposed algorithm **(b)** Path planning using decision algorithm

Fig. 14. Comparison of performance of proposed algorithm (scenario 3)

artificial potential filed algorithms.

The suggested algorithm is compared to the decision algorithm (Xidias, 2021), which takes both dynamic and static impediments into account. When the obstacle enters the threshold domain, the robot's velocity is reduced, and the robot must wait until the obstacle departs the threshold region, according to the decision algorithm. When the distance between the obstacle and robot exceeds the threshold value, the robot's velocity is boosted, allowing it to approach the goal. The path planning in scenario 3 using the decision algorithm is depicted in Fig. 14b. The robot must wait till the obstruction has passed, resulting in a longer computation time. The presented algorithm, as shown in Fig. 14a, re-plans the robot path when there is a collision between the obstacle and the robot path. The computation time in each of the algorithms is computed using MATLAB 2020a. The computation time is minimum for the proposed algorithm while compared to decision algorithm. In Table 4, a comparison of the suggested algorithm with the existing path planning algorithms is given. Analyzing the simulation results, it can be concluded that the SDAE based extended Kalman filter with Adam optimizer predict the obstacle path precisely. The proposed algorithm produced negligible error in the presence of both colored (brown, pink, and azure) and white noise. Also, the prediction uncertainty is less for the proposed algorithm which is a key factor in robot navigation. By accurately predicting the obstacle motion, the robot is able to achieve a collision free navigation in the dynamic environment. The developed algorithm outperforms the conventional Kalman filter and the denoising based extended Kalman filter. In comparison to the (Sedighi S., *et al.*, 2019), (Ge S.S., *et al.*, 2002), and (Xidias, 2021), path planning employing the developed methodology is faster and more robust in narrow passages.

Table 4. Comparison of path planning algorithms (dynamic environment)

Algorithm	Computation time (s)	Robustness in narrow passages
Proposed method (scenario 3)	104.643	yes
Decision algorithm (scenario 3)	247.867	yes
Proposed method (scenario 2)	64.342	yes
Hybrid A star (scenario 2)	78.249	yes
Artificial potential field (scenario 2)	68.214	no

5. Conclusion

A SDAE-based extended Kalman filter is proposed in this paper for predicting obstacle motion in dynamic scenarios. The SDAE is a deep neural network whose input is a noisy sensor data and output is the noise free data. The noisy and noise free data is used to get the measurement noise covariance matrix of the extended Kalman filter which is used to determine the path of a moving obstacle. To train the neural network, a set of noise free data are collected which are considered as the targets for the training purpose. The input of the SDAE during training is obtained by adding noises to the target data. Once the SDAE is trained then it can give the optimum measurement covariance matrix. The SDAE is capable of effectively denoising the measured data in the presence of both Gaussian noise and colored noise. MATLAB simulations are carried to predict the path of moving obstacle with conventional extended Kalman filter, Particle filter and by using the proposed SDAE based extended Kalman filter. The results illustrated that the extended Kalman filter using the SDAE gives a much accurate path for both linear and nonlinear obstacle paths. The simulation study also illustrated that the ISE, IAE, and MAE in the estimated obstacle path is very less with the SDAE based extended Kalman filter whose learning algorithm is Adam. But the training time is more for an Adam optimizer while compared to stochastic descent algorithm. Different scenarios are considered in MATLAB simulations to test the effectiveness of the proposed method for determining the optimal path in a dynamic environment with multiple impediments. Using MATLAB simulated testing environments, the performance of the proposed method in path planning is compared against hybrid A star, artificial potential field, and decision algorithms. The suggested methodology achieves an optimal collision-free path with minimal computing time in various testing scenarios.

ACKNOWLEDGMENT

This work was supported by All India Council for Technical Education-National Doctoral Fellowship (NDF-RPS) scheme.

References

Ariff, M.A.M., 2021. A new intelligent time-series prediction technique for coherency identification performance enhancement. Kuwait Journal of Science, 48(4).

Berntorp, K. & **Di Cairano, S., 2016, July.** Particle filtering for online motion planning with task specifications. In 2016 American Control Conference (ACC) (pp. 2123-2128). IEEE

Chen, J., Zhou, Y., Gong, J. & **Deng, Y., 2019, July.** An improved probabilistic roadmap algorithm with potential field function for path planning of quadrotor. In 2019 Chinese Control Conference (CCC) (pp. 3248-3253). IEEE.

Diversi, R., Guidorzi, R. & **Soverini, U., 2005.** Kalman filtering in extended noise environments. IEEE Transactions on Automatic Control, 50(9), pp.1396-1402.

Elnagar, A. 2001, July. Prediction of moving objects in dynamic environments using Kalman filters. In Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515) (pp. 414-419). IEEE.

Gao, J., He, Q., Gao, H., Zhan, Z. and Wu, Z., 2018. Design of an efficient multi-objective recognition approach for 8-ball billiards vision system. Kuwait Journal of Science 45.1 (2018).

Ge, S.S. & **Cui, Y.J., 2002.** Dynamic motion planning for mobile robots using potential field method. Autonomous robots, 13(3), pp.207-222.

Khan, M.S.A., Hussian, D., Ali, Y., Rehman, F.U., Aqeel, A.B. and Khan, U.S., 2021, November. Multi-Sensor SLAM for efficient Navigation of a Mobile Robot. In 2021 4th International Conference on Computing & Information Sciences (ICCIS) (pp. 1-5). IEEE.

Lin, X., Wang, Z.Q. & **Chen, X.Y., 2020, May.** Path Planning with Improved Artificial Potential Field Method Based on Decision Tree. In 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS) (pp. 1-5). IEEE.

Lin, Y. & **Saripalli, S., 2017.** Sampling-based path planning for UAV collision avoidance. IEEE Transactions on Intelligent Transportation Systems, 18(11), pp.3179-3192.

Liu, Z., Jiang, Z., Xu, T., Cheng, H., Xie, Z. & **Lin, L., 2018, May.** Avoidance of high-speed obstacles based on velocity obstacles. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7624-7630). IEEE.

Matisko, P. & **Havlena, V., 2010.**Noise covariances estimation for Kalman filter tuning. IFAC Proceedings Volumes, 43(10), pp.31-36.

Mehra, R. 1970. On the identification of variances and adaptive Kalman filtering. IEEE Transactions on automatic control, 15(2), pp.175-184.

Odelson, B.J., Rajamani, M.R. & **Rawlings, J.B., 2006.** A new autocovariance least-squares method for estimating noise covariances. Automatica, 42(2), pp.303-308.

Park, J.S. & Manocha, D., 2020. HMPO: human motion prediction in occluded environments for safe motion planning. arXiv preprint arXiv:2006.00424.

Park, S., Gil, M.S., Im, H. & **Moon, Y.S., 2019.** Measurement noise recommendation for efficient Kalman filtering over a large amount of sensor data. Sensors, 19(5), p.1168.

Prevost, C.G., Desbiens, A. & **Gagnon, E., 2007, July.** Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In 2007 American control conference (pp. 1805-1810). IEEE.

Ren, Z., Lai, J., Wu, Z. and Xie, S., 2021. Deep neural networks-based real-time optimal navigation for an automatic guided vehicle with static and dynamic obstacles. Neurocomputing, 443, pp.329-344.

Roggeman, H., Marzat, J., Derome, M., Sanfourche, M., Eudes, A. & **Le Besnerais, G., 2017.** Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 2090-2099).

Saricicek, I., Keser, S.B., Cibi, A. and Ozdemir, T., 2022. Energy Efficient Routing and Task Scheduling for Autonomous Transport Vehicles in Intra Logistics. Kuwait Journal of Science, 49(1).

Sedighi, S., Nguyen, D.V. &**Kuhnert, K.D., 2019, April.** Guided hybrid A-star path planning algorithm for valet parking applications. In 2019 5th international conference on control, automation and robotics (ICCAR) (pp. 570-575). IEEE.

Shumway, R.H. & **Stoffer, D.S., 2019.** Time series: a data analysis approach using R. Chapman and Hall/CRC.

Soydaner, D., 2020 A comparison of optimization algorithms for deep learning. International Journal of Pattern Recognition and Artificial Intelligence, 34(13), p.2052013.

Valappil, J. & **Georgakis, C., 2000.** Systematic estimation of state noise statistics for extended Kalman filters. AIChE Journal, 46(2), pp.292-308.

Van Den Berg, J.P. & **Overmars, M.H., 2005.** Roadmap-based motion planning in dynamic environments. IEEE Transactions on Robotics, 21(5), pp.885-897.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A. & **Bottou, L., 2010.** Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).

Völz, A. & **Graichen, K., 2019.** A predictive path-following controller for continuous replanning with dynamic roadmaps. IEEE Robotics and Automation Letters, 4(4), pp.3963-3970.

Wang, S.L. 2013. Research on key technology of multi-GNSS ground based augmentation system. Southeast Univ., Nanjing, China, Tech. Rep, pp.13-27.

Wei, H., Huang, Y., Hu, F., Zhao, B., Guo, Z. and Zhang, R., 2021. Motion Estimation Using Region-Level Segmentation and Extended Kalman Filter for Autonomous Driving. Remote Sensing, 13(9), p.1828.

Wu, F., Luo, H., Jia, H., Zhao, F., Xiao, Y. & **Gao, X., 2020.** Predicting the Noise Covariance With a Multitask Learning Model for Kalman Filter-Based GNSS/INS Integrated Navigation. IEEE Transactions on Instrumentation and Measurement, 70, pp.1-13.

Xidias, E.K., 2021. A Decision Algorithm for Motion Planning of Car-Like Robots in Dynamic Environments. Cybernetics and Systems, pp.1-20.

Xing, C., Ma, L. & **Yang, X., 2016.** Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. Journal of Sensors, 2016.

Yayan, U., Yazici, A. and Saricicek, I., 2021. Prognostics-aware multi-robot route planning to extend the lifetime. Kuwait Journal of Science.

Yuen, K.V., Liang, P.F. & **Kuok, S.C., 2013.** Online estimation of noise parameters for Kalman filter. Struct. Eng. Mech, 47(3), pp.361-381.

Zhu, Q., Han Y., Liu, P., Xiao, Y., Lu, P. and Cai, C., 2019. Motion planning of autonomous mobile robot using recurrent fuzzy neural network trained by extended Kalman filter. Computational intelligence and neuroscience, 2019.

Submitted: 29/01/2022
Revised: 30/03/2022
Accepted: 03/04/2022
DOI: 10.48129/kjs.18361