Kuwait J. Sci. **41** (3) pp. 129-138, 2014

On a local search algorithm for the capacitated max-k-cut problem

JINGLAN WU[*,**] AND WENXING ZHU[*]

[*] *Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China, Email: wxzhu@fzu.edu.cn*
[**] *Department of Computer Science, Minjiang University, Fuzhou 350108, China*

ABSTRACT

The local search algorithm for the capacitated max-k-cut problem proposed by Gaur *et al*. (2008) is not guaranteed to terminate. In this note, we modify the iterative step of their algorithm to make it terminate in a finite number of steps. The modified algorithm is pseudo-polynomial, and in a special case it is strongly polynomial. Moreover, we analyze the worst case bound of the modified algorithm and give some extensions.

Keywords: Approximation ratio; local search; max-k-cut problem.

INTRODUCTION

The capacitated max-k-cut problem was introduced by Gaur *et al*. (2008), which is to partition the vertices of a graph into k subsets, such that the sum of the weights of the edges connecting vertices in different subsets is maximized. In the problem, each subset has a possibly different capacity that imposes an upper bound on its size. This problem has many applications, e.g., the placement of containers on a ship with k bays (Avriel *et al*., 1998), the placement of television commercials in program breaks (Bollapragada & Garbiras, 2004), and optimal pooling for genome re-sequencing (Hajirasouliha *et al*., 2008).

It is well known that the capacitated max-k-cut problem is NP-hard. There are some approximation results for max-k-cut with given sizes of parts. Ageev *et al*. (2001) presented a $\frac{1}{2}$-approximation algorithm for the capacitated max-2-cut problem. Feige & Langberg (2001) designed a $0.5+\epsilon$-approximation algorithm for max-2-cut with given sizes of parts. Frieze & Jerrum (1997) presented a 0.65-approximation algorithm for the max-bisection problem, and this was improved to 0.699 by Ye (2001). For the max-k-cut problem, it was shown by Kann *et al*. (1997) that, no approximation algorithm can have a performance guarantee better than $1-\frac{1}{34k}$, unless $P=NP$. Currently the best known randomized approximation algorithm by Andersson (1999) for max-k-section, where each part has the same size, has a performance guarantee of $1-\frac{1}{k}+\theta(\frac{1}{k^3})$.

For the general capacitated max-k-cut problem, very few results have been obtained. Gaur *et al.* (2008) announced a first deterministic local search approximation algorithm for the problem. The algorithm begins with a random partitioning of the vertices, with each subset size not greater than its capacity, and then by swapping pairs of vertices that are in different subsets, and that increases the sum of the weights of the edges connecting vertices in different subsets. However, Gaur *et al.* (2011) subsequently proposed a counterexample to show that the local search algorithm in Gaur *et al.* (2008) is not guaranteed to terminate.

For tackling computationally intractable optimization problems, local search is one practical approach for finding nearly optimal solutions within a reasonable running time (Gendreau & Potvin, 2005). However, understanding the worst case performance or smoothed performance guarantees are of theoretical interest (Angel, 2006; Brunsch *et al.*, 2013). Hence in this note we reconsider the local search algorithm for the capacitated max-k-cut problem in Gaur *et al.* (2008). We modify their algorithm and prove that the modified algorithm will terminate in a finite number of steps, and the approximation ratio is $\frac{t(k-1)}{2(s-1)+t(k-1)}$, where t and s are the minimum and maximum sizes of the initially generated subsets respectively. Two interesting corollaries are also derived from this result.

LOCAL SEARCH ALGORITHM

All notations are based on those by Gaur *et al.* (2008). Let $G = (V, E)$ be a graph with $|V| = n$ vertices. Let $e = [u, v] \in E$ be an edge connecting vertices $u, v \in V$ with a weight $w(u, v) \in Z^+$. The capacitated max-k-cut problem is to partition the set V of vertices into k subsets $V_1, V_2, \cdots, V_k$ such that the i-th subset V_i contains at most S_i vertices, and $|V| \leq S_1 + S_2 + \cdots + S_k$. The objective is to find a partition of V, such that the sum of the weights of the edges connecting vertices in different subsets is as large as possible.

Let w_{uV_i} be the sum of the weights of the edges connecting a vertex u to the vertices in the subset V_i. We present the following deterministic local search algorithm.

Initialization. Partition V into nonempty subsets $V_1, V_2, \cdots, V_k$ by arbitrarily assigning $|V_i| \leq S_i$ vertices to V_i, $i = 1, 2, \cdots, k$.

Iterative step. Determine if there is a pair of vertices $u \in V_i$, and $v \in V_l$, $i \neq l$ such that

$$w_{uV_i} + w_{vV_l} > w_{uV_l} + w_{vV_i} - 2w(u, v). \qquad (1)$$

If such a pair of vertices exists, reassign vertex u to V_l, vertex v to V_i, and repeat this step.

Termination. When

$$w_{uV_i} + w_{vV_l} \leq w_{uV_l} + w_{vV_i} - 2w(u, v),$$

for all $u \in V_i$, and $v \in V_l$, the algorithm stops.

Remark 1. In equation (1) of the iterative step, $w_{uV_i} + w_{vV_l}$ is the sum of the weights of un-cut edges with $u \in V_i$ or $v \in V_l$, $i \neq l$ as one end. After swapping u and v, $w_{uV_l} + w_{vV_i} - 2w(u, v)$ is the sum of the weights of un-cut edges with u or v as one end. Since the edge connecting u and v remains a cross edge after swapping, and it is counted twice while calculating $w_{uV_l} + w_{vV_i}$, we have to deduct $2w(u, v)$ in the right hand side of equation (1).

Remark 2. In Gaur *et al.* (2008), the iterative step of the local search algorithm is

$$|V_i|w_{uV_i} + |V_l|w_{vV_l} > |V_i|w_{uV_l} + |V_l|w_{vV_i},$$

which is different from ours (see equation (1)). For a complete graph with n vertices and every edge weight $w(u, v) = 1$, where a bipartition is desired with capacities of 1 and $n - 1$ (Gaur *et al.*, 2011), this inequality always holds for $n \geq 5$ at any feasible bipartition of the vertex set, since the left hand side of this inequality is $(n - 1)(n - 2)$, and the right hand side is $2(n - 1)$. Note that for this instance every feasible bipartition of the vertex set is a maximal solution. Hence the algorithm by Gaur *et al.* (2008) does not terminate.

However, for our algorithm on every feasible bipartition of this instance, the left hand side of inequality (1) is $n - 2$, and the right hand side is also $n - 2$, which implies that inequality (1) does not hold and the algorithm will terminate.

Next we prove that our algorithm will terminate in a finite number of steps. Let W_{ii} be the sum of the weights of the edges with both ends in V_i. Let W_{il} be the sum of the weights of the edges that connect a vertex in V_i to a vertex in V_l. Let

$$ALG = \sum_{i=1}^{k-1} \sum_{l=i+1}^{k} W_{il},$$

where ALG denotes the sum of the weights of the cut edges.

Lemma 1. After any iterative step of the local search algorithm, let ALG' be the sum of the weights of the cut edges. We have $ALG' - ALG \geq 1$.

Proof. At any iterative step, there exists a pair of vertices $u \in V_i$ and $v \in V_l$, $i \neq l$, such that equation (1) holds, and we reassign u to V_l, vertex v to V_i. Then it is obvious that

$$ALG' - ALG = w_{uV_i} + w_{vV_l} - (w_{uV_l} + w_{vV_i} - 2w(u, v)).$$

By equation (1), we have $ALG' - ALG > 0$. Since all edge weights are nonnegative integers, it holds that $ALG' - ALG \geq 1$.

Theorem 1. The local search algorithm terminates in a finite number of steps. Its running time is $O(n^2 W)$, where n is the number of vertices of the graph, and $W = \sum_{[u,v] \in E} w(u, v)$.

Proof. By Lemma 1, after every iterative step, the weight sum of cut edges will increase by at least 1. Since the weight sum of all edges of graph G is W, the local search algorithm will terminate after at most W steps.

For every iterative step, it searches the neighborhood of the current partition by switching all pairs of vertices that are in different subsets, and compares the sum of edge weights before and after switching a pair of vertices. This will be finished in at most $O(n^2)$ time.

Hence the running time of the local search algorithm is $O(n^2 W)$.

According to Theorem 1, our local search algorithm is a pseudo-polynomial time algorithm for the capacitated max-k-cut problem. However, for a special case of the problem where the weights of all edges are the same, the time complexity of the algorithm becomes $O(n^4)$, which is strongly polynomial.

WORST CASE BOUND

Let OPT be the optimal value of the max-k-cut problem. The following theorem shows the worst case bound of our local search algorithm.

Theorem 2. The solution obtained by the local search algorithm has a value not smaller than $\frac{t(k-1)}{2(s-1)+t(k-1)}$ of the optimal value, where $t = \min_{1 \leq i \leq k} |V_i|$, $s = \max_{1 \leq i \leq k} |V_i|$, and V_i, $i = 1, 2, \cdots, k$ are the subsets of the initial partition.

Proof. The solution returned by the local search algorithm is locally optimal, i.e.,

$$w_{uV_i} + w_{vV_l} \leq w_{uV_l} + w_{vV_i} - 2w(u, v),$$

for all $u \in V_i$, and $v \in V_l$. We sum both sides of the above expression over all $u \in V_i$ and get

$$\sum_{u\in V_i}(w_{uV_i}+w_{vV_l})\le\sum_{u\in V_i}(w_{uV_l}+w_{vV_i}-2w(u,v)),$$

which is

$$\sum_{u\in V_i}w_{uV_i}+|V_i|w_{vV_l}\le\sum_{u\in V_i}w_{uV_l}+|V_i|w_{vV_i}-2\sum_{u\in V_i}w(u,v).$$

Next, we sum both sides of the above inequality over all $v\in V_l$ and get

$$|V_l|\sum_{u\in V_i}w_{uV_i}+|V_i|\sum_{v\in V_l}w_{vV_l}\le|V_l|\sum_{u\in V_i}w_{uV_l}+|V_i|\sum_{v\in V_l}w_{vV_i}-2\sum_{v\in V_l}\sum_{u\in V_i}w(u,v). \quad (2)$$

By noting that $\sum_{u\in V_i}w_{uV_i}=2W_{ii}$, and $\sum_{v\in V_l}w_{vV_l}=2W_{ll}$, equation (2) leads to

$$2|V_l|W_{ii}+2|V_i|W_{ll}\le|V_l|W_{il}+|V_i|W_{li}-2W_{li}.$$

Since $W_{li}=W_{il}$, the above inequality gives

$$2|V_l|W_{ii}+2|V_i|W_{ll}\le(|V_l|+|V_i|-2)W_{li}. \quad (3)$$

Summing both sides of inequality (3) over all $i,l=1,2,...,k,\ i\neq l$ gives

$$2\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(|V_l|W_{ii}+|V_i|W_{ll})\le\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(|V_l|+|V_i|-2)W_{li}. \quad (4)$$

Let $t=\min_{1\le i\le k}|V_i|$, $s=\max_{1\le i\le k}|V_i|$. Since

$$\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(|V_l|W_{ii}+|V_i|W_{ll})\ge t\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(W_{ii}+W_{ll}),$$

and

$$\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(|V_l|+|V_i|-2)W_{il}\le(2s-2)\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}W_{il},$$

inequality (4) leads to

$$t\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(W_{ii}+W_{ll})\le(2s-2)\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}W_{il}. \quad (5)$$

For the right-hand side of inequality (5),

$$(2s-2)\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k} W_{il} = 2(2s-2)ALG = 4(s-1)ALG. \tag{6}$$

For the left-hand side of inequality (5),

$$t\sum_{i=1}^{k}\sum_{l=1,l\neq i}^{k}(W_{ii}+W_{ll}) = t(k-1)\sum_{i=1}^{k}W_{ii} + t(k-1)\sum_{l=1}^{k}W_{ll} = 2t(k-1)\sum_{i=1}^{k}W_{ii}. \tag{7}$$

Combining (5), (6) and (7) gives

$$t(k-1)\sum_{i=1}^{k}W_{ii} \leq 2(s-1)ALG.$$

So

$$\sum_{i=1}^{k}W_{ii} \leq \frac{2(s-1)ALG}{t(k-1)}. \tag{8}$$

Since an optimal solution of the max-k-cut problem may contain at most all of the edges, we have

$$OPT \leq \sum_{i=1}^{k}W_{ii} + ALG \leq \frac{2(s-1)ALG}{t(k-1)} + ALG.$$

Therefore

$$\frac{ALG}{OPT} \geq \frac{t(k-1)}{2(s-1)+t(k-1)} = 1 - \frac{2(s-1)}{2(s-1)+t(k-1)}. \tag{9}$$

The result in equation (9) contains parameters s and t, which are dependent on the initial partition. To make the lower bound as large as possible, we should make the initial partition such that $\frac{s}{t}$ is as small as possible. Furthermore, we can derive some results without parameters from equation (9).

Corollary 1. Let $S = \max\{S_1, S_2, ..., S_k\}$, and without loss of generality, we suppose that $t \geq 1$. For the deterministic local search approximation algorithm, we have

$$\frac{ALG}{OPT} \geq \frac{(k-1)}{2(S-1)+(k-1)}.$$

Proof. Since $S = \max\{S_1, S_2, ..., S_k\} \geq s$, the corollary follows directly from equation (9) under the assumption that $t \geq 1$.

Corollary 2. For the max-k-section problem, where each part has the same size, the deterministic local search approximation algorithm gives

$$\frac{ALG}{OPT} > \frac{k-1}{k+1} = 1 - \frac{2}{k+1}.$$

Proof. For the max-k-section problem, we have $s = t$. Thus by equation (9),

$$\frac{ALG}{OPT} \geq \frac{t(k-1)}{2(s-1) + t(k-1)} = \frac{k-1}{k+1-\frac{2}{t}} > \frac{k-1}{k+1} = 1 - \frac{2}{k+1}.$$

Corollaries 1 and 2 are not interesting for the capacitated max-2-cut problem, but they are interesting for the corresponding capacitated max-k-cut problem, since our algorithm is deterministic, and at present we have not found approximation results of deterministic algorithms for the corresponding capacitated max-k-cut problem.

It must be remarked that our local search algorithm is not polynomial in the input size. However, since the max-2-cut problem without the capacity constraints is PLS-complete (Johnson *et al.*, 1988), it is obvious that the capacitated max-k-cut problem is also PLS-complete, which means that the problem is unlikely to have a polynomial time local search procedure.

However, a locally optimal approximation scheme with efficiently searchable neighborhood for every problem in PLS has been presented by Orlin *et al.* (2004). Since the neighborhood of our local search algorithm is efficiently searchable, i.e., we can verify that a solution is locally optimal in polynomial time, the result in Orlin *et al.* (2004) and Corollaries 1 and 2 imply that our local search algorithm can attain approximation ratio $\frac{(k-1)}{2(S-1)+(k-1)}(1-\epsilon)$ in time polynomial in the input size and $\frac{1}{\epsilon}$ for the capacitated max-k-cut problem, and can attain approximation ratio $(1-\frac{2}{k+1})(1-\epsilon)$ in time polynomial in the input size and $\frac{1}{\epsilon}$ for the max-k-section problem.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61170308.

REFERENCES

Ageev, A. A., Hassin, R. & Sviridenko, M. I. 2001. A 0.5-approximation algorithm for MAX DICUT with given sizes of parts. SIAM Journal on Discrete Mathematics **14**(2): 246-255.

Andersson, G. 1999. An approximation algorithm for max p-Section. Lecture Notes in Computer Science **1563**: 237-247, Springer.

Angel, E. 2006. A survey of approximation results for local search algorithms. Lecture Notes in Computer Science **3484**: 30-73, Springer.

Avriel, M., Penn, M., Shpirer, N. & Witteboon, S. 1998. Stowage planning for container ships to reduce the number of shifts. Annals of Operations Research **76**(0): 55-71.

Bollapragada, S. & Garbiras, M. 2004. Scheduling commercials on broadcast television. Operations Research **52**(3): 337-345.

Brunsch, T., Roglin, H., Rutten, C. & Vredeveld, T. 2013. Smoothed performance guarantees for local search. Mathematical Programming, accepted for publication, online at:
http://link.springer.com/article/10.1007 %2Fs10107-013-0683-7\#page-1.

Feige, U. & Langberg, M. 2001. Approximation algorithms for maximization problems arising in graph partitioning. Journal of Algorithms **41**(2): 174-211.

Frieze, A. & Jerrum, M. 1997. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. Algorithmica **18**(1): 67-81.

Gaur, D. R., Krishnamurti, R. & Kohli, R. 2008. The capacitated max-*k*-cut problem. Mathematical Programming **115**(1): 65-72.

Gaur, D. R., Krishnamurti, R. & Kohli, R. 2011. Erratum to the capacitated max k-cut problem. Mathematical Programming **126**(1): 191-191.

Gendreau, M. & Potvin, J.-Y. 2005. Metaheuristics in combinatorial optimization. Annals of Operations Research **140**(1): 189-213.

Hajirasouliha, I., Hormozdiari, F., Sahinalp, S. C. & Birol, I. 2008. Optimal pooling for genome re-sequencing with ultra-high-throughput short-read technologies. Bioinformatics **24**: i32-i40.

Johnson, D. S., Papadimitriou, C. H. & Yannakakis, M. 1988. How easy is local search? Journal of Computer and System Sciences **37**(1): 79-100.

Kann, V., Khanna, S., Lagergren, J. & Panconesi, A. 1997. On the hardness of approximating MAX k-CUT and its dual. Chicago Journal of Theoretical Computer Science **2**: 1-18.

Orlin, J. B., Punnen, A. P. & Schulz, A. S. 2004. Approximate local search in combinatorial optimization. SIAM Journal on Computing **33**(5): 1201-1214.

Ye, Y. 2001. A .699-approximation algorithm for Max-Bisection. Mathematical Programming **90**(1): 101-111.

Submitted : *29/08/2013*
Revised : *15/01/2014*
Accepted : *20/01/2014*

حول خوارزمية البحث المحلي لحل مشكلة أعلى قطع محدود لمجاميع ثابتة السعة

جينقلانوو* و وينزينغزو**

* مركز الرياضيات المتقطعة وعلم الحاسوب النظري – جامعة فوزو
wxzhu@fzu.edu.cn
** قسم علم الحاسوب – جامعة مينغجيانغفوزو 350108 – الصين

خلاصة

شرط توقف خوارزمية البحث المحلي المقترحة لحل مشكلة أعلى قطع محدود لمجاميع ثابتة السعة من قبل غاور وآخرون (2008) غير مؤكد. نقدم في هذه الورقة تعديل للخطوة التكرارية في خوارزميتهم لضمان توقفها بعد عدد محدود من الخطوات. الخوارزمية المعدلة هي متعددة حدود زائفة، وفي حالة خاصة يمكن جعلها متعددة حدود مؤكدة فعلياً. علاوة على ذلك،نقوم بتحليل الحد الأقصى في أداء الخوارزمية المعدلة ونعطي بعض الاقتراحات للتطوير.

الكلمات الرئيسية: نسبة التقريب؛ البحث المحلي؛ مشكلة أعلى قطع محدود.