RAIRO-Oper. Res. 50 (2016) 39–46
DOI: 10.1051/ro/2015010

RAIRO Operations Research
www.rairo-ro.org

THE PRIZE-COLLECTING CALL CONTROL PROBLEM ON WEIGHTED LINES AND RINGS

Weidong Li[1], Jianping Li[1], Li Guan[1] and Yaomin Shi[2]

Abstract. Given a set of request calls with different demands and penalty costs, the prize-collecting call control (PCCC) problem is to minimize the sum of the maximum load on the edges and the total penalty cost of the rejected calls. In this paper, we prove that the PCCC problem on weighted lines is NP-hard even for special cases, and design a 1.582-approximation algorithm using a randomized rounding technique. In addition, we consider some special cases of the PCCC problem on weighted lines and rings.

Mathematics Subject Classification. 90C27.

Received September 5, 2013. Accepted March 26, 2015.

1. Introduction

Due to extensive applications in various realistic areas such as bandwidth allocation, interval packing, multi-commodity flow and scheduling, the unsplittable flow problem (UFP) on lines receives significant attention in recent years. Given an undirected line $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$ and $E = \{e_i = (i, i + 1)| i = 1, 2, \ldots, n - 1\}$ and a set $\mathcal{P}$ of K request calls, each edge $e_i \in E$ has a capacity c_i, and each request call in $\mathcal{P}$ is specified by a subpath P_k between the source (*i.e.* leftmost) vertex $s_k \in V$ and the sink (*i.e.* rightmost) vertex $t_k \in V$, a demand $d_k > 0$, and a profit $p_k > 0$. The UFP on lines is to find a subset $\mathcal{P}' \subseteq \mathcal{P}$ of request calls with maximum total profit $\sum_{k:P_k\in\mathcal{P}'} p_k$, such that for each edge, the sum of the demands of all request calls in $\mathcal{P}'$ that use this edge does not exceed its capacity.

Throughout this paper, we call that a polynomial-time algorithm is a ρ-approximation algorithm for a minimization (or maximization) problem if it always outputs a feasible solution with objective value at most ρOPT (or at least OPT/ρ), where OPT denotes the optimal value. A polynomial time approximation scheme (PTAS) is a family of algorithms such that it can produce a $(1 + \epsilon)$-approximation solution, for any fixed real number $\epsilon > 0$. A fully polynomial time approximation scheme (FPTAS) is a PTAS whose running time is polynomial in $1/\epsilon$.

The UFP on lines is NP-hard even when $G = (V, E)$ is a single edge, as it specializes to the *knapsack* problem. When all edge capacities as well as all demands are 1, the UFP on lines corresponds to the maximum weight independent set on interval graphs, which can be solved in polynomial time. If all edges have a uniform

Keywords. Prize-collecting, call control, approximation algorithms.

[1] Yunnan University, Kunming 650091, P.R. China. {weidong,jianping,guanli}@ynu.edu.cn
[2] Chongqing Radio & TV University, Chongqing, P.R. China. 109481082@qq.com

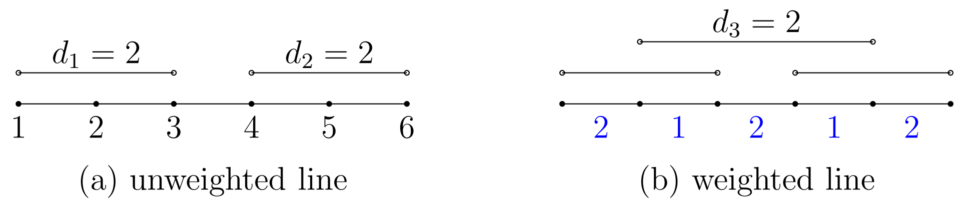

FIGURE 1. Comparison.

capacity (*i.e.* UCUFP), the UFP on lines is equivalent to the resource allocation problem [6], which is NP-hard. Calinescu *et al.* [10] presented a $(2+\epsilon)$-approximation algorithm for the UCUFP on lines.

As mentioned in [5], since the general UFP on lines is hard to approximate, most of the previous works have made some extra assumptions in order to obtain a reasonable approximation. With the no-bottleneck assumption which requires that $\max_k d_k \le \min_i c_i$, Chakrabarti *et al.* [11] presented the first constant-factor approximation algorithm. Chekuri *et al.* [12] presented the currently best known $(2+\epsilon)$-approximation algorithm, which matches the best known result for the UCUFP on lines [10]. If all the demands, edge capacities, and profits are quasi-polynomial in the number of request calls, Bansal [4] presented a quasi-polynomial time approximation scheme for the UFP on lines. Bansal [5] presented a polynomial-time $O(\log n)$-approximation algorithm for the UFP on lines without any assumption. Very recently, Bonsma *et al.* [8] introduced several novel algorithmic techniques and presented the currently best-known $(7+\epsilon)$-approximation algorithm for the UFP on lines without any assumption, and a $(2+\epsilon)$-approximation algorithm for the UFP on lines wherein the capacities can be slightly violated.

When generalizing the UFP on lines to rings, if all the demands are 1, Adamy *et al.* [1] presented a PTAS for the UFP on rings, while the NP-hardness still be open. When all demands and profits are 1, Adamy *et al.* [1] presented a strongly polynomial-time optimal algorithm for the UFP on rings. Assuming that $\max_k d_k \le \min_i c_i$, Chekuri *et al.* [12] presented the currently best known $(2+\epsilon)$-approximation algorithm for the UFP on rings. Bansal [5] presented a polynomial-time $O(\log n)$-approximation algorithm for the UFP on rings without any assumption. When generalizing the UFP on lines to other graphs, we refer to [3, 5, 8, 13–15].

Motivated by the recent increasing research on the prize-collecting combinatorial optimization problems, such as the prize-collecting Steiner tree problem [2, 7] and the min-sum clustering problem with penalties [17], we consider the *prize-collecting call control* (PCCC) problem on weighted lines, which is equivalent to the prize-collecting UFP on weighted lines. Given a weighted line $G = (V, E, w)$ and a subset $\mathcal{P} = \{P_k |\ k = 1, 2, \ldots, K\}$ of K request calls (paths), each edge $e_i \in E$ has a positive weight w_i, and each path P_k connecting a source vertex $s_k \in V$ and a sink vertex $t_k \in V$ has a positive demand d_k and a positive penalty cost p_k. The PCCC problem on weighted lines is to find a set $\mathcal{P}' \subseteq \mathcal{P}$ of paths such that the sum of the total penalty cost $\sum_{k:P_k \in \mathcal{P} \setminus \mathcal{P}'} p_k$ of the rejected paths and the maximum load of the edges in E is minimized, where the load of edge $e_i \in E$ is the total demand of the accepted paths which contain e_i multiplied by w_i.

In the previous conference version [20], we studied the PCCC problem on weighted lines where all weights of the edges are 1. Indeed, if the weights of the edges are different, the optimal solution for the PCCC problem on weighted lines will be changed. Consider an example with $n = 6$ and $K = 3$. All three paths $P_1 = \{e_1, e_2\}$, $P_2 = \{e_4, e_5\}$, and $P_3 = \{e_2, e_3, e_4\}$ have the same demand 2. The penalty costs are $p_1 = 4$, $p_2 = 4$ and $p_3 = 1$, respectively. If all weights of the edges are 1, the optimal solution is to select P_1 and P_2 with objective value $2+1 = 3$, as depicted in Figure 1a. If the weights of edges are 2, 1, 2, 1 and 2, respectively, the optimal solution is to select all three paths with objective value 4, where the load of every edge is 4, as depicted in Figure 1b.

When generalizing the PCCC problem on weighted lines to rings, we must find out the difference between lines and rings. Indeed, if there is an edge e_i in the ring such that every path in $\mathcal{P}$ does not contain it, the ring is equivalent to the line obtained by cutting e_i, as depicted in Figure 2a. If every edge is used by at least one

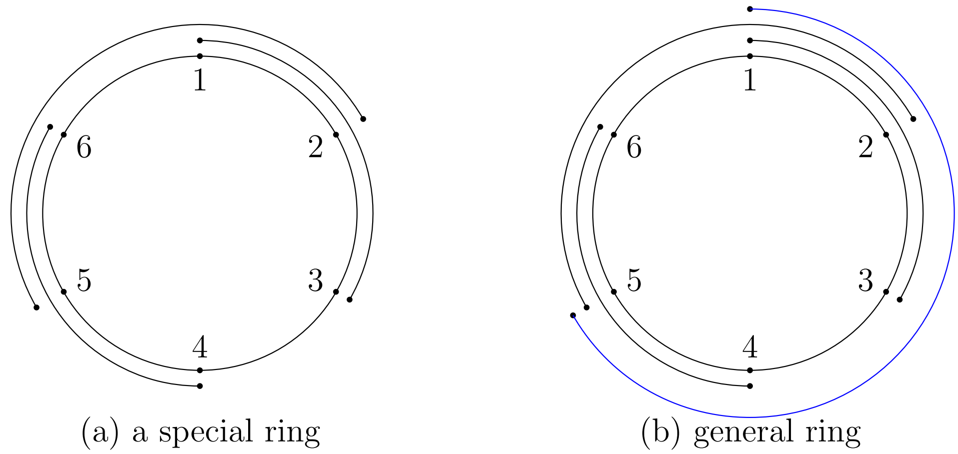

FIGURE 2. Rings.

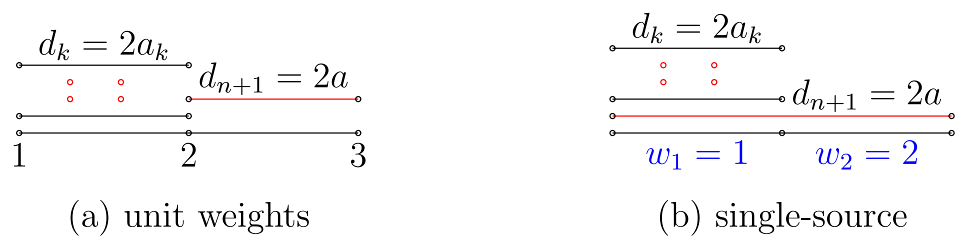

FIGURE 3. Reductions.

path in $\mathcal{P}$, the PCCC problem on weighted rings is more complicated than that on weighted lines, as depicted in Figure 2b.

This paper is organized as follows. In Section 2, we prove that the PCCC problem on weighted lines is NP-hard, and then design a 1.582-approximation algorithm. In Section 3, we prove that the PCCC problem on weighted rings possesses a PTAS when all demands are 1, and possesses an optimal algorithm in polynomial time when all demands and penalty costs are 1. In Section 4, we conclude our work with some remarks and discussion about future research directory.

2. The PCCC problem on weighted lines

In this section, we first prove that the PCCC problem on weighted lines is NP-hard even for two special cases, and then design two approximation algorithms. Finally, we design an optimal algorithm in polynomial time when all demands are 1.

2.1. NP-hardness

Although line is a very simple graph, we find that the PCCC problem on weighted lines is NP-hard even for two particular cases.

Theorem 2.1. *The PCCC problem on weighted lines is NP-hard, even when all weights of edges are 1.*

Proof. We construct a polynomial reduction from the *partition* problem [16]. Given an instance I consisting of a set $S = \{a_1, a_2, \ldots, a_n\}$ of positive numbers and $a = \sum_{k=1}^{n} a_k/2$ of the *partition* problem, construct an instance

$\tau(I)$ with 3 vertices and $n+1$ paths of the PCCC problem on weighted lines as follows. For each $k = 1, 2, \ldots, n$, the path $P_k = \{(1,2)\}$ has a demand $d_k = 2a_k$ and a penalty cost $p_k = a_k$. The path $P_{n+1} = \{(2,3)\}$ has a demand $d_{n+1} = 2a$ and a penalty cost $p_{n+1} = 4a$, as depicted in Figure 2a.

We claim that instance I of the *partition* problem has a feasible solution if and only if there is a feasible solution for instance $\tau(I)$ of the PCCC problem on weighted lines with objective value at most $3a$.

If instance I has a feasible solutio $S' \subseteq S$ satisfying $\sum_{a_k \in S'} a_k = a$, select P_{n+1} and the paths in $\{P_k | a_k \in S'\}$. The total penalty cost of the rejected paths is a, and the maximum load of the edges is $2a$. Thus, the objective value is $3a$.

If there is a feasible solution F for instance $\tau(I)$ with objective value at most $3a$, the path P_{n+1} must be accepted in F. Therefore, the total penalty cost X of the rejected paths satisfies $X \leq 3a - 2a = a$, and the total demand of the accepted paths P_k $(k \leq n)$ is $2(2a - X)$. Thus, the objective value of F is $2(2a-X)+X = 4a-X$. From the assumption, we have $4a - X \leq 3a$. Therefore, $X = a$, which implies that instance I has a feasible solution $S' = \{a_k | \ P_k \text{ is rejected } \}$.

Since the *partition* problem is NP-hard [16], so is the PCCC problem on weighted lines. □

Note that the single-source UFP [19] is NP-hard where all paths in $\mathcal{P}$ have the same source, even if G is an edge. When all weights of edges are 1, it is easy to verify that $\mathcal{P}' = \{P_k | d_k < p_k\}$ is an optimal solution for the single-source PCCC problem on unweighted lines. Not surprisingly, the single-source PCCC problem on weighted lines is NP-hard, even if $w_i \in \{1, 2\}$ for $1 \leq i \leq n-1$.

Theorem 2.2. *The PCCC problem on weighted lines is NP-hard, even when all paths in $\mathcal{P}$ have the same source.*

Proof. Similarly to the proof of Theorem 1, given an instance I of the *partition* problem, we construct an instance $\tau(I)$ with 3 vertices and $n+1$ paths of the PCCC problem on weighted lines as follows. For $k = 1, 2, \ldots, n$, the path $P_k = \{(1,2)\}$ has a demand $d_k = 2a_k$ and a penalty cost $p_k = a_k$. The path $P_{n+1} = \{(1,3)\}$ has a demand $d_{n+1} = 2a$ and a penalty cost $p_{n+1} = 5a + 1$. The weights of edges are $w_1 = 1$ and $w_2 = 2$, respectively, as depicted in Figure 2b. Similarly, it is easy to verify that instance I of the PARTITION problem has a feasible solution if and only if there is a feasible solution for instance $\tau(I)$ of the PCCC problem on weighted lines with objective value at most $5a$. Since the PARTITION problem is NP-hard [16], so is the PCCC problem on weighted lines. □

2.2. Approximation algorithms

For each edge $e_i \in E$, let $\Omega_i = \{P_k | e_i \in P_k\}$ be the set of paths containing edge e_i. We introduce a binary variable x_k for each $P_k \in \mathcal{P}$, where $x_k = 1$ if and only if P_k is accepted. Thus, the load of edge e_i is $Load(e_i) = \sum_{k:P_k \in \Omega_i} w_i d_k x_k$. We formulate the PCCC problem on weighted lines as the following integer linear program (ILP):

$$\begin{cases} \min \quad L + \displaystyle\sum_{k=1}^{K} p_k (1 - x_k) \\ Load(e_i) = \displaystyle\sum_{k:P_k \in \Omega_i} w_i d_k x_k \leq L, i = 1, 2, \ldots, n-1; \\ x_k \in \{0, 1\}, \quad k = 1, 2, \ldots, K. \end{cases}$$

Replacing the constraints $x_k \in \{0,1\}$ by $0 \leq x_k \leq 1$, we obtain the relaxation of the ILP, which is a linear program, and can be solved in polynomial time. Let $x^* = (x_1^*, x_2^*, \ldots, x_K^*)$ be an optimal solution for the relaxation of the ILP. Consider the 0-1 vector $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K)$ for the ILP, where $\bar{x}_k = 1$ if and only if $x_k^* \geq 1/2$.

Theorem 2.3. *The objective value of the solution $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K)$ is at most $2OPT$, where OPT is the objective value of the optimal solution.*

Proof. If $x_k^* \geq 1/2$, we have $\bar{x}_k = 1 \leq << 2x_k^*$, and $1 - \bar{x}_k = 0 \leq << 2(1 - x_k^*)$. Otherwise, we have $\bar{x}_k = 0 \leq << 2x_k^*$, and $1 - \bar{x}_k = 1 \leq << 2(1 - x_k^*)$. Hence,

$$
\begin{aligned}
&\max_i Load(e_i) + \sum_{k=1}^{K} p_k(1 - \bar{x}_k) \\
&= \max_i \sum_{k:P_k \in \Omega_i} w_i d_k \bar{x}_k + \sum_{k=1}^{K} p_k(1 - \bar{x}_k) \\
&\leq 2 \sum_{k:P_k \in \Omega_i} w_i d_k x_k^* + 2 \sum_{k=1}^{K} p_k(1 - x_k^*) \\
&\leq 2OPT,
\end{aligned}
$$

where the last inequality follows from the fact that the optimal value of the relaxation of the ILP is a lower bound on OPT. □

Chakrabarti *et al.* [11] proved that the nature linear programming (LP) formulation for the UFP on lines suffers from an integrality gap $\Omega(n)$. Theorem 3 implies the integrality gap of the nature LP formulation for the PCCC problem on weighted lines is at most 2. Moreover, we find a lower bound on the integrality gap of the nature LP formulation.

Theorem 2.4. *The integrality gap of the relaxation of the ILP is at least* $4/3$.

Proof. Consider an example with three vertices and two paths. All the weights of the edges are 1. The path $P_1 = \{(1, 2)\}$ has a demand 1 and a penalty cost 10. The path $P_2 = \{(2, 3)\}$ has a demand 2 and a penalty cost 1. It is no hard to verify that the optimal value of the relaxation of the ILP is $3/2$, and the optimal value of the ILP is 2. The integrality gap is $4/3$. □

Theorem 4 implies that we can not obtain a feasible solution by rounding x^* with approximation ratio less that $4/3$ for the ILP. Fortunately, a simple randomized rounding algorithm can produce a feasible solution with objective value no more that $1.582OPT$. We randomly choose a threshold α from the uniform distribution over $[1/e, 1]$. If $x_k^* > \alpha$, set $\hat{x} = 1$, and $\hat{x} = 0$, otherwise. Let $\hat{x} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_K)$ be the resulting solution.

Lemma 2.5. *The expected objective value of the solution* $\hat{x}$ *is at most* $\frac{e}{e-1}OPT$.

Proof. Let $L^* = \max_i \sum_{k:P_k \in \Omega_i} w_i d_k x_k^*$ and $\hat{L} = \max_i \sum_{k:P_k \in \Omega_i} w_i d_k \hat{x}_k$. Clearly, for any $\alpha \in [1/e, 1]$, we have $\hat{x}_k \leq x_k^*/\alpha$ and $\hat{L} \leq L^*/\alpha$. Therefore,

$$
E[\hat{L}] \leq \frac{\int_{\frac{1}{e}}^{1} \frac{L^*}{\alpha} d\alpha}{1 - \frac{1}{e}} = \frac{eL^*}{e-1} \ln \alpha |_{\frac{1}{e}}^{1} = \frac{e}{e-1} L^*.
$$

If $x_k^* \leq \frac{1}{e}$, we have

$$
E[p_k(1 - \hat{x}_k)] = p_k \leq \frac{(1 - x_k^*)p_k}{1 - \frac{1}{e}} = \frac{e}{e-1} p_k(1 - x_k^*).
$$

If $x_k^* > \frac{1}{e}$, we have

$$
\begin{aligned}
E[p_k(1 - \hat{x}_k)] &= p_k \cdot Pr[x_k^* \leq \alpha] + 0 \cdot Pr[x_k^* > \alpha] \\
&= p_k \cdot \frac{1 - x_k^*}{1 - \frac{1}{e}} = \frac{e}{e-1} p_k(1 - x_k^*).
\end{aligned}
$$

Thus,

$$E\left[\hat{L}+\sum_{k=1}^{K}p_k(1-\hat{x}_k)\right] \le \frac{e}{e-1}\left(L^*+\sum_{k=1}^{K}p_k(1-x_k^*)\right)$$
$$\le \frac{e}{e-1}OPT. \qquad \square$$

Note that there are at most K critical values x_k^*, $k = 1, 2, \dots, K$, for the threshold parameters α, which implies the randomized rounding algorithm can be derandomized in polynomial time. Therefore,

Theorem 2.6. *There exists a determinate $\frac{e}{e-1}$ (≈1.582)-approximation algorithm for the PCCC problem on weighted lines.*

2.3. Two special cases

Although the PCCC problem on weight lines is NP-hard, it can be solved optimally in polynomial time for some special cases.

Theorem 2.7. *When $d_k \equiv 1$, the PCCC problem on weighted lines admits a polynomial-time optimal algorithm.*

Proof. Given an instance I of the PCCC problem on weighted lines, let L be the maximum load of the edges in the optimal solution. For each edge e_i, $i = 1, 2, \dots, n-1$, set $c_i = \lfloor L/w_i \rfloor$. Consider the corresponding instance $\tau(I, L)$ of the UFP on lines, where the capacity of edge e_i is c_i, the profit of path P_k is p_k, and all demands of paths are 1. Although the UFP on lines is strongly NP-hard even if all edge capacities are uniform and all demands are either 1, 2, or 3 [8], when all demands are 1, the UFP on lines can be solved optimally in polynomial time by constructing an auxiliary instance for the minimum cost flow problem [1, 9]. For each possible value $L \in \{w_i, 2w_i, \dots, Kw_i | i = 1, 2, \dots, n-1\}$, let OPT_L be the optimal value for instance $\tau(I, L)$ of the UFP on lines. Clearly, $\min_L(L + \sum_{k=1}^{K} p_k - OPT_L)$ is the optimal value for instance I of the PCCC problem on weighted lines. $\square$

Theorem 2.8. *When the number of vertices n is a fixed number, the PCCC problem on weighted lines admits a FPTAS.*

Proof. Using the method in [18] with small modifications, we can obtain the FPTAS for the PCCC problem on weighted lines. We omit the details here. $\square$

3. The PCCC problem on weighted rings

It is easy to verify that the approximation algorithms in the last section can be extended to general graphs, including rings. We will consider two special cases of the PCCC problem on weighted rings.

Theorem 3.1. *When $d_k \equiv 1$, the PCCC problem on weighted rings admits a PTAS.*

Proof. We transform the PCCC problem on weighted rings to the weighted call control problem [1] on rings which is equivalent to the UFP on rings and defined as follows. Given a ring with edge capacities and a set of weighted paths on a ring, compute a subset of the paths with maximum weight without violating the capacity. For the weighted call control problem on rings, Adamy *et al.* [1] presented a polynomial-time algorithm $\mathcal{A}$ such that it computes a solution with objective value at least OPT and the edge capacity violated at most one.

Given an instance I of the PCCC problem on weighted rings, for each possible value L, we construct an instance $\tau(I, L)$ for the weighted call control problem on rings with edge capacity $c_i = \lfloor L/w_i \rfloor$ and the paths set $\mathcal{P}$, where the weight of the path $P_k \in \mathcal{P}$ is p_k. For each $L \in \{w_i, 2w_i, \dots, Kw_i | i = 1, 2, \dots, n-1\}$, implement the algorithm $\mathcal{A}$ for the instance $\tau(I, L)$, and select a solution satisfying that $L + \sum_{k=1}^{K} p_k - OPT_L$ is minimized, where OPT_L is the objective value of the produced solution. Clearly, $\min_L\{L+\sum_{k=1}^{K} p_k - OPT_L\}$ is the optimal value for instance I of the PCCC problem on weighted rings. As algorithm $\mathcal{A}$ produces a solution violating the edge capacity at most one, we obtain a solution with objective value at most $OPT + 1$.

To obtain a PTAS, for each $L \le \frac{1}{\epsilon}$, we use a substituted algorithm $\mathcal{A}_1$ for instance $\tau(I, L)$ which computes a maximum-weight set of paths such that no edge is violated. As in [1], we enumerate in polynomial time all subsets S_1 consisting of at most L paths containing the edge e_1. For each such subset S_1, we can use the optimal algorithm for the weighted call control problem on lines [1,9] to compute a maximum weight set S_1' of paths not containing e_1 such that $S_1 \cup S_1'$ is feasible. In the end, choose the solution $S_1 \cup S_1'$ with maximum weight. Adamy *et al.* [1] proved that this algorithm is polynomial and optimal. If $OPT \le \frac{1}{\epsilon}$, we obtain an optimal solution; otherwise, we can obtain a solution with objective value at most $OPT + 1 \le (1+\epsilon)OPT$, as $OPT > \frac{1}{\epsilon}$. Thus, the theorem holds. □

Theorem 3.2. *When $d_k \equiv 1$ and $p_k \equiv 1$, the PCCC problem on weighted rings admits a polynomial-time optimal algorithm.*

Proof. We transform the PCCC problem on rings to the call control problem on rings where all profits are 1, as in the proof of Theorem 3.1. When all weights are 1, the call control problem on rings admits a strongly polynomial optimal algorithm [1]. Hence, as discussed before, the PCCC problem on weighted rings admits a polynomial-time optimal algorithm, when $d_k \equiv 1$ and $p_k \equiv 1$. □

4. Conclusion

In this paper, we designed some polynomial-time algorithms for the PCCC problem on weighted lines and rings. However, the existence of the PTAS for the PCCC problem on weighted lines (or rings) still be open.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work is supported in part by the Tianyuan Fund for Mathematics of the National Natural Science Foundation of China [No. 11126315], the National Natural Science Foundation of China (Nos. 11301466, 11461081, 61170222), and the Project of First 100 High-level Overseas Talents of Yunnan Province.

References

[1] U. Adamy, C. Ambuhl, R.S. Anand and T. Erlebach, Call Control in Rings. *Algorithmica* **47** (2007) 217–238.

[2] A. Archer, M.H. Bateni, M.T. Hajiaghayi and H. Karloff, Improved approximation algorithms for prize-collecting Steiner tree and TSP. *SIAM J. Comput.* **40** (2011) 309–332.

[3] Y. Azar and O. Regev, Combinatorial algorithms for the unsplittable flow problem. *Algorithmica* **44** (2006) 49–66.

[4] N. Bansal, A. Chakrabarti, A. Epstein and B. Schieber, A quasi-PTAS for unsplittable flow on line graphs, in *Proc. of the thirty-eighth annual ACM Symposium on Theory of computing* (STOC) (2006) 721–729.

[5] N. Bansal, Z. Friggstad, R. Khandekar and M.R. Salavatipour, A logarithmic approximation for unsplittable flow on line graphs. *ACM Trans. Algorithms* **10** (2014) article no. 1.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor and B. Schieber, A unified approach to approximating resource allocation and scheduling. *J. ACM* **48** (2001) 1069–1090.

[7] M. Bateni, C. Chekuri, A. Ene, M.T. Hajiaghayi, N. Korula and D. Marx, Prize-collecting steiner problems on planar graphs, in *Proc. of the 22nd annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2011) 1028–1049.

[8] P. Bonsma, J. Schulz and A. Wiese, A constant-factor approximation algorithm for unsplittable flow on paths. *SIAM J. Comput.* **43** (2014) 767–799.

[9] M.C. Carlisle and E.L. Lloyd, On the k-coloring of intervals. *Discrete Appl. Math.* **59** (1995) 225–235.

[10] G. Calinescu, A. Chakrabarti, H. Karloff and Y. Rabani, An improved approximation algorithm for resource allocation. *ACM Trans. on Algorithms* **7** (2011) article no. 48.

[11] A. Chakrabarti, C. Chekuri, A. Kumar and A. Gupta, Approximation algorithms for the unsplittable flow problem. *Algorithmica* **47** (2007) 53–78.

[12] C. Chekuri, M. Mydlarz and F.B. Shepherd, Multicommodity demand flow in a tree and packing integer programs. *ACM Trans. Algorithms* **3** (2007) article no. 27.

[13] C. Chekuri, A. Ene and N. Korula, Unsplittable flow in paths and trees and column-restricted packing integer programs, in *Proc. of APPROX-RANDOM* (2009) 42–55.

[14] M.C. Costa, L. Letocart and F. Roupin, Minimal multicut and maximum integer multiflow: a survey. *Eur. J. Oper. Res.* **162** (2005) 55–69.

[15] K. Elbassioni, N. Garg, D. Gupta, A. Kumar, V. Narula and A. Pal, Approximation algorithms for the unsplittable flow problem on paths and trees. *Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)* (2012) 267–275.

[16] M.R. Garey and D.S. Johnson, Computer and Intractability: A guide to the theory of NP-completeness. W.H. Freeman and Company. San Francisco (1979).

[17] R. Hassin and E. Or, Min sum clustering with penalties. *Eur. J. Oper. Res.* **206** (2010) 547–554.

[18] O.H. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM* **22** (1975) 463–468.

[19] J.M. Kleinberg, Single-source unsplittable flow, in *Proc. of 37th Annual Symposium on Foundation of Computer Science* (FOCS) (1996) 68–77.

[20] W. Li, Y. Shi, Li. Guan and J. Li, The prize-collecting call control problem, in *Proc. of 2nd International Conference on Information Science and Engineering* (2010) 1609–1612.