Information Processing Letters 114 (2014) 56–59

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# Approximation algorithms for the ring loading problem with penalty cost

Weidong Li *, Jianping Li, Li Guan

*Yunnan University, Kunming 650091, PR China*

ARTICLE INFO

*Article history:*
Received 21 February 2012
Received in revised form 15 July 2013
Accepted 16 August 2013
Available online 26 August 2013
Communicated by F.Y.L. Chin

*Keywords:*
Approximation algorithms
Ring loading
Penalty cost

ABSTRACT

*The ring loading problem* and its variants have been extensively studied in the last fifteen years, under the assumption that all requests have to be satisfied. However, in many practical cases, one may wish to reject some requests, which results in a penalty cost. We introduce the *ring loading problem with penalty cost*, which generalizes the well-known ring loading problem (Schrijver et al., 1999 [14]). We prove that this problem is NP-hard even if the demand can be split, and design a 1.58-approximation algorithm for the integer demand splittable case and a $(1.58+\epsilon)$-approximation algorithm for the demand unsplittable case, for any given number $\epsilon > 0$.



## 1. Introduction

Given a set of connection requests with demands in SONET (Synchronous Optical NETwork) ring, each pair must be routed in one of the two possible ways around the ring. In order to compute the suitable capacity required for a proposed SONET ring, the ring loading problem (RLP, for short) is to minimize the maximum load on the ring, where the load of an edge is the sum of the demands of requests routed through that edge. RLP has been extensively studied in the literature under the assumption that all requests have to be satisfied [12–15].

Schrijver, Seymour and Winkler [14] developed an elegant LP-rounding method which can obtain a feasible solution with additive error at most 3/2 times the maximum demand. Khanna [12] designed a polynomial-time approximation scheme (PTAS) for RLP. Myung [13] presented an efficient algorithm for RLP with integer demand splitting where the request can be routed in two ways with the demand of each request in each direction restricted to integers. This result is improved by Wang [15] who presented a linear-time optimal algorithm. Wilfong and Winkler [16] designed an optimal algorithm for directed RLP with integer demand splitting. Recently, the online version was considered by Havill and Hutson [11]; they designed an online algorithm which achieves the best competitive ratio.

A closely related problem is the call admission control problem in rings which is to compute a maximum cardinality subset of the given paths in the ring such that no edge capacity is violated [1]. As the call admission control problem is an important combinatorial optimization problem encountered in the design and operation of communication networks, Anand et al. [3] considered another equivalent objective which is to minimize number of rejected requests for the call admission control problem in a general network. Some online related problems with the objective of minimizing the number of rejected requests can be found in [2,4,7].

In this paper, we consider a problem with a more integrated objective function, called the *ring loading problem with penalty cost* (RLPPC, for short), which is defined as follows. An $n$-node ring $C$ is an undirected graph $C=(V,E)$, where $V=\{1,2,\ldots,n\}$ is a set of nodes and $E=\{e_i=(i,i+1) \mid 1\leqslant i\leqslant n-1\}\cup\{e_n=(n,1)\}$ is a set of physical links. For each $j=1,2,\ldots,m$, we are given a connection request $r_j=(s_j,t_j)$, where $s_j,t_j\in V$ and $s_j<t_j$. We say that a request is routed in the clockwise

* Corresponding author.
*E-mail addresses:* weidong81@126.com, weidong@ynu.edu.cn (W. Li).

0020-0190/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2013.08.004

(counterclockwise) direction if a request passes through the node sequence $\{s_j, s_j+1, \ldots, t_j-1, t_j\}$ ($\{s_j, s_j-1, \ldots, 1, n, \ldots, t_j+1, t_j\}$). Each request $r_j$ has a demand $d_j$ and a penalty $p_j$, and it must be routed in one of the two possible ways around the ring or be rejected with penalty $p_j$. The objective is to minimize the sum of the maximum load of the edges on the ring $C$ and the total penalty cost, where the load of an edge is the sum of the demands of requests routed through that edge. RLPPC is closely related to the unsplittable flow problem on the ring [6], in which the requests are pre-routed, and the objective is to maximum the total profits of the accepted requests within the capacity limitations. Bansal et al. [6] designed an $O(\log n)$-approximation algorithm for it. A special case of the unsplittable flow problem on the ring is considered in [5,8,9].

If $p_j = \sum_{j=1}^{m} d_j + 1$ for each request $r_j$, which implies that no requests will be rejected, RLPPC is exactly the RLP problem in [14]. Thus, since RLP is NP-hard, RLPPC is also NP-hard. We also consider the RLPPC problem with integer demand splitting. This problem generalizes RLP with integer demand splitting [13,15], which possesses an optimal polynomial-time algorithm. We show that the RLPPC problem with integer demand splitting is NP-hard and present a 1.58-approximation algorithm for it by using a randomized rounding technique. Moreover, combining this method and the techniques in [12,14,16], we design a $(1.58+\epsilon)$-approximation algorithm for the RLPPC problem with demand unsplittable, where $\epsilon > 0$ is a fixed constant.

## 2. The RLPPC problem with integer demand splitting

In this section, we consider the RLPPC problem with integer demand splitting, where the demand of each accepted request $r_j$ can be split into two integer demands and routed in two ways satisfying total demand $d_j$. We prove that this problem is NP-hard, and present a 1.58-approximation algorithm for the RLPPC problem with *fractional demand splitting*. Then, using the LP-rounding method in [14,16], it is easy to obtain a 1.58-approximation algorithm for the RLPPC problem with integer demand splitting.

**Theorem 1.** *When the demand is integer splittable, RLPPC is NP-hard.*

**Proof.** We will construct a polynomial-time reduction from the partition problem [10]. Given a set $I = \{a_1, a_2, \ldots, a_n\}$ of positive integers and a positive integer $T = \sum_{j=1}^{n} a_j/2$, the partition problem is to decide whether there is a subset $I' \subseteq I$ satisfying $\sum_{a_j \in I'} a_j = T$. We construct an instance $\tau(I)$ for RLPPC as follows. Define a ring $C = (V, E)$ with $V = \{1, 2, 3\}$ and $E = \{(1,2), (2,3), (3,1)\}$. There are $n+2$ requests. For $j = 1, 2, \ldots, n$, the request $r_j = (1,2)$ has a demand $d_j = 4a_j$ and a penalty cost $p_j = a_j$. The requests $r_{n+1} = (2,3)$ and $r_{n+2} = (1,3)$ each has a demand $4T$ and a penalty cost $5T+1$.

We claim that instance $I$ has a feasible solution if and only if there is a feasible solution for instance $\tau(I)$ with objective value no more than $5T$.

If instance $I$ has a feasible solution $I' \subseteq I$ satisfying $\sum_{a_j \in I'} a_j = T$, each request $r_j$ corresponding to $a_j \in I'$ is rejected and each request $r_j$ corresponding to $a_j \in I \setminus I'$ is routed in the clockwise direction, for $j = 1, 2, \ldots, n$. The total penalty cost of the rejected requests is $T$. The requests $r_{n+1}$ and $r_{n+2}$ are routed in clockwise and counterclockwise direction, respectively. The maximum load of the edges is $4T$. Thus, we obtain a feasible solution with objective value $5T$.

If there is a feasible solution $\mathcal{F}$ for instance $\tau(I)$ with objective value at most $5T$, the requests $r_{n+1}$ and $r_{n+2}$ cannot be rejected. Clearly, the maximum load of the edges assigned to requests $r_{n+1}$ and $r_{n+2}$ in any feasible routing is at least $4T$. Without loss of generality, assume that, in the feasible solution $\mathcal{F}$, the requests $r_{n+1}$ and $r_{n+2}$ are routed in clockwise and counterclockwise direction, respectively. If not, it is easy to change the routing ways of $r_{n+1}$ and $r_{n+2}$ to satisfy the assumption without increasing the maximum load of the edges. The total penalty cost of the rejected requests denoted by $P$ satisfies

$$P \leqslant 5T - 4T = T. \tag{1}$$

Hence, the sum of demands of the accepted requests $r_j$ ($j \leqslant n$) is $4(2T-P) \geqslant 4T$. Combining with the fact that the loads of the edges $(2,3)$ and $(3,1)$ are $4T$ after assigning the requests $r_{n+1}$ and $r_{n+2}$, the maximum load of the edges is at least $4T + \frac{4(2T-P)-4T}{2} = 6T - 2P$, which implies that the objective value of $\mathcal{F}$ is at least $P + 6T - 2P = 6T - P$. From the assumption that the objective value of $\mathcal{F}$ is at most $5T$, we have $6T - P \leqslant 5T$, implying that $P \geqslant T$. Combining with (1), we have $P = T$, which implies that the instance $I$ has a feasible solution $I'$, where $I'$ contains all the elements $a_j$ corresponding to requests $r_j$ that are rejected. Since the partition problem is NP-hard [10], so is RLPPC. □

For the demand integer-splittable case, if the request $r_j$ is accepted, it can be routed in the clockwise direction with integer demand $d_j^{clockwise}$, and in the counterclockwise direction with integer demand $d_j^{counterclockwise}$, where $d_j^{clockwise} + d_j^{counterclockwise} = d_j$. To design an approximation algorithm for the RLPPC problem with integer demand splitting, we first construct a mixed integer program (MIP) for the RLPPC problem with *fractional demand splitting*, where $d_j^{clockwise}$ and $d_j^{counterclockwise}$ can be fractional numbers.

For each request $r_j$, let $P_1^j = \{e_i \mid s_j \leqslant i \leqslant t_j - 1\}$ be the set of edges on the path from $s_j$ to $t_j$ in the clockwise direction, and $P_0^j = E \setminus P_1^j$ the set of edges on the path from $s_j$ to $t_j$ in the counterclockwise direction. For every edge $e_i \in E$, let $C_i$ be the set of the aforementioned paths containing $e_i$, i.e., $C_i = \{P_k^j \mid e_i \in P_k^j$, for $j = 1, 2, \ldots, m$ and $k = 0, 1\}$.

For each request $r_j$, we introduce a 0–1 variable $z_j$ and two variables $x_1^j, x_0^j \in [0,1]$, where $z_j = 1$ ($z_j = 0$) indicates that the request $r_j$ is accepted (rejected). If $z_j = 1$, $x_1^j = d_j^{clockwise}/d_j$ ($x_0^j = d_j^{counterclockwise}/d_j$) implies that the request $r_j$ is routed in the clockwise (counterclockwise)