Theoretical Computer Science 590 (2015) 118–128

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Improved approximation algorithms for constrained fault-tolerant resource allocation

Kewen Liao [a], Hong Shen [a,b,*], Longkun Guo [c]

[a] School of Computer Science, The University of Adelaide, Adelaide, Australia
[b] School of Computer and Information Technology, Sun Yat-sen University, Guangzhou, China
[c] School of Mathematics and Computer Science, Fuzhou University, Fuzhou, China

ARTICLE INFO

Article history:
Received 13 December 2013
Received in revised form 3 February 2015
Accepted 18 February 2015
Available online 23 February 2015

Keywords:
Resource allocation
Approximation algorithms
LP-rounding
Reduction
Primal–dual
Time complexity

ABSTRACT

In Constrained Fault-Tolerant Resource Allocation (*FTRA*) problem, we are given a set of sites containing facilities as resources and a set of clients accessing these resources. Each site i can open at most R_i facilities with opening cost f_i. Each client j requires an allocation of r_j open facilities and connecting j to any facility at site i incurs a connection cost c_{ij}. The goal is to minimize the total cost of this resource allocation scenario. *FTRA* generalizes the Unconstrained Fault-Tolerant Resource Allocation (*FTRA*$_\infty$) [1] and the classical Fault-Tolerant Facility Location (*FTFL*) [2] problems: for every site i, *FTRA*$_\infty$ does not have the constraint R_i, whereas *FTFL* sets $R_i = 1$. These problems are said to be uniform if all r_j's are the same, and general otherwise. For the general metric *FTRA*, we first give an LP-rounding algorithm achieving an approximation ratio of 4. Then we show the problem reduces to *FTFL*, implying the ratio of 1.7245 from [3]. For the uniform *FTRA*, we provide a 1.52-approximation primal–dual algorithm in $O\left(n^4\right)$ time, where n is the total number of sites and clients.

1. Introduction

In the *Constrained Fault-Tolerant Resource Allocation* (*FTRA*) problem introduced in [1], we are given a set $\mathcal{F}$ of sites and a set $\mathcal{C}$ of clients, where $|\mathcal{F}| = n_f$, $|\mathcal{C}| = n_c$ and $n = n_f + n_c$. Each site $i \in \mathcal{F}$ contains at most R_i ($R_i \geq 1$) facilities to open as resources and each client $j \in \mathcal{C}$ is required to be allocated r_j ($r_j \geq 1$) different open facilities. Note that in *FTRA*, $\max_{j \in \mathcal{C}} r_j \leq \sum_{i \in \mathcal{F}} R_i$. Moreover, opening a facility at site i incurs a cost f_i and connecting j to any facility at i costs c_{ij}. The objective of the problem is to minimize the sum of facility opening and client connection costs under the resource constraint R_i. This problem is closely related to the *Unconstrained Fault-Tolerant Resource Allocation* (*FTRA*$_\infty$)[1] [1], the classical *Fault-Tolerant Facility Location* (*FTFL*) [2] and *Uncapacitated Facility Location* (*UFL*) [6] problems. Both *FTRA*$_\infty$ and *FTFL* are the special cases of *FTRA*: R_i is unbounded in *FTRA*$_\infty$, whereas $\forall i \in \mathcal{F}: R_i = 1$ in *FTFL*. These problems are said to be *uniform* if all r_j's are the same, and *general* otherwise. If $\forall j \in \mathcal{C}: r_j = 1$, they all reduce to *UFL*. We notice that both *FTRA* and *FTRA*$_\infty$ have potential applications in numerous distributed systems such as cloud computing, content delivery networks etc. Also,

* Corresponding author at: School of Computer Science, The University of Adelaide, Adelaide, Australia.
E-mail addresses: kewen.liao@gmail.com (K. Liao), hong@cs.adelaide.edu.au (H. Shen), longkun.guo@gmail.com (L. Guo).

[1] The problem was also called *Fault-Tolerant Facility Allocation* (*FTFA*) [4] and *Fault-Tolerant Facility Placement* (*FTFP*) [5]. Our names target the application-oriented resource allocation scenarios.

http://dx.doi.org/10.1016/j.tcs.2015.02.029
0304-3975/

we consider the problems in metric space, that is, the connection costs c_{ij}'s satisfy the metric properties that they are nonnegative, symmetric and satisfy triangle inequality. Note that even the simplest non-metric *UFL* is hard to approximate better than $O(\log n)$ unless $NP \subseteq DTIME\left[n^{O(\log\log n)}\right]$ [7].

Related work. Primal–dual and LP-rounding are two typical approaches in designing approximation algorithms for the facility location problems. Starting from the most basic and extensively studied *UFL* problem, there are JV [8], MMSV [9] and JMS [10] primal–dual algorithms obtaining approximation ratios of 3, 1.861 and 1.61 respectively. In addition, Mahdian et al. [11] improved that of the JMS algorithm to 1.52 using the standard cost scaling and greedy augmentation techniques. Shmoys et al. [6] first gave a filtering based LP-rounding algorithm achieving the constant ratio of 3.16. Later, Chudak and Shmoys [12] came up with the clustered randomized rounding algorithm which further reduces the ratio to 1.736. Based on their algorithm, Sviridenko [7] applied pipage rounding to obtain 1.582-approximation. Byrka and Aardal [13] achieved the ratio of 1.5 using a bi-factor result of the JMS algorithm. Recently, Li's more refined analysis in [14] obtained the current best ratio of 1.488, which is close to the 1.463 lower bound established by Guha and Khuller [15].

Comparing to *UFL*, *FTFL* seems more difficult to approximate. For the general *FTFL*, the primal–dual algorithm in [2] yields a non-constant factor $O(\log n)$. Constant approximation factors exist only for the uniform case. In particular, Jain et al. [16,17] showed their MMSV and JMS algorithms for *UFL* can be adapted to the uniform *FTFL* while preserving the ratios of 1.861 and 1.61 respectively. Swamy and Shmoys [18] improved the result to 1.52. On the other hand, LP-rounding approaches are more successful for the general *FTFL*. Guha et al. [19] obtained the first constant factor algorithm with the ratio of 2.408. Later, this was improved to 2.076 by Swamy and Shmoys [18] with several rounding techniques. Recently, Byrka et al. [3] used dependent rounding and laminar clustering techniques to get the current best ratio of 1.7245.

$FTRA_\infty$ was first introduced by Xu and Shen [4] and they claimed a 1.861 approximation algorithm which runs in pseudo-polynomial time for the general case. Liao and Shen [1] studied the uniform case of the problem and obtained a factor of 1.52 using a star-greedy approach. The general case of the problem was also studied by Yan and Chrobak. They first gave a 3.16-approximation LP-rounding algorithm [5], and then obtained the ratio of 1.575 [20] built on the work of [12,13,21,19]. Recently, Rybicki and Byrka [22] gave an elegant asymptotic approximation algorithm (with various better ratios depending on $\min_j r_j$) and some improved hardness results. For *FTRA*, the preliminary result is a pseudo-polynomial time 1.52-approximation algorithm [1] for the uniform case. Therefore, there is a need to provide a complete picture for this problem and discover the approximation gap between *FTRA* and *FTFL*.

In this paper, we strive to close this gap. However, there are several difficulties. First, despite the similar combinatorial structures of $FTRA_\infty$ and *FTRA*, the existing LP-rounding algorithms [5,20] for $FTRA_\infty$ cannot be adopted for *FTRA*. This is because these algorithms produce infeasible solutions that violate the constraint R_i in *FTRA*. In particular, the recent work of [20] requires liberally splitting facilities and randomly opening them. This cannot be done for both *FTRA* and *FTFL* as the splitting may cause more than R_i facilities to open, which is not a problem for $FTRA_\infty$. Second, in *FTFL*, $\max_{j\in\mathcal{C}} r_j \le n_f$, while r_j can be much larger than n_f in both $FTRA_\infty$ and *FTRA*. Therefore, the naive reduction idea of splitting the sites of an *FTRA* instance and then restricting each site to have at most one facility will create an equivalent *FTFL* instance with a possibly exponential size. Third, significantly more insights and heuristics are needed in addition to the previous work for solving *FTRA* (both the general and the uniform cases) in polynomial time.

Our contribution. For the general *FTRA*, we first develop a *unified LP-rounding algorithm* through modifying and extending the 4-approximation LP-rounding algorithm [18] for *FTFL*. The algorithm can directly solve *FTRA*, $FTRA_\infty$ and *FTFL* with the same approximation ratio of 4. This is achieved by: 1) constructing some useful *properties* of the unified algorithm which enable us to directly round the optimal fractional solutions with values that might exceed one while ensuring the feasibility of the rounded solutions and the algorithm correctness; 2) exploiting the primal and dual complementary slackness conditions of the *FTRA* problem's LP formulation. Then we show *FTRA* can reduce to *FTFL* using an *instance shrinking technique* inspired from the splitting idea of [23] for $FTRA_\infty$. It implies that these two problems may share the same approximability in weakly polynomial time. Hence, from the *FTFL* result of [3], we obtain the ratio of 1.7245. Note that, although the first rounding algorithm attains a worse approximation ratio, it could be more useful than the second to be adapted for other variants of the resource allocation problems.

For the uniform *FTRA*, we provide the first strongly polynomial time primal–dual algorithm. A carefully designed *acceleration heuristic* is presented and analyzed in order to improve upon the results of [4,1] to 1.61-approximation in $O(n^4)$. Moreover, by applying another similar heuristic to the greedy augmentation technique [19], the 3.16 ratio of [5] for the general $FTRA_\infty$ is improved to 2.408, and the previous 1.61 ratio reduces to 1.52.

The results shown in the paper directly hold for $FTRA_\infty$. For ease of analysis and implementation, the algorithms presented mostly follow the pseudocode style. Furthermore, we distinguish among pseudo-, weakly and strongly polynomial time algorithms w.r.t. the problem size n.

2. LP basics and properties

The *FTRA* problem has the following ILP formulation, in which solution variable y_i denotes the number of facilities to open at site i, and x_{ij} the number of connections between client j and site i. From the ILP, we can verify that the problem becomes the special cases *FTFL* if all R_i's are uniform and equal to 1, and $FTRA_\infty$ if the third resource constraint is removed.