SIAM J. DISCRETE MATH.
Vol. 27, No. 3, pp. 1584–1609

APPROXIMATION ALGORITHMS FOR FAULT TOLERANT FACILITY ALLOCATION*

HONG SHEN† AND SHIHONG XU‡

Abstract. Given n_f sites, each equipped with one facility, and n_c cities, *fault tolerant facility location* (*FTFL*) [K. Jain and V. V. Vazirani, *APPROX '00: Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, Spinger, New York, 2000, pp. 177–183] requires computing a minimum-cost connection scheme such that each city connects to a specified number of facilities. When each city connects to exactly one facility, FTFL becomes the classical *uncapacitated facility location problem* (*UFL*) that is well-known NP hard. The current best solution to FTFL admits an approximation ratio 1.7245 due to Byrka, Srinivasan, and Swamy applying the dependent rounding technique announced recently [*Proceedings of IPCO*, 2010, pp. 244–257], which improves the ratio 2.076 obtained by Swamy and Shmoys based on LP rounding [*ACM Trans. Algorithms*, 4 (2008), pp. 1–27]. In this paper, we study a variant of the FTFL problem, namely, *fault tolerant facility allocation* (*FTFA*), as another generalization of UFL by allowing each site to hold multiple facilities and show that we can obtain better solutions for this problem. We first give two algorithms with 1.81 and 1.61 approximation ratios in time complexity $O(mR \log m)$ and $O(Rn^3)$, respectively, where R is the maximum number of facilities required by any city, $m = n_f n_c$, and $n = \max\{n_f, n_c\}$. Instead of applying the dual-fitting technique that reduces the dual problem's solution to fit the original problem as used in the literature [K. Jain et al., *Journal of the ACM*, 50 (2003), pp. 795–824; K. Jain, M. Mahdian, and A. Saberi, *STOC'02: Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, New York, 2002, pp. 731–740; A. Saberi et al., *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, Springer, New York, 2001, pp. 127–137], we propose a method called inverse dual-fitting that alters the original problem to fit the dual solution and show that this method is more effective for obtaining solutions of multifactor approximation. We show that applying inverse dual-fitting and factor-revealing techniques our second algorithm is also (1.11,1.78)- and (1,2)-approximation simultaneously. These results can be further used to achieve solutions of 1.52-approximation to FTFA and 4-approximation to the *fault tolerant k-facility allocation* problem in which the total number of facilities is bounded by k. These are currently the best bifactor and single-factor approximation ratios for the problems concerned.

Key words. algorithms, theory, approximation algorithms, facility location, k-median problem

AMS subject classifications. 15A15, 15A09, 15A23

DOI. 10.1137/090781048

1. Introduction.

The classical facility location problem [26] has been widely studied in the field of operations research. In this problem, we are given a set $\mathcal{F}$ of n_f sites, each holding one facility, and a set $\mathcal{C}$ of n_c cities; each facility $i \in \mathcal{F}$ is associated with a nonnegative number f_i as the facility operating cost and each facility-city pair (i, j) is associated with a connection cost c_{ij} to access facility i from city j, $i \in \mathcal{F}, j \in \mathcal{C}$. The objective is to open a subset of the facilities in $\mathcal{F}$ and connect each city to an open facility so that the total cost is minimized. In this paper, we study a generalization of the facility location problem, namely, *fault tolerant facility allocation* (*FTFA*), in which each site allows multiple facilities (replicas) to be opened and each

*Received by the editors December 23, 2009; accepted for publication (in revised form) June 12, 2013; published electronically September 26, 2013.
http://www.siam.org/journals/sidma/27-3/78104.html
†Corresponding author. School of Information Science and Technology, Sun Yat-Sen University, China, and School of Computer Science, University of Adelaide, SA 5005, Australia (hong.shen@adelaide.edu.au).
‡School of Computer Science, University of Adelaide, SA 5005, Australia (shihong.xu@adelaide.edu.au).

city requires a desired number of connections for the purpose of fault tolerance and efficiency (parallel access). That is, city $j \in \mathcal{C}$ establishes r_j connections to open facilities. FTFA requires us to allocate to each site a proper number of facilities and further to each city the required number of facilities so that the total combined facility cost and connection cost is minimized. Because a city may require connection to an arbitrary number of facilities, each site is assumed to have an unlimited supply of facilities. The FTFA problem can be formulated by the following integer program:

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{C}} c_{ij} x_{ij} \\ \text{subject to} & \forall j \in \mathcal{C} : \sum_{i\in\mathcal{F}} x_{ij} \geq r_j, \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : y_i \geq x_{ij}, \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij}, y_i \in \mathbb{Z}^+. \end{array} \tag{1.1}$$

In this formulation, nonnegative integer y_i is the number of facilities deployed at site i and x_{ij} is the number of connections established between site i and city j. The first constraint ensures quality of service w.r.t. each city's connection requirement, i.e., city j's established connections must satisfy its required connection degree, achieving the desired fault tolerance. The second constraint ensures there are enough opened facilities at site i to be connected from city j. In this paper, we consider the metric version of the problem, i.e., the connection costs satisfy the triangle inequality.

DEFINITION 1.1. *An algorithm is a bifactor* (λ_f, λ_c)*-approximation or a single-factor* $\max(\lambda_f, \lambda_c)$*-approximation for FTFA iff for every instance* $\mathcal{I}$ *of FTFA and any feasible solution SOL of* $\mathcal{I}$ *with facility cost* F_{SOL} *and connection cost* C_{SOL}*, the total cost produced from the algorithm is at most* $\lambda_f F_{SOL} + \lambda_c C_{SOL}$*, where* λ_f *and* λ_c *are constants greater than or equal to one.*

FTFA is similar to the well-studied *fault tolerant facility location (FTFL)* problem [16, 10, 11, 30], which has the same objective function and constraints as FTFA except the range of variants: x_{ij} and y_i are nonnegative integers (i.e., $\mathbb{Z}^+$) in FTFA but binary integers (i.e., 0 or 1) in FTFL for all $i \in \mathcal{F}, j \in \mathcal{C}$. Without the restriction on the maximum number of facilities that can be opened at each site, FTFA is less constrained and hence incurs a smaller total cost than FTFL. The FTFA problem finds applications in grouped resource allocation in distributed systems and networks that require both reliability and efficiency, where reliability is provided through maintaining connections to different sites and efficiency is achieved by parallel access to different facilities within a group at the same site. These resources include data storage, server replicas/mirrors, virtual machines, and various kinds of services. For example, in a cloud system composed of multiple data centers (sites), each holding a large amount ($10^3 \sim 10^5$) of stored data organized in stacks and racks, to satisfy clients' different needs, the system should allow clients to access both data stores across different data centers for reliability and different data stores within a data center for efficiency. Because the number of data stores in each data center is huge, an unlimited supply of facilities at each site is a reasonable assumption. Other applications include surrogate server deployment in a content distribution network, where multiple facilities (surrogate servers or asynchronous transfer modes (ATMs) in an ATM network) can be deployed at one site if necessary and which share the duty to serve cities together. We notice that the FTFA problem becomes the classical *uncapacitated facility location (UFL)* problem when connectivity requirement $r_j = 1$ for all $j \in \mathcal{C}$. It is not hard to see that the hardness of UFL, FTFA, and FTFL complies with the following relation:

$$UFL \subseteq FTFA \subseteq FTFL.$$

Here, the second inclusion is implied by a special FTFL problem which has a set of facilities distributed by groups. Let $\mathcal{F}' = \mathcal{F} \times \{1, 2, \ldots, R\}$, where $R = \max_{j \in \mathcal{C}} r_j$ is the number of identical facilities in each group. Using this setting, the FTFA problem can be solved by FTFL algorithms because the number of facilities at any site is no more than R in any optimal solution of the FTFA problem. However, we notice that the existing algorithms for FTFL are not as efficient as the algorithms for UFL, in both approximation ratio and time complexity: most FTFL algorithms employ an LP rounding which is rather time consuming (typically requires $O(n^7)$ time) and the best known approximation ratio for FTFL is 1.725, which is worse than the ratio 1.5 for UFL. Therefore, in order to achieve a better result than that from applying FTFL algorithms directly, we must take full advantage of existing methods for UFL in solving the FTFA problem. In fact, we can transform the FTFA problem into a UFL problem by replacing $\mathcal{F}$ by the aforementioned $\mathcal{F}'$ and $\mathcal{C}$ by $\mathcal{C}' = \{(j, p), j \in \mathcal{C}, 1 \leq p \leq r_j\}$, where (j, p) is the pth port of city j, with an additional constraint. In this setting, we need to ensure no parallel connections are made between any facility-city pair, i.e., different ports of a city must be connected with different facilities. This constraint is nontrivial and as a result FTFA becomes harder to solve than UFL, yielding the first inclusion and hence implying its NP-hardness from that of UFL. In the subsequent sections, we will show how to deal with this constraint and obtain better single-factor and bifactor approximation solutions to FTFA than that for FTFL.

1.1. Related work. The facility location problem and its variants occupy a central place in operations research [26]. For the simplest problem—the maximization version of UFL to maximize the total profit when all demands (connections) are satisfied, Cornuejols, Fisher, and Nemhauser [9] obtained a $(1 - e^{-1})$-approximation algorithm. The first approximation algorithm for the minimization version to minimize the total cost for satisfying all demands is a greedy algorithm due to Hochbaum [12], which is $O(\log n)$-approximation in the general (nonmetric) case. The UFL problem has found extensive applications since these results and its metric version has been most widely studied. In metric UFL, the function of connection cost forms a metric, i.e., the connection costs between facilities and cities satisfy the triangle inequality. Existing algorithms for the *metric* UFL problem mainly apply two types of techniques, LP rounding and primal-dual, forming two groups of algorithms, respectively.

The first constant-factor approximation algorithm for the *metric* UFL problem was due to Shmoys, Tardos, and Aardal [28] based on the LP rounding technique. They gave a 3.16-approximation algorithm using the filtering technique of Lin and Vitter [20] to round the optimal solution of a linear programming relaxation. This ratio was improved by Chudak and Williamson to 1.736 [8] and by Sviridenko to 1.582 [29] through rounding an optimal fractional solution to a linear program.

In the line of primal-dual algorithms, three significant results were prosented during about the same period: Jain and Vazirani's algorithm (JV) [17], the algorithm of Mahdian, Markakis, and Saberi (MMS) [21], and the Jain–Mahdian–Saberi algorithm (JMS) [14], achieving approximation ratios of 3, 1.861, and 1.61, respectively. Different from the traditional primal-dual scheme [15, 31], dual-fitting relaxes the feasibility of the dual solution—if the dual solution becomes feasible after being shrunk by a factor, then this factor is the approximation ratio of the algorithm. Jain et al. studied the trade-off [13] between facility and connection cost and gave a series of bifactor approximation ratios. Charikar and Guha [5] improved the result of the JV algorithm to 1.853 and 1.728 applying a primal-dual and greedy augmentation technique. Mahdian, Ye, and Zhang [23] improved the JMS algorithm to 1.52-approximation by

adding a cost scaling and greedy augmentation procedure to it. Byrka [2] modified the Chudak and Shmoys algorithm [7] and obtained a new algorithm which is the first one that touches the approximability limit. Their new approach gave a 1.5-approximation algorithm, which is currently the best known for the problem.

FTFL [16] is a generalization of UFL, where connectivities at different cities (e.g., the number of distinct facilities that serve a city) are specified to meet fault-tolerant requirements. Guha, Meyerson, and Munagala obtained a 3.16-approximation algorithm by rounding the optimal fractional solution to the problem and further improved the result to 2.41 by employing a greedy local improvement step [11]. In the uniform connectivity case when all cities have the same connectivity requirement, Jain et al. [13] showed that MMS and JMS algorithms can be adapted to FTFL with the same approximation factors of 1.861 and 1.61. For the nonuniform connectivity case (general case), Swamy and Shmoys presented a 2.076-approximation algorithm applying LP rounding [30], and Byrka, Srinivasan, and Swamy [3] achieved the current best ratio of 1.7245 using dependent rounding.

When allowing each site to hold multiple facilities, FTFL becomes FTFA, which is defined in this paper as another generalization of UFL. For FTFA, we can transform it to FTFL by replacing each multiple-facility site with multiple single-facility sites and hence immediately have 1.61- and 1.7245-approximation solutions for the uniform-connection case and general case, respectively. There is no prior result that can achieve a ratio better than 1.7245 for solving the general case FTFA.

Guha and Khuller proved that the best possible approximation ratio (lower bound) for UFL is 1.463 [27], assuming NP $\not\subseteq$ DTIME$[n^{O(\log \log n)}]$. This result also holds for the fault tolerant version of the problem.

The *k-median* problem [20] has also been studied extensively [1, 4, 6] and the best known approximation ratio for this problem is $3 + \varepsilon$ [1]. Jain and Vazirani studied a new problem called *k-facility* which is a combination of the k-median and UFL problems and achieved a 6-approximation algorithm [17] and further a 4-approximation [13] based on the JMS algorithm [14, 13]. The *fault tolerant k-facility* problem was also studied by Swamy and Shmoys [30] and they achieved a 4-approximation algorithm for the uniform connectivity case. Performance of their algorithm is unknown for the general nonuniform connectivity case.

1.2. Our technique. Consider an integer program containing $k \geq 1$ items in the objective function. For the facility location problem, $k = 2$—the facility cost and connection cost. Suppose an optimal solution $OPT_1 = \sum_{p=1}^{k} I_p^*$, i.e., I_p^* is the cost for the pth item in the optimal solution. We say a solution SOL is $(\lambda_1, \ldots, \lambda_k)$-approximation if $SOL \leq \sum_{p=1}^{k} \lambda_p I_p^*$ for any optimal solution. When $k = 1$ or $\lambda_1 = \lambda_2 = \cdots = \lambda_k$, we say it is a single-factor approximation solution and otherwise a multifactor approximation solution.

Consider a minimization problem and a primal-dual algorithm—an algorithm that is iteratively making primal and dual updates using linear programming relaxation of the problem and its dual. Let the primal solution and dual solution have the same value for objective function in the process of evolution. As pointed out in the literature, the dual solution produced under this condition is, in general, infeasible to the dual problem (otherwise, we would be able to find an optimal solution for the primal problem). Instead of shrinking the dual solution by a factor in order to make it fit the primal problem as used in the dual-fitting technique [14, 13, 25], we propose a method called inverse dual-fitting that constructs an additional instance of the primal problem to make the problem fit its dual solution. Inverse dual-fitting has

the same effect as dual-fitting for single-factor approximation but is more powerful in multifactor approximation as shown below.

Formally, for a primal problem of minimization, we scale the coefficients in the objective function (i.e., the right side of the constraints in the dual problem) with constant λ_p for the pth item and obtain $OPT_2 \leq \sum_{p=1}^{k} \lambda_p I_p^*$, where OPT_2 is the optimal solution to the scaled instance of the primal problem. Actually, we can regard the primal optimal solution (now with total cost $\sum_{p=1}^{k} \lambda_p I_p^*$) as a feasible solution to the scaled instance. Due to the duality theory which states that the maximum of the dual problem is at most the minimum of the primal problem, we have $SOL_D \leq OPT_2$. As such, we only need to ensure that $SOL_P = SOL_D$ and SOL_D is a feasible dual solution to the scaled instance to achieve the result of $(\lambda_1, \ldots, \lambda_k)$-approximation, i.e., $SOL_P \leq \sum_{p=1}^{k} \lambda_p I_p^*$. The first condition is usually ensured by the algorithm—for our algorithms, the total cost is equal to the total credit paid by all cities. In order to ensure the feasibility of SOL_D (to the scaled instance), we need proper constants λ_p, $1 \leq p \leq k$. A factor-revealing technique is usually used to derive these constants.

Our inverse dual-fitting technique has shown successful applications in solving other relevant optimization problems in addition to FTFA. These problems include constrained fault-tolerant resource allocation, where each site has a limited supply of resources [18], and reliable resource allocation, where each city is associated with a fractional reliability for connection [19].

1.3. Our results. We use an inverse dual-fitting technique to design and analyze two algorithms for the metric FTFA problem in the general case. Both algorithms run through R phases and in each phase employ a subroutine to pick the most cost-effective star iteratively. The concept of cost efficiency is used by the MMS algorithm [25] which is a single-phase algorithm for UFL. The difference here is that our algorithms comprise multiple phases and in each phase deal with a distinct constraint to ensure the feasibility of the solution. To satisfy the constraint, our algorithms need to process three types of events: one for facilities opened in a previous phase, one for facilities opened in the current phase and another for opening a new facility in the current phase. Combined with factor-revealing LPs in the literature, our algorithms achieve 1.81 and 1.61 approximation factors within running time $O(mR \log m)$ and $O(Rn^3)$ respectively, where $m = n_f n_c$ and $n = \max\{n_f, n_c\}$.

The second algorithm aforementioned is also shown to be (1.11,1.78)- and (1,2)-approximation simultaneously by applying the techniques of inverse dual-fitting and factor-revealing. These results are further used to obtain respectively a 1.52-approximation algorithm to FTFA and a 4-approximation algorithm for the *fault tolerant k-facility allocation* (FTKFA) problem, which has an upper bound on the total number of facilities, i.e., k, on the basis of FTFA.

The remainder of the paper is organized as follows. In section 2 and section 3, we present the single-factor and bifactor approximation algorithms for solving the FTFA problem. In section 4, we show how to extend the bifactor approximation solution for FTFA to solve the problem of FTKFA in which the number of facilities has an upper bound k. Section 5 discusses some generalizations of FTFA for future study.

2. Single-factor approximation.

2.1. Problem formulation. In order to obtain a suitable dual for formulation (1.1) from which a satisfactory approximation ratio can be derived, we need to

first transform (1.1) into an equivalent formulation by applying a greedy approach of multiphase connection establishment as follows.

Assume each site has R facilities and city j has r_j ports. All ports of city j must be connected in the order 1 to r_j and all facilities at a site can be opened, if necessary, in the order 1 to R. We use a multiphase algorithm and connect one port for each city (if it is not fully connected) in one phase. More specifically, we establish one connection for all cities in $\mathcal{C}^p = \{j \in \mathcal{C} : r_j \geq p\}$ in phase $p \in \mathcal{R} = \{1, 2, 3, \ldots, R\}$. We use vector $y_i^p \in \{0, 1\}$ to denote whether the pth facility at site i is opened in phase p and $x_{ij}^p \in \{0, 1\}$ whether the pth port of a city is connected with a facility at site i. It is clear that $y_i = \sum_{p\in\mathcal{R}} y_i^p$, $x_{ij} = \sum_{p\in\mathcal{R}} x_{ij}^p$, and $x_{ij}^p = 0$ if $p > r_j$. With the above, program (1.1) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} \sum_{p\in\mathcal{R}} (f_i y_i^p + \sum_{j\in\mathcal{C}} c_{ij} x_{ij}^p) \\ \text{subject to} & \forall p \in \mathcal{R}, j \in \mathcal{C}^p : \sum_{i\in\mathcal{F}} x_{ij}^p \geq 1, \\ & \forall p \in \mathcal{R}, j \in \mathcal{C}^p, i \in \mathcal{F} : \sum_{p\in\mathcal{R}} y_i^p \geq \sum_{p\in\mathcal{R}} x_{ij}^p, \\ & \forall p \in \mathcal{R}, j \in \mathcal{C}^p, i \in \mathcal{F} : x_{ij}^p, y_i^p \in \{0, 1\}. \end{array} \tag{2.1}$$

Note that the second constraint does not necessarily imply $y_i^p \geq x_{ij}^p$ for any $p \in \mathcal{R}$ which otherwise will make the problem a simple aggregation of the UFL problem. Now consider the situation that $y_i^p < x_{ij}^p$, i.e., $y_i^p = 0$ and $x_{ij}^p = 1$ for some p. Since $\sum_{p\in\mathcal{R}} y_i^p \geq \sum_{p\in\mathcal{R}} x_{ij}^p$, for any (i, j, p) with $y_i^p < x_{ij}^p$ there must exist a $q < p$ that satisfies $y_i^q - x_{ij}^q > 0$. This observation implies that a connection can be established with a facility opened at an earlier phase. Let f_i^p be the cost paid to open a facility at site i in phase p. That is,

$$f_i^p = \begin{cases} f_i & \text{if a new facility of site } i \text{ must be opened in phase } p; \\ 0 & \text{otherwise.} \end{cases}$$

As an open facility can be accessed for free, the cost paid to the site is equal to zero if no new facility has to be opened. We use $z_i^p \in \{0, 1\}$ to denote whether site i is involved in the new established connections in phase p, i.e., $z_i^p = 1$ if $\sum_{j\in\mathcal{C}^p} x_{ij}^p > 0$ and 0 otherwise. Apparently, $y_i^p = 1 \Rightarrow z_i^p = 1$ because for any i, $\max_{j\in\mathcal{C}} \sum_{q=1}^{p} x_{ij}^q > \sum_{q=1}^{p-1} y_i^q \geq \max_{j\in\mathcal{C}} \sum_{q=1}^{p-1} x_{ij}^q$ implies $\sum_{j\in\mathcal{C}} x_{ij}^p > 0$, i.e., $z_i^p = 1$. Therefore, we have $z_i^p \geq y_i^p$ for any i and p. This yields $f_i^p = f_i$ if $y_i^p < z_i^p$ and 0 otherwise. Since the objective of program (2.1) carries a minimization function and $f_i^p z_i^p \geq f_i^p y_i^p \geq 0$, it is easy to see that with proper case reduction we can equivalently transform the objective to

$$\text{minimize} \sum_{p\in\mathcal{R}} \sum_{i\in\mathcal{F}} \left(f_i^p z_i^p + \sum_{j\in\mathcal{C}^p} c_{ij} x_{ij}^p \right)$$

and at the same time the second constraint to

$$z_i^p \geq x_{ij}^p \ \forall p \in \mathcal{R}, j \in \mathcal{C}^p, i \in \mathcal{F}.$$

As a result, program (2.1) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \sum_{p\in\mathcal{R}} \sum_{i\in\mathcal{F}} (f_i^p z_i^p + \sum_{j\in\mathcal{C}^p} c_{ij} x_{ij}^p) \\ \text{subjected to} & \forall p \in \mathcal{R}, j \in \mathcal{C}^p : \sum_{i\in\mathcal{F}} x_{ij}^p \geq 1, \\ & \forall p \in \mathcal{R}, j \in \mathcal{C}^p, i \in \mathcal{F} : z_i^p - x_{ij}^p \geq 0, \\ & \forall p \in \mathcal{R}, j \in \mathcal{C}^p, i \in \mathcal{F} : x_{ij}^p, z_i^p \in \{0, 1\}. \end{array} \tag{2.2}$$

Noticing $\sum_{p\in\mathcal{R}} z_i^p$ is not necessarily equal to y_i, we set $y_i = \max_{j\in\mathcal{C}} \sum_{q\in\mathcal{R}} x_{ij}^q$.

The LP-relaxation of this program can be obtained by allowing x_{ij} and y_i to be nonnegative real numbers. The dual problem of this LP relaxation can be easily derived as the following form of maximizing the aggregated credits offered from all the cities making connections in phase p:

$$\begin{array}{ll} \text{maximize} & \sum_{p\in\mathcal{R}}\sum_{j\in\mathcal{C}^p}\alpha_j^p \\ \text{subjected to} & \forall p\in\mathcal{R}, i\in\mathcal{F}: \sum_{j\in\mathcal{C}^p}\beta_{ij}^p \le f_i^p, \\ & \forall p\in\mathcal{R}, i\in\mathcal{F}, j\in\mathcal{C}^p: \alpha_j^p-\beta_{ij}^p\le c_{ij}, \\ & \forall p\in\mathcal{R}, i\in\mathcal{F}, j\in\mathcal{C}^p: \alpha_j^p,\beta_{ij}^p\ge 0. \end{array} \tag{2.3}$$

Here, following the same interpretation as in [25, 13], α_j^p stands for the total credit offered from city j and β_{ij}^p for the part of α_j^p contributed toward opening a facility at site i during phase p.

LEMMA 2.1. *For any feasible solution* (α,β) *to dual problem* (2.3) *and feasible solution* (x,y) *to primal problem* (2.1)*, we have* $\sum_{p\in\mathcal{R}}\sum_{j\in\mathcal{C}^p}\alpha_j^p \le \sum_{i\in\mathcal{F}}(\sum_{j\in\mathcal{C}}c_{ij}x_{ij}+f_iy_i)$.

Proof. We derive the inequality by applying all conditions in the constraints of the primal and dual:

$$\begin{aligned} &\sum_{p\in\mathcal{R}}\sum_{j\in\mathcal{C}^p}\alpha_j^p \le \sum_{p\in\mathcal{R}}\sum_{j\in\mathcal{C}^p}\left(\sum_{i\in\mathcal{F}}x_{ij}^p\right)\alpha_j^p \\ &\quad\left(\text{because } \sum_{i\in\mathcal{F}}x_{ij}^p\ge 1\,[(2.2)\,\text{constraint 1}]\right) \\ &\qquad\le \sum_{p\in\mathcal{R}}\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}^p}[(\alpha_j^p-\beta_{ij}^p)x_{ij}^p+\beta_{ij}^p z_i^p] \\ &\quad(\text{because } z_i^p\ge x_{ij}^p\,[(2.2)\,\text{constraint 2}]) \\ &\qquad\le \sum_{p\in\mathcal{R}}\sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{C}^p}c_{ij}x_{ij}^p+f_i^p z_i^p\right) \\ &\quad\left(\text{because } \alpha_j^p-\beta_{ij}^p\le c_{ij}\,[(2.3)\,\text{constraint 2}],\ \sum_{j\in\mathcal{C}^p}\beta_{ij}^p\le f_i^p\,[(2.3)\,\text{constraint 1}],\right. \\ &\quad\left.\text{and } x_{ij}^p, z_i^p\ge 0\,[(2.3)\,\text{constraint 3}]\right) \\ &\qquad= \sum_{i\in\mathcal{F}}\left(\sum_{j\in\mathcal{C}}c_{ij}x_{ij}+f_iy_i\right). \qquad \square \end{aligned}$$

According to the weak duality theorem, finding a feasible solution to the primal problem (2.2) is transformed into that to the dual problem (2.3), because the latter's approximation ratio (to the optimal solution of the primal) will never exceed the former's.

2.2. The algorithm. We now show how to construct an effective algorithm for solving the dual problem (2.3). An interesting observation is that we can extract $p\in\mathcal{R}$ in the constraints of the dual problem and this even holds for the objective function because $\alpha_j^p=0$ when $p>r_j$. We utilize this observation to design a high-level algorithm which decomposes the problem into R subproblems and processes them in order. Specifically, all ports are categorized into groups according to their port IDs, and $\mathcal{C}^p=\{j\in\mathcal{C}, r_j\ge p\}$. For simplicity, let $X_{ij}^b=\sum_{p=1}^b x_{ij}^p$ and $y_i^b=\sum_{p=1}^b y_i^p$, $1\le$

$b \le R$. Our algorithm evolves the solution from the initial stage (suppose $X_{ij}^0 = 0$ and $Y_i^0 = 0$), through R phases, to X_{ij}^R and Y_i^R. In each phase $p \in \mathcal{R}$, the algorithm establishes one connection for each city in $\mathcal{C}^p$. A facility opened in one phase at site i can be used for free by any city j in a later phase, suppose p, if this usage does not violate the constraint $X_{ij}^p \le Y_i^p$.

The process of our algorithm is presented in Algorithm 1. In the pth phase, the solution inherited from the last phase, i.e., $(X_{ij}^{p-1}, Y_i^{p-1})$, as well as $\mathcal{F}$ and $\mathcal{C}^p$, are used as the input of the subroutine. Note that cities with $r_j < p$ are already fully connected and therefore not included in $\mathcal{C}^p$. Suppose the new opened facilities and new established connections are denoted by (x_{ij}^p, y_i^p); then in the next phase, we have $X_{ij}^p = X_{ij}^{p-1} + x_{ij}^p$ and $Y_i^p = Y_i^{p-1} + y_i^p$ as part of the input. The algorithm ends when all R phases are finished.

ALGORITHM 1. 1.861-approximation FTFA.

Input: f_i, r_j, c_{ij}, $i \in \mathcal{F}, j \in \mathcal{C}$.

Output: x_{ij}, y_i, $i \in \mathcal{F}, j \in \mathcal{C}$.

(1) Initialization: $X_{ij}^0 \leftarrow 0, Y_i^0 \leftarrow 0, \mathcal{C}^1 = \mathcal{C}$, $p \leftarrow 1$.
(2) While $p \le R$:
  (a) Invoke *the pth phase connection* with input $(\{X_{ij}^{p-1}\}, \{Y_i^{p-1}\}, \mathcal{F}, \mathcal{C}^p)$ and produce output $(\{X_{ij}^p\}, \{Y_i^p\})$.
  (b) Set $p \leftarrow p + 1$.
(3) Set $x_{ij} \leftarrow X_{ij}^R$ and $y_i \leftarrow Y_i^R$, $i \in \mathcal{F}, j \in \mathcal{C}$.

In the pth phase algorithm, we use the notation of a star and a definition of cost efficiency. A star is composed of a facility and a group of cities that are connected with the facility. Consider the time before the new star is selected; the *cost efficiency* of a star is defined to be

$$\text{eff}(i, p, C') = \frac{f_i^p + \sum_{j \in C'} c_{i,j}}{|C'|}, \tag{2.4}$$

where f_i^p is the cost paid to open a facility at site i in phase p and C' is the set of member cities in the star. The two items in the numerator represent the total cost of the star and therefore the cost efficiency of a star is actually the average payment of all member cities to establish the star. Let $U \subseteq \mathcal{C}^p$ be the set of not fully connected cities in phase p, and $C' \subseteq U$ is a set of cities chosen by the algorithm to be connected with facility i. Because an open facility can be accessed for free under certain conditions, the cost paid to the facility is equal to zero if no new facility has to be opened.

ALGORITHM 2. The pth phase connection.

(1) Initially set $U \leftarrow \mathcal{C}^p$.
(2) While $U \neq \emptyset$:
  (a) Find the most cost-efficient star (i, p, C') according to formula (2.4).
  (b) Open a facility at site i if $\exists j \in C'$: $X_{ij}^{p-1} = Y_i^{p-1}$, and establish a connection to the facility for all cities in C'.
  (c) Set $f_i \leftarrow 0, U \leftarrow U \setminus C'$.

Now the dual variables, i.e., α_j^p and β_{ij}^p, can be used to find the most cost-efficient star. We use the same interpretation which was first used to interpret their

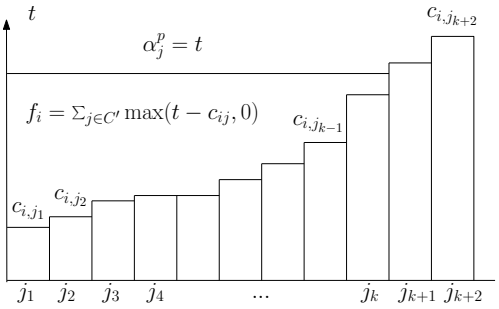


FIG. 2.1. *Credit offers for opening a facility.*

counterparts in UFL as in [25, 13]: α_j^p is the total cost (including the connection cost and the contribution to open facilities) paid by the pth port of city j and β_{ij}^p is the contribution received by facility i from the pth port of city j. As such, the most cost-effective star in each iteration of the subroutine can be found in this way: if the dual variables of all unconnected cities are raised simultaneously with time t, the most cost-effective star will be the first star (i, p, C') such that

$$\sum_{j \in C'} \max(t - c_{ij}, 0) = f_i^p,$$

where $\alpha_j^p = t$ and $\beta_{ij}^p = \max(t - c_{ij}, 0)$.

The pth phase connection opens the most cost-efficient star repeatedly until all the cities in $\mathcal{C}^p$ are *connected* with a facility. Once a city is connected, it is removed from U. In contrast, a facility is never removed; instead it can be reused for free under certain conditions. In fact, the subroutine is very close to the MMS algorithm [21] for the UFL problem. The difference is here we need to ensure the feasibility of the solution by maintaining $X_{ij}^p \le Y_i^p$ for any $i \in \mathcal{F}, p \in \mathcal{R}, j \in \mathcal{C}^p$. For simplicity, we set $y_i^p \leftarrow 1$ when a new facility at site i is opened and $x_{ij}^p \leftarrow 1$ when a new connection between city j and facility i is established. In order to maintain the feasibility of a solution, i.e., $X_{ij}^p \le Y_i^p$, we consider three cases for any $j \in \mathcal{C}^p$:

1. $X_{ij}^{p-1} < Y_i^{p-1}$. In this case, feasibility of the solution is maintained if we set $x_{ij}^p \leftarrow 1$. There is no need to open a new facility at site i and $f_i^p = 0$. We say facility i is available to be connected by city j.
2. $X_{ij}^{p-1} = Y_i^{p-1}$ and $y_i^p = 0$. In this case, we must open a new facility at site i, i.e., set $y_i^p \leftarrow 1$ in order to establish a new connection for city j and therefore $f_i^p = f_i$. The operating cost is shared between a set of cities in C' that need to connect to facility i.
3. $X_{ij}^{p-1} = Y_i^{p-1}$ and $y_i^p = 1$. In this case, a new facility has been opened by some cities before j in C' during the pth phase, so city j can access facility i without paying any operating cost. Feasibility of the solution is maintained if we set $x_{ij}^p \leftarrow 1$ and $f_i^p = 0$.

In these three cases, only the second one involves more than one city. Suppose each facility has a list of cities that are ordered according to their connection costs to the facility. As showed in Figure 2.1, the most cost-efficient star will consist of a facility and a set, containing the first k cities in this order, for some k. Therefore the algorithm can be finished efficiently in polynomial time.

Here, we use three types of events to process these cases respectively. Note that a city will contribute to opening a facility only when it has accumulated more credit

(increases in time) than the connection cost to an eligible open facility, and a star is formed only when a set of unconnected cities collectively makes a contribution f_i to open a facility at site i. We implement the pth phase connection as Algorithm 3 based on these observations.

ALGORITHM 3. Implementation of the pth phase connection.

(1) Initialization: $t \leftarrow 0, U \leftarrow \mathcal{C}^p$, $\alpha_j^p = 0$ for all $j \in U$.
(2) While $U \neq \emptyset$, increase time t until an instance of Event-1 or Event-2 or Event-3 occurs. If two events occur at the same time, process them in an arbitrary order.
   (a) Event-1: A city $j \in U$ has enough credit to be connected with an available site, suppose i, i.e., $t = c_{ij}$ and $X_{ij}^{p-1} < Y_i^{p-1}$. Set $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$.
   (b) Event-2: A site $i \in \mathcal{F}$ receives enough payment from cities in U to open its pth facility, i.e., $\sum_{j \in U} \max(t - c_{ij}, 0) = f_i$. Set $Y_i^p \leftarrow Y_i^{p-1} + 1$ and $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$ for any $j \in C' = \{j \in U : c_{ij} \leq t\}$.
   (c) Event-3: A city $j \in U$ has enough credit to be connected with a newly opened facility, i.e., $t = c_{ij}$. Set $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$.
   (d) For all cities $j \in U$, set $\alpha_j^p \leftarrow t$ and remove j from U if it is connected with a facility in phase p.

Remark 2.2. The output of Algorithm 1 is independent of the order of the cities processed. For example, we may also process cities in order $\mathcal{C}^R, \mathcal{C}^{R-1}, \ldots, \mathcal{C}^1$.

2.3. Analysis.

High-level analysis. We assume that the function of connection cost forms a metric. In order to analyze the performance of Algorithm 1, we first show that the maximum cost ratio in each phase is bounded by a constant for any instance of the problem. Formally, letting $\mathcal{I}$ be an instance of the FTFA problem and $p \in \mathcal{R}$ a phase when solving the problem, we define

$$\lambda_{p,\mathcal{I}} = \max_{i \in \mathcal{F}, p \in \mathcal{R}, C' \subseteq \mathcal{C}^p} \frac{\sum_{j \in C'} \alpha_j^p}{f_i + \sum_{j \in C'} c_{ij}}$$

as the maximum cost ratio with respect to any possible star (i, p, C').

CLAIM 2.3. *The cost of the solution in each phase is equal to* $\sum_{j \in \mathcal{C}^p} \alpha_j^p$ *and the maximum cost ratio* $\lambda_{p,\mathcal{I}}$ *is bounded by a constant* λ *for any phase* $p \in \mathcal{R}$ *and any instance* $\mathcal{I}$ *of the problem.*

We apply the inverse dual-fitting technique to analyze the approximation ratio of the high-level algorithm. We do this by composing an extra instance of the problem which has the same size as the original problem but different values of facility cost and connection cost. We achieve this by scaling the facility cost and connection cost by constant λ*:* $f_i' \leftarrow \lambda f_i$ *and* $c_{ij}' \leftarrow \lambda c_{ij}$*. Instead of shrinking the dual variable as in the dual-fitting* [25, 14, 13] *technique to achieve a feasible solution to the unscaled dual problem, we use the unshrunk dual solution which is feasible to the composed instance of the dual problem and achieve a* λ*-approximation ratio based on the claim. It is not hard to see that the same result can be achieved by applying the dual-fitting technique. However, we argue that our inverse dual-fitting technique is more powerful in multifactor approximation analysis, as shown in our other work* [32].

THEOREM 2.4. *If the pth phase connection fulfills Claim* 2.3*, the high-level algorithm, Algorithm* 1*, is a* λ*-approximation algorithm to the FTFA problem.*

Proof. First we check the feasibility of the solution. In the pth phase connection, in each phase we have for all $i \in \mathcal{F}, j \in \mathcal{C}^p : X_{ij}^p \leq Y_i^p$. In fact, this is required by the subproblem (2.1) in each phase. It is not hard to see that X_{ij}^p stops increasing when $p > r_j$ because a city j is included in $\mathcal{C}^p$ only when $p \leq r_j$ (it is already fully connected when $p > r_j$). Therefore, we have for all $i \in \mathcal{F}, j \in \mathcal{C} : X_{ij}^R \leq Y_i^R$ because Y_i^p is increasing monotonically (we never close a facility). Feasibility of the solution is proved.

In order to show the cost ratio, we compose an extra instance of the problem and its feasible dual solution. Letting $\beta_{ij}^p = \max(\alpha_j^p - \lambda c_{ij}, 0)$ for any $i \in \mathcal{F}, j \in \mathcal{C}, p \in \mathcal{R}$, and $C' = \{j \in \mathcal{C} : \alpha_j^p \geq \lambda c_{ij}\}$, we have $\sum_{j \in \mathcal{C}} \beta_{ij}^p = \sum_{j \in C'} \beta_{ij}^p = \sum_{j \in C'} (\alpha_j^p - \lambda c_{ij})$. According to Claim 2.3, we have

$$\sum_{j \in C'} (\alpha_j^p - \lambda c_{ij}) \leq \lambda f_i$$

for any $i \in \mathcal{F}, p \in \mathcal{R}$. That is, there exists dual variable $\beta_{ij}^p \geq 0$ such that

$$\forall p \in \mathcal{R}, i \in \mathcal{F} : \sum_{j \in \mathcal{C}} \beta_{ij}^p \leq \lambda f_i \tag{2.5}$$

$$\text{and } \forall p \in \mathcal{R}, i \in \mathcal{F}, j \in \mathcal{C} : \alpha_j^p - \beta_{ij}^p \leq \lambda c_{ij}. \tag{2.6}$$

From the definitions of α_i and β_{ij} apparently for all $p \in \mathcal{R}, i \in \mathcal{F}, j \in \mathcal{C} : \alpha_j^p, \beta_{ij}^p \geq 0$.

We note that the above inequalities are exactly the constraints of the dual problem (2.3) after λ-factor scaling on f_i and c_{ij}. Therefore, we can compose an instance of the dual, suppose $\mathcal{I}'$, with facility cost $f_i' = \lambda f_i$ and connection cost $c_{ij}' = \lambda c_{ij}$. Let OPT_2 be the optimal solution to the primal problem of $\mathcal{I}'$ and OPT_1 the optimal solution to the primal problem of $\mathcal{I}$. It is clear that

$$OPT_2 = \lambda OPT_1. \tag{2.7}$$

From inequalities (2.5) and (2.6), we know that (α, β) is a feasible solution to the dual problem of $\mathcal{I}'$. Due to the weak duality theorem, which states that the optimum of the dual problem (in the form of maximization) is no more than the optimum of the primal problem (in the form of minimization), we have

$$\sum_{j \in \mathcal{C}^p} \sum_{p \in \mathcal{R}} \alpha_j^p \leq OPT_2. \tag{2.8}$$

On the other hand, letting SOL be the solution derived by the algorithm, we have

$$SOL = \sum_{p \in \mathcal{R}} \sum_{j \in \mathcal{C}^p} \alpha_j^p \tag{2.9}$$

according to the first part of Claim 2.3. Combining (2.7), (2.8), and (2.9), we have

$$SOL \leq \lambda OPT_1.$$

The theorem follows. ∎

Remark 2.5. For many problems single-factor approximation analysis may be achieved by using the classical dual-fitting technique, i.e., shrinking the dual solution λ times to make it "fit" to the primal problem. Some greedy algorithms for the UFL

problem, like the MMS algorithm [25, 13] or the JMS algorithm [14, 13], are analyzed through decomposing the optimal solution into a group of stars because any solution to UFL can be decomposed into vertex-disjoint stars. Our proposed inverse dual-fitting provides an alternative way for single-factor analysis that is simpler and more effective in many cases.

From Algorithm 1, we can see that all payments are for either the connection cost or operating cost. Therefore the first part of the claim is satisfied by the algorithm, and we only need to find a proper value of $\lambda \geq 1$ such that for any instance $\mathcal{I}$ of the FTFA problem

$$\max_{i\in\mathcal{F},p\in\mathcal{R},C'\subseteq\mathcal{C}^p} \frac{\sum_{j\in C'} \alpha_j^p}{f_i + \sum_{j\in C'} c_{ij}} \leq \lambda.$$

It is clear that we only need to consider cities in $C' = \{j \in \mathcal{C}^p : \alpha_j^p \geq \lambda c_{ij}\}$ for each p. Without loss of generality, suppose there are k such cities in $\mathcal{C}^p$ and $\alpha_1^p \leq \alpha_2^p \leq \cdots \leq \alpha_k^p$. We consider some important properties of the pth phase connection in order to find a proper value of λ.

Analysis of the pth phase connection. First, we have the following lemma on the contribution received by a site according to Event-2 and Event-3.

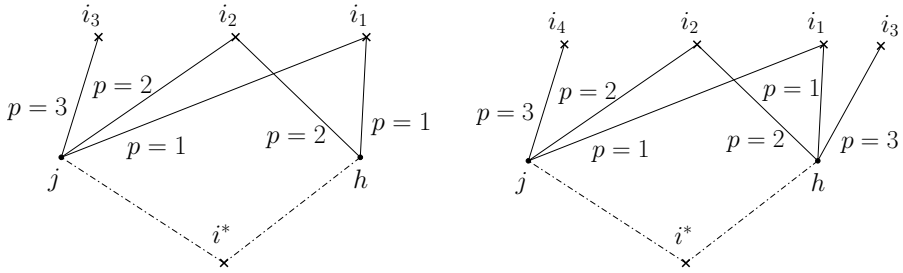
LEMMA 2.6. *For any instance $\mathcal{I}$ and phase $p \in \mathcal{R}$, $\sum_{j=h}^{k} \max(\alpha_h^p - c_{ij}, 0) \leq f_i$ for any site $i \in \mathcal{F}$ and city $h, 1 \leq h \leq k$.*

Proof. Assume $\sum_{j=h}^{k} \max(\alpha_h^p - c_{ij}, 0) > f_i$; then a new facility at site i must be opened at time $t = \alpha_h^p - \epsilon$ according to Event-2 because any j with $\alpha_j^p \geq \alpha_h^p$ still contributes to opening facilities at time t. According to the assumption, there is at least one city, suppose j', such that

$$\alpha_{j'}^p \geq \alpha_h^p \quad \text{and} \quad \alpha_h^p > c_{ij'}.$$

That is, $\alpha_{j'}^p > c_{ij'}$. Actually, j' can be connected with site i at least at time t according to Event-3 of the algorithm, which implies $\alpha_{j'}^p \leq c_{ij'}$. The contradiction establishes the lemma. ∎

It is natural to follow the approach proposed by Mahdian and others [13, 25] to obtain a property regarding the triangle inequality. However, in the fault-tolerant context, this becomes more complex. In fact, we are not able to conclude that a contribution is less than the connection cost to any open facility. As shown in Figure 2.2(a), neither $\alpha_j^3 \leq c_{i_1 j}$ nor $\alpha_j^3 \leq c_{i_2 j}$ can be achieved even if i_1 and i_2 are opened. This is because h is already connected with facilities i_1 and i_2 before making



(a) $p = 3, r_h = 2$ (b) $p = 2, r_h = 2$

FIG. 2.2. *Ranking of contributions.*

its third contribution. Fortunately, in our algorithm, only ports of the same rank are processed in a phase and this makes an important difference. In fact, if there are p open facilities, the pth contribution of a city is no more than the maximum connection cost from the city to these facilities. As shown in Figure 2.2(b), we have $\alpha_j^3 \le c_{i_3 j}$. Formally, we have the following lemma.

LEMMA 2.7. *For any instance $\mathcal{I}$ and phase $p \in \mathcal{R}$, $\alpha_j^p \le \alpha_h^p + c_{ij} + c_{ih}$ holds for any site $i \in \mathcal{F}$ and cities h and $j, 1 \le h, j \le k$.*

Proof. Assume $\alpha_j^p > \alpha_h^p$, since otherwise the lemma is obvious. Let H be the set of facilities that are connected with city h at time $t = \alpha_j^p - \epsilon$, so we have $|H| = p$ because h is already connected in the pth phase. Hence, there must exist a facility in H which is not connected with j at the moment in phase p. Suppose it is a facility at site i', and we have $X_{i'j}^{p-1} < Y_{i'}^p$. Therefore j can be connected with i' without paying operating cost. Considering two cases, i.e., the facility is opened respectively in an early phase or in phase p, we have $\alpha_j^p \le c_{i'j}$ for both cases according to Event-1 and Event-3. Further, we have $\alpha_j^p \le c_{ij}$ if $i = i'$ and $\alpha_j^p \le \alpha_h^p + c_{ij} + c_{ih}$ by the triangle inequality otherwise. Because $\alpha_h^p \ge c_{i'h}$ for any i' connected with city h, we have $c_{i'j} \le c_{i'h} + c_{ij} + c_{ih}$. The lemma follows. ∎

Performance of Algorithm 1. The above lemmas present some important properties of the pth phase connection and the following result shows that they are enough to bound the ratio of the total cost of a derived solution to that of an optimal solution. Let λ_k be the maximum of the following LP:

$$
\begin{array}{lll}
\text{maximize} & \dfrac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} c_j} & \\
\text{subjected to} & \forall 1 \le j < h \le k: \ \alpha_h \le \alpha_j + c_j + c_h, & \\
& \forall 1 \le h \le k: \ \sum_{j=h}^{k} \max(\alpha_h - c_j, 0) \le f, & \\
& \forall 1 \le j \le k: \ \alpha_j, c_j, f \ge 0. &
\end{array} \tag{2.10}
$$

If λ_k has an upper bound with respect to any integer k, we are able to choose this upper bound as the value of λ with respect to Claim 2.3. LPs like program (2.10) are also called factor-revealing LPs in the literature [24, 25, 14, 13].

COROLLARY 2.8. *Algorithm 1 is 1.861-approximation for the metric FTFA problem.*

Proof. For notational convenience, denote α_j^p by α_j, c_{ij} by c_j, and f_i by f. It is clear that Lemmas 2.6 and 2.7 imply the two constraints of program (2.10). As a result, we have

$$\lambda \le \sup_{k \ge 1} \{\lambda_k\}.$$

In fact, Mahdian and others [25, 13] showed that program (2.10) has an upper bound 1.861 in their analysis for the MMS algorithm. Combined with Theorem 2.4, the corollary follows. ∎

In each phase, there are at most $n_f \cdot |\mathcal{C}^p|$ events for which the algorithm needs $n_f \cdot |\mathcal{C}^p| \log(n_f \cdot |\mathcal{C}^p|)$ time to sort these events. Considering that the algorithm runs R phases and in each phase $|\mathcal{C}^p| \le |\mathcal{C}|$, we have the following lemma.

LEMMA 2.9. *The time complexity of Algorithm 1 is $O(mR \log m)$, where $m = n_c n_f$.*

3. Bifactor approximation. Interesting enough, the dual-fitting technique does not seem to have found use outside single-factor approximation analysis (i.e.,

1.861- and 1.61-approximation). For example, the JMS algorithm is also (1.11,1.78)-approximation [22] and (1,2)-approximation [14, 13] for UFL. However, these results were achieved through proving an upper bound of the total cost paid by all cities in any possible star s, i.e., $\lambda_f f_i + \lambda_c \sum_{j \in s \cap \mathcal{C}} c_{ij}$, rather than by applying the dual-fitting technique. This approach is straightforward because a solution to the UFL problem can be decomposed into a group of vertex-disjoint stars but is incapable of deriving bifactor ratios for FTFA effectively because of the difficulties in finding a best assignment of each city's multiple costs to the respective stars that balances the costs among all stars. In this section, we show that our inverse dual-fitting technique can be deployed for effective bifactor approximation analysis for FTFA to achieve a similar result to that of UFL. This technique also simplifies algorithm design through a consistent and intuitive explanation to the original dual variables during the process of fitting to the dual.

3.1. The algorithm. We note that once a city in Algorithm 1 is fully connected, it is not processed any more even if a facility is opened with a smaller connection cost. It is obvious that we are able to improve the algorithm by establishing connections, for each city, to facilities with smallest connection costs. We do this by switching two connections of a city, an old one with higher connection cost and a new one with smaller connection cost. Taking into account the reduction in total cost by connection switch, we redefine the cost efficiency of a star at the time before the new star is selected by

$$\text{eff}(i, p, C') = \frac{f_i^p + \sum_{j \in C'} c_{ij} - \sum_{j \in \mathcal{C}^p \setminus U} \max(c_{i'j} - c_{ij}, 0)}{|C'|}, \tag{3.1}$$

where $\mathcal{C}^p \setminus U$ is the set of cities which are already connected in phase p and $c_{i'j}$ the maximum connection cost of city j. The first two items in the numerator represent the total cost of the star, which is the same as in Algorithm 1. The third item is the contribution made by connected cities via connection switch.

The new algorithm for bifactor approximation, Algorithm 4, has the same structure as Algorithm 1, with the subroutine of the pth phase connection being replaced by the improved pth phase connection that uses a new cost efficiency as redefined by (3.1). With the same interpretation of dual variables as in Algorithm, the most cost-efficient star in each iteration of the algorithm can be found in a similar way: if the dual variables of all unconnected cities are raised simultaneously with time t, the most cost-efficient star will be the first star (i, p, C') for which

$$\sum_{j \in C'} \max(t - c_{ij}, 0) + \sum_{j \in \mathcal{C}^p \setminus U} \max(c_{i'j} - c_{ij}) = f_i^p.$$

When $f_i^p = 0$, the improved pth phase connection is the same as the original. When $f_i^p = f_i$, it receives payment from unconnected cities as well as payment from connected cities through a connection switch and, once its amount is equal to the facility cost, it opens a new facility at site i. Like the pth phase connection, it can be implemented through handling three types of events.

The above algorithm ensures that in each phase for all $i \in \mathcal{F}, j \in \mathcal{C}^p : X_{ij}^p \le Y_i^p$ as the result of handling three events. It is not hard to see that X_{ij}^p stops increasing when $p > r_j$ because a city j is included in $\mathcal{C}^p$ only when $p \le r_j$. Therefore, we have for all $i \in \mathcal{F}, j \in \mathcal{C} : X_{ij}^R \le Y_i^R$ because Y_i^p is increasing monotonically. (We never close a facility.) Feasibility of the solution is ensured.

ALGORITHM 4. Bifactor approximation FTFA.

(1) Initialization: $X_{ij}^0 \leftarrow 0, Y_i^0 \leftarrow 0, \mathcal{C}^1 = \mathcal{C}$, $p \leftarrow 1$.
(2) While $p \le R$:
   (a) Invoke the improved pth phase connection with input $(\{X_{ij}^{p-1}\}, \{Y_i^{p-1}\}, \mathcal{F}, \mathcal{C}^p)$ and produce output $(\{X_{ij}^p\}, \{Y_i^p\})$.
   (b) $p \leftarrow p + 1$.
(3) $x_{ij} \leftarrow X_{ij}^R$ and $y_i \leftarrow Y_i^R$, $i \in \mathcal{F}, j \in \mathcal{C}$.

ALGORITHM 5. The improved pth phase connection.

(1) Initialization: $t \leftarrow 0, U \leftarrow \mathcal{C}^p$.
(2) While $U \neq \emptyset$, increase time t until an instance of Event-1 or Event-2 or Event-3 occurs. If two events occur at the same time, process them in an arbitrary order.
   (a) Event-1: A city $j \in U$ has enough credit to be connected with an eligible site, suppose i. In this case $t = c_{ij}$ and $X_{ij}^{p-1} < Y_i^{p-1}$. Set $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$.
   (b) Event-2: A site $i \in \mathcal{F}$ receives enough credit from cities in U to open its pth facility. In this case $\sum_{j \in U} \max(t - c_{ij}, 0) + \sum_{j \in \mathcal{C}^p \setminus U} \max(c_{i'j} - c_{ij}) = f_i$. Set $Y_i^p \leftarrow Y_i^{p-1} + 1$ and $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$ for all $j \in C_1' = \{j \in U : c_{ij} \le t\}$ and $j \in C_2' = \{j \in \mathcal{C}^p \setminus U : c_{ij} \le c_{i'j}\}$, and set $X_{ij}^p \leftarrow X_{ij}^{p-1} - 1$ for all $j \in C_2'$.
   (c) Event-3: A city $j \in U$ has enough credit to be connected with a newly opened facility. In this case $t = c_{ij}$. Set $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$.
   (d) For all cities $j \in U$, set $\alpha_j^p \leftarrow t$ and remove city j from U if it is connected with a facility in phase p.

In Algorithm 5, the amount of credit paid for a connection can be divided further as part for a connection with a smaller cost and part for opening other facilities. Despite the difference from Algorithm 3, it is still true that all payments of a city are used to either open facilities or establish connections, and therefore the total cost of the solution is still $\sum_{j \in \mathcal{C}} \sum_{p=1}^{r_j} \alpha_j^p$. This results in the following lemma.

LEMMA 3.1. *A solution produced by Algorithm 4 is feasible to the FTFA problem and its total cost is equal to* $\sum_{j \in \mathcal{C}} \sum_{p=1}^{r_j} \alpha_j^p$.

3.2. Analysis. Let $\lambda_f \ge 1$ be a constant (to be fixed later), and define the maximum connection cost ratio with respect to any possible star (i, p, C') as

$$\lambda_{\mathcal{I}}' = \max_{i \in \mathcal{F}, p \in \mathcal{R}, C' \subseteq \mathcal{C}^p} \frac{\sum_{j \in C'} \alpha_j^p - \lambda_f \cdot f_i}{\sum_{j \in C'} c_{ij}}.$$

CLAIM 3.2. *The maximum cost ratio $\lambda_{\mathcal{I}}'$ is bounded by a constant λ_c for any instance $\mathcal{I}$ of the FTFA problem.*

THEOREM 3.3. *If the improved pth phase connection satisfies Claim 3.2, Algorithm 4 produces a solution within cost $\lambda_f F^* + \lambda_c C^*$, where F^* and C^* are respectively the facility cost and connection cost of an optimal solution to the FTFA problem.*

Proof. Let $\beta_{ij}^p = \max(\alpha_j^p - \lambda_c c_{ij}, 0)$ for any $i \in \mathcal{F}, j \in \mathcal{C}, p \in \mathcal{R}$, and $C' = \{j \in \mathcal{C} : \alpha_j^p \ge \lambda c_{ij}\}$, and we have $\sum_{j \in \mathcal{C}} \beta_{ij}^p = \sum_{j \in C'} \beta_{ij}^p = \sum_{j \in C'} (\alpha_j^p - \lambda_c c_{ij})$. According to Claim 3.2, we have

$$\sum_{j \in C'} (\alpha_j^p - \lambda_c c_{ij}) \le \lambda_f f_i$$

for any $i \in \mathcal{F}, p \in \mathcal{R}$. That is, there exists dual variable $\beta_{ij}^p \ge 0$ such that

$$\forall i \in \mathcal{F}, p \in \mathcal{R} : \sum_{j \in \mathcal{C}} \beta_{ij}^p \le \lambda_f f_i \tag{3.2}$$

$$\text{and } \forall i \in \mathcal{F}, j \in \mathcal{C}, p \in \mathcal{R} : \alpha_j^p - \beta_{ij}^p \le \lambda_c c_{ij}. \tag{3.3}$$

We note that the above inequalities have the same form as the constraints of the dual problem (2.3). Therefore, we can compose an instance of the FTFA, suppose $\mathcal{I}'$, with facility cost $f_i' = \lambda_f f_i$ and connection cost $c_{ij}' = \lambda_c c_{ij}$. Let OPT_2 be the optimal solution to the primal problem of $\mathcal{I}'$ and OPT_1 the optimal solution to the primal problem of $\mathcal{I}$. It is clear that

$$OPT_2 \le \lambda_f F^* + \lambda_c C^* \tag{3.4}$$

because the optimal solution to $\mathcal{I}$ is also a feasible solution to $\mathcal{I}'$. (Its cost is equal to the left side of the above inequality.) From inequality (3.2) and (3.3), we know (α, β) is a feasible solution to the dual problem of $\mathcal{I}'$. Due to the weak duality theorem, which states that the maximum of the dual problem is no more than the minimum of the primal problem, we have

$$\sum_{j \in \mathcal{C}} \sum_{p \in \mathcal{R}} \alpha_j^p \le OPT_2. \tag{3.5}$$

On the other hand, letting SOL be the solution produced by the algorithm, we have

$$SOL = \sum_{j \in \mathcal{C}} \sum_{p \in \mathcal{R}} \alpha_j^p \le OPT_2 \le \lambda_f F^* + \lambda_c C^* \tag{3.6}$$

according to Lemma 3.1. So the theorem follows. ∎

For an FTFA problem, we say a solution is (λ_f, λ_c)-approximation to FTFA if its cost is no more than $\lambda_f F^* + \lambda_c C^*$, where F^* and C^* are respectively the facility cost and connection cost of an optimal solution. Now, we only need to find a proper value of $\lambda_c \ge 1$ such that for any instance $\mathcal{I}$ of the FTFA problem

$$\max_{i \in \mathcal{F}, p \in \mathcal{R}, C' \subseteq \mathcal{C}^p} \frac{\sum_{j \in C'} \alpha_j^p - \lambda_f f_i}{\sum_{j \in C'} c_{ij}} \le \lambda_c.$$

Again, we only need to consider cities with $\alpha_j^p \ge \lambda_c c_{ij}$. Without loss of generality, suppose there are k such cities in $\mathcal{C}^p$ and further $\alpha_1^p \le \alpha_2^p \le \cdots \le \alpha_k^p$. We consider some important properties of the improved pth phase connection before finding a proper value of λ_c.

Consider time $t = \alpha_h^p - \epsilon$ $(\epsilon \to 0)$ and define

$$u_{jh} = \begin{cases} t, & \alpha_j^p = \alpha_h^p, \\ c_{i^* j}, & \alpha_j^p < \alpha_h^p, \end{cases}$$

for any $1 \le j \le h \le k$, where $c_{i^* j}$ is the maximum connection cost of city j at time t. We have the following properties for Algorithm 4.

LEMMA 3.4. *For a given instance $\mathcal{I}$ and any phase $p \in \mathcal{R}$, $\alpha_h^p \le u_{jh} + c_{ij} + c_{ih}$ for any $1 \le j < h \le k$.*

Proof. If $\alpha_j^p = \alpha_h^p$, $u_{jh} \to \alpha_h^p$ according to the definition, the inequality is obvious. So we only need to consider $\alpha_j^p < \alpha_h^p$. Letting H be the set of facilities connected

with city j at the moment t, we have $|H| = p$. Hence, there must exist a facility in H which is not connected with h. Suppose it is a facility at site i', and we have $X_{i'h}^{p-1} < Y_{i'}^{p}$. Therefore h can be connected with i' without paying the operating cost. Considering two cases when the facility is opened in a previous phase and in phase p, respectively, we have $\alpha_h^p \le c_{i'h}$ for both cases according to Event-1 and Event-3. Furthermore, if $i = i'$ we have $\alpha_h^p \le c_{ih}$. Otherwise, combining the triangle inequality $c_{i'h} \le c_{i'j} + c_{ij} + c_{ih}$ immediately yields $\alpha_h^p \le u_{jh} + c_{ij} + c_{ih}$ because u_{jh} is the maximum connection cost of city j at time t when $\alpha_j^p < \alpha_h^p$. The lemma follows. ∎

At time $t = \alpha_h^p$, the amount of contribution that city j offers to open a facility at site i is equal to

$$\begin{aligned} &\max(u_{jh} - c_{ij}, 0) \quad \text{if} \quad j < h \text{ and} \\ &\max(\alpha_h^p - c_{ij}, 0) \quad \text{if} \quad j \ge h. \end{aligned}$$

Note that by the definition of u_{jh} this holds even if $\alpha_j^p = \alpha_h^p$. It is clear that the total offer of cities to a site can never exceed the operating cost at this site. Therefore, we have $\sum_{j=1}^{h-1} \max(u_{jh} - c_{ij}, 0) + \sum_{j=h}^{k} \max(\alpha_h^p - c_{ij}, 0) \le f_i$. This results in the following lemma.

LEMMA 3.5. *For a given instance $\mathcal{I}$ and any phase $p \in \mathcal{R}$, $\sum_{j=1}^{h-1} \max(u_{jh} - c_{ij}, 0) + \sum_{j=h}^{k} \max(\alpha_h^p - c_{ij}, 0) \le f_i$ holds for any $1 \le h \le k$.*

The above lemmas present some properties of Algorithm 4, and the following theorem shows that they are enough to prove Claim 3.2. Define f and c_j to be the functions for f_i and c_{ij} on i, respectively, and let λ_c^k be the maximum of the following factor-revealing LP:

(3.7)

$$\begin{aligned} \text{maximize} \quad & \frac{\sum_{j=1}^{k} \alpha_j - \lambda_f \cdot f}{\sum_{j=1}^{k} c_j} \\ \text{subjected to} \quad & \forall 1 \le j \le h \le k: \ \alpha_h \le u_{jh} + c_j + c_h, \\ & \forall 1 \le h \le k: \ \sum_{j=1}^{h-1} \max(u_{jh} - c_j, 0) + \sum_{j=h}^{k} \max(\alpha_h - c_j, 0) \le f, \\ & \forall 1 \le j \le h \le k: \ \alpha_j, c_j, u_{jh}, f \ge 0. \end{aligned}$$

It is clear that Lemmas 3.4 and 3.5 imply the two constraints of program (3.7). As a result, we have

$$\lambda_c \le \sup_{k \ge 1} \{\lambda_c^k\}$$

corresponding to Claim 3.2. Actually, for different $\lambda_f \ge 1$, there is a unique upper bound $\sup_{k \ge 1} \{\lambda_c^k\}$ as showed on the approximation curve in [13]. Furthermore, we have the following theorem according to the existing results on the factor-evealing LP.

THEOREM 3.6. *For λ_c defined in Claim 3.2, we have*

1. *if $\lambda_f = 1.61$, then $\lambda_c \le 1.61$ (see [14]);*
2. *if $\lambda_f = 1.11$, then $\lambda_c \le 1.78$ (see [22]);*
3. *if $\lambda_f = 1$, then $\lambda_c \le 2$ (see [13]).*

Proof. Proofs of these results can be obtained directly from the cited references. Here, for the third result we give an alternative proof which is simpler than that given in [13]. We first relax the second constraint as

$$\forall 1 \le h \le k: \ \sum_{j=1}^{h-1} (u_{jh} - c_j) + \sum_{j=h}^{k} (\alpha_h - c_j) \le f.$$

According to the first constraint, we are able to use $\alpha_h - c_j - c_h$ to replace u_{jh} in the above inequality. After moving some items to the right side, we have

$$\forall 1 \le h \le k : \sum_{j=1}^{k}(\alpha_h - c_j) \le f + \sum_{j=1}^{h-1}(c_j + c_h).$$

For the above inequality combining all cases for $1 \le h \le k$, we have

$$\sum_{h=1}^{k}(k\alpha_h - kc_h) \le k \cdot f + \sum_{h=1}^{k}\sum_{j=1}^{h-1}(c_j + c_h).$$

Observing $\sum_{h=1}^{k}\sum_{j=1}^{h-1}(c_j + c_h) = (k-1)\sum_{h=1}^{k} c_h$, we have

$$k\sum_{h=1}^{k}\alpha_h \le kf + (2k-1)\sum_{h=1}^{k} c_h,$$

that is,

$$\frac{\sum_{j=1}^{k}\alpha_j - f}{\sum_{j=1}^{k} c_j} \le \frac{2k-1}{k} < 2.$$

This yields the third result. □

From Theorem 3.6 the following corollary is immediate.

COROLLARY 3.7. *Algorithm* 4 *is a* 1.61-, (1.11,1.78)- *and* (1,2)-*approximation algorithm for the* metric *FTFA problem.*

Different from Algorithm 1, Algorithm 4 has to traverse all *fully connected* cities for each site, i.e., $\mathcal{C}^p \setminus U$, to obtain their maximum connection costs. Therefore it needs $O(|\mathcal{C}^p| \cdot n_f)$ time to reach the time that the next event occurs. So, in total Algorithm 4 needs at most $O(|\mathcal{C}^p| \cdot n_f^2)$ steps to complete each phase because Event-2 occurs at most n_f times.

LEMMA 3.8. *The time complexity of Algorithm* 4 *is* $O(Rn^3)$*, where* n *is the maximum of* n_f *and* n_c*.*

3.3. Cost scaling and greedy augmentation. Guha and Khuller [27] and Charikar and Guha [4] showed that it is possible to improve the performance of the JMS algorithm by using cost scaling and greedy augmentation. Similarly, we use the same technique to improve Algorithm 4. The combined algorithm is given in Algorithm 6.

ALGORITHM 6. 1.52-approximation FTFA.

(1) Scale the facility costs by δ: $f_i \leftarrow \delta f_i$ for all $i \in \mathcal{F}$.
(2) Run Algorithm 4 on the scaled instance.
(3) Scale back the facility costs and perform greedy augmentation. Define the gain of opening a facility at site i, $gain(i)$, to be the reduction in total cost obtained by opening a facility at i to the current solution ($gain(i) = 0$ if the total cost does not decrease). While there exist facilities with positive gains, choose the facility at i for which $\frac{gain(i)}{f_i}$ is maximized and add it to the current solution.

The next lemma was first proved in [27, 4] for the UFL problem and then in [11] for the FTFL problem. Noticing that the FTFA problem is a special case of the FTFL problem, we have the next lemma.

LEMMA 3.9 (see [30, 11]). *Let* F^* *and* C^* *be the facility cost and connection cost, respectively, of an optimal solution to the FTFA problem. Greedy augmentation, when applied to a solution with initial facility cost* F *and connection cost* C*, produces a solution of cost at most* $F + \max\{0, \ln(\frac{C-C^*}{F^*})\} \cdot F^* + F^* + C^*$.

The above lemma implies the following result.

LEMMA 3.10 (see [5, 23, 30]). *If Algorithm* 4 *is a* (λ_f, λ_c)*-approximation, Algorithm* 6 *with parameter* $\delta \geq 1$ *gives a* $(\lambda_f + \ln\delta, 1 + \frac{\lambda_c - 1}{\delta})$*-approximation solution for any instance of the FTFA problem.*

As showed by Mahdian, Ye, and Zhang [22], we get $\lambda_f + \ln\delta = 1 + \frac{\lambda_c - 1}{\delta} = 1.52$ taking $(\lambda_f, \lambda_c) = (1.11, 1.78)$ and $\delta = 1.504$, which implies that Algorithm 6 is a 1.52-approximation.

THEOREM 3.11. *Algorithm* 6 *is a* 1.52*-approximation with running time* $O(Rn^3)$ *for FTFA.*

4. The FTKFA problem. In this section, we consider the FTKFA problem which can be seen as a combination of the k-median problem and the FTFA problem.

The k-median problem [20] has also been studied extensively [1, 4, 6]. This problem requires opening no more than k medians in a set of geographically distributed candidate sites and connecting each city with the closest open median so that the total connection cost of all cities is minimized. The k-facility problem differs from the k-median problem by considering the specified operating cost for each facility and minimizing the combined cost for facility operating and connection establishing. The FTKFA problem is a further generalization of the k-facility problem, where the connectivity at each city is not necessarily equal to one. FTKFA is also an extension of FTFA by applying an extra upper bound on total open facility numbers, i.e., k.

Jain and Vazirani [15] reduced the k-facility problem to the UFL in the following way: Suppose $\mathcal{A}$ is an approximation algorithm for the facility location problem. Consider an instance $\mathcal{I}$ of the problem with optimum cost OPT, and let F and C be the facility and connection costs of the solution found by $\mathcal{A}$. Algorithm $\mathcal{A}$ is called a Lagrangian multiplier preserving (LMP) λ-approximation if for every instance $\mathcal{I}, C/\lambda + F \leq OPT$. Jain and Vazirani [15] proposed that an LMP λ-approximation algorithm for the metric UFL problem gives rise to a 2λ-approximation algorithm for the metric k-facility problem. Here we consider the fault-tolerant version of the problem. Instead of using the concept of LMP λ-approximation, we use bifactor approximation. We use a $(1,\lambda)$-approximation algorithm to FTFA as a subroutine to obtain a $(\lambda + \frac{1}{n_f})(2 - \frac{1}{n_f})$-approximation algorithm for the metric FTKFA problem. This result is better than 2λ when $\lambda \geq 2$ but worse than 2λ otherwise. Applying the result on the bifactor approximation given in the previous section, we know Algorithm 4 is a (1,2)-approximation to FTFA and therefore the result we get has a $4 - 1/n_f^2$ approximation factor for the FTKFA problem. The algorithm has the virtue of simplicity and can be completed efficiently in strongly polynomial time.

We also assume each city contains r_j ports and let $\mathcal{P}$ denote the set of all ports of all cities. Let s be a star composed of a facility and a group of ports connected with the facility. Let $\mathcal{S}$ be all possible stars and $\mathcal{S}_i$ all possible stars centered at site i. The FTKFA problem can be formulated by

$$
\begin{array}{ll}
\text{minimize} & \sum_{s\in\mathcal{S}} c_s x_s \\
\text{subject to} & \sum_{s\in\mathcal{S}} x_s \leq k \\
& \forall l \in \mathcal{P} : \sum_{s:l\in s} x_s \geq 1, \\
& \forall i \in \mathcal{F}, j \in \mathcal{C} : \sum_{l\in\mathcal{P}_j} \sum_{s:l\in s} x_s \leq \sum_{s\in\mathcal{S}_i} x_s, \\
& \forall s \in \mathcal{S} : x_s \in \{0,1\}.
\end{array} \tag{4.1}
$$

In the above integer linear programming (ILP), $\mathcal{P}_j$ is the set of all ports of city j. The first constraint ensures at most k facilities are opened in total, the second

ensures at least one connection for each port, and the third constraint ensures enough open facilities at each location so that connections between any site-city pair can be assigned to distinct facilities.

Suppose the number of the facilities opened by an algorithm for FTFA is k'. It is clear that the solution can be used directly if $k' \leq k$. So we only need to consider $k' > k$. In this case, in order to minimize the total cost, we can always open k facilities, i. e. $\sum_{s\in\mathcal{S}} x_s = k$. Let $\tilde{x}_s$ be the optimal solution of the primal problem (with facility cost f_i). We set the cost of operating a facility at site i to $f_i + z$, and let $c_s^- = \sum_{l\in s\cap\mathcal{P}} c_{i,j}$ and $c_s = c_s^- + f_{i(s)}$. Supposing an algorithm $\mathcal{A}$ is $(1,\lambda)$-approximation and it happens to open k facilities, we have

$$\sum_{s\in\mathcal{S}}(c_s + z)x_s \leq \sum_{s\in\mathcal{S}}(f_{i(s)} + z)\tilde{x}_s + \lambda\sum_{s\in\mathcal{S}} c_s^-\tilde{x}_s$$

and

$$\sum_{s\in\mathcal{S}} x_s = k \geq \sum_{s\in\mathcal{S}}\tilde{x}_s.$$

That is,

$$\sum_{s\in\mathcal{S}} c_s x_s \leq \sum_{s\in\mathcal{S}}(c_s + z)x_s - \sum_{s\in\mathcal{S}} z\tilde{x}_s \leq \sum_{s\in\mathcal{S}} f_{i(s)}\tilde{x}_s + \lambda\sum_{s\in\mathcal{S}} c_s^-\tilde{x}_s \leq \lambda\sum_{s\in\mathcal{S}} c_s\tilde{x}_s. \tag{4.2}$$

We can conclude that the solution is λ-approximation. However, this result relies on the assumption that the algorithm for FTKFA opens exactly k facilities. In the rest of the thesis, we assume such an algorithm does not exist, and instead we combine two solutions with k_1 and k_2 facilities, respectively, $k_1 < k < k_2$, to achieve a solution with k facilities.

4.1. Bisection search and combination. Jain and Vazirani proposed an approach to get a 2λ-approximation algorithm for the metric k-facility problem by using an LMP λ-approximation algorithm for the metric UFL problem [15]. They achieved a 6-approximation algorithm using an LMP 3-approximation algorithm [17, 15] and further a 4-approximation algorithm for UFL in [14]. Their approaches are based on the concept of LMP λ-approximation and as a result need an extra step described in [14] to transform the JMS algorithm which is a $(1,\lambda)$-approximation to UFL into an LMP λ-approximation algorithm. Our approach simplifies this process by eliminating the middle step and using $(1,\lambda)$-approximation algorithms directly. Note that our approach is for the fault-tolerant extension of this problem. We first prove that two $(1,\lambda)$-approximation solutions to FTFA can be combined to achieve a $(\lambda + \frac{1}{n_f})$-approximation fractional solution to FTKFA. In the next subsection, we will round the fractional solution, losing a small factor.

Consider an algorithm using a bisection search to approximate the value of z, i.e., facility cost $f_i + z$. Let $c_{\max}$ be the maximum of all connection costs; it is clear that $\sum_{s\in\mathcal{S}} x_s = \max_{j\in\mathcal{C}} r_j \leq k$ when $z = n_c c_{\max}$ and $\sum_{s\in\mathcal{S}} x_s \geq k$ when $z = 0$. Instead of using $n_c c_{\max}$ and 0 directly, we find two values of z which are very close and then combine corresponding solutions together. Assume the solutions are $\boldsymbol{x}^1$, $\boldsymbol{x}^2$, respectively, for z_1 and z_2, and $\sum_{s\in\mathcal{S}} x_s^1 = k_1$ and $\sum_{s\in\mathcal{S}} x_s^2 = k_2$. The combined solution $\boldsymbol{x} = a\boldsymbol{x}^1 + b\boldsymbol{x}^2$, where $a = (k_2 - k)/(k_2 - k_1)$ and $b = (k - k_1)/(k_2 - k_1)$. Now the problem is how efficiently we can find the values of z_1 and z_2 such that they are close enough to ensure the quality of the combined solution and how we can get an

integer solution from the combined fractional solution. We have the following lemma for the first problem.

LEMMA 4.1. *The cost of fractional solution* $\boldsymbol{x}$ *is within a factor of* $(\lambda + \frac{1}{n_f})$ *of the cost of an optimal fractional solution to FTKFA if* $z_1 - z_2 \leq \frac{Rf_{\min}+n_c c_{\min}}{kn_f}$.

Proof. Suppose the primal solution and the dual solution derived by Algorithm 4 are $(\boldsymbol{x}^1, \boldsymbol{\alpha}^1)$ and $(\boldsymbol{x}^2, \boldsymbol{\alpha}^2)$, respectively. Let $\tilde{x}_s$ be the optimal solution of the original problem (with facility cost f_i). We have

$$\sum_{s\in\mathcal{S}}(c_s + z_1)x_s^1 \leq \sum_{s\in\mathcal{S}}(f_{i(s)} + z_1)\tilde{x}_s + \lambda\sum_{s\in\mathcal{S}} c_s^- \tilde{x}_s$$

according to the definition of $(1,\lambda)$-approximation. Considering $\sum_{s\in\mathcal{S}} \tilde{x}_s \leq k$, we have

$$\sum_{s\in\mathcal{S}} c_s x_s^1 \leq z_1(k - k_1) + \lambda\sum_{s\in\mathcal{S}} c_s\tilde{x}_s. \tag{4.3}$$

Similarly we have

$$\sum_{s\in\mathcal{S}} c_s x_s^2 \leq z_2(k - k_2) + \lambda\sum_{s\in\mathcal{S}} c_s\tilde{x}_s \tag{4.4}$$

and

$$(z_1 - z_2)(k - k_2) \leq \frac{1}{n_f}\sum_{s\in\mathcal{S}} c_s\tilde{x}_s$$

because $z_1 - z_2 \leq \frac{Rf_{\min}+n_c c_{\min}}{kn_f} \leq \frac{\sum_{s\in\mathcal{S}} c_s\tilde{x}_s}{kn_f}$.

Now, we replace z_2 with z_1 in the first item of inequality (4.4) and increase the coefficient of the second item by $\frac{1}{n_f}$. (The coefficient of z_2, i.e., $k - k_2$, is negative.) Combined with inequality (4.4), we have

$$\sum_{s\in\mathcal{S}} c_s x_s^2 \leq z_1(k - k_2) + \left(\lambda + \frac{1}{n_f}\right)\cdot\sum_{s\in\mathcal{S}} c_s\tilde{x}_s. \tag{4.5}$$

Multiplying inequality (4.3) with constant $a = (k_2 - k)/(k_2 - k_1)$ and inequality (4.5) with constant $b = (k - k_1)/(k_2 - k_1)$, we have the items with z_1 eliminated, i.e.,

$$\sum_{s\in\mathcal{S}} c_s x_s \leq \left[a\lambda + b\left(\lambda + \frac{1}{n_f}\right)\right]\sum_{s\in\mathcal{S}} c_s\tilde{x}_s \leq \left(\lambda + \frac{1}{n_f}\right)\sum_{s\in\mathcal{S}} c_s\tilde{x}_s.$$

The lemma follows. ∎

Since the total range of z is $f_{\max} + n_c c_{\max}$ and the interval between z_1 and z_2 is required to be smaller than $(Rf_{\min} + n_c c_{\min})/kn_f$, so the total number of probing steps is $\log\frac{kn_f n_c c_{\max}}{Rf_{\min}+n_c c_{\min}}$. Letting $L = c_{\max}/(Rf_{\min} + n_c c_{\min})$ and $n = \max(n_c, n_f)$, we have the following lemma.

LEMMA 4.2. *After* $O(\log(nL))$ *probes on* z *using a bisection search,* z_1 *and* z_2 *are so close that* $z_1 - z_2 \leq (Rf_{\min} + nc_{\min})/kn$ *and* $k_1 \leq k \leq k_2$.

We notice that Swamy and Shmoys [30] achieved a similar result. Our solution applies a similar approach to that in [30] but differs in three aspects:

1. Their result only applies the uniform connectivity case for the fault tolerant k-facility location problem where each site allows at most one facility while ours applies for both the uniform case and the general case and each site allows an unlimited number of facilities.
2. Their bisection search needs $O(\frac{poly(n)}{L'})$ steps, where $L' = \log c_{\max}$, while ours only needs $O(\log n + \log L)$, where $L = c_{\max}/(Rf_{\min} + nc_{\min})$. This is because we do not require the corresponding dual solutions are identical for the two primal solutions, and as a result the length of search interval is substantially greater.
3. Their approach depends on how to break ties between events in the primal-dual algorithm, while ours does not.

4.2. Randomized procedure for rounding.

Facility opening. We use the same randomized procedure as in [17] to open facilities. We show that a similar result can also be achieved in the fault tolerance context.

Let A and B be the sets of open facilities in the two solutions, $|A| = k_1$ and $|B| = k_2$. For each facility in A, we find the closest facility in B, which are not required to be distinct to each other. Let $B' \subset B$ be these facilities. If $|B'| \leq k_1$, we arbitrarily include additional facilities from $B \setminus B'$ into B' until $|B'| = k_1$. Now we open all facilities in A with probability a and open all facilities in B' with probability $b = 1 - a$. In addition, a set of cardinality $k - k_1$ is picked randomly from $B \setminus B'$ and facilities in this set are opened. Furthermore, each facility in B is opened with probability b. The procedure is demonstrated in Figure 4.1. For convenience, we use $\hat{y}_i$ and $\hat{x}_{ij}, i \in \mathcal{F}, j \in \mathcal{C}$, to denote the integer solution in which there are totally k open facilities; we have the following lemma.

LEMMA 4.3. *The expected facility cost* $E[\sum_{i\in\mathcal{F}} f_i\hat{y}_i]$ *is no more than* $a\sum_{i\in A} f_i x_i + b\sum_{i\in B} f_i x_i$.

Connection establishment. Instead of connecting a city with the r_j nearest open facilities, we consider a suboptimal approach for the sake of approximation factor revealing. The approach is first proposed in [30]. Our analysis follows the same idea but leads to a more strict result because we can only lose a factor $(2 - \frac{1}{n_f})$ instead of 2 to achieve an approximation ratio smaller than 4. This is because the factor we lost in the last step is $(2 + \frac{1}{n_f})$ for the sake of time complexity. We introduce the procedure

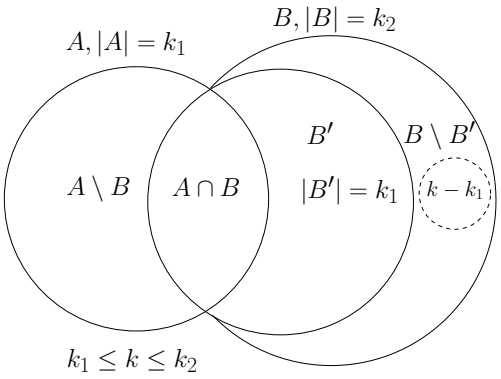


FIG. 4.1. *Randomized procedure.*

briefly as follows. Note that this approach is used only in the analysis because for the algorithm the optimal approach, i.e., connecting a city with the r_j nearest open facilities, is always preferred.

Let A_l be the set of facilities in A to which city $j \in \mathcal{C}$ is connected, namely, $A_j = \{i \in A : x_{ij} = 1\}$. Similarly, let B_j be the set of facilities in B that serve j. Clearly $|A_j| = |B_j| = r_j$. For each port l of city j, we define a set of facilities T_l, and l will only be connected to a facility in T_l. First, we arbitrarily assign each facility $i \in A_j$, and the facility in B_j to which it is matched (which could be the same as i), to a distinct set T_l. Observe the important fact that the sets T_l are disjoint, since distinct facilities in A_j are matched to distinct facilities in B. Let $m(A_j)$ denote the set of facilities that are matched to facilities in A_j. Then $|m(A_j)| = |A_j| = |B_j| \Rightarrow |m(A_j) \setminus B_j| = |B_j \setminus m(A_j)|$, so the number of T_l's not containing a facility from B_j after the first step is equal to the number of unmatched facilities in B_j. We assign a distinct unmatched facility of B_j to each set T_l which does not already contain a facility from B_j. Note that the sets T_l remain disjoint, so if we connect each port l to a facility in T_l, we will get a feasible solution.

LEMMA 4.4. *After the above randomized procedure, a city j is connected with r_j distinct facilities.*

Furthermore, we have the following lemma on the connection cost.

LEMMA 4.5. *The expected connection cost for a city j, denoted by $E[cost(j)]$, is no more than $(1 + \max(a, b)) \sum_{i \in \mathcal{F}} c_{ij} x_{ij}$.*

Proof. For convenience, if facility $i \in A_j$ is matched with i', we will consider i and i' as two different facilities even if $i = i'$. Let the service cost of city $j \in \mathcal{C}$ be $cost(j)$ and the service cost of port l be $cost(l)$. The set T_l contains at least one small facility $i_1 \in A_j$ and one large facility i_2 such that i_1 is matched with i_2.

If these are the only two facilities, then it must be that $i_2 \in B_j$. Either i_1 or i_2 is open, and we assign l to that open facility. So $E[cost(l)] = ac_{i_1 j} + bc_{i_2 j}$.

Otherwise, T_l contains a third facility $i_3 \in B_j$ such that i_3 is unmatched and $i_2 \in B_j$ .We assign l to i_3 if it is open and to i_1 or i_2, whichever is open, otherwise. So $E[cost(l)] = a(ac_{i_1 j} + bc_{i_2 j}) + bc_{i_3 j}$. Since i_1 is matched with i_2 and i_3 is unmatched, it must be that i_2 is closer to i_1 than i_3. So,

$$c_{i_2 j} \le c_{i_1 j} + c_{i_1 i_2} \le c_{i_1 j} + c_{i_1 i_3} \le 2c_{i_1 j} + c_{i_3 j}.$$

Therefore

$$\begin{aligned} E[cost(l)] &\le bc_{i_3 j} + a(ac_{i_1 j} + 2bc_{i_1 j} + bc_{i_3 j}) \\ &= a(1 + b)c_{i_1 j} + b(1 + a)c_{i_3 j}. \end{aligned}$$

Thus for every port l, if $i, i' \in T_l$, where $i \in A_j$ and $i' \in B_j$, we have

$$E[cost(l)] \le a(1 + b)c_{ij} + b(1 + a)c_{i'j}).$$

For both cases, we have $E[cost(l)] \le (1 + \max(a, b))(ac_{ij}x^1_{ij} + bc_{i'j}x^2_{i'j})$, since $x^1_{ij} = x^2_{i'j} = 1$. So, summing up the costs for all ports l, since the set of all facilities at i for the first component of the last item is precisely A_j and the set of all facilities at i for the second component is the set B_j, we get

$$\begin{aligned} E[cost(j)] &\le (1 + \max(a, b)) \left(\sum_{i \in A_j} ac_{ij}x^1_{ij} + \sum_{i \in B_j} bc_{ij}x^2_{ij} \right) \\ &= (1 + \max(a, b)) \sum_{i \in \mathcal{F}} c_{ij}(ax^1_{ij} + bx^2_{ij}), \end{aligned}$$

where the last equation holds since $x^1_{ij} = 0$ if $i \notin A_j$ and $x^2_{ij} = 0$ if $i \notin B_j$. The lemma follows because $x_{ij} = (ax^1_{ij} + bx^2_{ij})$. ∎

Approximation factor. We have the following theorem.

THEOREM 4.6. *A $(1, \lambda)$-approximation algorithm for FTFA can result in a $(\lambda + \frac{1}{n_f})(2 - \frac{1}{n_f})$-approximation stochastic algorithm for FTKFA.*

Proof. According to Lemma 4.4, we have

$$E\left[\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}\hat{x}_{ij}\right] \le (1 + \max(a, b))\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}x_{ij}.$$

According to Lemma 4.5, we have

$$E\left[\sum_{i\in\mathcal{F}} f_i\hat{y}_i\right] \le (a + b)\sum_{i\in A\cup B} f_i y_i \le (1 + \max(a, b))\sum_{i\in\mathcal{F}} f_i y_i.$$

Combining them together, we have

$$E\left[\sum_{s\in\mathcal{S}} c_s\hat{x}_s\right] \le (1 + \max(a, b))\sum_{s\in\mathcal{S}} c_s x_s.$$

On the other hand, it's easy to see that $a \le 1 - 1/n_f$ (this happens for $k_1 = k - 1$ and $k_2 = n_f$) and $b \le 1 - 1/k$ (this happens for $k_1 = 1$ and $k_2 = k + 1$). Therefore, $1 + \max(a, b) \le 2 - 1/n_f$. Combined with Lemma 4.1, the theorem follows. ∎

4.3. Derandomization. Because the randomization procedure is used only to open facilities (we always connect a city to the nearest open facilities), the derandomization technique proposed by Jain and Vazirani [17] can be applied here directly. We have the following result.

LEMMA 4.7. *The bisection search based deterministic algorithm which employs Algorithm 4 as a subroutine is a $(4 - \frac{1}{n_f^2})$-approximation algorithm for FTKFA and its time complexity is $O(Rn^3 \log(nL))$, where $L = c_{\max}/(Rf_{\min} + nc_{\min})$ and $n = \max(n_c, n_f)$.*

5. Discussion.

5.1. Dealing with demand. As mentioned, the FTFA problem with nonuniform demands (access frequencies) is equivalent to the FTFA problem with independent demands. Suppose the demand of city j is d_j; the problem becomes

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}d_j x_{ij} \\ \text{subjected to} & \forall j \in \mathcal{C} : \sum_{i\in\mathcal{F}} x_{ij} \ge r_j, \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij} \le y_i, \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij}, y_i \in \mathbb{Z}^+. \end{array}$$

When d_j is an integer, cost $d_j \cdot c_{i,j}$ implies that there are d_j copies of city j at the same location. It is clear that the new problem can be transformed into the FTFA problem with each city being replicated d_j copies. When d_j is not an integer, we multiply a large number with both d_j and f_i so that d'_j becomes an integer. It is not hard to show that the new problem has the same solution as the original problem. By applying the above approach, the problem can be transformed into an FTFA problem.

5.2. Fault tolerant network design. The FTFA problem also arises in the fault tolerant network design. Suppose the downtime ratio is uniformly σ for each facility and the usability required by city j is μ_j (percent of time that a city is serviced).

If the downtime of facilities (or links) is predicable (deterministic), for example, in a system where each facility needs a fraction of time to "rest," the corresponding network design problem can be modeled as an FTFA problem with r_j set to $\lceil \mu_j/(1-\sigma) \rceil$. If the downtime is unpredictable (stochastic), then r_j should be set to $\lceil \log_\sigma(1-\mu_j) \rceil$. In both cases, the proposed algorithms are able to solve the problem. However, if facilities or links have nonuniform downtimes, the constraint on connectivity becomes $\sum_{i\in\mathcal{F}}(1-\sigma_{ij})x_{ij} \geq \mu_j$ for the deterministic model and $\prod_{i\in\mathcal{F}:x_{ij}=1}\sigma_{ij} \leq 1-\mu_j$ for the stochastic model, where σ_{ij} is the downtime ratio of connection (i,j). For these cases our FTFA algorithms cannot be directly applied to solve the problem. We shall leave them as open problems for future study.

Acknowledgments. This work was supported by the National Science Foundation of China under general projects funding 61170232 and the 985 project funding of Sun Yat-Sen University.

REFERENCES

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, Local search heuristic for k-median and facility location problems, in *STOC '01: Proceedings of the 33rd Annual ACM symposium on Theory of Computing*, New York, NY, 2001, pp. 21–29.

[2] J. Byrka, *An optimal bifactor approximation algorithm for the metric uncapacited facility location problem*, in Proceedings of APPROX and RANDOM '07, 2007.

[3] J. Byrka, A. Srinivasan, and C. Swamy, *Fault-tolerant facility location: A randomized dependent lp-rounding algorithm*, in Proceedings of IPCO, 2010, pp. 244–257.

[4] M. Charikar and S. Guha, *Improved combinatorial algorithms for the facility location and k-median problems*, in FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, IEEE, Washington, DC, 1999, p. 378.

[5] M. Charikar and S. Guha, *Improved combinatorial algorithms for facility location problems*, SIAM J. Comput., 34 (2005), pp. 803–824.

[6] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys, *A constant-factor approximation algorithm for the k-median problem*, in Proceedings of the ACM Symposium on the Theory of Computing, 1999, pp. 1–10.

[7] F. A. Chudak and D. B. Shmoys, *Improved approximation algorithms for the uncapacitated facility location problem*, SIAM J. Comput., 33 (2004), pp. 1–25.

[8] F. A. Chudak and D. P. Williamson, *Integer programming and combinatorial optimization*, in Improved Approximation Algorithms for Capacitated Facility Location Problems, Lecture Notes in Computer Sci. 1610, Springer, Berlin, 1999, pp. 99–113.

[9] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, *Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms*, Management Sci., 23 (1977), pp. 789–810.

[10] S. Guha, A. Meyerson, and K. Munagala, *Improved algorithms for fault tolerant facility location*, in SODA '01: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, 2001, pp. 636–641.

[11] S. Guha, A. Meyerson, and K. Munagala, *A constant factor approximation algorithm for the fault-tolerant facility location problem*, J. Algorithms, 48 (2003), pp. 429–440.

[12] D. S. Hochbaum, *Heuristics for the fixed cost median problem*, Math. Program., 22 (1982), pp. 148–162.

[13] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, *Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP*, J. ACM, 50 (2003), pp. 795–824.

[14] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, in STOC '02: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, New York, 2002, pp. 731–740.

[15] K. Jain and V. V. Vazirani, *Primal-dual approximation algorithms for metric facility location and k-median problems*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, 1999, pp. 2–13.

[16] K. Jain and V. V. Vazirani, *An approximation algorithm for the fault tolerant metric facility location problem*, in APPROX '00: Proceedings of the Third International Workshop

on Approximation Algorithms for Combinatorial Optimization, London, Berlin, Springer-Verlag, 2000, pp. 177–183.

[17] K. Jain and V. V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation*, J. ACM, 48 (2001), pp. 274–296.

[18] K. Liao and H. Shen, *Unconstrained and constrained fault-tolerant resource allocation*, in COCOON '11: Proceedings of the 17th Annual International Computing and Combinatorics Conference, Dallas, TX, Springer, Berlin, 2011, pp. 555–566.

[19] K. Liao and H. Shen, *LP-based approximation algorithms for reliable resource allocation*, Computer J., DOI: 10.1093/compj/bxs164, Jan 11, 2013, to appear.

[20] J.-H. Lin and J. S. Vitter, *e-approximations with minimum packing constraint violation*, in STOC '92: Proceedings of the 24th Annual ACM Symposium on Theory of Computing, New York, 1992, pp. 771–782.

[21] M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani, *A greedy facility location algorithm analyzed using dual fitting*, In APPROX '01/RANDOM '01: Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 5th International Workshop on Randomization and Approximation Techniques in Computer Science, London, Springer-Verlag, Berlin, 2001, pp. 127–137.

[22] M. Mahdian, Y. Ye, and J. Zhang, *A 1.52-approximation algorithm for the uncapacitated facility location problem*, in Proceedings of APPROX '02, Lecture Notes in Comput. Sci. 2462, Springer-Verlag, Berlin, 2002, pp. 229–242.

[23] M. Mahdian, Y. Ye, and J. Zhang, *Approximation algorithms for metric facility location problems*, SIAM J. Comput., 36 (2006), pp. 411–432.

[24] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch, *New upper bounds on the rate of a code via the delsarte-macwilliams inequalities*, IEEE Trans. Inform. Theory, 23 (1977), pp. 157–166.

[25] A. Saberi, V. Vazirani, M. Mahdian, and E. Markakis, *A greedy facility location algorithm analyzed using dual fitting*, in Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, Springer-Verlag, Berlin, 2001, pp. 127–137.

[26] R. L. F. Pitu and B. Mirchandani, eds., *Discrete Location Theory*, John Wiley, New York, 1990.

[27] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms, 31 (1999), pp. 228–248.

[28] D. B. Shmoys, E. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 265–274.

[29] M. Sviridenko, *An improved approximation algorithm for the metric uncapacitated facility location problem*, in Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, London, Springer-Verlag, Berlin, 2002, pp. 240–257.

[30] C. Swamy and D. B. Shmoys, *Fault-tolerant facility location*, ACM Trans. Algorithms, 4 (2008), pp. 1–27.

[31] V. V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.

[32] S. Xu and H. Shen, *The Fault-Tolerant Facility Allocation Problem*, Proceedings of ISAAC 2009, Hawaii, 2009, pp. 689–698.