ROBUST FAULT TOLERANT UNCAPACITATED FACILITY LOCATION

SHIRI CHECHIK [1] AND DAVID PELEG [1]

[1] Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel
E-mail address: {shiri.chechik,david.peleg}@weizmann.ac.il

ABSTRACT. In the *uncapacitated facility location* problem, given a graph, a set of demands and opening costs, it is required to find a set of facilities R, so as to minimize the sum of the cost of opening the facilities in R and the cost of assigning all node demands to open facilities. This paper concerns the *robust fault-tolerant* version of the uncapacitated facility location problem (RFTFL). In this problem, one or more facilities might fail, and each demand should be supplied by the closest open facility that did not fail. It is required to find a set of facilities R, so as to minimize the sum of the cost of opening the facilities in R and the cost of assigning all node demands to open facilities that did not fail, after the failure of up to α facilities. We present a polynomial time algorithm that yields a 6.5-approximation for this problem with at most one failure and a $1.5 + 7.5\alpha$-approximation for the problem with at most $\alpha > 1$ failures. We also show that the $RFTFL$ problem is NP-hard even on trees, and even in the case of a single failure.

Introduction

The robust fault-tolerant facility location problem

For a given optimization problem, the robust fault-tolerant version of the problem calls for finding a solution that is still valid even when some components of the system fail. We consider the robust fault-tolerant version of the *uncapacitated facility location (UFL)* problem. In this problem, given a graph G, a demand $\omega(v)$ for every node v and a cost $f(v)$ for opening a facility at v, it is required to find a set of facilities R, so as to minimize the sum of the costs of opening the facilities in R and of shipping the demands of each node from the nearest open facility (at a cost proportional to the distance). In the robust fault-tolerant version of this problem (RFTFL), one or more facilities might fail. Subsequently, each demand should be supplied by the closest open facility that did not fail. It is required to select a set of facilities R, so as to minimize the sum of the costs of opening the facilities in R and the costs of assigning all node demands to open facilities that did not fail, after the failure of up to α facilities. We present a polynomial time algorithm that yields a 6.5-approximation for this problem with at most one failure and a $1.5 + 7.5\alpha$-approximation

for the problem with at most $\alpha > 1$ failures. We also show that the $RFTFL$ problem is NP-hard even on trees, and even in the case of a single failure.

Related Work

Many papers deal with approximating the UFL problem, cf. [3, 4, 7, 9, 12, 13]. The best approximation ratio known for this problem is 3/2, shown by Byrka in [2].

A fault-tolerant version of the facility location problem was first introduced by Jain and Vazirani [10], who gave it an approximation algorithm with ratio dependent on the problem parameters. The approximation ratio was later improved by Guha et al. to 2.41 [8] and then by Swamy and Shmoys to 2.076 [14]. However, the variant of the problem studied in these papers is different from the one studied here. In that version, every node j is assigned in advance to a number of open facilities, and pays in advance for all of them. More explicitly, every node j is assigned to r_j open facilities, and its shipping cost is some weighted linear combination of the costs of shipping its demand from all the facilities to which it is assigned. It is required to find a set of facilities R that minimizes the sum of the costs of opening the facilities in R and the sum of costs of shipping the demand of each node j from its r_j facilities in R. This approach is used to capture the expected cost of supplying all clients demand when some of the facilities fail. In contrast, in our definition a node j does not pay in advance for shipping its demand from a number of open facilities. Rather, it pays only for the cost of shipping its demand from the surviving facility that actually supplied its demand. Hence our definition for the fault-tolerant facility location problem requires searching for a set of facilities R that minimizes the sum of the costs of opening the facilities in R and the costs of assigning the demands of each node to one open facility that did not fail, for any failure of up to α facilities. Our approach is used to capture the worst case cost of supplying all clients demand when some of the facilities fail. We argue that our definition may be more natural in some cases, where after the failure of some facilities, each demand should still be supplied by a single supplier, preferably the closest surviving open facility, and each client should pay only for the cost of shipping its demand from that surviving facility, and not for all the other (possibly failed) facilities to which it was assigned originally. On the technical level, the approach taken in [8, 10, 14] is based on applying randomized rounding techniques and primal-dual methods to the corresponding integer linear program. This approach does not readily apply to our version of the problem, and we use a direct combinatorial algorithmic approach instead.

Two other closely related types of problems are the 2-stage stochastic and robust optimization problems (cf. [5, 6]). Both of these models involve two decision stages. In the first stage, some facilities may be purchased. This stage is followed by some scenario depending on the specifics of the problem at hand (in a facility location problem for example, the scenario may specify the clients and their corresponding demands). Subsequently, a second stage is entered, in which it is allowed to purchase additional facilities (whose cost might be much higher than in the first stage). In stochastic optimization there is a distribution over all possible scenarios and the goal is to minimize the expected total cost. In robust optimization the goal is to minimize the cost of the first stage plus the cost of the worst case scenario in the second stage. In contrast with these two models, in our variant the facilities must be selected and opened in advance, and these advance decisions must be adequate under all possible future scenarios.

Billionnet and Costa [1] showed a polynomial time algorithm for solving the ordinary (non-fault-tolerant) UFL problem on trees. In contrast, we show that the fault-tolerant variant RFTFL is NP-hard on trees.

1. Preliminaries

Let us start with common notation to be used later on. Consider an optimization problem Π over a universe V, which given an instance I, requires finding a solution consisting of a set of elements $R \subseteq V$. Denote by $C_\Pi(I, R)$ the cost of the solution R on the instance I of Π. Let $R_\Pi^*(I)$ denote the optimal solution to the problem Π on instance I, and let $C_\Pi^*(I) = C_\Pi(I, R_\Pi^*(I))$ be the cost of the optimal solution. We denote our algorithm for each problem Π studied later by $A_\Pi(I)$. The solution returned by the algorithm is referred to as $R_\Pi^{alg}(I)$ and its cost is $C_\Pi^{alg}(I) = C_\Pi(I, R_\Pi^{alg}(I))$.

Let us now define the *uncapacitated facility location (UFL)* problem. Let $I = \langle G, l, f, \omega \rangle$ be an instance of the problem, where $G = (V, E)$ is a graph with vertex set $V = \{1, ..., n\}$ and edge set E. Each node $v \in V$ hosts a client in need of service, and may host a facility, providing service to clients in nearby nodes. Each edge $e \in E$ has a positive length $l(e)$. The distance $d(u, v)$ between two points u and v on G is defined to be the length of the shortest path between them, where the length of a path is the sum of the lengths of its edges. For each node v, let $f(v)$ denote the *opening cost* associated with placing a facility at v, and let $\omega(v)$ denote the demand of the node v. The *shipping cost* of assigning the demand $\omega(u)$ of a client u to an open facility v is the product $SC_{u,v} = \omega(u)d(u, v)$. The shipping cost $SC_{u,R}$ from a set of open facilities R to a node u is the minimum cost of assigning u to a server in R, namely, $SC_{u,R} = \min\{SC_{u,v} \mid v \in R\}$. Defining the distance $d(v, R)$ between a set of points R and a point v on G to be the minimum distance between v and any node in R, i.e., $d(v, R) = \min_{r \in R} d(v, r)$, we also have $SC_{v,R} = \omega(v)d(v, R)$.

It is required to find a subset $R \subseteq V$ that minimizes the sum of costs of opening the facilities in R and the shipping costs from R to all other nodes. This problem can be formulated as searching for a subset $R \subseteq \{1, ..., n\}$ that minimizes the cost function

$$C_{UFL}(I, R) = C_{facil}(I, R) + C_{ship}(I, R), \tag{1.1}$$

where

$$C_{facil}(I, R) = \sum_{r \in R} f(r) \text{ and } C_{ship}(I, R) = \sum_{u=1}^{n} SC_{u,R} = \sum_{u=1}^{n} \omega(u) \cdot d(u, R).$$

Given a set R of open facilities and a facility $r \in R$, let $\varphi(I, r, R)$ denote the set of clients that are served by r under R, i.e., $\varphi(I, r, R) = \{u \mid d(v, r) \leq d(v, r') \; for \; every \; r' \in R\}$, or in other words, the nodes u that satisfy $d(u, R) = d(u, r)$, where ties are broken arbitrarily, i.e., if there is more than one open facility r such that $d(u, R) = d(u, r)$, then just choose one open facility r that satisfies $d(u, R) = d(u, r)$ and add u to $\varphi(I, r, R)$. (When the set R is clear from the context we omit it and write simply $\varphi(I, r)$, or even $\varphi(r)$ when the instance I is clear as well.)

The *robust fault-tolerant facility location (RFTFL)* problem is defined as follows. Each client is supplied by the nearest open facility, and in case this facility fails - it is supplied by the next nearest open facility. We would like to find a solution that is tolerant against a failure of one node. This problem can be formulated as searching for a subset $R \subseteq \{1, ..., n\}$ that minimizes the cost function

$$C_{RFTFL}(I, R) \ = \ C_{facil}(I, R) + C_{ship}(I, R) + C_{backup}(I, R), \tag{1.2}$$

where $C_{facil}(I, R)$ and $C_{ship}(I, R)$ are defined as above and

$$C_{backup}(I, R) \ = \ \max_{r \in R} \left\{ \sum_{v \in \varphi(I, r, R)} \omega(v) \cdot (d(v, R \backslash \{r\}) - d(v, r)) \right\} . \tag{1.3}$$

Note that

$$
\begin{aligned}
C_{RFTFL}(I, R) \ &= \ C_{facil}(I, R) + \max_{r \in R} \left\{ C_{ship}(I, R \setminus \{r\}) \right\} \\
&= \ C_{facil}(I, R) + \max_{r \in R} \left\{ \sum_{v=1}^{n} SC_{v, R \backslash \{r\}} \right\} \\
&= \ C_{facil}(I, R) + \max_{r \in R} \left\{ \sum_{v=1}^{n} \omega(v) \cdot d(v, R \backslash \{r\}) \right\} .
\end{aligned}
\tag{1.4}
$$

Again, when the instance I is clear from the context we omit it and write simply $C_{RFTFL}(R)$, $C_{facil}(R)$, $C_{ship}(R)$, $C_{backup}(R)$, etc.

We also consider the *robust α-fault-tolerant facility location (α_RFTFL)* problem, for integer $\alpha \geq 1$, where the solution should be resilient against a failure of up to α nodes. We define the α_RFTFL as follows. Each client is supplied by the nearest open facility which did not fail. We are looking for a subset $R \subseteq \{1, ..., n\}$ that minimizes the cost function

$$C_{\alpha\_RFTFL}(I, R) \ = \ C_{facil}(I, R) + \max_{|R'| \leq \alpha} \left\{ \sum_{v=1}^{n} \omega(v) \cdot d(v, R \setminus R') \right\} . \tag{1.5}$$

2. A constant approximation algorithm for RFTFL

2.1. The concentrated backup problem and its approximation

Towards developing a constant ratio approximation algorithm for RFTFL, we first consider a different problem, named *concentrated backup (conc_bu)*, defined as follows. An instance of the problem consists of a pair $\langle I, R_1 \rangle$ where $I = \langle G, l, f, \omega \rangle$ is defined as before and $R_1 = \{r_1, ..., r_k\}$ is a set of nodes. In this version, the nodes of R_1 act as both clients and servers (with open facilities), and all other nodes $v \notin R_1$ have zero demands. Informally, it is assumed that we have already paid for opening the facilities in R_1, and each $r \in R_1$ serves itself, at zero shipping cost. The problem requires to assign each client $r \in R_1$ to a backup server $v \neq r$, which may be either some server in R_1 or a new node from $V \backslash R_1$. For a set of nodes R_2, define the *backup cost*

$$C_{bu}(I, R_1, R_2) = \max_{r \in R_1} \left\{ SC_{r, R_1 \cup R_2 \backslash \{r\}} \right\} = \max_{r \in R_1} \left\{ \omega(r) d(r, R_1 \cup R_2 \backslash \{r\}) \right\}.$$

We are looking for a set R_2 minimizing

$$C_{conc\_bu}(I, R_1, R_2) = C_{facil}(R_2) + C_{bu}(R_1, R_2). \tag{2.1}$$

We denote this minimum cost by $C^*_{conc\_bu}(I, R_1)$. We show a 2-approximation algorithm for the concentrated backup problem.

The problems studied in this section and in section 3.1 are closely related to those considered in [11], and to solve them we use methods similar to the ones presented in [11].

Let us consider a simpler variant of the backup problem, named the *bounded backup (bb)* problem, which is defined on $\langle I, R_1, M \rangle$ and requires looking for a solution R_2 minimizing

$$C_{bb}(I, R_1, M, R_2) = C_{facil}(R_2)$$

subject to the constraint $C_{bu}(R_1, R_2) \leq M$, for integer M. We now present a relaxation algorithm that finds a set R_2 satisfying $C_{facil}(R_2) \leq C_{bb}^*(R_1, M)$ but obeying only the relaxed constraint $C_{bu}(R_1, R_2) \leq 2M$ instead $C_{bu}(R_1, R_2) \leq M$.

Algorithm $A_{bb}(I, R_1, M)$

    (1) $R_{bb}^{alg} \leftarrow \emptyset$

    (2) For $i = 1$ to k do:

        • $S_i \leftarrow \{v \mid \omega(r_i)d(v, r_i) \leq 2M\}\backslash\{r_i\}$ /* "relaxed" backup servers for r_i */

        • If $S_i \cap (R_1 \cup R_{bb}^{alg}) = \emptyset$ then add to R_{bb}^{alg} the node v in S_i with the minimum facility cost $f(v)$.

    (3) Return R_{bb}^{alg}.

Let us now prove the properties of algorithm A_{bb}. For every $r_i \in R_1$ let the set of feasible backup servers be $T_i = \{v \mid \omega(r_i)d(v, r_i) \leq M\}\backslash\{r_i\}$. Let the set of relaxed backup servers selected by the algorithm (namely, the final set R_{bb}^{alg} it returns) be $R_{bb}^{alg}(R_1, M) = \{q_1^{alg}, ..., q_J^{alg}\}$. Let ℓ_j be the phase in which the algorithm adds the new facility q_j^{alg} to R_{bb}^{alg}, for $1 \leq j \leq J$.

Lemma 2.1. $T_{\ell_i} \cap T_{\ell_j} = \emptyset$ for $1 \leq i, j \leq J$.

Proof: Assume otherwise, and let $v \in T_{\ell_i} \cap T_{\ell_j}$ for some $1 \leq i, j \leq J, i \neq j$. Assume without loss of generality that $\omega(r_{\ell_i}) \leq \omega(r_{\ell_j})$. Since $\omega(r_{\ell_j})d(v, r_{\ell_j}) \leq M$, necessarily $\omega(r_{\ell_i})d(v, r_{\ell_j}) \leq M$ as well, and by the definition of T_{ℓ_i}, also $\omega(r_{\ell_i})d(v, r_{\ell_i}) \leq M$, hence

$$\omega(r_{\ell_i})d(r_{\ell_i}, r_{\ell_j}) \leq \omega(r_{\ell_i})(d(v, r_{\ell_i}) + d(v, r_{\ell_j})) \leq 2M,$$

implying that $r_{\ell_j} \in S_{\ell_i} \cap R_1$, so the algorithm should not have opened a new facility in phase ℓ_i, contradiction. ∎

Lemma 2.2. $C_{facil}(R_{bb}^{alg}(R_1, M)) \leq C_{bb}^*(R_1, M)$.

Proof: Notice that there must be at least one node from every T_i in the optimal solution $R_{bb}^*(R_1, M)$. By Lemma 2.1 the sets $T_{\ell_1}, ..., T_{\ell_J}$ are disjoint, so there are at least J distinct nodes $q_j^* \in R_{bb}^*(R_1, M)$, one from each T_{ℓ_j}, for $1 \leq j \leq J$. In each phase i, the algorithm selects the cheapest node in $S_i \supseteq T_i$. Therefore, $f(q_j^{alg}) \leq f(q_j^*)$ for every $1 \leq j \leq J$. Hence

$$C_{facil}(R_{bb}^{alg}(R_1, M)) = \sum_{j=1}^{J} f(q_j^{alg}) \leq \sum_{j=1}^{J} f(q_j^*) \leq C_{bb}^*(R_1, M). \quad \blacksquare$$

Lemma 2.3. $C_{bu}(R_1, R_{bb}^{alg}(R_1, M)) \leq 2M$.

Proof: For each server r_i in R_1, the algorithm ensures that there is at least one open facility from the set S_i, so $\omega(r_i)d(r_i, R_1 \cup R_{bb}^{alg}(R_1, M) \setminus \{r_i\}) \leq 2M$. ∎

Now we present an approximation algorithm A_{conc_bu} for the concentrated backup problem using the relaxation algorithm A_{bb} for the bounded backup problem. First note that there can be at most nk possible values for the shipping costs $SC_{u,v} = \omega(u)d(u,v)$.

Algorithm $A_{conc\_bu}(I, R_1)$
  (1) For every $M \in \{SC_{u,v} \mid u,v \in V\}$ do:
    • let $R_{bb}^{alg}(R_1, M) \leftarrow A_{bb}(I, R_1, M)$.
  (2) Return the set $R_{bb}^{alg}(R_1, M)$ with the minimum cost $C_{conc\_bu}(R_1, R_{bb}^{alg}(R_1, M))$.

Lemma 2.4. $C_{conc\_bu}^{alg}(I, R_1) \leq 2C_{conc\_bu}^*(I, R_1)$.

Proof: Recall that, letting $R_2^* = R_{conc\_bu}^*(R_1)$,

$$C_{conc\_bu}^*(I, R_1) = C_{conc\_bu}(I, R_1, R_2^*) = C_{facil}(R_2^*) + C_{bu}(I, R_1, R_2^*).$$

Let $u \in R_1$ be the node that attains the maximum shipping cost $SC_{u,R_1 \cup R_2^* \setminus \{u\}}$, i.e., satisfies $\omega(u)d(u, R_1 \cup R_2^* \setminus \{u\}) = C_{bu}(I, R_1, R_2^*)$, and let $v \in R_1 \cup R_2^* \setminus \{u\}$ be its backup, i.e., the closest node to u. Then $C_{conc\_bu}^*(I, R_1) = C_{conc\_bu}(I, R_1, R_2^*) = C_{facil}(R_2^*) + SC_{u,v}$. Since the algorithm examines all possible values of M, it tests also $M_0 = SC_{u,v}$. For this value, the returned set $R_{bb}^{alg}(R_1, M_0)$ has opening cost at most $C_{bb}^*(R_1, M_0) = C_{facil}(R_2^*)$ and backup cost at most
$C_{bu}(I, R_1, R_{bb}^{alg}(R_1, M_0)) \leq 2M_0$ by Lemmas 2.2 and 2.3. Since the algorithm takes the minimum cost $C_{conc\_bu}(R_1, R_{bb}^{alg}(R_1, M))$ over all possible values of M, the resulting cost satisfies $C_{conc\_bu}^{alg}(I, R_1) \leq C_{facil}(R_2^*) + 2SC_{u,v} \leq 2C_{conc\_bu}^*(I, R_1)$, namely, an approximation ratio of 2. ∎

2.2. 6.5-approximation algorithm for RFTFL

We now present a polynomial time algorithm A_{RFTFL} that yields 6.5-approximation for the robust fault-tolerant uncapacitated facility location problem RFTFL. Consider an instance $I = \langle G, l, f, \omega \rangle$ of the problem. The algorithm consists of three stages.
Stage 1: Apply the 1.5-approximation algorithm of [2] to the original UFL problem in order to find an initial subset R_1 of servers. Notice that the cost of this solution satisfies

$$C_{UFL}(R_1) \leq 1.5C_{UFL}^* \leq 1.5C_{RFTFL}^* . \tag{2.2}$$

Each node is now assigned to a server in R_1. Next, we need to assign to each node a *backup server* which will serve it in case its original server fails.
Stage 2: Transform the given instance $I = \langle V, l, \omega, f \rangle$ of the problem into an instance $I' = \langle V, l, \omega', f' \rangle$ as follows. First, change the facility cost f by setting $f'(r) = 0$ for $r \in R_1$. Next, for each server $r \in R_1$, relocate all the demands of the nodes that are served by r, and place them at the server r itself, that is, set

$$\omega'(r) = \begin{cases} \displaystyle\sum_{v \in \varphi(I, r, R_1)} \omega(v), & for \ r \in R_1, \\ 0, & for \ r \notin R_1. \end{cases} \tag{2.3}$$

Stage 3: Invoke the 2-approximation algorithm A_{conc_bu} for the concentrated backup problem on the new instance I' and the set R_1. The approximation algorithm returns a new set R_2. We then return the set $R_1 \cup R_2$ as the final set of open facilities.

Lemma 2.5. *For every instance I and set $R_1 \subseteq V$, $C^*_{conc\_bu}(I', R_1) \leq C^*_{RFTFL}(I) + C_{UFL}(I, R_1)$.*

Proof: Consider some vertex $r \in R_1$ and let $\varphi(I, r, R_1) = \{v^r_1, ..., v^r_{k_r}\}$ be the nodes it serves. Consider the optimal solution $R^*_{RFTFL}(I)$ to the RFTFL problem. Let d^r_i be the distance from r to v^r_i for $1 \leq i \leq k_r$, and also let x^r_i be the distance from v^r_i to its optimal backup server, which is also its distance to $R^*_r \equiv R_1 \cup R^*_{RFTFL}(I) \backslash \{r\}$, i.e., $x^r_i = d(v^r_i, R^*_r)$. By the triangle inequality, $d(r, R^*_r) \leq d^r_i + x^r_i$, for every $1 \leq i \leq k_r$, so

$$
\begin{aligned}
\omega'(r) \cdot d(r, R^*_r) &= \sum_{l=1}^{k_r} \omega(v^r_l) \cdot d(r, R^*_r) \leq \sum_{l=1}^{k_r} \omega(v^r_l)(d^r_l + x^r_l) \\
&= \sum_{l=1}^{k_r} \omega(v^r_l) d(v^r_l, R_1) + \sum_{l=1}^{k_r} \omega(v^r_l) x^r_l \\
&\leq \sum_{v=1}^{n} \omega(v) \cdot d(v, R_1) + \sum_{v=1}^{n} \omega(v) \cdot d(v, R^*_r).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
C_{bu}(I', R_1, R^*_{RFTFL}(I)) &= \max_{r \in R_1} \left\{ \omega'(r) \cdot d(r, R^*_r) \right\} \\
&\leq C_{ship}(I, R_1) + \max_{r \in R_1} \left\{ \sum_{v=1}^{n} \omega(v) \cdot d(v, R^*_r) \right\}.
\end{aligned}
$$

Using (1.4) and (2.1) we now bound the cost of the optimal solution for problem *conc_bu* by

$$
\begin{aligned}
C^*_{conc\_bu}(I', R_1) &\leq C_{conc\_bu}(I', R_1, R^*_{RFTFL}(I)) \\
&= C_{facil}(I', R^*_{RFTFL}(I)) + C_{bu}(I', R_1, R^*_{RFTFL}(I)) \\
&\leq C_{facil}(I', R^*_{RFTFL}(I)) + \max_{r \in R_1} \left\{ \sum_{v=1}^{n} \omega(v) d(v, R^*_r) \right\} + C_{ship}(I, R_1) \\
&\leq C^*_{RFTFL}(I) + C_{ship}(I, R_1) \leq C^*_{RFTFL}(I, R_1) + C_{UFL}(I, R_1). \qquad \blacksquare
\end{aligned}
$$

Lemma 2.6. *For every instance I and sets $R_1, R_2 \subseteq V$,*
$$C_{RFTFL}(I, R_1 \cup R_2) \leq C_{UFL}(I, R_1) + C_{conc\_bu}(I', R_1, R_2).$$

Proof: The cost of opening the facilities in $R_1 \cup R_2$ is clearly at most the cost of opening the facilities in R_1 plus the cost of opening the facilities in R_2. For every facility $r \in R_1 \cup R_2$, in order to bound $C_{ship}(I, R_1 \cup R_2 \backslash \{r\})$, note that one can first move each client v to its closest open facility in R_1, and then move all the clients assigned to r (if $r \in R_1$) to the backup facility of r in R_2. The inequality follows. More formally we have the following. Recall that by (1.4),

$$C_{RFTFL}(I, R_1 \cup R_2) = C_{facil}(I, R_1 \cup R_2) + \max_{r \in R_1 \cup R_2} \left\{ C_{ship}(I, R_1 \cup R_2 \backslash \{r\}) \right\}.$$

Consider first the case that $\max\limits_{r \in R_1 \cup R_2} \{C_{ship}(I, R_1 \cup R_2 \setminus \{r\})\}$ is attained for some $r' \in R_2$. In this case, we get by (1.1) that

$$
\begin{aligned}
C_{RFTFL}(I, R_1 \cup R_2) &= C_{facil}(I, R_1 \cup R_2) + C_{ship}(I, R_1 \cup R_2 \setminus \{r'\}) \\
&\leq C_{facil}(I, R_1 \cup R_2) + C_{ship}(I, R_1) \\
&= C_{UFL}(I, R_1) + C_{facil}(I, R_2) \leq C_{UFL}(I, R_1) + C_{conc\_bu}(I', R_1, R_2).
\end{aligned}
$$

So now assume that $\max\limits_{r \in R_1 \cup R_2} \{C_{ship}(I, R_1 \cup R_2 \setminus \{r\})\}$ is attained for some $r' \in R_1$. Therefore,

$$
\begin{aligned}
C_{RFTFL}(I, R_1 \cup R_2) &= C_{facil}(I, R_1 \cup R_2) + C_{ship}(I, R_1 \cup R_2 \setminus \{r'\}) \\
&= C_{facil}(I, R_1) + C_{facil}(I, R_2) + \sum_{v=1}^{n} SC_{v, R_1 \cup R_2} \\
&\quad + \sum_{v \in \varphi(I, r', R_1 \cup R_2)} \omega(v) \cdot (d(v, R_1 \cup R_2 \setminus \{r'\}) - d(v, r')) \\
&\leq C_{UFL}(I, R_1) + C_{facil}(I, R_2) \\
&\quad + \max_{r \in R_1} \left\{ \sum_{v \in \varphi(I, r, R_1)} \omega(v) \cdot (d(r, R_1 \cup R_2 \setminus \{r\})) \right\} \\
&= C_{UFL}(I, R_1) + C_{facil}(I, R_2) + \max_{r \in R_1} \left\{ w'(r) \cdot (d(r, R_1 \cup R_2 \setminus \{r\})) \right\} \\
&= C_{UFL}(I, R_1) + C_{conc\_bu}(I', R_1, R_2). \quad \blacksquare
\end{aligned}
$$

Lemma 2.7. *Algorithm A_{RFTFL} yields a 6.5-approximation for the RFTFL problem.*

Proof: Consider the set of opened facilities $R_1 \cup R_2$. By Lemma 2.4, R_2 is a 2-approximation of the concentrated backup problem on the instance I', so

$$
C_{conc\_bu}(I', R_1, R_2) \leq 2C^*_{conc\_bu}(I', R_1).
$$

By Lemma 2.5, $C^*_{conc\_bu}(I', R_1) \leq C^*_{RFTFL}(I) + C_{UFL}(I, R_1)$, hence

$$
C_{conc\_bu}(I', R_1, R_2) \leq 2C^*_{RFTFL}(I) + 2C_{UFL}(I, R_1).
$$

Using Lemma 2.6 we get

$$
C_{RFTFL}(I, R_1 \cup R_2) \leq 3C_{UFL}(I, R_1) + 2C^*_{RFTFL}(I),
$$

and by (2.2), $C_{RFTFL}(I, R_1 \cup R_2) \leq 6.5C^*_{RFTFL}(I)$. $\blacksquare$

3. An approximation algorithm for α_RFTFL

3.1. The concentrated α_backup problem

As in the case of a single failure, we first consider a different problem, named *concentrated α_backup (conc_α_bu)*, defined as follows. An instance of the problem consists of a pair $\langle I, R_1 \rangle$ where $I = \langle G, l, f, \omega \rangle$ is defined as before and R_1 is a set of nodes. The nodes of R_1 act as both clients and servers (with open facilities), and all other nodes $v \notin R_1$ have zero demands. We are looking for a set R_2 minimizing

$$
C_{conc\_\alpha\_bu}(I, R_1, R_2) = C_{facil}(R_2) + C_{\alpha\_bu}(I, R_1, R_2), \tag{3.1}
$$

where C_{α_bu} is the *maximum α_backup cost* for a set of nodes R_2, defined as

$$C_{\alpha\_bu}(I, R_1, R_2) = \max_{|F| \leq \alpha} \left\{ \sum_{r \in (F \cap R_1)} \omega(r) \cdot d(r, R_1 \cup R_2 \backslash F) \right\}.$$

We will shortly present a 3α-approximation algorithm for the concentrated α-backup problem.

Towards this, let us first consider a simpler variant of the backup problem, named the *α-bounded backup (α_bb) problem*, which is defined on $\langle I, R_1, M \rangle$ and requires looking for a solution R_2 minimizing

$$C_{\alpha\_bb}(R_1, M, R_2) = C_{facil}(R_2)$$

subject to the constraint $C_{light\_\alpha\_bu}(R_1, R_2) \leq M$ for some integer M, where

$$C_{light\_\alpha\_bu}(R_1, R_2) = \max_{r \in R_1, |F| \leq \alpha} \{\omega(r)d(r, R_1 \cup R_2 \backslash F)\}.$$

We now present a relaxation algorithm that finds a set R_2 satisfying $C_{facil}(R_2) \leq C^*_{\alpha\_bb}(R_1, M)$ but allowing the relaxed constraint $C_{light\_\alpha\_bu}(R_1, R_2) \leq 3M$ instead of $C_{light\_\alpha\_bu}(R_1, R_2) \leq M$.

Algorithm $\boldsymbol{A_{\alpha\_bb}(I, R_1, M)}$

(1) $R^{alg}_{\alpha\_bb} \leftarrow \emptyset$

(2) Let $r_1, ..., r_k$ be the servers in R_1 sorted by nonincreasing order of demands.

(3) $Z \leftarrow \emptyset$ /* The set of servers r_i where the algorithm opens facilities in phase i */

(4) For $i = 1$ to k do:

(5)   • $S_i \leftarrow \{v \mid \omega(r_i)d(v, r_i) \leq 2M\} \backslash \{r_i\}$.

     • $T_i \leftarrow \{v \mid \omega(r_i)d(v, r_i) \leq M\} \backslash \{r_i\}$

     • If $S_i \cap Z = \emptyset$ then:

       – Add to $R^{alg}_{\alpha\_bb}$, the $\alpha - |T_i \cap (R_1 \cup R^{alg}_{\alpha\_bb})|$ nodes in $T_i \backslash (R_1 \cup R^{alg}_{\alpha\_bb})$ with the lowest facility costs.

       – $Z \leftarrow Z \cup \{r_i\}$

(6) Return $R^{alg}_{\alpha\_bb}$.

Let us now prove the properties of Alg. A_{α_bb}. Let $\{\ell_j \mid 1 \leq j \leq J\}$ be the phases in which the algorithm adds new facilities to $R^{alg}_{\alpha\_bb}$. By a proof similar to that of Lemma 2.1, we have the following.

Lemma 3.1. $T_{\ell_i} \cap T_{\ell_j} = \emptyset$ for $1 \leq j < i \leq J$.

Lemma 3.2. $C_{facil}(R^{alg}_{\alpha\_bb}(R_1, M)) \leq C^*_{\alpha\_bb}(R_1, M)$.

Proof: There must be at least α nodes in every T_{ℓ_j} in the optimal solution $R^*_{\alpha\_bb}(R_1, M)$. By Lemma 3.1 the sets T_{ℓ_j} for $1 \leq j \leq J$ are disjoint, so the only nodes that the algorithm adds to $R^{alg}_{\alpha\_bb}$ from the set T_{ℓ_j} are added at phase ℓ_j. The algorithm selects the cheapest nodes in T_{ℓ_j} in order to complete to α nodes. Therefore, $C_{facil}(R^{alg}_{\alpha\_bb}(R_1, M) \cap T_{\ell_j}) \leq$

$C_{facil}(R^*_{\alpha\_bb}(R_1, M) \cap T_{\ell_j})$ for every $1 \leq j \leq J$. Hence

$$
\begin{aligned}
C_{facil}(R^{alg}_{\alpha\_bb}(R_1, M)) &= \sum_{j=1}^{J} C_{facil}(R^{alg}_{\alpha\_bb}(R_1, M) \cap T_{\ell_j}) \leq \sum_{j=1}^{J} C_{facil}(R^*_{\alpha\_bb}(R_1, M) \cap T_{\ell_j}) \\
&\leq C^*_{\alpha\_bb}(R_1, M). \quad \blacksquare
\end{aligned}
$$

Lemma 3.3. $C_{\alpha\_bu}(R_1, R^{alg}_{\alpha\_bb}(R_1, M)) \leq 3M$.

Proof: For each server $v_i \in R_1$, the algorithm ensures that either there are at least α open facilities from the set T_i or v_i is at distance at most $2M$ from another $v_j \in R_1$ that has α open facilities from the set T_j. In the first case the distance is at most M and in the second - at most $3M$. $\quad \blacksquare$

Now we present an approximation algorithm $A_{conc\_\alpha\_bu}$ for the concentrated α_backup problem, using the relaxation algorithm A_{α_bb} for the α_bounded backup problem.

Algorithm $\boldsymbol{A_{conc\_\alpha\_bu}(I, R_1)}$
(1) For every subset $T \subseteq \{SC_{v,u} \mid v, u \in V\}$ such that $|T| \leq \alpha$ do:
   - $M(T) \leftarrow \sum_{m \in T} m$
   - let $R^{alg}_{\alpha\_bb}(R_1, M(T)) \leftarrow A_{\alpha\_bb}(I, R_1, M(T))$.
(2) Return the set $R^{alg}_{\alpha\_bb}(R_1, M(T))$ with the minimum cost
   $C_{conc\_\alpha\_bu}(R_1, R^{alg}_{\alpha\_bb}(R_1, M(T)))$.

Lemma 3.4. $C^{alg}_{conc\_\alpha\_bu}(I, R_1) \leq 3\alpha C^*_{conc\_\alpha\_bu}(I, R_1)$.

Proof: Denote the optimal solution for $conc\_\alpha\_bu$ on $\langle I, R_1 \rangle$ by $R^*_2 = R^*_{conc\_\alpha\_bu}(R_1)$. Then

$$
C^*_{conc\_\alpha\_bu}(I, R_1) = C_{conc\_\alpha\_bu}(I, R_1, R^*_2) = C_{facil}(R^*_2) + C_{\alpha\_bu}(I, R_1, R^*_2).
$$

Let $\{u_1, ..., u_j\} \subseteq R_1$ and $\{v_1, ..., v_j\} \subseteq R_1 \cup R^*_2$ for some $j \leq \alpha$ be the sets of nodes that attain the maximum shipping cost, i.e., satisfy $C_{\alpha\_bu}(I, R_1, R^*_2) = M_0$ for $M_0 = \sum_{i=1}^{j} SC_{u_i,v_i} = \sum_{i=1}^{j} \omega(u_i)d(u_i, v_i)$. Then $C^*_{conc\_\alpha\_bu}(I, R_1) = C_{facil}(R^*_2) + M_0$. Notice that there must be at least α nodes in the set $R^*_2 \cup R_1$ at distance at most M_0 from every server r in R_1. Clearly $C_{facil}(R^*_{\alpha\_bb}(R_1, M_0)) \leq C_{facil}(R^*_2)$. Since the algorithm examines all possible values of $M(T)$, it tests also M_0. For this value, the returned set $R^{alg}_{\alpha\_bb}(R_1, M_0)$ has opening cost at most $C^*_{\alpha\_bb}(R_1, M_0) \leq C_{facil}(R^*_2)$ and backup cost at most $C_{\alpha\_bu}(I, R_1, R^{alg}_{\alpha\_bb}(R_1, M_0)) \leq 3M_0$ by Lemmas 3.2 and 3.3. Since the algorithm takes the minimum cost $C_{conc\_\alpha\_bu}(R_1, R^{alg}_{\alpha\_bb}(I, R_1, M(T)))$ over all possible subsets T, the resulting cost is at most

$$
\begin{aligned}
C^{alg}_{conc\_\alpha\_bu}(I, R_1) &\leq C_{conc\_\alpha\_bu}(I, R_1, R^{alg}_{\alpha\_bb}(R_1, M_0)) \\
&\leq C_{facil}(R^*_2) + \max_{|F| \leq \alpha} \left\{ \sum_{r \in (F \cap R_1)} \omega(r)d(r, R_1 \cup R^{alg}_{\alpha\_bb}(R_1, M_0) \backslash F) \right\} \\
&\leq C_{facil}(R^*_2) + 3\alpha M_0 \leq 3\alpha C^*_{conc\_\alpha\_bu}(I, R_1). \quad \blacksquare
\end{aligned}
$$

3.2. $(1.5 + 7.5\alpha)$-approximation algorithm to the α_RFTFL

We now present a polynomial time algorithm named A_{α_RFTFL}, yielding a $(1.5 + 7.5\alpha)$-approximation for the robust fault-tolerant uncapacitated facility location problem α_RFTFL against a failure of α nodes, for constant $\alpha > 1$. Consider an instance $I = \langle G, l, f, \omega \rangle$ of the problem. The algorithm is similar to Algorithm RFTFL, except for the third stage. Instead of invoking the 2-approximation algorithm A_{conc_bu} for the concentrated backup problem on the new instance I' and the set R_1, invoke the 3α-approximation algorithm $A_{conc\_\alpha\_bu}$ for the concentrated $\alpha\_$backup problem on the new instance I' and the set R_1. Algorithm $A_{conc\_\alpha\_bu}$ returns a new set R_2^{alg}. Algorithm A_{α_RFTFL} now returns the set $R_1 \cup R_2^{alg}$. Proof of the following lemma is deferred to the full paper.

Lemma 3.5. *Algorithm A_{α_RFTFL} yields a $(1.5 + 7.5\alpha)$-approximation for the α_RFTFL problem.*

4. Robust Fault-tolerant uncapacitated facility location on trees

In this section we show that the RFTFL problem is NP-hard even on trees. The claim holds even in the case where only the edge lengths or only the node demands are variable and the other parameters are uniform. An instance of the RFTFL problem is $\langle T, l, f, \omega, P \rangle$, where T is a tree, l, f and ω are defined as before and P is an integer. It is required to decide if the cost of the optimal solution to the RFTFL problem on the instance $\langle T, l, f, \omega \rangle$ is P or less.

The proofs, via reductions from subset sum and from a variant of the partition problem, are deferred to the full paper. The following results are established.

Theorem 4.1. *$RFTFL$ on trees is NP-complete even with*

(1) *unit edge lengths and opening costs (but variable node demands),*
(2) *unit node demands and opening costs (but variable edge lengths).*

References

[1] A. Billionnet and M. Costa, Solving the uncapacited plant location problem on trees, *Discrete Applied Mathematics* **49**, (1994), 51–59.
[2] J. Byrka, An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Proc. 10th Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2007, 29–43.
[3] M. Charikar and S. Guha, Improved combinatorial algorithms for facility location problems, *SIAM J. Comput.* **34**, (2005), 803–824.
[4] F. A. Chudak and D. B. Shmoys, Improved approximation algorithms for the uncapacitated facility location problem, *SIAM J. Comput.* **33**, (2003) 1–25.
[5] K. Dhamdhere, V. Goyal, R. Ravi, and M. Singh, How to Pay, Come What May: Approximation Algorithms for Demand-Robust Covering Problems, In *Proc. 46th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2005, 367–378.
[6] U. Feige, K. Jain, M. Mahdian, and V.S. Mirrokni, Robust combinatorial optimization with exponential scenarios, In *Proc. 12th Int. Conf. on Integer Programming and Combinatorial Optimization (IPCO)*, Ithaca, NY, 2007, 439–453.
[7] S. Guha and S. Khuller, Greedy strikes back: Improved facility location algorithms, *J. of Algorithms* **31**, (1999), 228–248.

[8] Sudipto Guha, Adam Meyerson, and Kamesh Munagala, A constant factor approximation algorithm for the fault-tolerant facility location problem, *J. Algorithms* **48** ,(2003) , 429–440.

[9] K. Jain, M. Mahdian, and A. Saberi, A new greedy approach for facility location problems, In *Proc. 34th ACM Symposium on Theory of Computing (STOC)*, 2002, 731–740.

[10] K. Jain, V. Vazirani, An approximation algorithm for the fault tolerant metric facility location problem, In *Proc. APPROX*,LNCS Vol. 1913, 2000, 177–183.

[11] S. Khuller, R. Pless, and Y. Sussmann, Fault tolerant k-center problems. *Theoret. Comput. Sci.* 242 12 (2000), 237-245

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

[13] D.B. Shmoys, Approximation algorithms for facility location problems, In *Proc. 3rd Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, LNCS, Vol. 1913, (2000), 265–274.

[14] C. Swamy, D. Shmoys, Fault-tolerant facility location, In *Proc. 14th ACM-SIAM SODA*, 2003, 735–736.

[15] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem, *Combinatorica* **2**, (1982), 385–393.