# An Approximation Algorithm for the Fault Tolerant Metric Facility Location Problem ⋆

Kamal Jain and Vijay V. Vazirani

College of Computing,Georgia Institute of Technology, Atlanta, GA 30332–0280
{kjain,vazirani}@cc.gatech.edu

**Abstract.** We consider a fault tolerant version of the metric facility location problem in which every city, $j$, is required to be connected to $r_j$ facilities. We give the first non-trivial approximation algorithm for this problem, having an approximation guarantee of $3 \cdot H_k$, where $k$ is the maximum requirement and $H_k$ is the $k$-th harmonic number. Our algorithm is along the lines of [2] for the generalized Steiner network problem. It runs in phases, and each phase, using a generalization of the primal-dual algorithm of [4] for the metric facility location problem, reduces the maximum residual requirement by 1.

## 1 Introduction

Given costs for opening facilities and costs for connecting cities to facilities, the uncapacitated facility location problem seeks a minimum cost solution that connects each city to a specified number of open facilities. In the fault tolerant version, each city must be connected to a specified number of facilities. Formally, we are given a set of cities and a set of facilities. For each city we are given its connectivity requirement and for each facility we are given its opening cost. For each city-facility pair, we are given the cost of connecting the city to the facility. We assume that the connection costs satisfy the triangle inequality. We want to open facilities and connect each city to as many open facilities as its connectivity requirement such that the total cost of opening facilities and connecting cities is minimized. This problem has potential industrial applications where the facilities and the connections are susceptible to failure.

We give a $3 \cdot H_k$ factor approximation algorithm, where $k$ is the maximum requirement and $H_k = 1 + 1/2 + 1/3 + \cdots + 1/k$. Our algorithm is along the lines of [2] for the generalized Steiner network problem. It runs in phases, and in each phase, reduces the maximum residual requirement by 1. In each phase it considers only those cities which have the maximum residual requirement. The procedure for a phase will give each of these cities one more connection to open facilities. In contrast to the usual facility location problem, a facility may not provide a new connection to every city. We show that a generalization of primal-dual algorithm in [4] works for each phase with a performance factor of 3. In the case of the generalized Steiner network problem, adapting the primal-dual algorithm for the Steiner forest problem to a phase of generalized Steiner network problem took significant work [5]. In contrast, in the case of facility location problem, this adaptation is straight forward, demonstrating a strength of primal-dual schema in facility location problem [4].

⋆ Research supported by NSF Grant CCR-9820896.

K. Jansen and S. Khuller (Eds.): APPROX 2000, LNCS 1913, pp. 177–182, 2000.

## 2 The Fault Tolerant Metric Uncapacitated Facility Location Problem

The *uncapacitated facility location problem* seeks a minimum cost way of connecting cities to open facilities. It can be stated formally as follows: Let $G$ be a bipartite graph with bipartition $(F, C)$, where $F$ is the set of *facilities* and $C$ is the set of *cities*. Let $f_i$ be the cost of opening facility $i$, $r_j$ be the number of facilities city $j$ should be connected to, and $c_{ij}$ be the cost of connecting city $j$ to (opened) facility $i$. The problem is to find a subset $I \subseteq F$ of facilities that should be opened, and a function $\phi$ :C$\rightarrow 2^I$ assigning cities to a set of open facilities in such a that each city $j$ is assigned to a set of cardinality $r_j$ and the total cost of opening facilities and connecting cities to open facilities is minimized. We will consider the *metric* version of this problem, i.e., the $c_{ij}$'s satisfy the triangle inequality.

Consider the following integer program for this problem. In this program, $y_i$ is an indicator variable denoting whether facility $i$ is open, and $x_{ij}$ is an indicator variable denoting whether city $j$ is connected to the facility $i$. The first constraint ensures that each city,$j$, is connected to at least $r_j$ facilities and the second ensures that each of these facilities must be open.

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \qquad (1) \\ \text{subject to} \quad & \forall j \in C: \ \sum_{i \in F} x_{ij} \geq r_j \\ & \forall i \in F, j \in C: \ y_i - x_{ij} \geq 0 \\ & \forall i \in F, j \in C: \ x_{ij} \in \{0, 1\} \\ & \forall i \in F: \ y_i \in \{0, 1\} \end{aligned}$$

An LP-relaxation of this program is:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in F, j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \qquad (2) \\ \text{subject to} \quad & \forall j \in C: \ \sum_{i \in F} x_{ij} \geq r_j \\ & \forall i \in F, j \in C: \ y_i - x_{ij} \geq 0 \\ & \forall i \in F, j \in C: \ x_{ij} \geq 0 \\ & \forall i \in F: \ 1 \geq y_i \geq 0 \end{aligned}$$

The dual program is:

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in C} r_j \alpha_j - \sum_{i \in F} z_i \qquad (3) \\ \text{subject to} \quad & \forall i \in F, j \in C: \ \alpha_j - \beta_{ij} \leq c_{ij} \end{aligned}$$

$$\forall i \in F: \ \sum_{j \in C} \beta_{ij} \leq f_i + z_i$$
$$\forall j \in C: \ \alpha_j \geq 0$$
$$\forall i \in F, j \in C: \ \beta_{ij} \geq 0$$

We will adopt the following notation: $n_c = |C|$ and $n_f = |F|$. The total number of vertices $n_c + n_f = n$ and the total number of edges $n_c \times n_f = m$. The maximum of $r_j$'s is $k$. Optimum solution of the integer program is $OPT$ and of linear program is $OPT_f$.

### 2.1 The High Level Algorithm

Our algorithm opens facilities and assign them to cities in $k$ phases numbered from $k$ down to 1. Each phase decreases the maximum *residual requirement*, which is the maximum number of further facilities needed by a city, by 1. Hence at the beginning of the $p$-th phase maximum residual requirement is $p$ and at the end of it the maximum residual requirement is $p - 1$.

The algorithm starts with an empty solution ($I_k$,$\mathrm{C}_k$). The $p$-th phase of the algorithm takes the solution ($I_p$,$\mathrm{C}_p$) and extend it to ($I_{p-1}$,$\mathrm{C}_{p-1}$) such that the maximum residual requirement is decreased by one, thereby maintaining the loop invariant that the maximum residual requirement with respect to solution ($I_p$,$\mathrm{C}_p$) is $p$. Hence, ($I_0$,$\mathrm{C}_0$) is a feasible solution. In the next section, we will show the following theorem.

**Theorem 1.** *Cost of* $(I_{p-1}, C_{p-1})$ *minus the cost of* $(I_p, C_p)$ *is at most* $3 \cdot OPT/p$.

**Corollary 1.** *Cost of* $(I_0, C_0)$ *is at most* $3 \cdot H_k OPT$.

## 3 The $p$-th Phase

This phase extends the solution ($I_p$,$\mathrm{C}_p$) to ($I_{p-1}$,$\mathrm{C}_{p-1}$) so that the each city, $j$, with residual requirement of $p$ with respect to the solution ($I_p$,$\mathrm{C}_p$) gets connected to at least one more open facility. This can happen in two ways, first a new facility is opened in ($I_{p-1}$,$\mathrm{C}_{p-1}$) and $j$ is connected to that. Second, $j$ is connected to already open facility in ($I_p$,$\mathrm{C}_p$) to which it was not connected. In the first case, both the facility and the connection must be paid in this phase itself whereas in the second case only the connection needs to be paid.

So in this phase, facilities are of two types, *free* and *priced*. The set of free facilities is $I_p$. A priced facility if opened can be used by any city whereas a free facility can be used by only those cities which are not already using it. So denote the set of cities with residual requirement of $p$ by $C_p$. The problem of this phase can be written as the following integer program.

$$\text{minimize} \quad \sum_{i \in F, j \in C_p} c_{ij} x_{ij} + \sum_{i \in F - I_p} f_i y_i \tag{4}$$

$$\text{subject to} \quad \forall j \in C_p: \quad \sum_{i \in F - \mathrm{C}_p(j)} x_{ij} \geq 1$$

$$\forall i \in F - I_p, j \in C_p : \; y_i - x_{ij} \geq 0$$
$$\forall i \in F, j \in C : \; x_{ij} \in \{0, 1\}$$
$$\forall i \in F - I_p : \; y_i \in \{0, 1\}$$

An LP-relaxation of this program is:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in F, j \in C_p} c_{ij} x_{ij} + \sum_{i \in F - I_p} f_i y_i \\ \text{subject to} \quad & \forall j \in C_p : \sum_{i \in F - \mathrm{C}_p(j)} x_{ij} \geq 1 \\ & \forall i \in F - I_p, j \in C_p : \; y_i - x_{ij} \geq 0 \\ & \forall i \in F, j \in C : \; x_{ij} \geq 0 \\ & \forall i \in F : \; y_i \geq 0 \end{aligned} \tag{5}$$

The dual program is:

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in C_p} \alpha_j \\ \text{subject to} \quad & \forall i \in F - I_p, j \in C_p : \; \alpha_j - \beta_{ij} \leq c_{ij} \\ & \forall i \in I_p, j \in C_p : \; \alpha_j \leq c_{ij} \\ & \forall i \in F - I_p : \; \sum_{j \in C} \beta_{ij} \leq f_i \\ & \forall j \in C : \; \alpha_j \geq 0 \\ & \forall i \in F, j \in C : \; \beta_{ij} \geq 0 \end{aligned} \tag{6}$$

**Theorem 2.** *Optimum solution of LP 5 is at most $OPT_f/p$.*

*Proof.* Let optimum solution of LP 5 is $OPT_p$. By strong duality theorem of linear programming theory, there is a dual feasible solution for LP 6 of value $OPT_p$. Let $(\alpha, \beta)$ be one such solution satisfying LP 6. Following procedure extends this solution to a feasible dual solution to LP 3 of value $p \cdot OPT_p$, hence proves the theorem.

1. $\forall j \in C - C_p$, $\alpha_j \leftarrow 0$.
2. $\forall j \in C - C_p, i \in F$, $\beta_{ij} \leftarrow 0$.
3. $\forall j \in C_p, i \in \mathrm{C}_p(j)$, $\beta_{ij} \leftarrow \alpha_j$.
4. $\forall i \in I_p$, $z_i = \sum_{j \in C_p} \beta_{ij}$.

Denote this extended solution by $(\alpha, \beta, z)$. One can easily check that $(\alpha, \beta)$ is a feasible solution to LP 3. Its value is $\sum_{j \in C} r_j \alpha_j - \sum_{i \in F} z_i = \sum_{j \in C_p} r_j \alpha_j - \sum_{i \in F} \sum_{j \in C_p} \beta_{ij} = \sum_{j \in C_p} r_j \alpha_j - \sum_{j \in C_p} \sum_{i \in \mathrm{C}_p(j)} \beta_{ij} = \sum_{j \in C_p} r_j \alpha_j - \sum_{j \in C_p} \sum_{i \in \mathrm{C}_p(j)} \alpha_j = \sum_{j \in C_p} r_j \alpha_j - \sum_{j \in C_p} |\mathrm{C}_p(j)| \alpha_j = \sum_{j \in C_p} (r_j - |\mathrm{C}_p(j)|) \alpha_j = \sum_{j \in C_p} p \alpha_j = p \sum_{j \in C_p} \alpha_j = p \cdot OPT_p$.

In the next section we will adapt the primal-dual algorithm of [4] to show the following theorem.

**Theorem 3.** *Cost of* $(I_{p-1}, C_{p-1})$ *minus the cost of* $(I_p, C_p)$ *is at most* $3 \cdot OPT_p$.

**Corollary 2.** *Cost of* $(I_{p-1}, C_{p-1})$ *minus the cost of* $(I_p, C_p)$ *is at most* $3 \cdot OPT/p$.

## 4 Primal-Dual Algorithm for the *p*-th Phase

Our algorithm is essentially the same as the primal-dual algorithm in [4] except for the following differences.

1. Duals of only those cities which have residual requirement of $p$ will be raised.
2. Facilities in $I_p$ are free, others carry there original costs.
3. Connection already used in ($I_p$,$C_p$) are of infinite costs. Cost of other connections remain the same.

For completeness, we are reproducing the primal-dual algorithm of [4] with the above mentioned changes. The algorithm runs in two phases. The first phase runs in a primal-dual fashion to find a tentative solution and the second modifies it so that the primal becomes at most the thrice of the dual. The algorithm has a notion of *time*. It begins at time zero with a zero primal and a zero dual solution. At time zero, all cities in $C_p$ are *unconnected*, all facilities except free facilities are *closed*. Free facilities are *open*.

As the time passes the algorithm *raises* the dual variable $\alpha_j$ for each unconnected city uniformly at rate one. Now the following two kinds of events can happen:

1. Dual constraint corresponding to a connection, $ij$, goes *tight* i.e., $\alpha_j - \beta_{ij} = c_{ij}$. Such a connection is declared *tight*. The algorithm performs one of the following step according to the state of facility $i$.
   (a) If facility $i$ is (tentatively) open then city $j$ is declared *(tentatively) connected* to facility $i$. Dual variable for this city will not be raised any further.
   (b) If facility $i$ is closed then $\beta_{ij}$ will begin responding to the raise of $\alpha_j$ i.e., whenever $\alpha_j$ will be raised $\beta_{ij}$ will also be raised by the same amount to maintain the feasibility of $\alpha_j - \beta_{ij} \leq c_{ij}$.
2. Dual constraint corresponding to a facility $i$ goes tight i.e., $\sum_{j \in C} \beta_{ij} = f_i$. This facility is declared *tentatively opened*. Every unconnected city have a tight edge to this facility is declared tentatively connected to this facility.

The first phase of the algorithm ends when there is no more unconnected city. A city $j$ is said to be *overpaying* if there are at least two tentatively open facilities $i_1$ and $i_2$ such that both $\beta_{i_1 j}$ and $\beta_{i_2 j}$ are positive. The second phase picks a maximal set of tentatively open facilities such that no city is overpaying. All facilities in this maximal set are opened and all other tentatively opened facilities are closed. Any city having a tight edge to an open facility is declared connected to it. The next lemma follows by this construction.

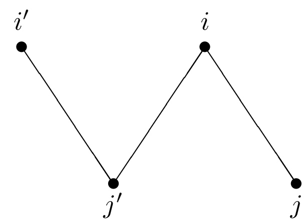
**Lemma 1.**

$$\sum_{i\in F, j\in C_p \text{ and } j \text{ is connected}} c_{ij}x_{ij} + \sum_{i\in F-I_p} f_i = \sum_{j\in C_p \text{ and } j \text{ is connected}} \alpha_j.$$

The performance gap of factor 3 comes from the tentatively connected cities. Consider a tentatively connected city $j$. Suppose it was tentatively connected to facility $i$, which got closed. Since we picked a maximal set of tentatively opened facility such that no city is overpaying, there must be a city $j'$ which was paying to this facility $i$ and an opened facility say $i'$. City $j$ is connected to the facility $i'$. The next lemma establishes the performance guarantee of 3.

**Lemma 2.** *For any tentatively connected city, connection cost is at most the three times the dual raised by it.*

*Proof.*




Let $t_i$ and $t_{i'}$ respectively be the times at which the facilities $i$ and $i'$ are declared tight. The proof follows from the following three observations and the triangle inequality. Note that the facility $i' \notin I_p$, hence the triangle inequality is maintained for this situation.

1. Since $j$ is declared tentatively connected to $i$, $\alpha_j \geq t_i$ and $\alpha_j \geq c_{ij}$.
2. Since connection $ij'$ and $i'j'$ both are tight, $\alpha_{j'} \geq c_{ij'}$ and $\alpha_{j'} \geq c_{i'j'}$.
3. Since, during the first phase, $\alpha_{j'}$ is stopped being raised as soon as one of the facilities $j'$ has a tight edge to is tentatively opened, $\alpha_{j'} \leq \min(t_i, t_{i'})$.

Using first and the last we have $\alpha_{j'} \leq \alpha_j$, which together with the second gives, $c_{ij} + c_{ij'} + c_{i'j'} \leq 3 \cdot \alpha_j$. Hence by triangle inequality we get $c_{i'j} \leq 3 \cdot \alpha_j$.

## References


1. A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. on Computing*, 24:440-456, 1995.
2. M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. Williamson. Improved approximation algorithms for network design problems. *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms*, 223-232, 1994.
3. M. X. Goemans, D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal of Computing*, 24:296-317, 1995.
4. K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and $k$-median problems using the primal-dual schema and Lagrangian relaxation. *To appear in JACM.*
5. D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:435-454, December 1995.