International Journal of Mathematics and
Computer Science, **16**(2021), no. 4, 1087–1101

An Integer Programming Formulation of Capacitated Facility Location Problem

Eiman Jadaan Alenezy

Department of Mathematics
College of Basic Education
Public Authority for Applied Education and Training
Kuwait, Kuwait

email: ej.alenezy@paaet.edu.kw

(Received December 5, 2020, Accepted January 10, 2021)

Abstract

We study the three approximation algorithms developed for solving Uncapacitated Facility Location Problem (UCFLP). These are: the algorithms based on linear programming rounding, the primal-dual algorithm of Jain and Vazirani [7], and the algorithms based on dual-fitting. They give the reader an insight into the main approach to solve the Facility Location Problem (FLP).

1 Introduction

The facility location problem (FLP) seeks to locate a number of facilities to serve a number of customers. Thus, there is a set of potential facility locations F; opening a facility at location $i \in F$ has an associated nonnegative fixed cost f_i and has either a limited or unlimited capacity S_i of available supply. There is a set of customers (clients) or demand points D that require service; customer $j \in D$ has a demand d_j that must be satisfied by the open facilities. If a facility at location $i \in F$ is used to satisfy part of the demand of client $j \in D$, then there is a service or transportation cost incurred which is often proportional to the distance from i to j, denoted by c_{ij}.

Key words and phrases: Primal-dual algorithm, Dual-fitting, Linear programming rounding, Facility Location Problem, Uncapacitated Facility Location Problem.
AMS (MOS) Subject Classifications: 90B08, 90B04.
ISSN 1814-0432, 2021, http://ijmcs.future-in-tech.net

Then, our formulation of the Capacitated Facility Location Problem (CFLP) as a mixed-integer programming problem, which we will refer to as (IP), is as follows:

$$\text{Minimize} \sum_{j\in D}\sum_{i\in F} d_j c_{ij} x_{ij} + \sum_{i\in F} f_i y_i$$

subject to

$$\sum_{i\in F} x_{ij} = 1, \qquad \forall j \in D \tag{1.1}$$

$$\sum_{j\in D} d_j x_{ij} \leq S_i y_i, \qquad \forall i \in F \tag{1.2}$$

$$x_{ij} \leq y_i, \qquad \forall i \in F\ ,\ j \in D \tag{1.3}$$

$$y_i \in \{0,1\}, \qquad \forall i \in F \tag{1.4}$$

$$p \leq \sum_{i\in F} y_i \leq p+2 \tag{1.5}$$

$$x_{ij} \geq 0, \qquad \forall i \in F\ ,\ j \in D \tag{1.6}$$

There are two variants of the capacitated problem, depending on whether each location's demand must be assigned to only one facility, or the demand may be fractionally split among more than one (completely) open facility. For a review of this problem, see [9].

The CFLP and variations of it were extensively treated in the literature, see for example [1, 2, 3, 4, 11].

In this article, we describe three ways of using linear programming in designing approximation algorithms to solve UCFLP. We present the main ideas of the approximation algorithms developed for solving the (UFLP). First, we focus on the algorithms based on linear programming rounding. Secondly, we present the primal-dual algorithm of Jain and Vazirani [7]. Finally, we briefly discuss the algorithms based on dual-fitting.

2 The Uncapacitated Facility Location Problem (UCFLP)

The uncapacitated facility location problem is one of the most widely studied problems in Operations Research. The problem is considered from different perspectives such as probabilistic analysis of average case performance, empirical investigation of heuristics, polyhedral characterization (see [14] for an extensive survey).

There are two measures for the UCFLP: the non-metric case; that is, where the distances need not to be symmetric and not satisfy the triangle inequality which is known to be as hard as the set-covering problem in terms of problem complexity. The other, the metric case; that is, where most of the work on approximating the UCFLP has concentrated. The metric version is also NP-hard in terms of computational complexity (see [6]).

3 An Integer Programming Formulation

In this section we will focus on the approximation algorithms for the metric UCFLP. In order to illustrate the various solution methods, we assume that each $j \in D$ has a demand of one unit. But we can easily extend this to arbitrary demand. In addition, we assume that $f_i > 0$ for every $i \in F$, because if there is a facility i with no opening cost, then we can open it and assign to it all the demand points that are near to this facility. An integer programming formulation of this problem together with its linear programming relaxation help in designing approximation algorithms to solve this problem.

For the formulation of an integer programming, let $(0, 1)$ variables of y_i ($i \in F$) indicate if facility i is open, and the $(0, 1)$ variables for x_{ij} ($j \in D,\ i \in F$) indicate if demand point j is served by facility i or not. Therefore, we define the following cost functions:
Let $c(x) = \sum_{i \in F,\, j \in D} c_{ij} x_{ij}$ and $f(y) = \sum_{i \in F} f_i y_i$, Thus $c(x)$ represents the service cost and $f(y)$ represents the cost of opening facilities. Using these functions, the UCFLP integer formulation is:

(P_{int}) Minimize $c(x) + f(y)$

subject to:

$$\sum_{i \in F} x_{ij} = 1 \qquad \text{for each } j \in D \tag{3.7}$$

,

$$x_{ij} \leq y_i \qquad \text{for each } i \in F \text{ and } j \in D, \tag{3.8}$$

$$x_{ij}\ ,\ y_i \in \{0,1\} \qquad \text{for each } i \in F \text{ and } j \in D.$$

Constraints (3.7) ensure that each demand point $j \in D$ is assigned to a facility, while constraint (3.8) ensures that demand points are serviced only by an open facility. We denote the optimum value of (P_{int}) by C_{OPT}.

Now, consider the following LP-relaxation of (P_{int}):

(P) Minimize $c(x) + f(y)$

subject to:

$$(3.7), (3.8)$$

$$\text{and } x_{ij} \qquad \text{for each } i \in F \text{ and } j \in D$$

Note that, $x_{ij} \leq 1$ and $y_i \leq 1$ hold automatically for any optimal solution (x, y) of (P), and inequalities (3.8) implies that $y_i \geq 0$ for any solution of (P). Let C_{LP} denote the optimum value of (P). Now, let (x, y) be a solution of (P) for every $x_{ij} > .0$ We say that j is fractionally served by i in (x, y).

The dual problem of (P) can be formulated as follows:

(D) maximize $\sum_{j \in D} v_j$

subject to:

$$v_j - t_{ij} \leq c_{ij}\ , \qquad \text{for each } i \in F \text{ and } j \in D \tag{3.9}$$

$$\sum_{j \in D} t_{ij} \leq f_i\ , \qquad \text{for each } i \in F \tag{3.10}$$

$$\text{and, } v_j\ ,\ t_{ij} \geq 0\ , \qquad \text{for each } i \in F \text{ and } j \in D$$

We interpret the dual solution based on the complementary slackness condition, as in [7].

Let (x^*, y^*) and (v^*, t^*) be an optimal primal solution and an optimal dual solution, respectively. Now, the primal dual complementary slackness conditions are:

(S1) $\forall i \in F\ ,\ j \in D : x^*_{ij} > 0$ implies that $v^*_j = c_{ij} + t^*_{ij}$.

(S2) $\forall i \in F\ : y^*_i > 0$ implies that $\sum_{j \in D} t^*_{ij} = f_i$.

(S3) $\forall j \in D\ :\ v^*_j > 0$ implies that $\sum_{i \in F} x^*_{ij} = 1$.

(S4) $\forall i \in F\ ,\ j \in D\ :\ t^*_{ij} > 0$ implies that $y^*_i = x^*_{ij}$.

If i is a facility with $y^*_i > 0$, then $\sum_{j \in D} t^*_{ij} = f_i$ by (S2). Thus, t^*_{ij} can be considered as the amount j is willing to pay for the opening of facility i. To open facility i, the demand point j has to pay all the opening cost f_i. The following fact needs to be considered in this interpretation. Let $j \in D$ be a demand point that was not served by facility i in (x^*, y^*); i.e. $x^*_{ij} = 0$. (S4) implies that since $x^*_{ij} \neq y^*_i$, then $t^*_{ij} = 0$, so a demand point j is not willing to pay for a facility that it is not assigned to. Now, if a demand point $j \in D$ was fractionally served by facility i in (x^*, y^*), then (S1) implies that $v^*_j = c_{ij} + t^*_{ij}$. Therefore, if we consider c_{ij} as the cost j has to pay for being connected to facility i, then v^*_j can be considered as the total price the demand point j is willing to pay.

4 Approximation Algorithms Based On Linear Programming

In this section we describe three ways of using linear programming in designing approximation algorithms to solve the UCFLP: linear programming rounding, the primal-dual algorithm of Jain and Vazirani [7], and the dual-fitting algorithm.

All the approximation algorithms based on linear programming contain the following general steps:

1. Formulate the problem as integer linear program (ILP), denote the optimal value by C_{OPT}.

2. Relax the (ILP) to a linear program (LP), denote the optimal value by C_{LP}.

3. Use the linear program (LP) in order to obtain in polynomial time a feasible solution to the (ILP) bounded by ρC_{LP}.

4.1 Linear Programming Rounding Algorithm

All the algorithms that are based on linear programming, find an optimal solution to (P) first, and then round it to an integer solution (x, y) that satisfies

$$c(x) + f(y) \leq \rho C_{LP} \leq \rho C_{OPT},$$

where ρ is the approximation guarantee factor of the algorithm.

We will use a simplified version of Shmoys, Tardos and Aardal's algorithm [15], which is due to Chudak [5], to present the main idea of these algorithms. First, we need to introduce some notations and definitions:
Let (x^*, y^*) and (v^*, t^*) be optimal solutions to (P) and (D) respectively. For every $j \in D$, we call the neighborhood of j the set of facilities that fractionally serve j; i.e., $N_j = \{i \in F \mid x^*_{ij} > 0\}$. The following lemma consider some properties of this neighborhood.

Lemma 4.1. *1. For every $j \in D$ and $i \in N_j$, , $c_{ij} \leq v^*_j$.*

2. For every $j \in D$, $\sum_{i \in N_j} x^*_{ij} = 1$ and $\sum_{i \in N_j} y^*_i \geq 1$.

3. Let f_{i_0} be the facility with the minimum opening cost in the neighborhood N_j of a demand point j. Then $f_{i_0} \leq \sum_{i \in N_j} f_i y^*_i$.

4. Let j, j' be two demand points with $N_j \cap N_{j'} \neq \phi$ and $v^*_j \leq v^*_{j'}$. Then $c_{ij} \leq 3v^*_{j'}$ for every $i \in N$.

Proof.

1. Since $x^*_{ij} > 0$, $\forall i \in N_j$, $v^*_j = c_{ij} + t^*_{ij}$. However, $t^*_{ij} \geq 0$. Therefore, $v^*_j = c_{ij} + t^*_{ij} \geq c_{ij}$, and $c_{ij} \leq v^*_j$, for every $j \in D$ and $i \in N_j$.

2. In the primal (P) we have $\sum_{i \in F} x_{ij} \geq 1$, for all $j \in D$, but $x_{ij} = 0$, $i \notin N_j$. Therefore, $\sum_{i \in N_j} x_{ij} \geq 1$, $\forall j \in D$, but $x_{ij} > 0$, $i \in N_j$. Then $v_j > 0$

by (S1)and so $\sum_{i\in N_j} x_{ij} = 1$ by (S3), and $y_{ij} \geq x_{ij}$, $i \in N_j$, $\forall j \in D$. Thus $\sum_{i\in N_j} y_i \geq \sum_{i\in N_j} x_{ij} = 1$, $\forall j \in D$.

3. From (2), we have $f_{i_0} \leq f_i$ and $\sum_{i\in N_j} y_i^* \geq 1$, for every $i \in N_j$. This implies that $\sum_{i\in N_j} f_i y_i^*$.

4. Let $i' \in N_j \cap N_j$. By the triangle equality, we have:

$$c_{ij'} \leq c_{i'j} + c_{i'j'} + c_{ij},$$

but $c_{ij} \leq v_j^*$, $c_{i'j} \leq v_j^*$, $c_{i'j'} \leq v_{j'}^*$, and $v_j^* \leq v_{j'}^*$. Consequently, $c_{ij'} \leq 3v_{j'}^*$.

We can now describe the rounding algorithm as follows:
It starts by grouping (partitioning) the demand points into clusters (blocks). Each cluster has a unique cluster center $j \in D$. We will denote the set of centers by $C \subseteq D$, and their construction is by passing through the demand points in increasing order of v_j^*. So

$$v_1^* \leq ... \leq v_n^*$$

The node j becomes a new cluster center if $N_j \cap N_{j_0} = \phi$ for all cluster centers $j_0, j_0 \leq j$ constructed so far. Otherwise, if $N_j \cap N_{j_0} \neq \phi$ for some center j_0, then we add j to the cluster center at j_0. Having the clustering constructed, we open the cheapest facility $i \in N_{j_0}$ for each center $j_0 \in C$ and assign demand points in this cluster to $i \in F$. Let C_j be a cluster centered at j.

Remark 4.2. *The clustering procedure assigns every demand point j to a cluster centered in some points j_0 with $v_{j_0}^* \leq v_j^*$. The facilities, which are in a cluster centered at j_0 are in fact facilities in N_{j_0}. Therefore, $\sum_{i\in C_j} x_{ij}^* = 1$. Furthermore, the neighborhoods of any two centers are disjoint.*

As a result of this algorithm, we obtain the following integer solution:

$$y_i = \begin{cases} 1 & , \quad i \text{ is opened by the algorithm} \\ 0 & , \quad \text{otherwise} \end{cases}$$

And

$$x_{ij} = \begin{cases} 1 & , \quad j \text{ is assigned to } i \text{ by the algorithm} \\ 0 & , \quad \text{otherwise} \end{cases}$$

We know that the neighborhood of two cluster centers in C are disjoint. By Remark 4.2, the cost of opening facilities incurred by (x, y) can be bounded by

$$f(y) \leq \sum_{j_0 \in C} f_{i_0(j_0)} \leq \sum_{j_0 \in C} \sum_{i \in N_{j_0}} f_i y_i^* \leq C_{LP},$$

where $i_0(j_0)$ is the facility with the minimum cost in the cluster centered at j_0, in cluster C_{j_0}.

By Lemma 4.1 (4) and the way of choosing cluster centers, for every demand point j there is an open facility within distance $3v_j^*$. As a result, the service cost incurred by (x, y) can be bounded by

$$c(x) \leq 3 \sum_{j \in D} v_j^* = 3C_{LP}.$$

Notice that the above solution (x, y) satisfies $c(x) + f(y) \leq 4C_{LP} \leq 4C_{OPT}$.

4.2 The Primal-Dual Algorithm of Jain and Vazirani

Jain and Vazirani considered a very simple combinatorial algorithm [7]. They based their algorithm on the primal-dual technique. In polynomial time, they constructed an integer solution (x, y) of (P) and a dual feasible solution (v, t) of (D) which satisfy the dual complementary slackness conditions and satisfy a relaxation of the primal ones.

The relaxed complementary slackness conditions are:

$$(S1') \ \forall i \in F \text{ and } j \in D \text{ with } x_{ij} > 0, 1/3c_{ij} \leq v_j - t_{ij} \leq c_{ij}.$$

Remark 4.3. *Once we obtain (x, y) and (v, t), the cost of the solution (x, y) can be bounded by*

$$\begin{aligned} c(x) + 3f(y) &= \sum_{i \in f,\, j \in D} c_{ij} x_{ij} + 3 \sum_{i \in F} f_i y_i \\ &\leq \sum_{i \in F,\, j \in D} c_{ij} x_{ij} + 3 \sum_{i \in F,\, j \in D} t_{ij} x_{ij} \\ &= \sum_{j \in D} \sum_{i \in F} (c_{ij} + 3t_{ij}) x_{ij} \\ &\leq 3 \sum_{j \in D} v_j. \end{aligned}$$

Proof.
Consider $\sum_{j\in D} c_{ij}x_j + 3\sum_{i\in F} f_i y_i$. If $y_i > 0$, then $f_i = \sum_{j\in D} t_{ij}$ by (S2). Thus,

$$\sum_{j\in D} c_{ij}x_j + 3\sum_{i\in F} f_i y_i = \sum_{j\in D} c_{ij}x_j + 3\sum_{i\in F}\sum_{j\in D} t_{ij}y_i.$$

If $t_{ij} > 0$, then $y_i = x_i$ by (S4). Therefore,

$$\sum_{j\in D} c_{ij}x_j + 3\sum_{i\in F} f_i y_i = \sum_{j\in D} c_{ij}x_j + 3\sum_{i\in F}\sum_{j\in D} t_{ij}x_i.$$

If $x_{ij} > 0$, then $t_{ij} \leq v_j - 1/3c_{ij}$ by (S1'). This implies that

$$\sum_{i\in F}\sum_{j\in D}(c_{ij} + 3v_j - c_{ij})x_{ij} \leq 3\sum_{i\in F}\sum_{j\in D} v_j x_{ij}.$$

If $v_j > 0$, then $\sum_{i\in F} x_{ij} = 1$ by (S3) and so

$$3\sum_{j\in D} v_j = 3C_{LP}.$$

Moreover, since

$$c(x) + 3f(y) \leq 3\sum_{j\in D} v_j \leq 3C_{LP} \leq 3C_{OPT},$$

the performance guarantee of Jain and Vazirani algorithm is 3.

Before we present the construction of (x, y) and (v, t) with the above properties, motivated by the interpretation of the dual variables of Section 3, we introduce some notations with respect to an arbitrary feasible solution (v, t) of (D).

- A demand point j is willing to contribute towards opening a facility i if $t_{ij} > 0$. Recall that t_{ij} represents the contribution it is willing to pay to open facility i.

- A facility $i \in F$ is fully paid when

$$\sum_{j\in D} t_{ij} = f_i.$$

- A demand point $j \in D$ has reached a facility i, when $v_j \geq c_{ij}$. If i is fully paid, then we say that j gets connected. We will call i a connecting facility for j.

Phase 1 First, the algorithm starts at time 0, with $v \equiv t \equiv 0$; initially. Each demand point is defined to be unconnected. We increase all v_j for each unconnected j uniformly at unit rate; i.e., v_j will grow by 1 in unit time. When some $j \in D$ reaches a not fully paid facility $i \in F$, we start increasing t_{ij} uniformly until f_i is fully paid. When j gets connected, we stop increasing v_j and t_{ij}, for every $i \in F$.

More simply, the algorithm proceeds as follows:

Until all $j \in D$ are connected DO
  Increase v_j for all $j \in D$ not yet connected
  Increase t_{ij} for all $i \in F$, $j \in D$ satisfying
  - j has reached i
  - j is not yet connected
  - j is not yet fully paid

Notice that, the algorithm has the property that at time T, the dual variable v_j which is still increasing equals T.

Next, we construct a feasible integer solution (x, y) to (P_{int}). A normal integer solution is the one obtained by opening all the fully paid facilities and then assigning each demand point to the nearest open facility. However, this solution may not satisfy the dual complementary slackness condition (S4), because for a $j \in D$ there will be only one facility i with $x_{ij} > 0$, but there is a possibility of having more facilities $i' \in F$ with $y_(i') > 0$ and $t_{i'j} > 0$. In order to satisfy condition (S4), we need to open facilities such that each demand point is willing to pay for opening at most one facility. The way to do this is as follows:

- For every fully paid facility i, let T_i be the time when i became fully paid.
- We associate a set of clients to every fully paid facility i, which is the set of the demand points willing to pay for opening i; i.e.,

$$D_i = \{j \in D \mid t_{ij} > 0\}$$

Lemma 4.4. *(i) For every demand point j that reached a facility i, we have $c_{ij} \leq v_j$.*

(ii) If $j \in D$, then $v_j \leq T_i$.

(iii) If i is a connecting facility for some j, then $T_i \leq v_j$.

(iv) If i and i' are two fully paid facilities with $D_i \cap D(i') \neq 0$ and i' is a connecting facility for some demand points j, then $1/3c_{ij} \leq v_i - t_{ij} \leq c_{ij}$.

Proof.
(i) and (ii) follow from the fact that if j has reached i, then $v_j = c_{ij} + t_{ij}$ and j get connected at most at time T_i.

(iii) If $j \in D_i$ and i are connected facilities for j, then $v_j = T_i$. Otherwise, j would have reached i after T_i which implies that $T_i \leq v_j$.

(iv) If j has reached i, then $v_j - t_{ij} = c_{ij}$. Therefore, both inequalities are satisfied.
If j did not reach i, then $t_{ij} = 0$ and $v_j \leq c_{ij}$ and as a result the second inequality follows immediately.
Now, let j' be a point in $D_i \cap D_{i'}$. By the triangle inequality we conclude that

$$c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij1}.$$

Since the demand point $j' \in D_i \cap D_{i'}$, and j' reached i and i', (i) implies that $c_{i'j'} \leq v_{j'}$ and $c_{ij'} \leq v_{j'}$. In addition, by (ii) $v_{j'} \leq min\{T_{i'}, T_i\}$. Now since i' is a connecting facility for j, $c_{i'j} \leq v_j$ and $T_{i'} \leq v_j$, $v_{j'} \leq v_j$. So we can now bound c_{ij} by

$$c_{ij} \leq 2v_{j'} + v_j \leq 3v_j.$$

When all the demand points j get connected to i, where i is declared the connecting facility for each of these demand points j, which is declared by the algorithm temporarily open, the first phase terminates.

Phase 2 The purpose of phase 2 is to deal with the following situation: demand points $j \in D$ are connected to more than one temporarily open facility $i \in F$. In this case, the algorithm wants each demand point j to get connected to only one of the temporarily open facilities.

We give the following example to show how the algorithm works:
Let $D_{i_{m'}}$ denote the set of demand points that are connected to a temporarily open facility $i_{m'}$, where $m = |F|$ and $m' \subseteq m$. Assume the

following sets from phase 1:
$D_{i_1} = \{1, 3, 6, 7, 8\}$
$D_{i_2} = \{1, 3, 6\}$
$D_{i_3} = \{2, 6, 7, 9\}$
$D_{i_4} = \{1, 2, 3\}$
$D_{i_4} = \{1, 10, 20\}$
Since $D_{i_2} \subseteq D_{i_i}$, we can close facility i_2 and keep i_1 open. Remove the demand points 6 and 7 in D_{i_3}, because they are already connected to an open facility i_1. Also remove 1 and 3 in D_{i_4} because they are already connected to facility i_1, and remove 2 in D_{i_4} because it is already connected to i_3. Now close facility i_4 and keep facility i_3 open. Finally, remove 1 of D_{i_5} because it is already connected to i_1. Keep doing this until $D_{i_i} \cap D_{i_2} \cap D_{i_3} \cap D_{i_4} \cap D_{i_5} = \phi$.

Remark 4.5. *The set of clients associated to two open facilities are disjoint and each demand point is assigned to a cluster. Each demand point pays for the opening cost of at most one facility. Therefore, there is an open facility for each cluster.*

The primal solution obtained this way is

$$y_i = \begin{cases} 1 & , \quad i \text{ is open} \\ 0 & , \quad \text{otherwise} \end{cases}$$

And

$$x_{ij} = \begin{cases} 1 & , \quad j \text{ was assigned to } i \text{ by the algorithm} \\ 0 & , \quad \text{otherwise} \end{cases}$$

Remark 4.6. *Each demand point j is associated to an open facility i that became fully paid at some time $T_i \leq v_j$. Therefore, if we stop increasing v_j at time T_i, then from the dual obtained this way we can construct the same primal solution (x, y). To see this, by Lemma 4.4, it is easy to verify that (x, y) and (v, t) are feasible to (P)and (D) respectively and that they satisfy (S'), (S2) and (S4).*

Jain and Vazirani [7] proved that the analysis of their algorithm is tight; it is a 3–approximation algorithm. The running time is $O(n^2 \log n)$, where $n = n_d + n_f$, $|D| = n_d$ and $|F| = n_f$.

The combinatorial algorithm design by Mettu and Laxton [13] is very similar to the algorithm above as they find a quantity T_i for every facility i such that

$$\sum_{j \in D} \max\{T_i - c_{ij}, 0\} = f_i. \tag{4.11}$$

They use a clustering similar to the one of Jain and Vazirani to attain a 3-performance guarantee.

4.3 The Dual-Fitting Algorithm

The idea of the dual fitting algorithm is as follows:
If for a vector v and for every facility i and a set of k of demand points the following holds:

$$\sum_{j=1}^{k} v_j \leq \beta \left(f_i + \sum_{j=1}^{k} c_{ij} \right), \tag{4.12}$$

then the pair $\beta^{-1}(v, \overline{t})$ with $\overline{t_{ij}} = \max\{v_j - \beta c_{ij}, 0\}$ is a dual feasible solution. By the weak duality theorem, $\beta^{-1} \sum\limits_{j \in D} v_j$ is a lower bound for the optimal solution and therefore the approximation ratio of the algorithm is β.

The first algorithm for the UCFLP using this technique was developed by Mahdian, Markakis, Saberi, and Vazirani [12]. They construct the pair (v, t) in the same way of Jain and Vazirani's algorithm; the only difference is that a facility i is considered fully paid when $\sum\limits_{j \text{ unconnected}} t_{ij} = f_i$. Once a facility becomes fully paid, it keeps this property until the end of the algorithm. All the fully paid facilities are opened and each demand point is assigned to the closest open facility. Using this technique Mahdian [12] shows that relation (4.12) is satisfied for $\beta = 1.86$, Consequently, his algorithm has a 1.86 performance guarantee.

This algorithm was improved by Jain, Mahdian, and Saberi [8] who, inspired by the interpretation of the dual program, made changes to the algorithms of [7] and [12] based on a more realistic way of contributing:

A demand point j has to contribute towards opening i' with the gain that j will have if i' is opened and j will be served by i' instead of i. More precisely, a facility i is fully paid when $\sum\limits_{j \in D} t_{ij} = f_i$. We construct (v, t) as in Jain and Vazirani [7] with the difference that after a demand point gets connected, say to a facility i, we set $t_{i'j} = \max\{c_{ij} - c_{i'j}, 0\}$ for all the facilities $i' \neq i$. Due to this improvement, the performance guarantee increased to 1.61 while in [12] the performance guarantee was 1.52.

5 Conclusion

In this article we presented an integer programming formulation together with the linear programming relaxation formulation for solving the UCFLP. We discussed, in details, the known approximation algorithms for the metric UCFLP. In addition, we reviewed the main results of these algorithms. The different approaches to solving the UCFLP provided an insight into solution techniques for the more complex CFLP.

References

[1] K. Aardal, Capacitated facility location: Separation algorithms and computational experience, Mathematical Programming, **81,** (1998), 149–175.

[2] K. Aardal, Reformulation of capacitated facility location problems: How redundant information can help, *Annals of Operations Research*, **82,** (1998), 289–308.

[3] J. E. Beasley, An algorithm for solving large capacitated warehouse location problems, *European Journal of Operational Research*, **33,**(1988), 314–325.

[4] J. E. Beasley, Lagrangean heuristics for location problems, European Journal of Operational Research, **65,** (1993), 383–399.

[5] F. Chudak, Improved Approximation Algorithms for the Uncapacitated Facility Location Problem, Ph.D. thesis, Cornell University, 1998.

[6] G. Cornuejols, G. L. Nemhauser, L. A. Wosley, The Uncapacitated Facility Location Problem, In P. Mirchandani, R. Francis (editors), Discrete Location Theory, John Wiley and Sons, New York, 1990, 119–171.

[7] K. Jain, V. Vazirani, Approximation Algorithms for metric facility Location and K-median problems using the primal-dual schema and Lagrangean relaxation, Journal of the ACM, **48,** 274–296.

[8] K. Jain, M. Mahdian, A. Saberi, A new greedy approach for facility location problem, In proceedings of the 34th Annual ACM Symposium on theory of Computing, 2002.

[9] E. L. Lawler, Combinatorial Optimisation: Networks and Matroids, Holt, Rinehart, and Winston, New York, 1976.

[10] J. Lin, J. Vitter, ε-Approximation with minimum packing constraint violation, In proceedings of 24th Annual ACM Symposium on Theory of Computing, (1992), 771-782.

[11] M. Mahdian, E. Markakis, A. Saberi, V. Vazirani, A greedy facility Location algorithm analysed using dual fitting, In Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, Springer-Verlag LNCS, **2129,** (2001), 127–137.

[12] M. Mahdian, Y. Ye, J. Zhang, Approximation algorithm for the Metric facility Location problem, SIAM Journal on Computing, **36**, no. 2, (2006), 411–432.

[13] R. Mettu, C. Plaxton, The online median problem, In proceeding of the 41st IEEE Symposium on Foundation of Computer Science, (2000), 339–348. 2000.

[14] P. Mirchindani, R. Francis (editors), Discrete Location Theory, John Wiley and sons, New York, 1990.

[15] D. Shmoys, E. Tardos, K. Aardal, Approximation Algorithm for Facility Location problem, Proceedings of 29th ACM Stoc., (1997), 265–274.