



Uncapacitated facility location problem with self-serving demands

E Monabbati*

Received: 25 September 2012; Revised: 21 February 2013; Accepted: 13 March 2013

Abstract

In classical uncapacitated facility location problems (UFLP) the goal is to satisfy requirements of some demand points by setting up some servers, among potential facility locations, such that the total cost including service costs and fixed costs are minimized. In this paper a generalization of UFLP is considered in which some demand points, called self-serving, could be served exclusively by a new server at that point. Numerical experiments show that near optimal solutions are achieved by the proposed method.

Key words: Location problems, generalization of uncapacitated facility location problem, dual-ascent.

1 Introduction

In the uncapacitated facility location problem (UFLP) there are m customers indexed by $i \in \mathcal{I} := \{1, 2, \dots, m\}$ which may be served by n potential facilities (servers) indexed by $j \in \mathcal{J} := \{1, 2, \dots, n\}$. Requirements of a demand point i could be served by a nonnegative cost c_{ij} from facility j . Setting up a server in facility location j has a fixed cost $f_j > 0$. The goal is to choose some facility locations to set up servers, such that a total cost including service costs and fixed costs are minimized. This NP-hard problem has several interesting mathematical formulations, which can be found in survey papers on this topic such as [5, 9, 14, 15], as well as in the references therein. Many researchers have handled this problem and devised heuristic, approximation and exact algorithms to solve this problem (see [8, 14] for some comparisons of these algorithms).

In this paper a simple generalization of UFLP, called UFLP-SS, has been investigated. Suppose the requirements of some demand points, called *self-serving demands*, could be satisfied by setting up a server there. We call these potential facility locations, *demand-side*. Note that a server established in a self-serving demand point, could only fulfill the requirements of this point. Denote by $\mathcal{K} = \{1, \dots, p\} \subseteq \mathcal{I}$ the set of self-serving demands and also let a fixed cost of setting up a server in such demand points be $g_k > 0$, $k \in \mathcal{K}$.

*Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Islamic Republic of Iran, 91775-1159, email: eh_mo236@stu-mail.um.ac.ir

UFLP-SS has a linear zero-one programming formulations similar to UFLP. The objective is to

$$\text{minimise } \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} + \sum_{k \in \mathcal{K}} g_k z_k + \sum_{j \in \mathcal{J}} f_j y_j \quad (1)$$

$$\text{subject to } \sum_{j \in \mathcal{J}} x_{kj} + z_k = 1, \quad k \in \mathcal{K}, \quad (2)$$

$$\sum_{j \in \mathcal{J}} x_{lj} = 1, \quad \ell \in \mathcal{L}, \quad (3)$$

$$x_{ij} - y_j \leq 0, \quad i \in \mathcal{I}, j \in \mathcal{J}, \quad (4)$$

$$z_k, y_j, x_{ij} \in \{0, 1\}, \quad i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}, \quad (5)$$

where $\mathcal{L} = \mathcal{I} \setminus \mathcal{K}$, x_{ij} is one if demand point i is served by a server placed at the potential facility location j and is zero otherwise, y_j is one if a server is set up at potential location j and zero otherwise, z_k is one, only if self-serving demand point k is served by a server placed at this point.

Constraint set (2) shows that a self-serving demand point may be served by either a demand-side server or a server from potential facility locations. Constraint sets (3) and (4) are exactly the same as for UFLP. Constraint set (3) ensures that each demand point must be served by a server and constraint set (4) reveal that if no server is placed at location j , (*i.e.* $y_j = 0$), then no demand point could be served from this facility location. It is easy to see that if $\mathcal{K} = \emptyset$ then UFLP-SS reduces to UFLP.

Recently, a feasible region of this problem, when equalities are replaced by inequalities, was studied by Baiou & Barahona [2] on a directed graph. They give conditions under which the region has integral vertices. Vasko *et al.* [16] introduced this model as partial coverage UFLP and investigated some large scale applications in inventory sizing. They also show that this problem simply could be converted to a UFLP.

In UFLP-SS, (1)–(5), replacing (2) with

$$\sum_{j \in \mathcal{J}} x_{kj} + z_k \geq 1, \quad k \in \mathcal{K}$$

we have a problem which is known as *uncapacitated facility location problem with penalties*, when $\mathcal{K} = \mathcal{I}$. It is easy to see that the later is equivalent to UFLP-SS. Several approximation algorithms are proposed for this problem and its variants (see [1, 4, 7, 10, 11, 12, 17, 18]).

This section is concluded by describing another relation between UFLP and UFLP-SS. Consider a UFLP in which a location of a subset $\mathcal{J}^P \subseteq \mathcal{J}$ of servers are determined, that is, there exist servers at each potential facility location in the set \mathcal{J}^P . Therefore the aim is to serve all demand points using some new servers among $\mathcal{J} \setminus \mathcal{J}^P$ potential facility locations together with servers in \mathcal{J}^P . Define decision variables x_{ij} and y_j as in UFLP-SS, and change the definition of z_k as follows: z_i is one, only if a server in \mathcal{J}^P satisfies the requirements of demand point i . On the other hand, if $z_i = 1$ then a server from \mathcal{J}^P with minimum service cost is assigned to the demand point i . Thus by the

assumption $g_i = \min_{j \in \mathcal{J}^P} c_{ij}$ we have a UFLP-SS. In the next section an implementation of the Erlenkotter algorithm proposed in [6] is modified to find near optimal solutions for UFLP-SS.

2 Solution procedure

The Erlenkotter algorithm is a heuristic weak-dual based method for UFLP. In this method considering weak-dual of the problem and complementary slackness conditions, a near-optimal primal solution is given. The Erlenkotter method is modified and applied to UFLP-SS, directly.

Consider UFLP-SS as formulated in equations (1)–(5). An LP-relaxation of the problem could be achieved by relaxing the integrality assumption on x_{ij} , y_j and z_k such that the objective is to

$$\begin{aligned} \text{minimise } z_p(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} + \sum_{k \in \mathcal{K}} g_k z_k + \sum_{j \in \mathcal{J}} f_j y_j \\ \text{subject to } \sum_{j \in \mathcal{J}} x_{kj} + z_k &= 1, & k \in \mathcal{K}, \\ \sum_{j \in \mathcal{J}} x_{\ell j} &= 1, & \ell \in \mathcal{L}, \\ -x_{ij} + y_j &\geq 0, & i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}, \\ z_k, y_j, x_{ij} &\geq 0, & i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}. \end{aligned}$$

A dual of this LP, using dual-variables \mathbf{u} , \mathbf{v} , \mathbf{w} , is to

$$\text{maximise } \sum_{k \in \mathcal{K}} u_k + \sum_{\ell \in \mathcal{L}} v_\ell \quad (6)$$

$$\text{subject to } u_k - w_{kj} \leq c_{kj}, \quad j \in \mathcal{J}, k \in \mathcal{K}, \quad (7)$$

$$v_\ell - w_{\ell j} \leq c_{\ell j}, \quad \ell \in \mathcal{L}, j \in \mathcal{J}, \quad (8)$$

$$u_k \leq g_k, \quad k \in \mathcal{K}, \quad (9)$$

$$\sum_{i \in \mathcal{I}} w_{ij} \leq f_j, \quad j \in \mathcal{J}, \quad (10)$$

$$w_{ij} \geq 0, \quad i \in \mathcal{I}, j \in \mathcal{J}. \quad (11)$$

From (7), (8) and (11) we have

$$w_{kj} \geq (u_k - c_{kj})^+, \quad k \in \mathcal{K}, j \in \mathcal{J},$$

$$w_{\ell j} \geq (v_\ell - c_{\ell j})^+, \quad \ell \in \mathcal{L}, j \in \mathcal{J}.$$

where $(a)^+ = \max\{0, a\}$. Assume that $w_{kj} = (u_k - c_{kj})^+$, $k \in \mathcal{K}$ and $w_{\ell j} = (v_\ell - c_{\ell j})^+$, $\ell \in \mathcal{L}$, $j \in \mathcal{J}$. Eliminating variables w_{ij} we get the following form of the dual problem,

called the *condensed dual form*:

$$\text{maximise } z_d(\mathbf{u}, \mathbf{v}) = \sum_{k \in \mathcal{K}} u_k + \sum_{\ell \in \mathcal{L}} v_\ell \quad (12)$$

$$\text{subject to } \rho_j(\mathbf{u}, \mathbf{v}) \leq 0, \quad j \in \mathcal{J}, \quad (13)$$

$$u_k \leq g_k, \quad k \in \mathcal{K}, \quad (14)$$

where

$$\rho_j(\mathbf{u}, \mathbf{v}) = \sum_{k \in \mathcal{K}} (u_k - c_{kj})^+ + \sum_{\ell \in \mathcal{L}} (v_\ell - c_{\ell j})^+ - f_j.$$

This pair of primal-dual problems is solved by finding a primal and dual feasible solutions $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and (\mathbf{u}, \mathbf{v}) , respectively, satisfying the following complementary slackness (CS) conditions (after some simplifications)

$$x_{kj}(u_k - c_{kj})^- = 0, \quad j \in \mathcal{J}, k \in \mathcal{K}, \quad (15)$$

$$x_{\ell j}(v_\ell - c_{\ell j})^- = 0, \quad j \in \mathcal{J}, \ell \in \mathcal{L}, \quad (16)$$

$$z_k(u_k - g_k) = 0, \quad k \in \mathcal{K}, \quad (17)$$

$$y_j \rho_j(\mathbf{u}, \mathbf{v}) = 0, \quad j \in \mathcal{J}, \quad (18)$$

$$(u_k - c_{kj})^+(y_j - x_{kj}) = 0, \quad j \in \mathcal{J}, k \in \mathcal{K}, \quad (19)$$

$$(v_\ell - c_{\ell j})^+(y_j - x_{\ell j}) = 0, \quad j \in \mathcal{J}, \ell \in \mathcal{L}, \quad (20)$$

where $(a)^- = \min\{0, a\}$. Note that equations (15) and (16) are obtained considering the fact that $a - (a)^+ = (a)^-$.

Given a dual feasible solution, (\mathbf{u}, \mathbf{v}) , using complementary slackness conditions (15)–(20), one can obtain a feasible integer primal solution. To this end new notation are introduced. Define $\bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ and $\bar{\mathcal{K}}(\mathbf{u})$ as follows

$$\bar{\mathcal{J}}(\mathbf{u}, \mathbf{v}) = \{j \in \mathcal{J} \mid \rho_j(\mathbf{u}, \mathbf{v}) = 0\},$$

$$\bar{\mathcal{K}}(\mathbf{u}) = \{k \in \mathcal{K} \mid u_k = g_k\}.$$

If $j \in \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ then the corresponding primal solution, y_j , could be equal to 1. This is true for z_k if $k \in \bar{\mathcal{K}}(\mathbf{u})$. Choose a subset $\mathcal{J}^+ \subseteq \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ as indices of candidate primal variables which may be 1. A condition under which a primal solution, $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, exists and satisfies all CS conditions, but not necessarily (19) and (20) are now derived.

Let $\ell \in \mathcal{L}$. Since $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is feasible, then from (3), there is an index $j \in \mathcal{J}^+$ such that $x_{\ell j}$ is one and (16) implies that $(v_\ell - c_{\ell j})^- = 0$, *i.e.* $v_\ell \geq c_{\ell j}$.

Now, let $u_k < g_k$ for some $k \in \mathcal{K}$, then $z_k = 0$ from (17). Thus from (2) we have $\sum_{j \in \mathcal{J}} x_{kj} = 1$. Therefore, similar to the previous arguments, we must have an index $j \in \mathcal{J}^+$ such that $u_k \geq c_{kj}$.

Consequently if (\mathbf{u}, \mathbf{v}) has the properties

$$\begin{aligned} &\text{for each } k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u}) \text{ there is an index } j \in \mathcal{J}^+ \text{ such that } u_k \geq c_{kj}, \\ &\text{for each } \ell \in \mathcal{L} \text{ there is an index } j \in \mathcal{J}^+ \text{ such that } v_\ell \geq c_{\ell j}, \end{aligned} \quad (21)$$

then the following primal feasible solution is obtained

$$\begin{aligned} y_j &= \begin{cases} 1 & \text{if } j \in \mathcal{J}^+, \\ 0 & \text{otherwise,} \end{cases} & x_{kj} &= \begin{cases} 1 & \text{if } k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u}) \text{ and } j = \alpha(k), \\ 0 & \text{otherwise,} \end{cases} \\ z_k &= \begin{cases} 1 & \text{if } k \in \bar{\mathcal{K}}(\mathbf{u}), \\ 0 & \text{otherwise,} \end{cases} & x_{\ell j} &= \begin{cases} 1 & \text{if } \ell \in \mathcal{L} \text{ and } j = \alpha(\ell), \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (22)$$

where $\alpha(i)$ is such that $c_{i\alpha(i)} = \min_{j \in \mathcal{J}^+} c_{ij}$. This pair of primal-dual solutions satisfy all complementary slackness conditions except (19) and (20).

Hence the duality gap is given by the expression in Lemma 1

Lemma 1 *Given dual feasible solution (\mathbf{u}, \mathbf{v}) satisfying (21) and corresponding primal solution (22) we have*

$$z_p(\mathbf{x}, \mathbf{y}, \mathbf{z}) - z_d(\mathbf{u}, \mathbf{v}) = \sum_{k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u})} \sum_{\substack{j \in \mathcal{J}^+ \\ j \neq \alpha(k)}} (u_k - c_{kj})^+ + \sum_{k \in \bar{\mathcal{K}}} \sum_{j \in \mathcal{J}^+} (u_k - c_{kj})^+ + \sum_{\ell \in \mathcal{L}} \sum_{\substack{j \in \mathcal{J}^+ \\ j \neq \alpha(\ell)}} (v_\ell - c_{\ell j})^+.$$

Proof: Primal and dual solutions are constructed such that they were feasible and satisfied all complementary slackness conditions except (19) and (20). A demand point i is assigned to a server $\alpha(i) \in \mathcal{J}^+$, that is $x_{i,\alpha(i)} = 1$. If there is an index $\alpha(i) \neq j \in \mathcal{J}^+$ such that $u_i > c_{ij}$ then $y_j = 1$, $x_{ij} = 0$ and $(u_i - c_{ij})^+ > 0$. Thus $(u_i - c_{ij})^+(y_j - x_{ij}) = (u_i - c_{ij})^+$; The first and the last term in the result are drawn from this fact. Now we focus on the second summation. If $k \in \bar{\mathcal{K}}(\mathbf{u})$ then $z_k = 1$, therefore $x_{kj} = 0$ by (2), $j \in \mathcal{J}$. On the other hand for each $j \in \mathcal{J}^+$ we have $y_j = 1$, so, $(u_k - c_{kj})^+(y_j - x_{kj}) = (u_k - c_{kj})^+$, $j \in \mathcal{J}^+$. This completes the proof. ■

Using optimality conditions for LPs (primal and dual feasibility and complementary slackness), we endeavor to solve the LP-relaxation of UFLP-SS. Our algorithm like Erlenkotter's, consists of two procedures: dual ascent and dual adjustment. The first one tries to find a good dual feasible solution that satisfies (21), and the second attempts to reduce the infeasibility of complementary slackness conditions (19) and (20).

2.1 Dual ascent

The dual ascent procedure starts with a dual feasible solution $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ and a set $\mathcal{I}' = \mathcal{K}' \cup \mathcal{L}'$ of candidate indices for ascending. It tries to achieve a better dual solution by increasing the components of \bar{v}_ℓ , $\ell \in \mathcal{L}'$, \bar{u}_k , $k \in \mathcal{K}'$. Increasing is based on two simple facts:

- If $\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}}) < 0$ and $\bar{v}_\ell \geq c_{\ell j}$ for some $\ell \in \mathcal{L}'$ then increasing v_ℓ by $\Delta > 0$ increases $\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ by the amount of Δ ,
- If $\bar{v}_\ell \leq c_{\ell j}$ for some $\ell \in \mathcal{L}'$ then increasing v_ℓ makes no change in $\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}})$.

These facts are also true if we replace \bar{v}_ℓ by \bar{u}_k for some $k \in \mathcal{K}'$. Algorithm 1 illustrates the steps of the dual ascent algorithm.

The following lemma shows that a dual solution obtained by the dual ascent algorithm satisfies condition (21).

Algorithm 1: Dual ascent procedure

Input : A dual feasible solution $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ and the sets \mathcal{K}' , \mathcal{L}' .
Output : A dual feasible solution $(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ that satisfies condition (21).

```

1 while  $\mathcal{L}' \neq \emptyset$  do
2   Select  $\ell$  from  $\mathcal{L}'$ 
3    $\Delta_1 \leftarrow \min\{-\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}}) : \rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \leq 0, \bar{v}_\ell \geq c_{\ell j}\}$ 
4    $\Delta_2 \leftarrow \min\{c_{\ell j} - \bar{v}_\ell : \bar{v}_\ell < c_{\ell j}\}$ 
5    $\Delta \leftarrow \min\{\Delta_1, \Delta_2\}$ 
6   if  $\Delta > 0$  then
7      $\bar{v}_\ell \leftarrow \bar{v}_\ell + \Delta$ 
8     Update  $\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ ,  $j \in \mathcal{J}$ 
9   else
10     $\mathcal{L}' \leftarrow \mathcal{L}' \setminus \{\ell\}$ 
11 while  $\mathcal{K}' \neq \emptyset$  do
12   Select  $k$  from  $\mathcal{K}'$ 
13    $\Delta_1 \leftarrow \min\{-\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}}) : \rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}}) \leq 0, \bar{u}_k \geq c_{kj}\}$ 
14    $\Delta_2 \leftarrow \min\{c_{kj} - \bar{u}_k : \bar{u}_k < c_{kj}\}$ 
15    $\Delta_3 \leftarrow g_k - \bar{u}_k$ 
16    $\Delta \leftarrow \min\{\Delta_1, \Delta_2, \Delta_3\}$ 
17   if  $\Delta > 0$  then
18      $\bar{u}_k \leftarrow \bar{u}_k + \Delta$ 
19     Update  $\rho_j(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ ,  $j \in \mathcal{J}$ 
20   else
21     $\mathcal{K}' \leftarrow \mathcal{K}' \setminus \{k\}$ 

```

Lemma 2 *Algorithm 1 ends at a dual feasible solution which satisfies condition (21).*

Proof: Without loss of generality we assume that $\mathcal{K}' = \mathcal{K}$ and $\mathcal{L}' = \mathcal{L}$ in Algorithm 1. Let (\mathbf{u}, \mathbf{v}) be a dual solution at an iteration of the dual ascent procedure and suppose that condition (21) is not satisfied, that is

$$v_{\ell_0} < c_{\ell_0 j} \text{ for all } j \in \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v}) \text{ and some } \ell_0 \in \mathcal{L}, \quad (23)$$

or

$$u_{k_0} < c_{k_0 j} \text{ for all } j \in \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v}) \text{ for some } k_0 \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u}). \quad (24)$$

It is shown that in each case, Δ , which is computed in Steps 3–5 or Steps 13–16 in Algorithm 1, is positive, and therefore Algorithm 1 does not terminate at (\mathbf{u}, \mathbf{v}) .

First suppose v satisfies condition (23). Consider Steps 3 through 5. Since $v_{\ell_0} < c_{\ell_0 j}$, then $\Delta_2 > 0$. If $\Delta_1 = 0$ then $\rho_{j_0}(\mathbf{u}, \mathbf{v}) = 0$ for some $j_0 \in \mathcal{J}$ for which $v_{\ell_0} \geq c_{\ell_0 j_0}$, meanwhile $j_0 \in \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ and this contradicts condition (23). Thus $\Delta > 0$ in this case.

Assume that u satisfies condition (24). The proof of this part is very similar to the previous one. Since $k_0 \notin \bar{\mathcal{K}}(\mathbf{u})$ thus $\Delta_3 > 0$, by definition. If $\Delta_1 = 0$ then $\rho_{j_0}(\mathbf{u}, \mathbf{v}) = 0$ for some j_0 for which $u_{k_0} \geq c_{k_0 j_0}$ which is impossible. It is clear that Δ_2 is also positive (see Steps 13–16 from Algorithm 1). Thus $\Delta = \min\{\Delta_1, \Delta_2, \Delta_3\} > 0$. ■

Observe that in general we cannot replace $\bar{\mathcal{K}}(\mathbf{u})$ by one of its subsets, as we do for $\bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$, since the upper bound of variable u_k may prevent it to increase, *e.g.* when $g_k < \min_{j \in \mathcal{J}^+} c_{kj}$.

2.2 Dual adjustment

Algorithm 1 gives a good feasible dual solution. However, this solution could cause some violations in the complementary slackness conditions. If there is more than one $j \in \mathcal{J}^+$ for which $v_\ell \geq c_{\ell j}$ then the duality gap is increased by $(v_\ell - c_{\ell j})^+ > 0$ for $j \in \mathcal{J}^+$, $j \neq \alpha(\ell)$. Similarly, if there is more than one $j \in \mathcal{J}^+$ for which $u_k \geq c_{kj}$ then the duality gap is increased by $(u_k - c_{kj})^+ > 0$ for $j \in \mathcal{J}^+$, $j \neq \alpha(k)$ (see Lemma 1). To state how the dual adjustment procedure could improve the duality gap, the following notation are introduced

$$\begin{aligned}
 \mathcal{J}_{1k}^* &= \{j \in \mathcal{J}^+ : u_k \geq c_{kj}\}, & k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u}), \\
 \mathcal{J}_{2\ell}^* &= \{j \in \mathcal{J}^+ : v_\ell \geq c_{\ell j}\}, & \ell \in \mathcal{L}, \\
 \mathcal{J}_{1k}^+ &= \{j \in \mathcal{J}^+ : u_k > c_{kj}\}, & k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u}), \\
 \mathcal{J}_{2\ell}^+ &= \{j \in \mathcal{J}^+ : v_\ell > c_{\ell j}\}, & \ell \in \mathcal{L}, \\
 \mathcal{I}_{1j}^+ &= \{k : \mathcal{J}_{1k}^* = \{j\}\}, & j \in \mathcal{J}^+, \\
 \mathcal{I}_{2j}^+ &= \{\ell : \mathcal{J}_{2\ell}^* = \{j\}\}, & j \in \mathcal{J}^+, \\
 \mathcal{I}_{1j}^- &= \{k \in \mathcal{K} : u_k - c_{kj} > 0\}, & j \in \mathcal{J}, \\
 \mathcal{I}_{2j}^- &= \{\ell \in \mathcal{L} : v_\ell - c_{\ell j} > 0\}, & j \in \mathcal{J}, \\
 c_{k\beta(k)} &= \min_{\substack{j \in \mathcal{J}^+ \\ j \neq \alpha(k)}} c_{kj}, & k : |\mathcal{J}_{1k}^+| > 1, \\
 c_{\ell\beta(\ell)} &= \min_{\substack{j \in \mathcal{J}^+ \\ j \neq \alpha(\ell)}} c_{\ell j}, & \ell : |\mathcal{J}_{2\ell}^+| > 1.
 \end{aligned}$$

Suppose (\mathbf{u}, \mathbf{v}) satisfies property (21). Let $k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u})$ and $j_0, j_1 \in \mathcal{J}^+$ be such that $u_k > c_{kj_h}$, $h = 0, 1$. Thus $w_{kj_h} = u_k - c_{kj_h} > 0$, $h = 0, 1$. Let x be a corresponding primal solution for (\mathbf{u}, \mathbf{v}) . Since x is primal feasible then $\sum_j x_{kj} = 1$ (observe that since $k \in \mathcal{K} \setminus \bar{\mathcal{K}}(\mathbf{u})$ then $z_k = 0$) and at least one of x_{kj_h} , $h = 0, 1$, say x_{kj_0} , is zero, meanwhile $j_0 \in \mathcal{J}^+$, so $y_{j_0} = 1$. Therefore $w_{kj_0}(y_{j_0} - x_{kj_0}) > 0$. This shows that complementary condition (18) is violated if $|\mathcal{J}_{1k}^+| > 1$.

To decrease these violations a decrease in u_k is needed. By definition $\rho_j(\mathbf{u}, \mathbf{v}) = \sum_{\ell \in \mathcal{L}} (v_\ell - c_{\ell j})^+ + \sum_{k \in \mathcal{K}} (u_k - c_{kj})^+ - f_j$. Thus a change in u_k affects $\rho_j(\mathbf{u}, \mathbf{v})$ if and only if $k \in \mathcal{I}_{1j}^-$. On the other hand if u_k decreases to u'_k by $\Delta > 0$ then $(u'_k - c_{kj})^+ < (u_k - c_{kj})^+$ for $j \in \mathcal{J}_{1k}^+$. Thus a decrease in u_k makes all $\rho_j(\mathbf{u}, \mathbf{v})$ negative with $j \in \mathcal{J}_{1k}^+$. Since $|\mathcal{J}_{1k}^+| > 1$, then \mathcal{J}_{1k}^+ has at least two elements $\alpha(k)$ and $\beta(k)$, *i.e.* after changing u_k to $u_k + \Delta$ we have $\rho_{\alpha(k)} < 0$, $\rho_{\beta(k)} < 0$. These slacks could be compensated by an increase in variable u_s , $s \in \mathcal{I}_{1\alpha(k)}^- \cup \mathcal{I}_{1\beta(k)}^-$ provided that this variable does not change any $\rho_j(\mathbf{u}, \mathbf{v})$ other than $\rho_{\alpha(k)}$ and $\rho_{\beta(k)}$, that is $\mathcal{J}_{1s}^* = \{\alpha(k)\}$ or $\mathcal{J}_{1s}^* = \{\beta(k)\}$, in other words $s \in \mathcal{I}_{1\alpha(k)}^+ \cup \mathcal{I}_{1\beta(k)}^+$. Here we choose a variable from \mathcal{K} to compensate for the decrease in u_k . One can remove this restriction and select the remunerated variable from \mathcal{L} .

A similar argument help us to choose an index $\ell \in \mathcal{L}$ as a candidate variable among v_ℓ , $\ell \in \mathcal{L}$. The amount that a variable's value can decrease in the dual adjustment algorithm is discussed next. In Algorithm 2, which shows the dual adjustment steps, a candidate

variable u_k (v_ℓ) is decreased to the largest c_{kj} ($c_{\ell j}$) not greater than u_k (v_ℓ) (line 5 or 20). Next in lines 6 and 21 all $\rho_j(\mathbf{u}, \mathbf{v})$, with $u_k > c_{kj}$ ($v_\ell > c_{\ell j}$) are updated.

In lines 8–13 the dual ascent algorithm is called three times. The first call attempts to increase the dual variables in the set $\mathcal{I}_{1\alpha(k)}^+ \cup \mathcal{I}_{1\beta(k)}^+$ to remove the slack of constraints $\rho_h(\mathbf{u}, \mathbf{v}) < 0$, $h = \alpha(k), \beta(k)$. The second call is prompted from decreasing u_k discontinuously to the next c_{kj} and the last call ensures that (\mathbf{u}, \mathbf{v}) is dual feasible and satisfies (21).

Algorithm 2: Dual adjustment procedure

Input : A dual feasible solution (\mathbf{u}, \mathbf{v}) that satisfies (21)
Output : Tries to reduce duality gap.

```

1 for  $k \in \mathcal{K} \setminus \overline{\mathcal{K}}(\mathbf{u})$  do
2   if  $|\mathcal{J}_{1k}^+| > 1$  then
3     if  $\mathcal{I}_{1\alpha(k)}^+ \cup \mathcal{I}_{1\beta(k)}^+ \neq \emptyset$  then
4        $c_k^* \leftarrow \max_{j: u_k > c_{kj}} c_{kj}$ 
5        $\Delta \leftarrow u_k - c_k^*$ 
6       For each  $j$  such that  $u_k > c_{kj}$  set  $\rho_j(\mathbf{u}, \mathbf{v}) \leftarrow \rho_j(\mathbf{u}, \mathbf{v}) - \Delta$ 
7        $u_k \leftarrow c_k^*$ 
8        $\mathcal{K}' \leftarrow \mathcal{I}_{1\alpha(k)}^+ \cup \mathcal{I}_{1\beta(k)}^+$ ,  $\mathcal{L}' \leftarrow \emptyset$ 
9       apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
10       $\mathcal{K}' \leftarrow \mathcal{K}' \cup \{k\}$ 
11      apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
12       $\mathcal{K}' \leftarrow \mathcal{K} \setminus \overline{\mathcal{K}}(\mathbf{v})$ 
13      apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
14      Update  $\mathcal{J}^+$ ,  $\overline{\mathcal{K}}(\mathbf{u})$ ,  $\mathcal{J}_{1j}^+$ 's and  $\mathcal{I}_{1j}^+$ 's
15      If  $u_k$  doesn't become its original value then go to 2

16 for  $\ell \in \mathcal{L}$  do
17   if  $|\mathcal{J}_{2\ell}^+| > 1$  then
18     if  $\mathcal{I}_{2\alpha(\ell)}^+ \cup \mathcal{I}_{2\beta(\ell)}^+ \neq \emptyset$  then
19        $c_\ell^* \leftarrow \max_{j: v_\ell > c_{\ell j}} c_{\ell j}$ 
20        $\Delta \leftarrow v_\ell - c_\ell^*$ 
21       for each  $j$  such that  $v_\ell > c_{\ell j}$  set  $\rho_j(\mathbf{u}, \mathbf{v}) \leftarrow \rho_j(\mathbf{u}, \mathbf{v}) - \Delta$ 
22        $v_\ell \leftarrow c_\ell^*$ 
23        $\mathcal{K}' \leftarrow \emptyset$ ,  $\mathcal{L}' \leftarrow \mathcal{I}_{2\alpha(\ell)}^+ \cup \mathcal{I}_{2\beta(\ell)}^+$ 
24       apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
25        $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\ell\}$ 
26       apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
27        $\mathcal{L}' \leftarrow \mathcal{L}$ 
28       apply Algorithm 1 on  $(\mathcal{K}', \mathcal{L}')$ 
29       Update  $\mathcal{J}^+$ ,  $\mathcal{J}_{2j}^+$ 's and  $\mathcal{I}_{2j}^+$ 's
30       If  $v_\ell$  doesn't become its original value then go to 17.
```

Figure 1 shows a combination of the proposed procedures which may be used to solve UFLP-SS. Note that an initial dual feasible solution of the problem is $u_k = \min\{g_k, \min_j c_{kj}\}$, $k \in \mathcal{K}$, $v_\ell = \min_j c_{k\ell}$, $\ell \in \mathcal{L}$. The subset $\mathcal{J}^+ \subseteq \overline{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ is constructed using a simple drop heuristic method such as the one presented in Algorithm 3.

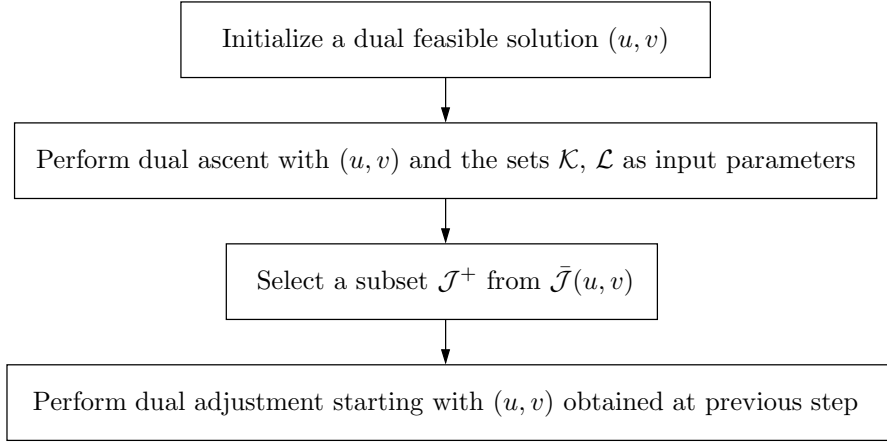


Figure 1: A procedure to produce near optimal solution of UFLP-SS.

Algorithm 3: A procedure to construct a subset \mathcal{J}^+

Input : A set $\bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$
Output : Construct a subset $\mathcal{J}^+ \subseteq \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ satisfying (21)

```

1  $\mathcal{J}^+ \leftarrow \bar{\mathcal{J}}(\mathbf{u}, \mathbf{v})$ 
2 improvement  $\leftarrow$  true
3 while improvement do
4   for  $j \in \mathcal{J}^+$  do
5      $\mathcal{J}^0 \leftarrow \mathcal{J}^+ \setminus \{j\}$ 
6     if there is a primal solution satisfying (22) using the set  $\mathcal{J}^0$  then
7        $t_j \leftarrow$  objective value at the primal solution
8     else
9        $t_j \leftarrow +\infty$ 
10   $t_{j^*} \leftarrow \min t_j$ 
11  if  $t_{j^*} = +\infty$  then
12    improvement  $\leftarrow$  false
13  else
14     $\mathcal{J}^+ \leftarrow \mathcal{J}^+ \setminus \{j^*\}$ 
  
```

3 Computational results

A method proposed in the previous section is applied on randomly generated test cases. A method similar to the one used in Barahona & Chudak [3] and Mladenovic *et al.* [13] is used to generate random UFLP-SS's. The number of $m + n$ point (m potential facility locations and n demand points) are uniformly generated in rectangle $[0, 0.5] \times [0, 1]$. The service cost c_{ij} is computed by rounding 1000 times the Euclidean distance between points i and j . All installation costs f_j are set to $\lceil 1000\sqrt{n}/m \rceil$ (type I), $\lceil 100\sqrt{n}/m \rceil$ (type II) and $\lceil 10\sqrt{n}/m \rceil$ (type III) for each n, m .

A demand-side server fixed set up costs g_k is set to be $\lceil 100\sqrt{m}/n \rceil$. For each m , reported in Table 1, we run the code for $n = m/2$ and two values of $p = m, m/2$. This allows us to probe the performance of the algorithm concerning the relation of potential servers and potential demand-side servers.

m	f_j	p	$1 - \text{gap}/z_d(\mathbf{u}, \mathbf{v})$	run time (sec)
200	Type I	100	0.99	6.70
		200	1.00	0.94
	Type II	100	0.99	2.35
		200	0.99	0.89
	Type III	100	0.99	2.29
		200	0.99	0.77
400	Type I	200	0.99	56.47
		400	1.00	5.79
	Type II	200	0.99	21.82
		400	0.99	5.48
	Type III	200	0.99	18.50
		400	0.99	5.23
800	Type I	400	0.98	436.63
		800	1.00	53.41
	Type II	400	0.99	171.80
		800	0.99	54.89
	Type III	400	0.99	151.80
		800	0.99	46.76
1600	Type I	800	0.99	2 897
		1 600	0.99	295.95
	Type II	800	0.99	1 243
		1 600	0.99	248.77
	Type III	800	0.99	1 072
		1 600	0.99	261.92

Table 1: Algorithm performance and run time.

Table 1 contains the results of applying the proposed algorithm on randomly generated UFLP-SS. It was implemented in MATLAB 7.10 and run on an Intel ®Core 2 Duo CPU 2.00 GHz, with 3 GB of RAM. Ten and five problems for $m < 500$ and $m > 500$ were respectively generated.

Column 4 of the table contains the average solution quality ($1 - \text{gap}/z_d(\mathbf{u}, \mathbf{v})$) and shows the performance of the algorithm to find optimal solutions of the LP-relaxation. Column 5 displays the average run time on each test case in seconds.

From this results it can be observed that the run-time is depend on the number of demand-side potential facilities versus the number of demand points. Experiments show that the run times of the algorithm substantially decrease when p approaches m .

4 Conclusions and future work

In this paper we considered a generalization of UFLP in which some demand points could be satisfied by servers, called demand-side, located at that point. It is shown that a version of UFLP in which the location of some facilities are pre-determined, could be converted to UFLP-SS. Finally a dual-based heuristic algorithm like Erlenkotter's is presented. Computational results show that this method gives near optimal solutions for randomly generated UFLP-SS's. Another consequence is that the run time of the proposed procedure drops when p , the number of demand-side servers, approaches m .

It seems that the idea of the proposed algorithm could be applied on the problems with more general structure. If demand constraints are in the form

$$\sum_{j \in \mathcal{J}} x_{ij} + \sum_{k \in \mathcal{K}} a_{ik} z_k = 1, \quad i \in \mathcal{K},$$

where $a_{ik} \in \mathbb{R}$ and $i, k \in \mathcal{K}$, then the condensed dual reads

$$\text{maximise } z_d(\mathbf{u}, \mathbf{v}) = \sum_{k \in \mathcal{K}} u_k + \sum_{\ell \in \mathcal{L}} v_\ell \quad (25)$$

$$\text{subject to } \rho_j(\mathbf{u}, \mathbf{v}) \leq 0, \quad j \in \mathcal{J}, \quad (26)$$

$$A^T \mathbf{u} \leq \mathbf{g}, \quad (27)$$

where $A = (a_{ik})_{i,k \in \mathcal{K}}$ (see equations (13) and (14)). The framework of dual ascent and dual adjustment plus a projection on the null space of A can be used. Note that if $A = I$ then we have a UFLP-SS.

Acknowledgment

The author thanks three anonymous referees for their constructive comments. The author also gratefully appreciates the help of Martin P. Kidd and Stephan E. Visagie for improving the presentation of the paper.

References

- [1] ASADI M, NIKNAFS A, & GHODSI M, 2009, *An approximation algorithm for the k-level uncapacitated facility location problem with penalties*, in SARBAZI-AZAD H, PARHAMI B, MIREMADI SG & HESSABI S (EDS), *Advances in Computer Science and Engineering*, Volume 6 of *Communications in Computer and Information Science*, pp. 41–49, Springer, Heidelberg.
- [2] BAIYOU M & BARAHONA F, 2009, *On the integrality of some facility location polytopes*, *Journal of Discrete Mathematics*, **23**(2), pp. 665–679.
- [3] BARAHONA F & CHUDAK F, 1999, *Solving large scale uncapacitated location problems*, (Unpublished), Technical Report RC21515, IBM, New York (NY).
- [4] CHARIKAR M, KHULLER S, MOUNT DM & NARASIMHAN G, 2001, *Algorithms for facility location problems with outliers*, *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, Washington (DC), pp. 642–651.
- [5] CORNUEJOLS G, NEMHAUSER GL & WOLSEY LA, 1990, *The uncapacitated facility location problem*, pp. 119–171 in MIRCHANDANI PB & FRANCIS RL (EDS), *Discrete location theory*, Wiley-Interscience, New York (NY).
- [6] FRANCIS RL & MCGINNIS F & WHITE JA, 1992, *Facility layout and location: An analytical approach*, Prentice Hall, Upper Saddle River (NJ).
- [7] HAYRAPETYAN A, SWAMY C, & TARDOS E, 2005, *Network design for information networks*, *Proceedings of the 17th annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia (PA), pp. 933–942.
- [8] HOEFER M, 2003, *Experimental comparison of heuristic and approximation algorithms for uncapacitated facility location*, pp. 165–178 in JANSEN K, MARGRAF M, MASTROLILLI M & ROLIM J (EDS), *Experimental and Efficient Algorithms*, Springer Berlin, Heidelberg.
- [9] KRARUP J & PRUZAN PM, 1983, *The simple plant location problem: Survey and synthesis*, *European Journal of Operational Research*, **12**(1), pp. 36–81.

- [10] LI Y, DU D, XIU N & XU D, 2012, *Improved approximation algorithms for the facility location problems with linear/submodular penalty*, Working paper series 2012, [Online], [Cited October, 15th 2013], Available at http://www.optimization-online.org/DB_HTML/2012/02/3343.html.
- [11] LI Y, DU D, XIU N & XU D, 2012, *A unified dual-fitting approximation algorithm for the facility location problems with linear/submodular penalties*, Journal of Combinatorial Optimization, August, pp. 1–12.
- [12] LI G, WANG Z, & XU D, 2012, *An approximation algorithm for the k-level facility location problem with submodular penalties*, Journal of Industrial and Management Optimization, **8(3)**, pp. 521–529.
- [13] MLADENOVIC N, BRIMBERG J & HANSEN P, 2006, *A note on duality gap in the simple plant location problem*, European Journal of Operational Research, **174(1)**, pp. 11–12.
- [14] THIZY JM, WASSENHOVE LNV & KHUMUWALA BM, 1985, *Comparison of exact and approximate methods of solving the uncapacitated plant location problem*, Journal of Operations Management, **6(1)**, pp. 23–34.
- [15] VERTER V, 2011, *Uncapacitated and capacitated facility location problems*, pp. 25–37 in EISELT HA & MARIANOV V (EDS), *Foundations of location analysis*, Springer, New York (NY).
- [16] VASKO FJ, NEWHART DD, STOTT KL & WOLF FE, 2003, *A large-scale application of the partial coverage uncapacitated facility location problem*, Journal of Operational Research Society, **54(1)**, pp. 11–20.
- [17] XU G & XU J, 2005, *An LP rounding algorithm for approximating uncapacitated facility location problem with penalties*, Information Processing Letters, **94(3)**, pp. 119–123.
- [18] XU G & XU J, 2009, *An improved approximation algorithm for uncapacitated facility location problem with penalties*, Journal of Combinatorial Optimization, **17(4)**, pp. 424–436.