

# Integrating Facility Location and Production Planning Decisions

**H. Edwin Romeijn**

*Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109-2117*

**Thomas C. Sharkey**

*Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, New York 12180-3590*

**Zuo-Jun Max Shen**

*Department of Industrial Engineering and Operations Research, University of California, 4129 Etcheverry Hall, Berkeley, California 94720-1777*

**Jiawei Zhang**

*IOMS-Operations Management, Stern School of Business, New York University, New York, New York 10012*

**We consider a metric uncapacitated facility location problem where we must assign each customer to a facility and meet the demand of the customer in future time periods through production and inventory decisions at the facility. We show that the problem, in general, is as hard to approximate as the set cover problem. We therefore focus on developing approximation algorithms for special cases of the problem. These special cases come in two forms: (i) specialize the production and inventory cost structure and (ii) specialize the demand pattern of the customers. In the former, we offer reductions to variants of the metric uncapacitated facility location problem that have been previously studied. The latter gives rise to a class of metric uncapacitated facility location problems where the facility cost function is concave in the amount of demand assigned to the facility. We develop a modified greedy algorithm together with the idea of cost-scaling to provide an algorithm for this class of problems with an approximation guarantee of 1.52. © 2009 Wiley Periodicals, Inc. NETWORKS, Vol. 55(2), 78–89 2010**

**Keywords:** facility location; production planning; approximation algorithms; computational complexity

Received February 2007; accepted November 2008

Correspondence to: T. C. Sharkey; e-mail: sharkt@rpi.edu

Contract grant sponsor: National Science Foundation; Contract grant number: DMI-0355533

Contract grant sponsor: National Science Foundation; Contract grant numbers: CMMI-0621433 and CMMI-0727640

Contract grant sponsor: National Science Foundation Graduate Research Fellowship

DOI 10.1002/net.20315

Published online 6 May 2009 in Wiley InterScience (www.interscience.wiley.com).

© 2009 Wiley Periodicals, Inc.

## 1. INTRODUCTION

Traditional facility location problems focus on determining a set of open facilities and assigning each customer to an open facility, where the goal is to minimize the sum of facility opening costs and connection costs of the customers to the facilities. When, as is often the case, the company needs to meet the demand of the customers in future time periods through production and inventory decisions at its facilities, using a traditional facility location model may be inappropriate because it cannot accurately represent all incurred costs. In particular, the connection costs often represent transportation costs only, and even if they attempt to capture production and inventory holding costs, this can only be done at a very coarse level. Therefore, using traditional facility location models may lead to very (and unnecessarily) high production and inventory costs. In this article, we introduce a new class of integrated facility location and production planning problems that generalizes traditional facility location problems by taking into account the demands of each customer in future time periods. Our goal is then to minimize the sum of facility opening costs, connection costs, and production/inventory costs. We will focus on developing approximation algorithms for this class of problems.

The class of problems studied in this article can be described as follows. We are given a set of  $m$  facilities, a set of  $n$  customers, and  $T$  time periods. The demand of customer  $j$  in period  $t$  is given by  $d_{jt}$  ( $j = 1, \dots, n; t = 1, \dots, T$ ). It will be convenient to also define the cumulative demand of customer  $j$  as  $d_j = \sum_{t=1}^T d_{jt}$ . We wish to assign a customer

$j$  to an open facility  $i$  and meet the demand of the customer through production and inventory decisions at the facility. There is a connection cost,  $c_{ij}$ , associated with facility  $i$  and customer  $j$ , which is expressed as a cost per unit of demand. We will assume that connection costs are symmetric, i.e., the cost of shipping from facility  $i$  to customer  $j$  is equal to the cost of shipping from customer  $j$  to facility  $i$ , and satisfy the triangle inequality, i.e., for any facilities  $i, i'$  and customers  $j, j'$ ,  $c_{ij} \leq c_{ij'} + c_{i'j} + c_{ij}$ . Although these assumptions on the connection costs are restrictive, they are common in developing constant factor approximation algorithms for facility location problems. Each facility has an opening cost of  $f_i$ , which we must pay if we assign any customers to the facility. Each facility  $i$  has a concave function representing the cost of producing  $p$  units in time period  $t$ ,  $P_{it}(p)$ , and a concave function representing the cost of holding  $I$  units in time period  $t$ ,  $H_{it}(I)$ . Our *uncapacitated facility location and production planning* (UFLPP) problem can be formulated as:

$$\begin{aligned} \text{minimize } & \sum_{i=1}^m \left( f_i y_i + \sum_{t=1}^T (P_{it}(p_{it}) + H_{it}(I_{it})) \right) \\ & + \sum_{i=1}^m \sum_{j=1}^n d_j c_{ij} x_{ij} \end{aligned}$$

subject to

$$\sum_{i=1}^m x_{ij} = 1 \quad \text{for all } j = 1, \dots, n \quad (1)$$

$$x_{ij} \leq y_i \quad \text{for all } i = 1, \dots, m; j = 1, \dots, n \quad (2)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i = 1, \dots, m; j = 1, \dots, n \quad (3)$$

$$y_i \in \{0, 1\} \quad \text{for all } i = 1, \dots, m \quad (4)$$

$$I_{i,t-1} + p_{it} = \sum_{j=1}^n d_j x_{ij} + I_{it} \quad \text{for all } i = 1, \dots, m; t = 1, \dots, T \quad (5)$$

$$I_{i0} = 0 \quad \text{for all } i = 1, \dots, m \quad (6)$$

$$p_{it}, I_{it} \geq 0 \quad \text{for all } i = 1, \dots, m; t = 1, \dots, T. \quad (7)$$

Constraints (1)–(4) are traditional facility location problems and constraints (5)–(7) are production planning constraints at each facility to ensure that, in each time period, we meet the demand of all customers assigned to the facility. If we know the set of customers assigned to a given facility, then we simply need to solve an uncapacitated production planning problem at that facility. Note that we have assumed that production and inventory variables are uncapacitated. If the production variables are capacitated at each facility, it becomes NP-hard to even determine if the UFLPP problem has a feasible solution. This can be seen since if  $T = 1$ , then the UFLPP problem is a capacitated facility location problem with unsplittable demands. It is relatively straightforward to show that the Partition problem has a solution if and only if an associated capacitated facility location problem

with unsplittable demands has a feasible solution. Therefore, a study of approximation algorithms for the UFLPP problem should either focus on models where the production and inventory variables are uncapacitated or constraints (3) are relaxed to allow for splittable demands. In this paper, we focus on the former class of problems.

The field of designing constant factor approximation algorithms for facility location problems and variants has been extremely active since Shmoys et al. [22] gave the first constant factor guarantee for the metric uncapacitated facility location (UFL) problem. Their algorithm used the idea of rounding the optimal solution to a linear programming relaxation of the facility location problem. Sviridenko [23] used an LP-rounding technique to develop an approximation algorithm with a factor of 1.58. Recently, Byrka [3] used LP-rounding, the algorithm of Chudak and Shmoys [5], and the algorithm of Mahdian et al. [17] to achieve an algorithm with a factor of 1.50. These algorithms have high running times because they must solve a linear program. Primal-dual algorithms for the UFL problem are computationally attractive and can achieve similar approximation guarantees. Jain et al. [13] offer primal-dual algorithms with approximation guarantees of 1.861 and 1.61. The current best-known approximation factor for the metric UFL problem using a primal-dual algorithm achieves a guarantee of 1.52 and is due to Mahdian et al. [17]. This algorithm uses the 1.61-approximation algorithm of Jain et al. [13] and the idea of scaling (see Charikar and Guha [4]). Guha and Khuller [11] show that it is not possible to achieve an approximation guarantee of better than 1.463 for the UFL problem unless  $NP \subseteq TIME[O(n^{\log \log n})]$ .

In Section 3, we will show that an important subclass of the UFLPP problem can be formulated as a generalization of the UFL problem where the facility costs are general concave functions of the amount of demand assigned to the facility. We will refer to this problem as the *concave cost facility location* (CCFL) problem. Hajiaghayi et al. [12] generalize the 1.861-approximation algorithm of Jain et al. [13] for the UFL problem to the special case of the CCFL problem in which each customer has unit demand; this algorithm runs in  $O(n^3 \log n)$  time. In addition, they show that the CCFL problem with unit demands can be converted to a UFL problem with  $n$  customers and  $nm$  facilities and then use the algorithm of Mahdian et al. [17] to obtain a 1.52-approximation algorithm that runs in  $O(n^6)$  time. For the CCFL problem with general integral demand, this approach has a pseudopolynomial running time. In particular, the UFL problem would have  $O(nD)$  customers and  $O(nmD)$  facilities where  $D$  is equal to the sum of the demands of the customers. Further, for the CCFL problem with general integral demand, we can develop a  $(1.52 + \epsilon)$ -approximation algorithm by approximating the concave functions with piecewise linear functions and using a similar reduction as Hajiaghayi et al. [12] to the UFL problem. However, this approach has a running time of  $O\left(n^3 \left(\frac{\ln D}{\ln(1+\epsilon)}\right)^3\right)$ , which is dependent on both  $D$  and on  $\epsilon$ . In this paper, we provide a generalization

TABLE 1. Relevant approximation algorithms for facility location problems.

Reference	Problem	Approximation factor	Running time
Jain et al. [13]	UFL	1.861	$O(n^2 \log n)$
Jain et al. [13]	UFL	1.61	$O(n^3)$
Mahdian et al. [17]	UFL	1.52	Quasi-linear
Hajiaghayi et al. [12]	CCFL with unit demands	1.861	$O(n^3 \log n)$
Hajiaghayi et al. [12]	CCFL with unit demands	1.52	$O(n^6)$
Consequence of [12]	CCFL with general demands	1.52	Pseudo-polynomial
Consequence of [12]	CCFL with general demands	$1.52 + \epsilon$	$O\left(n^3 \left(\frac{\ln D}{\ln(1+\epsilon)}\right)^3\right)$
This paper	CCFL with general demands	1.52	$O(n^4 \log n)$

of the 1.61-approximation algorithm of Jain et al. [13] for the UFL problem to the CCFL problem with general integral (or, as we discuss in Section 3.1, rational) demands that runs in  $O(n^4 \log n)$  time. We then use this algorithm and a scaling idea to generalize the 1.52-approximation algorithm of Mahdian et al. [17] to the CCFL problem that also runs in  $O(n^4 \log n)$  time. Independent of this work, Magnanti and Stratila [16] provided a 1.61-approximation algorithm for the CCFL problem with general integral demands with a running time of  $O(n^4)$ . A summary of past approximation algorithms for facility location problems that are relevant to the work in this paper appears in Table 1.

Given a set of customers assigned to a facility in the UFLPP problem, we must manage the production and inventory decisions at the facility to ensure we meet the cumulative demand of the set of customers assigned to the facility in each time period. The production planning problem that is faced by a facility is a generalization of the classical economic lot-sizing problem (see Wagner and Whitin [27]) where concave production cost functions replace fixed-charge plus linear production costs and concave holding cost functions replace linear holding costs. This production planning problem can be solved in  $O(T^2)$  time (see Wagner [26] and Veinott [24]). The economic lot-sizing problem can be solved in  $O(T \log T)$  time and can be solved in  $O(T)$  time in the case of non-speculative motives (see Aggarwal and Park [1], Federgruen and Tzur [7], and Wagelmans et al. [25]). Krarup and Bilde [14] used a primal-dual algorithm to show that the facility location formulation of the economic lot-sizing problem yielded an integral solution. Levi et al. [15] developed primal-dual algorithms for three important classes of deterministic inventory problems, including the economic lot-sizing problem. Their primal-dual algorithm solved the economic lot-sizing problem to optimality. They also developed a 2-approximation algorithm for the joint-replenishment problem (see Zangwill [28], Veinott [24], and Arkin et al. [2]) and a 2-approximation algorithm for the multistage assembly problem (see Roundy [19]).

The remainder of this article is organized as follows. In Section 2 of this article, we show that the UFLPP problem is as hard as the set cover problem and conclude that, in general, it is highly unlikely that a constant-factor approximation algorithm exists. Therefore, we will focus on approximating special cases of the UFLPP problem. The cases that we

will study fall into two categories: (i) specializing the production and inventory cost structure at the facilities (Section 2) and (ii) specializing the demand pattern of the customers (Section 3). For problems belonging to (i), we will offer reductions from the UFLPP problem to previously studied variants of the metric UFL problem. For the problems in (ii), we will show that they belong to the CCFL problem with general rational demand. We will develop an approximation algorithm with a guarantee of 1.52 for the CCFL problem with general rational demand.

## 2. APPROXIMATING THE UFLPP PROBLEM WITH GENERAL DEMANDS

Before developing an approximation algorithm for the UFLPP problem, it is necessary to determine the complexity of approximating the UFLPP problem. We begin with a result that shows that the UFLPP problem is as hard as the Set Cover problem. The definition of the Set Cover problem is: Given a collection of items  $1, \dots, n$  and a collection of sets  $S_i \subseteq \{1, \dots, n\}$  with associated cost  $f_i$  for  $i = 1, \dots, m$ , determine a minimum cost set  $\mathcal{S} \subseteq \{S_i : i = 1, \dots, m\}$  such that for each  $j \in \{1, \dots, n\}$  there exists  $S_i \in \mathcal{S}$  such that  $j \in S_i$ .

**Theorem 2.1.** *If there exists an  $\alpha$ -approximation algorithm for the UFLPP problem, then there also exists an  $\alpha$ -approximation algorithm for the Set Cover problem.*

**Proof.** Consider an instance of the Set Cover problem, and define an instance of the UFLPP problem as follows.

- The set of customers is  $\{1, \dots, n\}$  and the set of facilities is  $\{1, \dots, m\}$ .
- Set  $c_{ij} = 0$  for all facility/customer pairs  $(i, j)$  and the facility opening costs to  $f_i$ .
- Set the number of time periods to be  $T = n$  and set the demand of customer  $j$  to be  $d_{jt} = 0$  if  $j \neq t$  and  $d_{jt} = 1$  if  $j = t$ .
- Set the inventory cost function for each time period at each facility to be  $H_{it}(I_{it}) = I_{it} \cdot \max_{i=1, \dots, m}(f_i + 2)$ .
- Define the production functions as  $P_{it}(p_{it}) = 0$  if  $t = j \in S_i$  and  $P_{it}(p_{it}) = p_{it} \cdot \max_{i=1, \dots, m}(f_i + 1)$  if  $t = j \notin S_i$ .

The inventory holding costs ensure that it is most cost-effective to satisfy demand in period  $t$  with production in

period  $t$ . This implies that if we assign customer  $j$  to facility  $i$ , it will (optimally) cost  $P_{it}(1)$ . It is easy to see that it is always cheaper to assign customer  $j$  to a facility  $i$  such that  $j \in S_i$  than to assign customer  $j$  to a facility  $i$  where  $j \notin S_i$ . Therefore, an optimal set of open facilities in the UFLPP problem will be a minimum cost cover. Now if we have an  $\alpha$ -approximation algorithm for the UFLPP problem, we can easily convert the solution returned by the algorithm to a feasible cover with a cost no worse than the returned solution. This then would yield an  $\alpha$ -approximation algorithm for the Set Cover problem. ■

As Feige [8] showed a hardness result about approximating the set cover problem, Theorem 2.1 implies that we cannot develop an approximation algorithm with a guarantee of better than  $(1 - \epsilon) \log n$  unless  $NP \subseteq TIME[O(n^{\log \log n})]$ . Therefore, it is interesting to identify special cases of the UFLPP problem that can be approximated to within a constant factor. The following two theorems deal with two different cost structures for the UFLPP problem for which this is the case. Both results assume that the production cost functions and inventory cost functions are linear and can be written as  $P_{it}(p_{it}) = b_{it}p_{it}$  ( $i = 1, \dots, m; t = 1, \dots, T$ ) and  $H_{it}(I_{it}) = h_{it}I_{it}$  ( $i = 1, \dots, m; t = 1, \dots, T$ ) respectively. The first result, in addition, assumes that the production and inventory holding costs do not depend on the facility.

**Theorem 2.2.** *If there exists an  $\alpha$ -approximation algorithm for the metric UFL problem, then there exists an  $\alpha$ -approximation algorithm for the class of instances of the UFLPP problem with linear and facility-invariant production and inventory holding costs.*

**Proof.** We will first define a UFL problem based on this class of instances of the UFLPP problem. In this conversion, the facility opening costs will remain the same as the facility opening costs in the UFLPP problem. We will focus on defining the connection cost of a customer to a facility. Denote the unit production and inventory holding costs by  $b_{it} = b_t$  and  $h_{it} = h_t$  for  $i = 1, \dots, m$  and  $t = 1, \dots, T$ , respectively. First, as production and inventory costs are linear and production/inventory is uncapacitated, we can determine the optimal cost of meeting demand in period  $t$ , which we denote  $C_t^*$ , through a simple recursion:

$$C_t^* = \min \{b_t, h_{t-1} + C_{t-1}^*\}.$$

Once we have  $C_t^*$  for  $t = 1, \dots, T$ , the optimal production/inventory costs of customer  $j$  are given by  $\sum_{t=1}^T d_{jt}C_t^*$ . It is clear that if we are given a solution to the UFLPP problem where the production and inventory costs corresponding to customer  $j$  are higher than  $\sum_{t=1}^T d_{jt}C_t^*$ , then this solution cannot be optimal since we can lower the production/inventory costs. Therefore, to solve UFLPP problem, we can restrict ourselves to solutions where the production and inventory costs are  $\sum_{t=1}^T d_{jt}C_t^*$ .

We define a UFL problem by defining the demands to be equal to the aggregate demands  $d_j$  and the connection costs

to be  $\bar{c}_{ij} = c_{ij} + \gamma_j$  where  $\gamma_j = \sum_{t=1}^T d_{jt}C_t^*/d_j$ . Now consider any pair of facilities  $i, i'$  and pair of customers  $j, j'$ :

$$\bar{c}_{ij} = c_{ij} + \gamma_j \leq c_{ij} + c_{i'j} + c_{ij'} + \gamma_j \leq \bar{c}_{i'j} + \bar{c}_{ij'} + \bar{c}_{ij'}$$

so that the connection costs satisfy the triangle inequality. Given an assignment of customers to facilities in this facility location problem, consider the assignment costs:

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n d_j \bar{c}_{ij} x_{ij} &= \sum_{i=1}^m \sum_{j=1}^n d_j (c_{ij} + \gamma_j) x_{ij} \\ &= \sum_{i=1}^m \sum_{j=1}^n d_j c_{ij} x_{ij} + \sum_{j=1}^n d_j C_t^* x_{ij}, \end{aligned}$$

which are the connection costs plus the optimal production/inventory costs in the same assignment of customers to facilities in the UFLPP problem. Similarly, given an assignment of customers to facilities in the UFLPP problem, the connection costs plus the optimal production/inventory costs are the same as the assignment costs in the facility location problem. Therefore, this metric UFL problem solves this special class of the UFLPP problem and the parameters of the problem can be determined in  $O(nT + nm)$  time. ■

The second result again deals with instances for which the production cost and holding cost functions are linear. However, we now assume that an ordering of the facilities exists such that, in every time period, it is as cost-effective to produce (or hold) a unit of demand at facility  $i$  than it is at facility  $i' > i$ . Note that we do *not* make any assumptions regarding the facility opening costs, so it may be very expensive to open a facility with cheap production/inventory costs and very cheap to open a facility with expensive production/inventory costs.

**Theorem 2.3.** *There exists a 6-approximation algorithm for the class of instances of the UFLPP problem with linear production and inventory holding costs if there exists an ordering of the facilities such that, if  $i$  and  $i'$  are two facilities, then  $i < i'$  implies that for all  $t$ ,  $b_{it} \leq b_{i't}$  and  $h_{it} \leq h_{i't}$ .*

**Proof.** We will convert this class of the UFLPP problem to a facility location problem with service installation costs. This problem generalizes the facility location problem where each customer  $j$  has an associated service  $g(j)$ . We can install service  $g(j)$  at facility  $i$  at a cost of  $f_i^{g(j)}$ . We then wish to assign customers to facilities while minimizing the sum of the facility opening costs, service installation costs, and connection costs. Shmoys et al. [21] offer a primal-dual approximation algorithm for this problem with an approximation guarantee of 6, when there exists an ordering of the facilities such that if  $i < i'$  then for any service  $\ell$ , we have  $f_i^\ell \leq f_{i'}^\ell$ . Note that this ordering makes no assumptions on the values of the facility opening costs.

In our conversion, the facility opening costs and connection costs remain the same as in the UFLPP problem. We

now describe how to determine the service installation costs at each facility. We can determine the optimal cost of meeting demand in period  $t$  at facility  $i$ ,  $C_{it}^*$ , through a similar recursion as in the proof of Theorem 2.2 above. It is clear that if  $i < i'$  then  $C_{it}^* \leq C_{i't}^*$  for all  $t = 1, \dots, T$ . Now define the (optimal) production/inventory costs associated with assigning customer  $j$  to facility  $i$ :

$$f_i^j = \sum_{t=1}^T C_{it}^* d_{jt}.$$

Similar to the proof of Theorem 2.2, we can restrict our search for the optimal solution to the UFLPP problem to solutions where the production/inventory costs of customer  $j$  are equal to  $f_i^j$  if  $j$  is assigned to facility  $i$ . We define  $g(j) = j$  and require that service  $g(j)$  must be installed at the facility to which customer  $j$  is assigned. It is clear that if we assign a customer  $j$  to a facility  $i$  that the service installation cost,  $f_i^j$ , is the optimal production/inventory costs for meeting the demand of customer  $j$  at facility  $i$ . Similarly, if customer  $j$  is assigned to facility  $i$  in a solution to the (UFLPP) problem, it is clear that optimal production/inventory costs associated with this assignment will equal the service installation cost associated with customer  $j$ . This conversion can be done in  $O(nmT)$  time and we can apply the algorithm of Shmoys et al. [21] to approximate this special case of the UFLPP problem within a factor of 6. ■

### 3. APPROXIMATING THE UFLPP PROBLEM WITH SEASONAL DEMANDS

In this section, we will study the important subclass of the UFLPP problem in which the demands follow a *seasonal pattern*. In particular, we say that a problem in the class has seasonal demands if each customer's aggregate demand is distributed among the  $T$  periods using a common (nonnegative) vector of multiplicative seasonal effects,  $\sigma^\top = (\sigma_1, \dots, \sigma_T)$ , so that the demand of customer  $j$  in period  $t$  is given by  $d_{jt} = \sigma_t d_j$ . For convenience and without loss of generality we will assume that the seasonal effects are normalized so that  $\sum_{t=1}^T \sigma_t = 1$ .

Consider the optimal production and inventory holding costs required at facility  $i$  to meet a vector of demands equal to a nonnegative scalar multiple  $z$  of the vector of seasonal effects  $\sigma$  as given by the optimal solution value of the following optimization problem:

$$\text{minimize } \sum_{t=1}^T (P_{it}(p_{it}) + H_{it}(I_{it}))$$

subject to (PP( $i$ ))

$$\begin{aligned} I_{i,t-1} + p_{it} &= \sigma_t z + I_{it} \quad \text{for } t = 1, \dots, T \\ I_{i0} &= 0 \\ p_{it}, I_{it} &\geq 0 \quad \text{for } t = 1, \dots, T. \end{aligned}$$

We denote the optimal value function of this problem by  $g_i(z)$ . It is clear that we can restrict our search for the optimal solution to the UFLPP problem to solutions of the UFLPP problem where the production and inventory costs at a facility are equal to  $g_i(\sum_{j=1}^n d_j x_{ij})$  since otherwise we could improve the cost of the current solution by altering the production and inventory variables. With a slight abuse of notation, we define the function

$$f_i(z) = \begin{cases} f_i + g_i(z) & \text{if } z > 0 \\ 0 & \text{if } z = 0. \end{cases}$$

We can then represent the total facility opening costs and optimal production and inventory costs of the UFLPP problem with seasonal demands as

$$\sum_{i=1}^m f_i \left( \sum_{j=1}^n d_j x_{ij} \right).$$

Let us now consider the connection costs associated with the UFLPP problem with seasonal demands. We have:

$$\begin{aligned} \sum_{j=1}^n \sum_{i=1}^m \left( \sum_{t=1}^T d_{jt} \right) c_{ij} x_{ij} &= \sum_{j=1}^n \sum_{i=1}^m \left( d_j \sum_{t=1}^T \sigma_t \right) c_{ij} x_{ij} \\ &= \sum_{j=1}^n \sum_{i=1}^m d_j c_{ij} x_{ij} \end{aligned}$$

since  $\sum_{t=1}^T \sigma_t = 1$ . Recalling that the connection costs  $c_{ij}$  are metric, the UFLPP problem is thus an instance of the following class of generalized metric facility location problems:

$$\text{minimize } \sum_{i=1}^m f_i \left( \sum_{j=1}^n d_j x_{ij} \right) + \sum_{j=1}^n \sum_{i=1}^m d_j c_{ij} x_{ij}$$

subject to (P)

$$\sum_{i=1}^m x_{ij} = 1 \quad \text{for all } j = 1, \dots, n$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i = 1, \dots, m; j = 1, \dots, n.$$

The following lemma shows that in any instance of (P) resulting from the UFLPP problem with seasonal demands the functions  $f_i$  are concave by showing that the functions  $g_i$  are concave.

**Lemma 3.1.** *The functions  $g_i$ ,  $i = 1, \dots, m$ , are concave.*

**Proof.** It is well known that the only candidate solutions that need to be considered for (PP( $i$ )) are those that are characterized by a set of periods in which production takes place, together with the so-called zero-inventory property that says that the ending inventory in a period preceding a production period is equal to zero. It is easy to see that, given a particular choice of production periods, the value of (PP( $i$ )) as a

function of  $z$  is a concave function, so that  $g_i$  is the minimum of a family of concave functions, which implies that  $g_i$  itself is concave. ■

As mentioned in the introduction, we will refer to the problem class (P) with concave functions  $f_i$  as the concave cost facility location (CCFL) problem. Before studying this problem in more detail, note that it has other applications besides the UFLPP problem with seasonal demands. For example, Shen et al. [20] and Daskin et al. [6] consider a joint inventory-location problem, special cases of which belong to the class of the CCFL problem. In Section 3.1, we will develop a 1.61-approximation algorithm for the CCFL problem and use this algorithm and the idea of cost-scaling to ultimately give a 1.52-approximation algorithm for it. In Section 3.2, we examine the problem (P) for other structures of the functions  $f_i$ ,  $i = 1, \dots, m$ .

### 3.1. Approximation Algorithms for the CCFL Problem

Before we begin developing an approximation algorithms for the CCFL problem, it will be necessary to define the time required to evaluate the function  $f_i(z)$  for some fixed value of  $z$ .

**Definition 3.2.** We let  $\phi$  denote the time required to evaluate the function  $f_i(z)$  for a fixed  $z$ .

It is important to note that this evaluation may not be trivial for all concave functions  $f_i(z)$ . For example, in the CCFL problem that arises from the UFLPP problem with seasonal demands, we have that  $\phi = O(T^2)$  for general concave production and holding cost functions and  $\phi = O(T \log T)$  for fixed-charge plus linear production costs and linear holding costs (the cost structure of the economic lot-sizing problem).

Further, for the remainder of this section, we will assume that the demand level of the customers ( $d_j$  for  $j = 1, \dots, n$ ) are integral. We can make this assumption without loss of generality for the CCFL problem with rational demand levels, since, if the demand levels are rational, we can multiply the demand levels of the customers by a large integer while dividing the connection costs and facility cost functions by the same large integer. In almost all situations, the base demand level of each customer will be rational, so, therefore, the integrality assumption is not very restrictive.

We will begin by generalizing the 1.61-approximation algorithm of Jain et al. [13]. We now informally describe this algorithm. At any point in the algorithm we will have two sets of customers: connected customers and unconnected customers. Each customer will then make an offer to each facility. The offer of a connected customer to a facility is equal to the amount the customer would save in paying the connection to this facility as opposed to paying the connection cost to the facility it is currently assigned. The offer of an unconnected customer to a facility will be based on the customer's budget and the connection cost to this facility. If a set of customers offer enough to cover the opening cost of

a facility, the facility will open and each customer in the set will be assigned to it. If this event occurs, then the amount offered to the facility by a customer can be thought of as the customer's contribution to the opening of the facility. If no facility is offered enough to open, we raise the budget of each of the unconnected customers.

There are two crucial elements to this generalization that differ from the algorithm of Jain et al. [13]: (i) the idea of "contribution withdrawal," i.e., if a customer switches facilities, it withdraws some of its contribution to the facility it is initially connected to and offers it to the facility it switches to and (ii) efficiently solving a nonlinear fractional binary programming problem that is necessary at each iteration of the algorithm. We then use this algorithm to generalize the two-phase algorithm of Mahdian et al. [17] to the CCFL problem, deriving an approximation algorithm for the CFL problem with a guarantee of 1.52.

We will now describe the notation used in our algorithm to approximate the CCFL problem. Let  $D = \{1, \dots, n\}$  be the set of customers. At any point in the algorithm,  $U \subseteq \{1, \dots, n\}$  will denote the set of unconnected customers. Moreover, for any facility  $i$ ,  $A_i$  will denote the set of customers assigned to it by the algorithm so far, and  $T_i = \sum_{j \in A_i} d_j$  will be the corresponding total demand currently assigned. There is a notion of time  $\tau$  associated with the algorithm. For any  $j \in U$ , we set the budget of customer  $j$  at time  $\tau$  equal to  $\alpha_j = \tau$ . Each customer will offer some money from its budget to a facility  $i$ , which we denote  $o_{ji}$ . The offer of customer  $j$  to facility  $i$  (where  $j \notin A_i$ ) depends on whether customer  $j$  has been assigned to a facility earlier than time  $\tau$ . In particular, at time  $\tau$ , if  $j \in U$ , then  $o_{ji} = d_j \max\{\alpha_j - c_{ij}, 0\}$ ; if  $j \in A_{i'}$ , then  $o_{ji} = d_j \max\{w_{ji'} + c_{ij} - c_{ij}, 0\}$  where  $w_{ji'} = (f_{i'}(T_{i'}) - f_{i'}(T_{i'} - d_j))/d_j$ . If  $j \in A_{i'}$ , then  $w_{ji'}$  is the amount of customer  $j$ 's contribution to facility  $i'$  that would be withdrawn from facility  $i'$  if customer  $j$  switches facilities. Note that the introduction of  $w_{ji'}$  is significantly different than the algorithm of Jain et al. [13], which does not allow a customer to withdraw some of its contribution from the facility it is assigned. The idea behind the algorithm is to, as time progresses, assign a subset of customers to a facility if the total offer by this set covers the additional facility costs. We are now in a position to present the algorithm.

---

### Greedy Algorithm

---

**Step 0:** Set  $U = \{1, \dots, n\}$ ,  $A_i = \emptyset$ , and  $T_i = 0$  for all  $i = 1, \dots, m$ . Initialize  $\tau = 0$ .

**Step 1:** If  $U = \emptyset$ , terminate the algorithm. Otherwise, increase  $\tau$  until there exists a facility  $i$  and a set  $S \subseteq D \setminus A_i$  such that

$$\sum_{j \in S} o_{ji} = f_i \left( T_i + \sum_{j \in S} d_j \right) - f_i(T_i).$$

**Step 2:** Connect the clients in  $S$  to facility  $i$ , setting  $A_i = A_i \cup S$  and  $T_i = T_i + \sum_{j \in S} d_j$ . For each  $j \in S \cap U$ , freeze  $\alpha_j = \tau$ . Set  $U = U \setminus S$ . For each  $i' \neq i$ , set  $A_{i'} = A_{i'} \setminus S$  and  $T_{i'} = \sum_{j \in A_{i'}} d_j$ . Return to Step 1.

---

We will now show that the set  $S$  in Step 1 (and thereby the next time at which the Greedy Algorithm assigns a set of customers to a facility) can be found by solving a particular optimization problem.

**Lemma 3.3.** *The set  $S$  in Step 1 (and thereby the next time at which the Greedy Algorithm assigns a set of customers to a facility) can be found by solving the following minimization problem for each  $i = 1, \dots, m$ :*

$$\min_{S_1 \in D \setminus (A_i \cup U), S_2 \in U} \frac{f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i) + \sum_{j \in S_1 \cup S_2} d_j a_j}{\sum_{j \in S_2} d_j} \quad ((FP(i)))$$

where  $a_j = c_{ij}$  if  $j \in U$  and  $a_j = -(c_{ij} + w_{j'j} - c_{ij})$  if  $j \in A_i$ .

**Proof.** Suppose that we arrive at Step 1 in the algorithm. We wish to determine the next time in which a set of customers (potentially) would be assigned to facility  $i$ . We have two types of customers that may be assigned to facility  $i$ : (i) customers currently connected to other facilities and (ii) customers currently unconnected. Let  $S_1 \subseteq D \setminus (A_i \cup U)$  be a set of customers currently connected to other facilities and  $S_2 \subseteq U$  be a set of customers currently unconnected. To determine the next time an assignment would occur at facility  $i$ , we can solve

$$\text{minimize } \tau$$

subject to (SP(i))

$$\begin{aligned} & \sum_{j \in S_1} d_j \max\{w_{j'j} + c_{ij} - c_{ij}, 0\} + \sum_{j \in S_2} d_j \max\{\tau - c_{ij}, 0\} \\ & = f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i) \end{aligned} \quad (8)$$

$$S_1 \subseteq D \setminus (A_i \cup U)$$

$$S_2 \subseteq U,$$

since (8) ensures that the amount offered to a facility is equal to the additional amount incurred by the facility in serving customers in  $S_1 \cup S_2$ . As we are minimizing  $\tau$ , we can disregard solutions to this problem where we select a customer that does not offer anything to facility  $i$ . In other words, for potentially optimal solutions to (SP(i)), (8) can be rewritten as

$$\begin{aligned} & \sum_{j \in S_1} d_j (w_{j'j} + c_{ij} - c_{ij}) + \sum_{j \in S_2} d_j (\tau - c_{ij}) \\ & = f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i). \end{aligned}$$

If we let  $a_j = -(w_{j'j} + c_{ij} - c_{ij})$  if  $j \in D \setminus (A_i \cup U)$  and  $a_j = c_{ij}$  if  $j \in U$ , then we see that (8) can be written as

$$\sum_{j \in S_2} d_j \tau = f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i) + \sum_{j \in S_1 \cup S_2} a_j d_j.$$

This means that we have a closed form expression for  $\tau$  given  $S_1$  and  $S_2$  by dividing the previous equation by  $\sum_{j \in S_2} d_j$ . Therefore, we can write (SP(i)) as a problem (FP(i)):

$$\min_{S_1 \in D \setminus (A_i \cup U), S_2 \in U} \frac{f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i) + \sum_{j \in S_1 \cup S_2} d_j a_j}{\sum_{j \in S_2} d_j}.$$

If we let  $\tau_i$  be the optimal solution to (SP(i)) or, equivalently, the optimal solution value of (FP(i)), then the next time that any set of customers can be connected to some facility will be  $\tau = \min_{i=1, \dots, m} \tau_i$ . Therefore, by solving (FP(i)) for facilities  $i = 1, \dots, m$ , we can determine the next time an event occurs. ■

Lemma 3.3 shows that it suffices to solve the problems (FP(i)) for  $i = 1, \dots, m$  to implement the Greedy Algorithm. Our next result deals with the complexity of solving a problem of the form (FP(i)).

**Lemma 3.4.** *The problem (FP(i)) can, for a given facility  $i$ , be solved in  $O(n^2 \max\{\log n, \phi\})$  time, where the time required to evaluate  $f_i(z)$  for a given  $z$  is  $\phi$ .*

**Proof.** Let  $\lambda^*$  be the value of the optimal solution to (FP(i)). It can be shown that  $\lambda^*$  is equal to the value of  $\lambda$  for which there exists a nontrivial, i.e.,  $S_1 \cup S_2 \neq \emptyset$ , optimal solution with value 0 to the following optimization problem:

$$\begin{aligned} & \min_{S_1 \subseteq D \setminus (A_i \cup U), S_2 \subseteq U} f_i \left( T_i + \sum_{j \in S_1 \cup S_2} d_j \right) - f_i(T_i) \\ & \quad + \sum_{j \in S_1} d_j a_j + \sum_{j \in S_2} d_j (a_j - \lambda). \end{aligned} \quad ((KP(\lambda)))$$

Note further that the optimal solution of (FP(i)) is equal to a nontrivial optimal solution of (KP( $\lambda^*$ )). We will use this fact, and the structure of the maximal optimal solution (i.e., the optimal solution where  $|S_1 \cup S_2|$  is largest), to develop an efficient algorithm to solve (FP(i)).

Suppose that we wish to solve (KP( $\lambda$ )) for a fixed  $\lambda$ . If we index the customers in  $\{1, \dots, n\} \setminus A_i$  in nondecreasing order of  $\bar{a}_j(\lambda)$ , where  $\bar{a}_j(\lambda) = a_j - \lambda$  if  $j \in U$  and  $\bar{a}_j(\lambda) = a_j$  if  $j \notin U$ , it can be shown that there exists an optimal solution that selects the first  $\ell$ , for some  $\ell = 1, \dots, n - |A_i|$ , customers in the order (see Shen et al. [20]). It can also be shown that if  $\bar{a}_j(\lambda) = \bar{a}_{j'}(\lambda)$  for customers  $j, j'$ , then there exists an optimal solution to (KP( $\lambda$ )) where we select both  $j$  and  $j'$  or we select neither and therefore we can view  $j, j'$  as a single entity. To solve (FP(i)), it is sufficient to know the ordering of the customers based on  $\bar{a}_j(\lambda^*)$ , since we may then evaluate  $n$  potential solutions to (FP(i)) to determine the optimal solution. Therefore, we turn our attention to determining the number of distinct orderings of the customers and, more importantly, the number of distinct candidate solutions arising from the orderings.

First, note that the ordering of the customers  $j \in D \setminus (A_i \cup U)$  is independent of  $\lambda$ , i.e., it is only based on  $a_j$ . Similarly,

the ordering of customers  $j \in U$  is independent of  $\lambda$ . As we increase  $\lambda$ , new orderings only arise when a customer in  $j \in D \setminus (A_i \cup U)$  and a customer  $j \in U$  switch places. If there exist  $j, j' \in D \setminus (A_i \cup U)$  (or  $j, j' \in U$ ) such that  $a_j = a_{j'}$ , then  $\bar{a}_j(\lambda) = \bar{a}_{j'}(\lambda)$  for any  $\lambda$ . Therefore, as mentioned earlier, we may essentially merge customers  $j$  and  $j'$ , meaning that we will either select them both or select neither in evaluating candidate optimal solutions to (FP( $i$ )). This immediately yields a bound of  $O((n - |A_i| - |U|)|U|) = O(n^2)$  on the number of distinct orderings of the variables. Without further analysis, this leads to  $O(n^3)$  candidate solutions for (FP( $i$ )) by evaluating the solutions where we choose the first  $\ell = 1, \dots, n - |A_i|$  customers in each ordering. It turns out, however, that many of these candidate solutions are counted multiple times in this coarse analysis.

A deeper analysis of the number of candidate solutions that need to be considered to solve (FP( $i$ )) is based on the following observations. Let  $I(\lambda)$  be the indexing of the customers based on  $\bar{a}_j(\lambda)$  and suppose that we know all the candidate solutions based on this ordering. Let  $\bar{\lambda}$  be the next value that changes the indexing of the customers. If exactly two customers have swapped places in going from  $I(\lambda)$  to  $I(\bar{\lambda})$ , we need to only evaluate a single new candidate solution. In particular, if customers  $\ell$  and  $\ell + 1$  swapped places, then we only need to consider the set consisting of the first  $\ell - 1$  customers with the  $(\ell + 1)$ st customer in  $I(\lambda)$ . Therefore, it is necessary to evaluate a number of candidate solutions equal to the number of swaps from  $I(\lambda)$  to  $I(\bar{\lambda})$  to evaluate all new candidate solutions in  $I(\bar{\lambda})$ . Further, if customers  $\ell$  and  $\ell + 1$  swapped places, to evaluate the the new candidate solution based on this swap, we need to know the sums  $\sum_{j=1}^{\ell-1} d_j + d_{\ell+1}$  and  $\sum_{j=1}^{\ell-1} a_j d_j + a_{\ell+1} d_{\ell+1}$ . Once we know these sums, we need to evaluate the function

$$f_i \left( T_i + \sum_{j=1}^{\ell-1} d_j + d_{\ell+1} \right)$$

to evaluate the candidate solution. Therefore, if we know the partial sums  $\sum_{j=1}^{j'} d_j$  and  $\sum_{j=1}^{j'} a_j d_j$  for  $j' = 1, \dots, n$  before the swap, we can evaluate the candidate solution in  $O(\phi)$  time. Note further that we can update the new partial sums in  $O(1)$  time. Therefore, given an ordering and all the partial sums corresponding to this ordering, if we perform a swap, we can evaluate a candidate solution and update the necessary partial sums based on this swap in  $O(\phi)$  time.

The algorithm to solve (FP( $i$ )) is then as follows. We view any  $j, j' \in D \setminus (A_i \cup U)$  (or  $j, j' \in U$ ) with  $a_j = a_{j'}$  as a single entity. First, for each  $j \in U$  and  $j' \in D \setminus (A_i \cup U)$ , we determine the value of  $\lambda_{jj'}$  where  $\bar{a}_j(\lambda_{jj'}) = \bar{a}_{j'}(\lambda_{jj'})$ . It is only necessary to consider values of  $\lambda_{jj'}$  that are non-negative. We sort these values in nondecreasing order, which requires  $O(n^2 \log n)$  time. Then the actual algorithm starts by determining the indexing of the customers based on the values  $\bar{a}_j(0)$  and evaluating the  $O(n)$  corresponding candidate solutions. As we evaluate each of these candidate solutions, we record the sum of the demands for the first  $\ell$  customers as well as the sum of

the  $a_j d_j$ -values for the first  $\ell$  customers, for all values of  $\ell$ . At this point, we have a particular customer indexing and have evaluated all candidate solutions according to this indexing. We then continue the algorithm by examining the minimum value of  $\lambda_{jj'}$  not yet considered. This yields a new indexing by swapping the places of customer  $j'$  and  $j''$  for each pair  $j', j''$  that achieves the minimum value. This leads to a number of new candidate solutions that must be evaluated (in particular, one for each pair  $j', j''$  for which  $\lambda_{jj'}$  attains the minimum). As we have recorded the partial sums of demands and  $a_j d_j$ -values we can evaluate the new candidate solution and update the sums appropriately in  $O(\phi)$  time. As we raise  $\lambda$ , each new candidate solution that must be evaluated is induced by a swapping of customers. As there are  $O(n^2)$  such swaps, the algorithm requires  $O(n^2 \max\{\log n, \phi\})$  time. ■

As the Greedy Algorithm can be implemented in a way that solves (FP( $i$ )) for  $i = 1, \dots, m$  at each iteration, and since we assign at least one unconnected customer in each iteration, Lemma 3.4 implies a running time of  $O(mn^3 \max\{\log n, \phi\})$ . We will now focus on determining the approximation guarantee of the algorithm. This analysis generalizes the algorithm for the metric uncapacitated facility location problem with an approximation guarantee of 1.61 presented in Jain et al. [13]. We begin with a property of the variables  $\alpha_j$ .

**Lemma 3.5.** *The total cost of the solution produced by the Greedy Algorithm is no more than  $\sum_{j=1}^n d_j \alpha_j$ .*

**Proof.** At some point in time, consider facility  $i$  and  $A_i$ . We will first show that the total savings in disconnecting any subset  $S \subseteq A_i$  is at least  $\sum_{j \in S} d_j (w_{ji} + c_{ij})$ . By the definition of  $w_{ji}$  and the concavity of  $f_i$ , we have

$$\begin{aligned} d_j w_{ji} &= f_i(T_i) - f_i(T_i - d_j) \\ &\leq \frac{d_j}{\sum_{j \in S} d_j} \left( f_i(T_i) - f_i \left( T_i - \sum_{j \in S} d_j \right) \right) \end{aligned}$$

implying

$$\begin{aligned} \sum_{j \in S} d_j (w_{ji} + c_{ij}) &\leq f_i(T_i) - f_i \left( T_i - \sum_{j \in S} d_j \right) + \sum_{j \in S} d_j c_{ij} \\ &= f_i(T_i) + \sum_{j \in A_i} d_j c_{ij} - f_i \left( T_i - \sum_{j \in S} d_j \right) - \sum_{j \in A_i \setminus S} d_j c_{ij} \end{aligned}$$

where the last term is the savings of disconnecting the customers in the set  $S$ . We will now prove our desired result by an inductive argument based on  $\tau$ . We will show that the total cost of serving customers in  $D \setminus U$  is at most  $\sum_{j \in D \setminus U} d_j \alpha_j$  at time  $\tau$ . It is clearly true for  $\tau = 0$ . Assume that it holds for time  $\tau$  and consider the first  $\tau' > \tau$  that an event occurs in



the Greedy Algorithm, i.e., there exists a facility  $i$  and set  $S \subseteq D \setminus A_i$  such that

$$\sum_{j \in S} o_{ji} = \sum_{j \in S} f_i \left( T_i + \sum_{j \in S} d_j \right) - f_i(T_i).$$

Let  $S_1 \subset S$  be the set of clients that switched to  $i$  from another facility and  $S_2 = S \setminus S_1$ . The additional cost of serving customers in  $S$  at facility  $i$  is

$$\begin{aligned} \sum_{j \in S} d_j c_{ij} + f_i \left( T_i + \sum_{j \in S} d_j \right) - f_i(T_i) \\ &= \sum_{j \in S} (d_j c_{ij} + o_{ji}) \\ &= \sum_{j \in S_1} (d_j c_{ji} + o_{ji}) + \sum_{j \in S_2} (d_j c_{ji} + o_{ji}) \\ &= \sum_{j \in S_1} d_j (c_{ij} + w_{ji}) + \sum_{j \in S_2} d_j \alpha_j \end{aligned}$$

where  $i_j$  is the facility to which  $j \in S_1$  was previously connected. As previously shown, the cost in savings in disconnecting the customers  $j \in S_1$  from the facilities they were previously connected to is at least  $\sum_{j \in S_1} d_j (c_{ij} + w_{ji})$  and therefore the cost is increased by at most  $\sum_{j \in S_2} d_j \alpha_j$ . Our result follows from the induction hypothesis. ■

Consider the optimal solution to the CCFL problem and let  $A_i^*$  denote the set of customers assigned to facility  $i$ . If we can show the existence of a pair of constants  $(R_f, R_c)$  such that for every  $i$

$$\sum_{j \in A_i^*} d_j \alpha_j \leq R_f f_i \left( \sum_{j \in A_i^*} d_j \right) + R_c \sum_{j \in A_i^*} d_j c_{ij}$$

then the Greedy Algorithm is an  $(R_f, R_c)$ -approximation algorithm. We will now focus on a facility  $i$  and the customers indexed  $1, \dots, k$  such that  $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_k$ . We will let  $e_j$  denote the connection cost of the  $j$ -th customer in the ordering to facility  $i$  and  $f = f_i(\sum_{j=1}^k d_j)$ . For each  $j \in A_i$ , define the critical time of  $j$  as the time  $\tau$  right before  $j$  is connected to a facility. For any customer  $\ell < j$ , we define  $r_{\ell j} = \alpha_\ell = \alpha_j$  if  $\ell$  has not been connected to a facility at the critical time of  $j$  and  $r_{\ell j} = w_{\ell i} + c_{i\ell}$  if  $\ell$  is connected to facility  $i_\ell$  at the critical time of  $j$ . We will now derive a set of inequalities involving the variables  $\alpha_j$  and  $r_{\ell j}$ . The concept of ‘‘contribution withdrawal’’ in our algorithm plays a very important role in deriving these inequalities.

**Lemma 3.6.** *For any pair of customers  $j, \ell$ ,  $\alpha_j \leq r_{\ell j} + e_i + e_j$ .*

**Proof.** Consider the critical time of customer  $j$ . If  $\ell$  is not connected at this time, then this inequality holds trivially.

Otherwise, let  $i'$  be the facility to which  $\ell$  is connected. By definition, we have

$$r_{\ell j} = c_{i'\ell} + \frac{f_{i'}(T_{i'}) - f_{i'}(T_{i'} - d_\ell)}{d_\ell} \geq c_{i'\ell} + \frac{f_{i'}(T_{i'} + d_j) - f_{i'}(T_{i'})}{d_j} \quad (9)$$

where the inequality holds due to the concavity of  $f_{i'}$ . It must be true that

$$\alpha_j \leq c_{i'j} + \frac{f_{i'}(T_{i'} + d_j) - f_{i'}(T_{i'})}{d_j} \quad (10)$$

as otherwise we would have connected customer  $j$  to facility  $i'$  at an earlier time. Combining Equations (9) and (10) yields

$$\alpha_j \leq c_{i'j} + r_{\ell j} - c_{i'\ell}.$$

Applying the triangle inequality to this relation proves our desired result. ■

Lemma 3.6 relates the triangle inequality to our algorithm. The following lemma derives a set of inequalities involving the facility opening cost based upon the concavity of the facility cost function.

**Lemma 3.7.** *For every  $j = 1, \dots, k$ ,*

$$\sum_{\ell=1}^{j-1} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell=j}^k d_\ell \max\{\alpha_\ell - e_\ell, 0\} \leq f.$$

**Proof.** Consider the time at which customer  $j$  gets connected to a facility, i.e.,  $\tau = \alpha_j$ . Let  $A_i$  and  $T_i$  be the set of clients served at facility  $i$  and the total demand served at  $i$ . The amount that client  $\ell$  offers to facility  $i$  is  $d_\ell \max\{\alpha_j - e_\ell, 0\}$  if  $\ell \geq j$  and  $d_j \max\{r_{\ell j} - e_\ell, 0\}$  if  $\ell < j$ . Let  $D_1$  and  $D_2$  be the set of connected and unconnected customers in  $\{1, \dots, k\}$ . We have

$$\begin{aligned} \sum_{\ell \in D_1 \setminus A_i} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell \in D_2} d_\ell \max\{\alpha_\ell - e_\ell, 0\} \\ \leq f_i \left( T_i + \sum_{\ell \in (D_1 \cup D_2) \setminus A_i} d_\ell \right) - f_i(T_i) \quad (11) \end{aligned}$$

since otherwise we would have connected customer  $k$  to facility  $i$  at an earlier time. By the concavity of  $f_i$  and the fact that  $T_i \geq \sum_{\ell \in D_1 \cap A_i} d_\ell$ , we have

$$\begin{aligned} f_i \left( T_i + \sum_{\ell \in (D_1 \cup D_2) \setminus A_i} d_\ell \right) - f_i(T_i) \\ \leq f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell + \sum_{\ell \in (D_1 \cup D_2) \setminus A_i} d_\ell \right) - f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right) \\ \leq f_i \left( \sum_{\ell=1}^k d_\ell \right) - f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right). \quad (12) \end{aligned}$$

Combining Equations (11) and (12) yields

$$\begin{aligned} & \sum_{\ell \in D_1 \setminus A_i} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell \in D_2} d_\ell \max\{\alpha_\ell - e_\ell, 0\} \\ & \leq f_i \left( \sum_{\ell=1}^k d_\ell \right) - f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right). \end{aligned} \quad (13)$$

We now turn our attention to deriving an inequality for  $\sum_{\ell \in D_1 \cap A_i} d_\ell \max\{r_{\ell j} - e_\ell, 0\}$ . Because of the concavity of  $f_i$ , we have, for each  $\ell' \in A_i$ ,

$$\begin{aligned} f_i(T_i) - f_i(T_i - d_{\ell'}) & \leq f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right) - f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell - d_{\ell'} \right) \\ & \leq \frac{d_{\ell'} f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right)}{\sum_{\ell \in D_1 \cap A_i} d_\ell}. \end{aligned}$$

It follows that

$$\begin{aligned} & \sum_{\ell \in D_1 \cap A_i} d_\ell \max\{r_{\ell j} - e_\ell, 0\} \\ & = \sum_{\ell \in D_1 \cap A_i} d_\ell (r_{\ell j} - e_\ell) = \sum_{\ell \in D_1 \cap A_i} d_\ell w_{\ell, j} \\ & = \sum_{\ell \in D_1 \cap A_i} (f_i(T_i) - f_i(T_i - d_\ell)) \\ & \leq \sum_{\ell \in D_1 \cap A_i} \frac{d_\ell f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right)}{\sum_{\ell \in D_1 \cap A_i} d_\ell} = f_i \left( \sum_{\ell \in D_1 \cap A_i} d_\ell \right). \end{aligned} \quad (14)$$

Therefore, by combining equations (13) and (14), we have

$$\begin{aligned} & \sum_{\ell \in D_1} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell \in D_2} d_\ell \max\{r_{\ell j} - e_\ell, 0\} \\ & \leq f_i \left( \sum_{\ell=1}^k d_\ell \right) = f. \end{aligned}$$

Our result follows by noticing that

$$\begin{aligned} & \sum_{\ell=1}^{j-1} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell=j}^k d_\ell \max\{\alpha_\ell - e_\ell, 0\} \\ & = \sum_{\ell \in D_1} d_\ell \max\{r_{\ell j} - e_\ell, 0\} + \sum_{\ell \in D_2} d_\ell \max\{r_{\ell j} - e_\ell, 0\}. \end{aligned}$$

Lemma 3.6 and Lemma 3.7 lead to our main result about the Greedy Algorithm.

**Theorem 3.8.** *For any  $R_f \geq 1$ , the Greedy Algorithm is an  $(R_f, R_c)$ -approximation algorithm, where  $R_c$  is an upper bound on the solution of*

$$\text{maximize } \frac{\sum_{j=1}^k d_j \alpha_j - R_f f}{\sum_{j=1}^k d_j e_j}$$

subject to

(FLP)

$$\begin{aligned} & \alpha_j \leq \alpha_{j+1} \quad \text{for all } j = 1, \dots, k-1 \\ & r_{\ell j+1} \leq r_{\ell j} \quad \text{for all } \ell = 1, \dots, j-1; \\ & \quad j = 2, \dots, k \\ & \alpha_j \leq r_{\ell j} + e_j + e_\ell \quad \text{for all} \\ & \quad \ell = 1, \dots, j-1; j = 2, \dots, k \end{aligned}$$

$$\begin{aligned} & \sum_{\ell=1}^{j-1} d_\ell \max\{r_{\ell j} - e_\ell, 0\} \\ & + \sum_{\ell=j}^k d_\ell \max\{\alpha_\ell - e_\ell, 0\} \leq f \quad \text{for all } j = 1, \dots, k \end{aligned}$$

$$\begin{aligned} & \alpha_j, d_j, f, r_{\ell j} \geq 0 \quad \text{for all } \ell = 1, \dots, j-1; \\ & \quad j = 2, \dots, k. \end{aligned}$$

Problem (FLP) is called the factor revealing linear program (see Jain et al. [13]). It was shown by Jain et al. [13] that for the case of unit demands ( $d_j = 1$  for  $j = 1, \dots, k$ ) if  $R_f = 1.61$  then  $R_c = 1.61$  and if  $R_f = 1$  then  $R_c = 2$ . Further, Mahdian et al. [17] show that if  $R_f = 1.11$  then  $R_c = 1.78$ , which will be important in developing a 1.52-approximation algorithm for the CCFL problem. These results still hold for general integral  $d_j$  by replicating each  $\alpha_j$  by  $d_j$  copies and each  $r_{\ell j}$  by  $d_\ell \times d_j$  copies. It can easily be seen that the replicated copies still satisfy the constraints of (FLP). This leads to the following result.

**Lemma 3.9.** *The Greedy Algorithm is a 1.61-approximation algorithm for the CCFL problem with a running time of  $O(mn^3 \max\{\log n, \phi\})$ .*

We can use the Greedy Algorithm and Theorem 3.8 to generalize the algorithm of Mahdian et al. [17].

**Theorem 3.10.** *There exists a 1.52-approximation algorithm for the CCFL problem with a running time of  $O(mn^2 \max\{\log n, \phi\} \max\{m, n\})$ .*

**Proof.** We develop a two-phase algorithm for the CCFL problem. In the first phase, we use the concept of scaling the facility costs that was introduced by Charikar and Guha [4]. Specifically, we scale up the facility opening costs by a factor of  $\delta$ , i.e.,  $f_i(\sum_{j=1}^n d_j x_{ij}) = \delta f_i(\sum_{j=1}^n d_j x_{ij})$ , and apply the Greedy Algorithm. Given the solution returned by the first phase, we scale down the facility cost functions back to the original facility cost functions at the same rate. If at any point in this phase, we can reassign a set of customers to a different facility without increasing the cost of the solution, we perform the reassignment. If we are at a point in the second phase of the algorithm, say the facility cost functions are scaled up by a factor of  $\delta^c$ , where no reassignments can be performed, then it can be shown that determining the next factor  $\delta^*$  where a

reassignment can be performed is equivalent to determining the maximum value of  $\delta^*$  that satisfies

$$\sum_{j \in S} d_j \max\{w_{ij} + c_{ij} - c_{ij}, 0\} = \delta^* f_i \left( T_i + \sum_{j \in S} d_j \right) - \delta^* f_i(T_i)$$

for some facility  $i = 1, \dots, m$  and set  $S \subseteq D \setminus A_i$ . For a facility  $i$ , define the problem

$$K_i(\delta') = \max_{S \subseteq D \setminus A_i} \sum_{j \in S} d_j \max\{w_{ij} + c_{ij} - c_{ij}, 0\} - \delta' f_i \left( T_i + \sum_{j \in S} d_j \right) - f \delta' f_i(T_i)$$

where  $i_j$  is the facility  $j$  is currently assigned to and  $w_{ij}$  is defined as above. For a fixed  $\delta'$ , this problem belongs to the same class as the problem  $KP(\lambda)$  in the proof of Lemma 3.4. If we sort the customers in  $D \setminus A_i$  according to  $d_j \max\{w_{ij} + c_{ij} - c_{ij}, 0\}$  in non-increasing order, then an optimal solution to  $K_i(\delta')$  contains the first  $k$  customers in the ordering. Note that the ordering of the customers is independent of  $\delta'$ . Define  $\delta'_k$  to be the value of  $\delta'$  such that the solution containing the first  $k$  customers in the ordering has its objective value equal to 0. Therefore, the largest value of  $\delta'_k$  will be the first time we could perform a reassignment at facility  $i$ . This implies that we can determine  $\delta^*$  by applying the above procedure for each facility, in  $O(mn \max\{\log n, \phi\})$  time. If  $\delta^* \leq 1$ , then we terminate the second phase of the algorithm. In each reassignment, at least one customer is switched to a different facility. Each customer can be reassigned to each facility a constant number of times throughout the second phase of the algorithm, and therefore we have a bound of  $O(mn)$  reassignments. Therefore, the second phase of the algorithm runs in  $O(m^2 n^2 \max\{\log n, \phi\})$  time.

The derivation of the approximation guarantee of the algorithm uses the results of Mahdian et al. [17]. Mahdian et al. [17] showed that their two-phase algorithm is an  $\left(R_f + \ln \delta + \epsilon, 1 + \frac{R_c - 1}{\delta}\right)$ -approximation algorithm for any  $(R_f, R_c)$  given by Theorem 3.8. This analysis relied on deriving a factor revealing linear program by analyzing an algorithm that scales down  $\delta$  in  $L$  discrete steps rather continuously. We can apply a similar analysis as this to show that our two-phase algorithm is an  $\left(R_f + \ln \delta + \epsilon, 1 + \frac{R_c - 1}{\delta}\right)$ -approximation algorithm. If we set  $(R_f, R_c) = (1.11, 1.78)$  and  $\delta = 1.504$ , then we have a  $(1.11 + \ln(1.504) + \epsilon, 1 + \frac{1.78 - 1}{1.504}) = (1.5181 + \epsilon, 1.518)$ -approximation algorithm for the CCFL problem. Therefore, the two-phase algorithm has a guarantee of 1.52. ■

This immediately leads to the following result for the UFLPP problem with seasonal demands.

**Corollary 3.11.** *There exists a 1.52-approximation algorithm for the UFLPP problem with seasonal demands with a*

*running time of  $O(mn^2 \max\{\log n, T^2\} \max\{m, n\})$  in the case of general concave production and holding cost functions and  $O(mn^2 \max\{\log n, T \log T\} \max\{m, n\})$  for fixed-charge plus linear production costs and linear holding costs.*

**Proof.** For general concave production and holding cost functions, we can use the algorithm of Wagner [26] or Veinott [24] to evaluate the function  $f_i(z)$  in  $O(T^2)$  time. For fixed-charge plus linear production costs and linear holding costs, we can use the algorithm of Aggarwal and Park [1], Federgruen and Tzur [7], or Wagelmans et al. [25] to evaluate the function  $f_i(z)$  in  $O(T \log T)$  time. ■

### 3.2. Results on Generalizations of the CCFL Problem

In this subsection, we will discuss approximation algorithms and results for generalized facility location problems (P), where the functions  $f_i, i = 1, \dots, m$ , have different structures. In particular, we examine the problem (P) where the functions  $f_i, i = 1, \dots, m$ , are subadditive. Recall that a function  $f$  is *subadditive* if  $f(x + y) \leq f(x) + f(y)$  for any non-negative  $x$  and  $y$ . This class of problems was recently proposed by Gabor and van Ommeren [10]. They discuss three examples of facility location problems with stochastic demands that belong to this class of problems. Other examples of facility location problems with subadditive facility costs can be found in Gabor and van Ommeren [9] and Rodolakis et al. [18]. Gabor and van Ommeren [10] develop an approximation algorithm with a guarantee of  $2(1 + \epsilon)$ , for any  $\epsilon > 0$ . Also, for a special class of subadditive cost functions, they develop an approximation algorithm with a guarantee of 2. It was observed in both Gabor and van Ommeren [10] and Rodolakis et al. [18] that the concave envelope of a subadditive function is a 2-approximation of the function. As the Greedy Algorithm that we developed earlier is a (1, 2)-approximation algorithm for the CCFL problem, this yields a (2, 2)-approximation algorithm for the facility location problem with discrete subadditive cost functions, given that we can evaluate the concave envelope of a subadditive function at a single point in polynomial time. Note that this is less restrictive than needing to construct the entire concave envelope of a subadditive function in polynomial time.

## 4. SUMMARY AND CONCLUDING REMARKS

In this article, we have studied a model for integrating facility location and production planning decisions. It was shown that, in general, this problem is as hard as the set cover problem. Therefore, we have focused on identifying special cases of this problem class that can be approximated within a constant factor. One of the classes of problems for which we derived a new approximation algorithm can be viewed as a metric facility location problem where the facility costs are a concave function of the amount of demand assigned to the facility. We developed a greedy algorithm for this class of functions that generalizes an algorithm of Jain et al. [13]. We

then were able to use this greedy algorithm together with the idea of cost-scaling to develop an approximation algorithm with a guarantee of 1.52.

An important direction for future research is to identify additional problems in our class that can be approximated within a constant factor. For example, it will be interesting to investigate whether constant factor approximation algorithms exist for generalizations of the two problem classes examined in Section 2 that include production setup costs. It may also be important to develop algorithms, such as a column generation approach, to solve the problem the UFLPP problem exactly.

## Acknowledgments

We thank the anonymous referees for their helpful suggestions that led to a better presentation of the paper.

## REFERENCES

- [1] A. Aggarwal and J. Park, Improved algorithms for economic lot size problems, *Oper Res* 41 (1993), 549–571.
- [2] E. Arkin, D. Joneja, and R. Roundy, Computational complexity of uncapacitated multi-echelon production planning problems, *Oper Res Lett* 8 (1989), 61–66.
- [3] J. Byrka, An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem, *Proceedings of the 10th Int Workshop Approximation Algorithms Combinatorial Optim Prob (APPROX)*, Princeton, NJ, 2007, pp. 29–43.
- [4] M. Charikar and S. Guha, Improved combinatorial algorithms for facility location problems, *SIAM J Comput* 34 (2005), 803–824.
- [5] F. Chudak and D. Shmoys, Improved approximation algorithms for the uncapacitated facility location problem, *SIAM J Comput* 33 (2003), 1–25.
- [6] M. Daskin, C. Coullard, and Z.J. Shen, An inventory-location model: Formulation, solution algorithm, and computational results, *Ann Oper Res* 10 (2001), 83–106.
- [7] A. Federgruen and M. Tzur, A simple forward algorithm to solve general dynamic lot sizing models with  $n$  periods in  $O(n \log n)$  or  $O(n)$  time, *Manage Sci* 37 (1991), 909–925.
- [8] U. Feige, A threshold of  $\ln n$  for approximating set cover, *J ACM* 45 (1998), 634–652.
- [9] A. Gabor and J. van Ommeren, An approximation algorithm for a facility location problem with stochastic demands and inventories, *Oper Res Lett* 34 (2006), 257–263.
- [10] A. Gabor and J. van Ommeren, Approximation algorithms for facility location problems with a special class of subadditive cost functions, *Theoret Comput Sci* 363 (2006), 289–300.
- [11] S. Guha and S. Khuller, Greedy strikes back: Improved facility location algorithms, *J Algorithms* 31 (1999), 228–248.
- [12] M. Hajiaghayi, M. Mahdian, and V. Mirrokni, The facility location problem with general cost functions, *Networks* 42 (2003), 42–47.
- [13] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani, Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP, *J ACM* 50 (2003), 795–824.
- [14] J. Krarup and O. Bilde, “Plant location, set covering, and economic lot size: An  $O(mn)$ -algorithm for structured problems,” *Numerische methoden bei optimierungsaufgaben, band 3: Optimierung bei graph-theoretischen und ganzzahligen problemen*, L. Collatz, W. Wetterling (Editors), *Int Ser Numer Math* vol. 36, Birkhäuser, 1977, pp. 155–180.
- [15] R. Levi, R. Roundy, and D. Shmoys, Primal-dual algorithms for deterministic inventory problems, *Math Oper Res* 31 (2006), 267–284.
- [16] T. Magnanti and D. Stratila, Strongly polynomial primal-dual algorithms for concave cost combinatorial optimization problems, Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2007.
- [17] M. Mahdian, Y. Ye, and J. Zhang, Approximation algorithms for metric facility location problems, *SIAM J Comput* 36 (2006), 411–432.
- [18] G. Rodolakis, S. Siachalou, and L. Georgiadis, Qual Service in Multiservice IP Networks: Third Int Workshop, QoS-IP 2005, LNCS 3375 (2005), pp. 207–220.
- [19] R. Roundy, Efficient, effective lot-sizing for multi-product, multi-stage production systems, *Oper Res* 41 (1993), 371–386.
- [20] Z. Shen, C. Coullard, and M. Daskin, A joint location-inventory model, *Transportation Sci* 37 (2003), 40–55.
- [21] D. Shmoys, C. Swamy, and R. Levi, Facility location with service installation costs, *Proc 15th ACM-SIAM Symp Discr Algorithms (SODA)*, New Orleans, LA, 2004, pp. 1088–1097.
- [22] D. Shmoys, E. Tardos, and K. Aardal, Approximation algorithms for facility location problems, *Proc 29th Ann ACM Symp Theory Comput (STOC)*, El Paso, TX, 1997, pp. 265–274.
- [23] M. Sviridenko, A 1.582-approximation algorithm for the metric uncapacitated facility location problem, *Proceedings 9th Conference Integer Program Combinatorial Optim (IPCO)*, Cambridge, MA, 2002, pp. 240–257.
- [24] A. Veinott, Minimum concave cost solutions of Leontief substitution models of multi-facility inventory systems, *Oper Res* 17 (1969), 262–291.
- [25] A. Wagelmans, S. van Hoesel, and A. Kolen, Economic lot sizing: An  $O(n \log n)$  algorithm that runs in linear time in the Wagner-Whitin case, *Oper Res* 40-S1 (1992), S145–S156.
- [26] H. Wagner, A postscript to dynamic problems of the theory of the firm, *Naval Res Logist Q* 7 (1960), 7–12.
- [27] H. Wagner and T. Whitin, Dynamic version of the economic lot size model, *Manage Sci* 5 (1958), 89–96.
- [28] W. Zangwill, A deterministic multi-product, multi-facility production and inventory model, *Oper Res* 17 (1966), 486–507.