SIAM J. Discrete Math.
Vol. 24, No. 2, pp. 538–551

APPROXIMATION ALGORITHMS FOR MULTICOMMODITY FACILITY LOCATION PROBLEMS*

R. RAVI† AND AMITABH SINHA‡

Abstract. Multicommodity facility location refers to the extension of facility location to allow for different clients' demands for different goods from among a finite set of goods. This leads to several optimization problems, depending on the cost of opening a facility (now a function of the commodities it serves). In this paper, we introduce and provide approximation algorithms for two fairly general versions of multicommodity facility location. We formulate integer programming models for these problems, and use them to provide approximation algorithms for the problems that are close to the inapproximability thresholds.

Key words. facility location, approximation algorithm, LP rounding

AMS subject classifications. 68W25, 90C27, 90C08

DOI. 10.1137/080732857

1. Introduction.

The location of facilities (manufacturing plants, warehouses, distribution centers, collection centers) is an essential component of the design of production and supply networks in the value chain. According to the 14th Annual State of Logistics report [7], U.S. companies spent \$910 billion (or 8.7% of the gross domestic product) on business logistics in 2002. Therefore, the question of locating facilities is of critical importance, and has spurred a long line of research. Much of this work was extensively surveyed by Cornuéjols, Nemhauser, and Wolsey [5], and more recently by Daskin, Snyder, and Berger [6].

The classic *uncapacitated facility location* (UFL) problem is the simplest model of facility location. This problem is defined in a metric space, a subset of whose points are clients and another subset consists of facilities. Each facility has an opening cost, and the objective is to compute a solution that minimizes the total cost of open facilities and the sum of the distances from each client to its nearest open facility.

In this paper, we study a natural extension of the facility location model where there is a finite set of commodities, and each client demands a subset of those commodities. We call this *multicommodity facility location* (MCFL). Facility costs are now a function of the location and the commodities served. This model has many practical applications. For example, a consumer goods manufacturer who produces five different goods might choose to locate three plants to manufacture them, to utilize the economies of scale arising from having one plant producing two commodities instead of a separate plant for each commodity.

*Received by the editors August 15, 2008; accepted for publication (in revised form) March 21, 2010; published electronically May 14, 2010. A preliminary version of this paper appeared in [15]. The preliminary version contains an error: Theorem 3.1 is incorrect, and does not appear in this version.

http://www.siam.org/journals/sidma/24-2/73285.html

†Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 (ravi@cmu.edu). This author's research was supported in part by NSF grant CCR-0105548 and ITR grant CCR-0122581 (the Aladdin project).

‡Ross School of Business, University of Michigan, 701 Tappan Street, Ann Arbor, MI 48109 (amitabh@umich.edu). This author conducted his research while a graduate student at the Tepper School of Business (then GSIA), Carnegie Mellon University, and was supported in part by a Carnegie Bosch Institute graduate fellowship, NSF grant CCR-0105548, and ITR grant CCR-0122581 (the Aladdin project).

To take one specific example, as of June 2005, Toyota Motor North America [21] had five assembly plants (California, Texas, Indiana, Kentucky, and Ontario) in North America, which produced nine different vehicles (Camry, Avalon, Camry Solara, Corolla, Matrix, Sequoia, Sienna, Tacoma, and Tundra). Not all facilities produced all vehicles: The Tundra was assembled both at Indiana and at Texas, while the Indiana plant also assembled Sequoias and Siennas. The decisions about where each plant is located and what vehicles are assembled there are critical to Toyota's profitability, and must take into account various considerations of demand, manufacturing cost, etc. Our models aim to capture this kind of reality and provide solutions for such problems.

Daskin, Snyder, and Berger [6] provide a comprehensive survey of the operations research literature in facility location. However, surprisingly little exists on facility location models with multiple commodities or items. Recently, Shi et al. [17] studied a two-echelon location/routing model with multiple commodities. However, in their model, the location of the plants and consumers is given, as is the commodities to be manufactured at each plant. The only decisions are the locations of the warehouses, whose costs are independent of the commodities served.

Our work falls within the framework of *approximation algorithms*; that is, heuristics with guarantees on both solution quality and running time. The NP-completeness of even the UFL problem means that the MCFL and all its variants are computationally intractable, thus necessitating efficient approximation algorithms. Within the approximation algorithms literature, the uncapacitated (single commodity) facility location problem has received a lot of attention. The first constant factor approximation uses LP rounding and is due to Shmoys, Tardos, and Aardal [19]; our algorithm builds on their results and those of Lin and Vitter [11]. Other work includes the current best approximation ratio of 1.5 due to Byrka [4]. The work of Guha and Khuller [8] combined with an observation by Sviridenko (personal communication, reported in [10]) established that the problem is inapproximable beyond 1.46 unless P = NP.

Several extensions of the facility location model have been considered in the approximation algorithms literature. Using local search, Pál, Tardos, and Wexler [14] provided an approximation algorithm for the facility location problem with capacity constraints on facilities. Meyerson [13] considered an online version of the facility location problem, wherein client requests are received sequentially, and provided an algorithm with a bounded performance ratio. The k-median problem is a variant where facilities have no costs, but one is constrained to opening no more than k of them. The current best approximation algorithm also uses a local search routine [2]. Ravi and Sinha [16] consider the problem of combining facility location with designing a network in the presence of economies of scale, and provide a constant-factor approximation algorithm.

However, none of these models incorporate multiple commodities. The first papers to provide nontrivial approximation algorithms for different versions of MCFL were (to the best of our knowledge) the conference version of this manuscript [15] and an independent paper by Shmoys, Swamy, and Levi [18] that appeared in the same conference. Shmoys, Swamy, and Levi [18] considered somewhat restricted classes of MCFL problems, and provided constant-factor approximation algorithms for them. For example, they obtained a 6-approximation if the following restriction is imposed: There is an ordering on the facilities wherein if facility A precedes facility B, the incremental cost for serving any commodity is no greater at facility A than at facility B. If, additionally, the facility cost structure is the same in all facilities, the approximation ratio is improved to 2.391. Subsequent to their work, Svitkina and Tardos [20] generalized their model to one with a hierarchical structure on the cost function, and used local search to obtain a constant-factor approximation for the special case when the facility cost structure is identical in all facilities. We consider more general facility cost functions, but our results are weaker: The approximation ratio is only logarithmic in the number of commodities. We point out (see the discussion in section 3) that an approximation ratio that is logarithmic in the number of clients is relatively straightforward for certain cost structures; however, one would typically expect that the number of clients is much greater than the number of commodities.

1.1. Our results and paper outline. We first consider the MCFL problem, where the cost of producing a set of commodities at a given facility can be expressed as the sum of a fixed cost for opening the facility, and commodity-wise incremental costs for producing specific commodities. In section 2, we provide a $(\log k + 4)$-approximation algorithm for this problem, where k is the number of commodities. Our model for MCFL can be extended to allow for more general facility costs. A similar approximation ratio is obtained, albeit with an (arguably unavoidable) increase in the computational complexity. This extension is described in section 3. We also study the hardness of these problems in section 4, and show that our approximation ratios are close to the inapproximability thresholds, thus supporting our assertion that our algorithms are close to the best possible (in terms of approximability).

2. Multicommodity facility location.

2.1. Model. Our model and notation extend the well-studied *uncapacitated facility location* model. There is a set of clients D, and a set of facilities F, with a distance function $c : D \times F \to \mathbb{R}^+$, which is a metric. There is also a set of commodities S, and each client j has demand for one unit of commodity $d_j \in S$. Each client demands exactly one unit of exactly one commodity, although the same commodity could be demanded by several different clients. Relaxing this assumption to allow different commodities to be demanded by the same client can be incorporated by making multiple copies of the client, each demanding one commodity, provided one permits a solution where different commodities to a client are allowed to be provided by possibly different facilities. Relaxing the assumption to allow the quantity demanded to be nonunit is less straightforward. We briefly discuss relaxing the single-commodity-per-client assumption in section 4.3.

The number of clients $|D|$, facilities $|F|$, and commodities $|S|$ are abbreviated n, m, and k, respectively. We use 2^S to represent the set of all subsets of S. A *configuration* $\sigma(i) \in 2^S$ for facility i is simply a specification of the commodities one has chosen to manufacture there. In general, the cost of opening facility i in configuration $\sigma(i)$ may be an arbitrary general function. In this section, however, we consider a *linear* specification of this cost. Each facility i has an opening cost (fixed cost) $f_i^0 \geq 0$, and for each commodity s, there is an incremental cost $f_i^s \geq 0$, so that the cost of opening facility i in configuration σ is $f_i(\sigma) = f_i^0 + \sum_{s \in \sigma} f_i^s$. The idea is as follows: Often, installing a facility involves a substantial start-up cost *independent of the commodities manufactured*, while the cost of configuring the facility to produce any commodity involves an incremental cost, particularly if all commodities are somewhat similar to each other (for example, if an automotive plant is retooled to produce an additional line of vehicles). Even if the facility costs do not obey this structure exactly, the costs can often be well-approximated by such a structure. The more general case of facility costs which depend arbitrarily on the commodities installed is discussed in

	Algorithm MCFL(S, D, d, F, f, c)
Given:	(i) Set of commodities S. (ii) Set of clients D, with client j demanding commodity $d_j \in S$. (iii) Set of facilities F, with the cost of facility i defined by the fixed opening cost f_i^0 and the incremental cost f_i^s to serve commodity s. (iv) Distance matrix c, with c_{ij} denoting the distance from facility i to client j.
Returns:	(i) A set of facilities $F' \subseteq F$ to open, and for each facility $f \in F'$, the set of commodities $\sigma(f)$ to be manufactured. (ii) For each client j, an assignment to an open facility $\phi(j) \in F'$ that manufactures the commodity required at j; that is, $d_j \in \sigma(\phi(j))$.
1:	Formulate the integer program (2.1–2.6), and let (x^*, y^*) be an optimal solution to its linear relaxation.
2:	Obtain the "filtered" solution $(\overline{x}, \overline{y})$ as described in section 2.4.
3:	Select a subset of clients R as representatives, as described in section 2.5. All clients not in R are assigned a representative in R.
4:	Solve the k-set cover instance induced by $(\overline{x}, \overline{y})$ and R as described in section 2.6. Let $\hat{z}$ denote the integer solution to this instance.
5:	Construct an integer solution $(\hat{x}, \hat{y})$ for the problem using $\hat{z}$, as described in section 2.7. The assignment ϕ is given by $\hat{x}_{ij}$ for clients in R, and the assignment of the representatives for clients not in R. The facilities to open and their configurations are given by $\hat{y}$.

FIG. 2.1. *Algorithm* **MCFL(S, D, d, F, f, c)**.

section 3.

A feasible solution is specified by a set of facilities F', together with a configuration $\sigma(i)$ of each facility $i \in F'$, such that each commodity which has at least one client demanding it is served by at least one facility in F'. Given F', each client j is assigned to $\phi(j) \in F'$, the nearest open facility which includes commodity d_j in its configuration. The total facility cost is $\sum_{i \in F'} f_i(\sigma(i))$, the sum of the costs incurred in opening each facility in F' in its chosen configuration. The total service cost is $\sum_{j \in D} c_{j,\phi(j)}$, the sum of distances from each client to the facility assigned to it. The total cost is the sum of these two, and the objective is to minimize the total cost.

2.2. Overview of the algorithm for MCFL. We begin by formulating an integer program for the problem, and solve its linear relaxation. We then filter the solution so that each client is fractionally assigned only to facilities that are close to it. Following this, we select a set of representative clients, and assign all other clients to these representatives. We then view the fractional solution consisting of the fractionally opened facilities and the representatives as a fractional solution to an instance of a variant of the set-cover problem. We then exploit the special structure of the fractional solution to efficiently round this set-cover instance, which results in the selection of facilities to be opened and the specification of their configurations. All representative clients are served by the facilities given by the set-cover solution, while all other clients are assigned to the same facilities as their representatitives. The rounding process resembles the technique used by Shmoys, Tardos, and Aardal [19], but with suitable adaptations to incorporate the multiple commodities.

An overview of the algorithm is provided in Figure 2.1. The details of the algorithm and the analysis are presented in the remainder of this section.

2.3. IP formulation. The starting point of our algorithm is an integer programming formulation for the MCFL problem. The integer program (IP) formulation (2.1–2.6) shown below extends the basic UFL formulation of Balinski [3] to incorporate multiple commodities, while also exploiting the linear cost structure of the facilities. Variable y_i^0 indicates whether or not facility i is opened, and variable y_i^s indicates whether or not facility i is serving commodity s. Variable x_{ij} indicates whether client j is served by facility i. Recall that d_j is the commodity demanded by client j.

$$\min \sum_{i\in F} \sum_{j\in D} c_{ij}x_{ij} + \sum_{i\in F} \sum_{s=0}^{k} f_i^s y_i^s, \tag{2.1}$$

$$\sum_{i\in F} x_{ij} \geq 1 \qquad \forall j \in D, \tag{2.2}$$

$$x_{ij} \leq y_i^{d_j} \qquad \forall j \in D,\ \forall i \in F, \tag{2.3}$$

$$y_i^s \leq y_i^0 \qquad \forall i \in F,\ \forall s \in S, \tag{2.4}$$

$$x_{ij}, y_i^s \geq 0 \qquad \forall j \in D,\ \forall i \in F,\ \forall s \in S, \tag{2.5}$$

$$x_{ij}, y_i^s \in \{0,1\} \qquad \forall j \in D,\ \forall i \in F,\ \forall s \in S. \tag{2.6}$$

The objective function is the sum of the total "service" costs (the transportation costs from each client to the facility serving it) and the total "facility" costs (the cost of opening each facility and installing commodities to be manufactured there). The constraints (2.2) ensure that each client is served by at least one facility. Constraints (2.3) ensure that if client j (which requires one unit of commodity d_j) is served by facility i, then commodity d_j is installed at facility i. Constraints (2.4) ensure that if a facility is opened to serve any commodity, then its fixed cost of opening f_i^0 must be incurred (by forcing the variable y_i^0 to one).

The linear relaxation of the IP formulation above is therefore (2.1–2.5). Observe that since this is a minimization program with all costs nonnegative, we do not need constraints of the form $x_{ij}, y_i^s \leq 1$ in order to compute the optimal fractional solution. Since the size of this linear program is polynomial in (n, m, k), an optimal solution can be found efficiently in time polynomial in (n, m, k). Let (x^*, y^*) denote an optimal solution of the linear relaxation, with objective function value z^*. We use z^* as our lower bound, and the next steps round the fractional solution to an integer solution with total cost within a bounded factor of z^*.

2.4. Filtering. The next step of the algorithm uses the filtering technique of Lin and Vitter [11], as was also done in [19]. We fix a constant $0 < \alpha < 1$. For every client j, we define its optimal fractional service cost to be $c_j^* = \sum_{i\in F} c_{ij}x_{ij}^*$. Order the facilities which serve client j according to nondecreasing distance from j. The α point for client j is the smallest distance c_j^α such that $\sum_{i\in F: c_{ij}\leq c_j^\alpha} x_{ij}^* \geq \alpha$.

We now create a new fractional solution $(\overline{x}, \overline{y})$, defined as follows. For each facility i and commodity s, define $\overline{y}_i^s = \min\{1, y_i^{s*}/\alpha\}$, and also define $\overline{y}_i^0 = \min\{1, y_i^{0*}/\alpha\}$. Define $\overline{x}$ as follows:

$$\overline{x}_{ij} = \begin{cases} \min\{1, x_{ij}^*/\alpha\} & \text{if } c_{ij} \leq c_j^\alpha, \\ 0 & \text{otherwise.} \end{cases}$$

LEMMA 2.1. *The fractional solution* $(\overline{x}, \overline{y})$ *is feasible for* (2.1–2.5). *Moreover,* $c_j^\alpha \leq \frac{1}{1-\alpha} c_j^*$.

Proof. The definition of $(\overline{x}, \overline{y})$ ensures that the constraints (2.2–2.5) are satisfied. Also, since $c_j^* = \sum_{i \in F} c_{ij} x_{ij}^*$, we have $c_j^* \geq \sum_{i \in F: c_j^\alpha \leq c_{ij}} c_{ij} x_{ij}^*$. This term, by definition of c_j^α, is at least $(1-\alpha) c_j^\alpha$. This gives us $c_j^* \geq (1-\alpha) c_j^\alpha$, which results in $c_j^\alpha \leq \frac{1}{1-\alpha} c_j^*$. ∎

For the uncapacitated, single-commodity facility location problem, Shmoys, Tardos, and Aardal [19] round the solution $(\overline{x}, \overline{y})$ into an integer solution by considering clients in nondecreasing order of c_j^α, and opening a facility each time with client j as well as all other clients sufficiently close to j assigned to the opened facility. However, we are unable to do this because of the presence of multiple commodities: Nearby clients might require different commodities, and we have to choose in which configuration to open each facility. This requires the introduction of a number of new steps in our algorithm, which are described in sections 2.5–2.7.

2.5. Selection of representatives. The existence of multiple commodities presents several difficulties if we attempt to round the fractional solution $(\overline{x}, \overline{y})$. Hence we introduce a new step where we select a set of clients as representatives such that no two representatives of the same commodity are fractionally served by the same facility. We do this representative selection independently for each commodity.

Fix a commodity s, and consider all clients that require commodity s in increasing order of c_j^α. Let $D_s = \{j_1, j_2, \ldots, j_{n_s}\}$ be the clients in this order. Iteratively, mark the smallest index (smallest c_j^α) client $j \in D_s$ as a representative. All clients $j' \in D_s$ such that there exists a facility i such that $\overline{x}_{ij} > 0$ and $\overline{x}_{ij'} > 0$ are removed from the list D_s. We do not consider client j' as a candidate for being a representative, and instead mark client j as the representative for client j'. Let R denote the set of representatives over all commodities.

PROPOSITION 2.2. *If client* $j' \in D_s$ *is represented by* $j \in D_s$, *then* $c_{jj'} \leq 2c_{j'}^\alpha$.

Proof. By definition of $\overline{x}$, we have $\overline{x}_{ij} > 0$ only if $c_{ij} \leq c_j^\alpha$. We let j' be represented by j only if there exists a facility i such that $\overline{x}_{ij} > 0$ and $\overline{x}_{ij'} > 0$. This means that $c_{jj'} \leq c_{ij} + c_{ij'} \leq c_j^\alpha + c_{j'}^\alpha$. However, since clients are considered for inclusion in R in nondecreasing order of their α points, we must have $c_j^\alpha \leq c_{j'}^\alpha$. Therefore, $c_{jj'} \leq 2c_{j'}^\alpha$. ∎

2.6. Interpretation as a fractional *k*-set cover solution. The next step of our algorithm is to cast the fractional solution (restricted to the set R) as the fractional solution to an instance of the set-cover problem. Informally, the set-cover problem is to choose a minimum weight subcollection of sets (from a given collection) such that every element in the given universe appears in at least one set in the chosen subcollection. Further discussion of the relationship of MCFL with set-cover is provided in section 4.

The k-set-cover problem is a special case of the set-cover problem when each set has cardinality no more than k. We use our fractional solution to construct a k-set-cover instance and show that $\overline{y}$ is a fractional solution of the k-set-cover instance. We use the following IP formulation for our instance of k-set-cover. Recall that we use the term configuration to denote a set of commodities, and a facility i is said to be opened in configuration $\sigma \in 2^S$ if it serves the commodities in σ. We create a set (i, σ) with cost $f_i(\sigma)$ for every facility-configuration pair (i, σ). Let z_i^σ be 1 if set (i, σ) is included in our solution. Our universe consists of all clients in R, and a client $j \in R$ is included in a set (i, σ) if and only if $\overline{x}_{ij} > 0$ and $d_j \in \sigma$. The k-set-cover instance is defined below; subsequently, we will establish that it is indeed a k-set-cover instance (that is, every set has cardinality no more than k), and we will convert our current solution $(\overline{x}, \overline{y})$ into a feasible fractional solution for the instance.

$$\min \sum_{i\in F}\sum_{\sigma\in 2^S} f_i(\sigma) z_i^\sigma \tag{2.7}$$

$$\text{s.t.} \sum_{(i\in F,\sigma\in 2^S):\overline{x}_{ij}>0, d_j\in\sigma} z_i^\sigma \geq 1 \quad \forall j \in R, \tag{2.8}$$

$$z_i^\sigma \in \{0,1\} \quad \forall i \in R, \sigma \in 2^S. \tag{2.9}$$

Our k-set-cover instance is defined as in (2.7–2.9), with an element for every client $j \in R$ and a set for every facility-configuration pair. This by itself is an instance that is exponential in the number of commodities. However, the following lemma casts our fractional solution $\overline{y}$ as a fractional solution to the k-set-cover instance so that only polynomially many facility-configuration pairs have nonzero fractional variables. This allows us to restrict our attention to a k-set-cover instance (and fractional solution) of size polynomial in (n, m, k).

Lemma 2.3. *There is a fractional solution z to the k-set-cover instance* (2.7–2.9) *such that the following hold:* (i) $\sum_{(i\in F,\sigma\in 2^S):\overline{x}_{ij}>0,d_j\in\sigma} z_i^\sigma \geq 1$ *for all* $j \in R$; (ii) *for every facility i, there are at most k configurations for which $z_i^\sigma > 0$;* (iii) *the total cost of the fractional solution is no more than* $\sum_{i\in F}\sum_{s=0}^k f_i^s \overline{y}_i^s$.

Proof. Consider facility i, and order the commodities so that $\overline{y}_i^0 \geq \overline{y}_i^1 \geq \overline{y}_i^2 \geq \cdots \geq \overline{y}_i^k$. For notational convenience, we also define $\overline{y}_i^{k+1} = 0$. Observe that without loss of feasibility and with only a decrease in costs, we can assume that $\overline{y}_i^0 = \overline{y}_i^1$. Let $[s] = \{1, 2, \ldots, s\}$. We open facility i in configuration $[s]$ to extent $z_i^{[s]} = \overline{y}_i^s - \overline{y}_i^{s+1}$ for $s = 1, 2, \ldots, k-1$, and $z_i^{[k]} = \overline{y}_i^k$.

We now verify (i). Consider a client $j \in R$. Define s_j to be the index of commodity d_j in the ordering of $\overline{y}_i^s$ for facility i. Observe that $\sum_{(i\in F,\sigma\in 2^S):\overline{x}_{ij}>0,d_j\in\sigma} z_i^\sigma \geq \sum_{i\in F:\overline{x}_{ij}>0}\sum_{s=s_j}^k z_i^{[s]}$, because the sum in the right-hand side of the inequality is over a subset of the (set of all possible) configurations considered in the sum on the left-hand side. By the definition of $z_i^{[s]}$, we then have $\sum_{i\in F:\overline{x}_{ij}>0} \sum_{s=s_j}^k z_i^{[s]} \geq \sum_{i\in F:\overline{x}_{ij}>0} \sum_{s=s_j}^k (\overline{y}_i^s - \overline{y}_i^{s+1})$. This telescopic sum reduces to $\sum_{i\in F:\overline{x}_{ij}>0} \overline{y}_i^{d_j}$, which by constraint (2.3) must be at least $\sum_{i\in F} \overline{x}_{ij}$. Using constraint (2.2), this quantity must be at least 1, proving (i).

Each facility i is opened only in the configurations $[1], [2], \ldots, [k]$, so (ii) holds.

Consider facility i, and consider all sets (i, σ) corresponding to facility i. The total cost of these sets in the fractional k-set cover solution is $\sum_{s=1}^k f_i([s]) z_i^{[s]} = \sum_{s=1}^k f_i([s])(\overline{y}_i^s - \overline{y}_i^{s+1}) = \sum_{s=1}^k \overline{y}_i^s f_i^s + \overline{y}_i^0 f_i^0$. The last equality follows using $\overline{y}_i^1 = \overline{y}_i^0$ and the observation that $f_i([s]) = f_i^0 + f_i^1 + \cdots + f_i^s$, since that is the total cost incurred if facility i is opened to serve commodities $1, 2, \ldots, s$. This proves (iii). ∎

We know that the integrality gap of k-set-cover is no more than $\log k + 1$ (see [22]). Hence we can find an integer solution $\hat{z}$ of (2.7–2.9) with cost no more than $(\log k + 1)\sum_{i\in F}\sum_{s=0}^k f_i^s \overline{y}_i^s$. We now use the solution $\hat{z}$ to obtain an integer solution $(\hat{x}, \hat{y})$.

2.7. Constructing the final integer solution. The final integer solution is guided by the preceding two steps. We start with the integer solution $\hat{z}$ to the k-set-cover instance (2.7–2.9). The facilities to open are given by

$$\hat{y}_i^\sigma = \begin{cases} 1 & \text{if } \hat{z}_i^\sigma = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The set of open facilities is therefore given by $F' = \{i \in F : \hat{y}_i^\sigma = 1 \text{ for some } \sigma \in 2^S\}$, and the configuration (set of commodities manufactured) of an open facility i is $\cup_{\sigma \in 2^S : \hat{y}_i^\sigma = 1} \sigma$. We convert $\hat{y}$ into integer variables $\hat{y}^s$ in the obvious way: Define $\hat{y}_i^s = 1$ for any (i, s) pair if and only if there exists σ such that $\hat{y}_i^\sigma = 1$ and $s \in \sigma$; otherwise, define $\hat{y}_i^s = 0$. Define $\hat{y}_i^0 = \max_s\{\hat{y}_i^s\}$.

We now describe the assignment ϕ of clients to open facilities by first describing the integer solution $\hat{x}_{ij}$. First, consider clients in R. For each such client j, because of the fact that $\hat{z}$ is an integer solution to the k-set-cover instance, there must be at least one facility $i \in F'$ such that $\overline{x}_{ij} > 0$ and $d_j \in \sigma(i)$; define $\hat{x}_{ij} = 1$ for the closest such facility, also defining $\phi(j) = i$. Next, consider a client j not in R. For this client, there must be a representative $j' \in R$; define $\phi(j) = \phi(j')$ and $\hat{x}_{ij} = 1$ for $i = \phi(j)$. Define $\hat{x}_{ij} = 0$ for all other cases.

LEMMA 2.4. *The solution $(\hat{x}, \hat{y})$ is a feasible integer solution for the formulation* (2.1–2.6).

Proof. For every client $j \in R$, the fact that $\hat{z}$ is an integer solution to the k-set-cover instance (2.7–2.9) guarantees that there is a facility in F' which serves the commodity demanded by j. This ensures that the constraints (2.2–2.3) are satisfied for all such clients. For clients $j \notin R$, the fact that $\phi(j) = \phi(j')$ for some client $j' \in R$ which demands the same commodity, as j guarantees feasibility of the same constraints. Nonnegativity and integrality are trivially satisfied; hence $(\hat{x}, \hat{y})$ is a feasible solution for (2.1–2.6). ☐

2.8. Bounding the cost of the solution. We now prove our bound on the cost of the solution $(\hat{x}, \hat{y})$. The overall idea is similar to that of [19], although several adaptations have to be made to account for the intermediate steps of our algorithm.

THEOREM 2.5. *The solution $(\hat{x}, \hat{y})$ produced by Algorithm MCFL in Figure* 2.1 *is a* $(\log k + 4)$*-approximation for the MCFL instance. Specifically, the following hold:*

(i) $\sum_{i \in F} \sum_{j \in D} c_{ij} \hat{x}_{ij} \le \frac{3}{1-\alpha} \sum_{j \in D} c_j^*$;

(ii) $\sum_{i \in F} \sum_{s=0}^k f_i^s \hat{y}_i^s \le \frac{\log k + 1}{\alpha} \sum_{i \in F} \sum_{s=0}^k f_i^s y_i^{s*}$.

Proof. We first show (i). For each client $j \in R$, Lemma 2.1 guarantees that $\overline{x}_{ij} > 0$ only if $c_{ij} \le \frac{c_j^*}{1-\alpha}$, so we also have $\hat{x}_{ij} = 1$ only if $c_{ij} \le \frac{c_j^*}{1-\alpha}$. Next, consider a client $j' \notin R$, where $j \in R$ is its representative. By Proposition 2.2, $c_{jj'} \le 2c_{j'}^\alpha \le \frac{2}{1-\alpha} c_{j'}^*$. Since client j' is assigned to the facility $\phi(j)$ that serves j, we have $c_{\phi(j)j'} \le c_{\phi(j)j} + c_{jj'} \le c_j^\alpha + \frac{2}{1-\alpha} c_{j'}^* \le \frac{3}{1-\alpha} c_{j'}^*$. Summing the bound on $c_{\phi(j)j}$ over all clients $j \in D$ results in (i); that is, $\sum_{i \in F} \sum_{j \in D} c_{ij} \hat{x}_{ij} \le \frac{3}{1-\alpha} \sum_{j \in D} c_j^*$.

Next, we establish (ii). From the discussion in section 2.6, it follows that the solution $\hat{y}$ is a $(\log k + 1)$-approximation to the k-set-cover instance; hence, $\sum_{i \in F} \sum_{s=0}^k f_i^s \hat{y}_i^s \le (\log k + 1) \sum_{i \in F} \sum_{s=0}^k f_i^s \overline{y}_i^s$. Since we defined $\overline{y}_i^s = \min\{1, y_i^{s*}/\alpha\}$, we have (ii).

Therefore, we find that the cost of the integer solution $(\hat{x}, \hat{y})$ is bounded from above by $\max\{\frac{3}{1-\alpha}, \frac{\log k + 1}{\alpha}\} z^*$. This quantity is minimized when $\frac{3}{1-\alpha} = \frac{\log k + 1}{\alpha}$, which is attained at $\alpha = \frac{\log k + 1}{\log k + 4}$. This results in the approximation ratio being $\log k + 4$. ☐

Notice that in the above result we might be able to trade off α with respect to k to get improved guarantees for specific instances depending on the relative contributions of facility costs and service costs in an LP relaxation. Furthermore, if additional constraints are imposed to strengthen the linear relaxation, one might be able to obtain a better integer solution. It is an open question whether such strengthenings can also help improve the approximation ratio, although in the light of our hardness results in section 4, improvements beyond some constant times $\log k$ are unlikely.

3. General (nonlinear) facility costs. Recall that in our model for MCFL, we impose a linearity requirement on the cost structure of the facilities. A more general class of cost functions is *subadditive* functions. Here, apart from $f_i(\sigma) \geq 0$, the only requirement is that for any two configurations σ and σ' and any facility i, we have $f_i(\sigma \cup \sigma') \leq f_i(\sigma) + f_i(\sigma')$. Observe that the linear cost functions considered in section 2 are subadditive, so that this is a larger class of cost functions.

The main problem with allowing more general cost functions is that the input size may no longer be polynomial in the number of commodities. If $f_i(\sigma)$ must be specified for every configuration σ, then for each facility we need a total of 2^k numbers to fully specify its cost structure. Therefore, the size of the input is polynomial in $(n, m, 2^k)$.

We find that some minor modifications of our MCFL algorithm allow us to extend our algorithm to provide a $(\log k + 4)$-approximation for this more general class of MCFL problems as well. However, the running time of the algorithm, while polynomial in the input size, is now exponential in the number of commodities. In the rest of this section, we provide details of the algorithm and analysis for this general case.

The MCFL problem is closely related to the set-cover problem—both because our algorithm uses a set-cover subroutine, and because the hardness result follows from the hardness of set-cover. Note that the weighted set-cover with up to k elements per set and a total of n elements to cover was shown by Halldórsson [9] to be MAX-SNP-hard even when we allow the algorithm running time polynomial in $(n, m, 2^k)$, implying the same for our general MCFL instance. This provides further evidence about the intractability of MCFL with general cost functions.

The most general structure on facility cost functions is obtained by removing the subadditivity requirement. An example of a facility cost structure that is not subadditive is seen when the cost of manufacturing 2 or fewer commodities is finite, but manufacturing 3 or more commodities is infinitely costly. One observes that such a cost function can be used to model capacity constraints on facilities, for which no IP formulation with a bounded integrality gap is known. We note, however, that there are constant factor approximation algorithms for capacitated facility location that use local search [14, 12, 23]; it is an open question whether one can effectively use local search for MCFL. The MCFL problem with no restriction on facility costs can be modeled as a general nonmetric UFL problem. For such problems, depending on the cost structure, it may be possible to recast the MCFL problem as a set-cover problem, which would yield an $O(\log n)$-approximation. An example of such a cost structure is seen when all client-facility transportation costs are identical. It is an open question whether one can obtain tighter approximation ratios for these more general versions of MCFL.

3.1. Algorithm for MCFL with general subadditive costs. We now show how our algorithm for MCFL can be extended to the more general case when facility costs are subadditive, but not necessarily linear. We label this problem S-MCFL (where the S indicates subadditivity). Recall that the cost function is subadditive if, for every facility i and every pair of configurations σ and σ', we have $f_i(\sigma \cup \sigma') \leq f_i(\sigma) + f_i(\sigma')$. The model is otherwise identical to the description in section 2.1.

	Algorithm S-MCFL(S, D, d, F, f, c)
Given:	(i) Set of commodities S. (ii) Set of clients D, with client j demanding commodity $d_j \in S$. (iii) Set of facilities F, with the cost of facility i to serve commodities $\sigma \subseteq 2^S$ given by $f_i(\sigma)$. (iv) Distance matrix c, with c_{ij} denoting the distance from facility i to client j.
Returns:	(i) A set of facilities $F' \subseteq F$ to open, and for each facility $f \in F'$, the set of commodities $\sigma(f)$ to be manufactured. (ii) For each client j, an assignment to an open facility $\phi(j) \in F'$ that manufactures the commodity required at j; that is, $d_j \in \sigma(\phi(j))$.
1:	Formulate the integer program (3.1–3.5), and let (x^*, y^*) be an optimal solution to its linear relaxation.
2:	Obtain the "filtered" solution $(\overline{x}, \overline{y})$ as described in section 3.2.
3:	Select a subset of clients R as representatives, as described in section 3.2. All clients not in R are assigned a representative in R.
4:	Solve the k-set-cover instance induced by $(\overline{x}, \overline{y})$ and R as described in section 3.3. Let $\hat{z}$ denote the integer solution to this instance.
5:	Construct an integer solution $(\hat{x}, \hat{y})$ for the problem using $\hat{z}$, as described in section 3.3. The assignment ϕ is given by $\hat{x}_{ij}$ for clients in R, and the assignment of the representatives for clients not in R. The facilities to open and their configurations are given by $\hat{y}$.

FIG. 3.1. *Algorithm* **S-MCFL**(S, D, d, F, f, c).

Our algorithm for S-MCFL is very similar to that of MCFL. There are two main differences: First, the IP formulation is different because the formulation for MCFL relies critically on the linear nature of the cost function; second, the k-set-cover rounding is different, once again because we cannot exploit the linear nature of the cost function. In the rest of this section, we describe these differences in detail. The algorithm is described briefly in Figure 3.1.

3.2. IP formulation and initial rounding. The IP formulation (3.1–3.5) shown below extends Balinski's formulation [3] of UFL to S-MCFL, and is a minor modification of the formulation (2.1–2.6) for MCFL. We maintain an indicator variable y_i^σ for each configuration σ of each facility i. Variable x_{ij}^σ is 1 if and only if client j is served by facility i in configuration σ, as follows:

$$\min \sum_{i \in F} \sum_{j \in D} \sum_{\sigma \in 2^S} c_{ij} x_{ij}^\sigma + \sum_{i \in F} \sum_{\sigma \in 2^S} f_i(\sigma) y_i^\sigma, \tag{3.1}$$

$$\sum_{i \in F} \sum_{\sigma : d_j \in \sigma} x_{ij}^\sigma \geq 1 \qquad \forall j \in D, \tag{3.2}$$

$$x_{ij}^\sigma \leq y_i^\sigma \qquad \forall j \in D,\ \forall i \in F,\ \forall \sigma \in 2^S, \tag{3.3}$$

$$x_{ij}^\sigma, y_i^\sigma \geq 0 \qquad \forall j \in D,\ \forall i \in F,\ \forall \sigma \in 2^S, \tag{3.4}$$

$$x_{ij}^\sigma, y_i^\sigma \in \{0, 1\} \qquad \forall j \in D,\ \forall i \in F,\ \forall \sigma \in 2^S. \tag{3.5}$$

The objective function is the sum of the total "service" costs (the transportation costs from each client to the facility serving it) and the total "facility" costs (the cost of opening each facility in its desired configuration). The constraints (3.2) ensure that each client is served by at least one facility, while the constraints (3.3) ensure that a facility must be open if it is serving some client.

Observe that we do not have constraints of the form $\sum_{\sigma \in 2^S} y_i^\sigma \leq 1$ for facilities i. That is, we do not explicitly enforce that each facility be opened in no more than one configuration. Instead, this is implicitly enforced in the optimal solution, since subadditive costs guarantee that if a facility is opened in two or more configurations, the cost of opening it in the union of all open configurations is no greater.

By dropping the integrality constraints (3.5) from the integer program above, we obtain its linear relaxation (3.1–3.4). The size of the linear program is polynomial in n, m, and 2^k; therefore, it can be solved to optimality in running time polynomial in its size. We use the optimal solution of the linear relaxation, denoted (x^*, y^*) with the objective function value z^*, as our lower bound.

We then convert the solution (x^*, y^*) to the filtered solution $(\overline{x}, \overline{y})$ as described in section 2.4, with a few appropriate modifications. Lemma 2.1 continues to hold.

We also select the set of representatives R as described in section 2.5. Proposition 2.2 also continues to hold. So far, nothing has changed because until this point in the execution of our algorithm for MCFL, the linearity of the facility costs has not been used.

3.3. Interpretation as a fractional *k*-set cover solution. The fractional solution $(\overline{x}, \overline{y})$ can be used to construct a k-set-cover instance as described in section 2.6 and the formulation (2.7–2.9). The following lemma, stated here without proof, is a counterpart of Lemma 2.3 and allows us to use the k-set-cover rounding procedure to construct an integer solution.

LEMMA 3.1. *The formulation* (2.7–2.9) *is an instance of k-set-cover, and $z = \overline{y}$ is a feasible solution for its linear relaxation of cost no more than* $\sum_{i \in F} \sum_{\sigma \in 2^S} f_i(\sigma) \overline{y}_i^\sigma$.

In the case of linear costs, we found that z could be transformed into a fractional solution to the k-set-cover instance of size polynomial in (n, m, k), which allowed for rounding in time polynomial in (n, m, k) (Lemma 2.3). However, we no longer have linear costs, so we cannot find a fractional solution of size polynomial in k. Instead, we must resort to standard set-cover rounding algorithms [22], where the running time is polynomial in the input size (therefore, polynomial in 2^k). Fortunately, the integrality ratio of these general rounding procedures is still $(\log k + 1)$. As before, let $\hat{z}$ denote this integer solution.

The final integer solution for the MCFL problem is given as before. We set $\hat{y}_i^\sigma$ to 1 if $\hat{z}_i^\sigma = 1$, and 0 otherwise. Each client is assigned to the nearest open facility which manufactures the commodity required by the client; this gives us the $\hat{x}$ component of the final solution.

Observe that Theorem 2.5 also does not rely in any way on the special cost structure of the facilities. Therefore, its analog for S-MCFL continues to hold and bound the cost of the solution. For completeness, we state the theorem here, but omit the proof since it is identical to that of Theorem 2.5.

THEOREM 3.2. *The solution $(\hat{x}, \hat{y})$ produced by Algorithm S-MCFL in Figure* 3.1 *is a* $(\log k + 4)$*-approximation for the S-MCFL instance. Specifically, the following hold:*

(i) $\sum_{i \in F} \sum_{j \in D} \sum_{\sigma \in 2^S} c_{ij} \hat{x}_{ij}^\sigma \leq \frac{3}{1-\alpha} \sum_{j \in D} \sum_{i \in F} \sum_{\sigma \in 2^S} c_{ij} x_{ij}^{*\sigma}$ *and*

(ii) $\sum_{i \in F} \sum_{\sigma \in 2^S} f_i(\sigma) \hat{y}_i^\sigma \leq \frac{\log k + 1}{\alpha} \sum_{i \in F} \sum_{\sigma \in 2^S} f_i(\sigma) y_i^{*\sigma}$.

Therefore, Algorithm S-MCFL is an $(\log k + 4)$-approximation algorithm for multicommodity facility location with subadditive costs.

4. Hardness and extensions.

4.1. Hardness of approximation. While the logarithmic approximation ratio we obtain for the multi-commodity facility location problem may not be entirely satisfactory, we find that it is the best one can hope for up to constant factors. The problem is not only NP-complete, it is also inapproximable to a factor better than $\Omega(\log k)$, where k is the total number of different commodities in the problem instance. In this section, we prove this inapproximability by reducing the set-cover problem to MCFL. It is well known that the set-cover problem cannot be approximated better than $\Omega(\log k)$, where k is the total number of elements in the set-cover instance (see, for example, Arora and Sudan [1]).

An instance of a (weighted) set-cover is specified by a collection C of subsets of S, each with a weight w_c. The objective is to find a minimum weight subcollection $C' \subseteq C$ such that $\cup_{c \in C'} c = S$.

THEOREM 4.1. *Any ρ-approximation algorithm for MCFL can be used to derive a ρ-approximation algorithm for weighted set cover.*

Proof. Given an instance of a weighted set-cover specified by (S, C, w), we transform it into an instance of MCFL. The set of commodities is the set of elements S. We also have one client $d \in D$ for each commodity $s \in S$, and a facility i for every set $c \in C$. The distance between any client and any facility is zero. The cost function of facility i is $f_i(\sigma) = w_c$ if $\sigma \subseteq c$ and $f_i(\sigma) = \infty$ otherwise. Observe that this facility cost structure is linear, using the following specification: $f_i^0 = w_c$; if $s \in c$, then $f_i^s = 0$; and if $s \notin c$, then $f_i^s = \infty$.

Any solution to this MCFL instance with cost less than ∞ is a feasible solution to the set-cover instance. Moreover, any minimal solution selects each open facility in only one configuration (namely, its maximal configuration), so that the total cost of the MCFL solution is the same as the cost of the associated set-cover solution. Hence any ρ approximation for MCFL yields a ρ approximation for weighted set-cover. ∎

4.2. MCFL with restrictions on the number of commodities per facility. The MCFL problem studied in section 2 assumes that all commodities can be manufactured at all facilities. Frequently, the situation is such that any given facility can manufacture only a subset of the commodities, and one must choose which commodities to manufacture only from that subset. We find that if this is the case, our algorithm has a better performance ratio.

In particular, suppose that while the total number of commodities is k (as before), each facility has a list of at most t commodities that can be manufactured there, where $t < k$. If we run the same algorithm as before, we find that instead of a k-set-cover instance we now have to solve a t-set-cover instance. Since the approximation ratio of t-set-cover is $(\log t + 1)$, the approximation ratio of our solution is $(\log t + 4)$. If t is much smaller than k, this could represent a substantial improvement in the approximation ratio.

4.3. Extensions. When defining our model in section 2.1, we had established that each client requires exactly one unit of exactly one commodity. As noted, relaxing this assumption to allow more than one commodity to be demanded by a single client is straightforward provided we allow solutions where different commodities at a client are served by possibly different facilities. To do this, one would simply create as many copies of the client as the commodities it requires, and then have each client require a single commodity. The increase in the size of the problem by this construction is still polynomial, and the algorithm would work exactly as described.

5. Conclusion. We have introduced and provided approximation algorithms for MCFL problems. Our algorithm for MCFL matches the best known approximation ratio for the appropriate set covering variant it generalizes. Since the number of commodities t involved in a facility is likely to be much smaller than the number of clients n, the $O(\log t)$-approximation ratio represents a somewhat credible heuristic for this problem.

Several open questions remain. The empirical performance of our algorithms, possibly combined with other heuristics, might reveal further insights into the important issues one faces when locating facilities to serve multiple commodities. There may be other natural restrictions on the facility cost functions that enable constant-factor approximation algorithms; some such restrictions were studied by Shmoys, Swamy, and Levi [18]. Our study focuses exclusively on the facility location component of the supply chain; combining this work with routing and inventory decisions would significantly advance the study of the design of supply chains.

Acknowledgments. We thank the participants of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms conference as well as several referees and readers of this manuscript for several suggestions that have improved the paper significantly.

REFERENCES

[1] S. Arora and M. Sudan, *Improved low degree testing and its applications*, Combinatorica, 23 (2003), pp. 365–426.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit, *Local search heuristics for k-median and facility location problems*, SIAM J. Comput., 33 (2004), pp. 544–562.

[3] M. Balinski, *On finding integer solutions to linear programs*, in Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems, IBM Data Process Division, White Plains, NY, 1966, pp. 225–248.

[4] J. Byrka, *An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem*, in Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization and Combinatorial Optimization, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 29–43.

[5] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey, *The uncapacitated facility location problem*, in Discrete Location Theory, P. Mirchandani and R. Francis, eds., Wiley, 1990, pp. 119–171.

[6] M. Daskin, L. V. Snyder, and R. T. Berger, *Facility location in supply chain design*, in Logistics Systems: Design and Operation, A. Langevin and D. Riopel, eds., Springer, New York, 2005, pp. 39–73.

[7] R. Delaney and R. Wilson, 14*th Annual State of Logistics Report*, Cass/ProLogis, Council of Supply Chain Management Professionals, Lombard, IL, 2003.

[8] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms, 31 (1999), pp. 228–248.

[9] M. M. Halldórsson, *Approximating k-set cover and complementary graph coloring*, in Proceedings of the 5th International Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1084, Springer, Berlin, 1996, pp. 118–131.

[10] K. Jain and V. V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation*, J. ACM, 48 (2001), pp. 274–296.

[11] J.-H. Lin and J. S. Vitter, *ϵ-approximations with minimum packing constraint violation*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, 1992, pp. 771–782.

[12] M. Mahdian and M. Pál, *Universal facility location*, in Proceedings of the 11th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 2832, Springer, Berlin, 2003, pp. 409–421.

[13] A. Meyerson, *Online facility location*, in Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Soc., Los Alamitos, CA, 2001, pp. 426–431.

[14] M. Pal, E. Tardos, and T. Wexler, *Facility location with nonuniform hard capacities*, in Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Soc., Los Alamitos, CA, 2001, pp. 329–338.

[15] R. Ravi and A. Sinha, *Multicommodity facility location*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2004, pp. 342–349.

[16] R. Ravi and A. Sinha, *Approximation algorithms for problems combining facility location and network design*, Oper. Res., 54 (2006), pp. 73–81.

[17] L. Shi, R. R. Meyer, M. Bozbay, and A. J. Miller, *A nested partitions framework for solving large-scale multicommodity facility location problems*, J. Systems Science and Systems Engineering, 13 (2004), pp. 158–179.

[18] D. B. Shmoys, C. Swamy, and R. Levi, *Facility location with service installation costs*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2004, pp. 1088–1097.

[19] D. B. Shmoys, E. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, ACM, New York, 1997, pp. 265–274.

[20] Z. Svitkina and E. Tardos, *Facility location with hierarchical facility costs*, in Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2006, pp. 153–161.

[21] Toyota, *Toyota Motor Manufacturing North America, Inc. About Toyota: Operations–Manufacturing*, 2005, available online at http://www.toyota.com/about/operations/manufacturing/index.html.

[22] V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.

[23] J. Zhang, B. Chen, and Y. Ye, *A multiexchange local search algorithm for the capacitated facility location problem*, Math. Oper. Res., 30 (2005), pp. 389–403.