DIMACS Technical Report 98-30
June 1998

Analysis of a Local Search Heuristic for Facility Location Problems[1]

by

Madhukar R. Korupolu[2] *C. Greg Plaxton*[3] *Rajmohan Rajaraman*[4]

[1]A preliminary version of this paper appears in Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pages 1–10.
[2]Department of Computer Science, University of Texas, Austin TX 78712. Supported by the National Science Foundation under Grant No. CCR–9504145. Email: madhukar@cs.utexas.edu.
[3]Department of Computer Science, University of Texas, Austin TX 78712. Supported by the National Science Foundation under Grant No. CCR–9504145. Email: plaxton@cs.utexas.edu.
[4]DIMACS Center, Rutgers University, Piscataway, NJ 08854. Email: rraj@dimacs.rutgers.edu.

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Bellcore and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

In this paper, we study approximation algorithms for several NP-hard facility location problems. We prove that a simple local search heuristic yields polynomial-time constant-factor approximation bounds for the metric versions of the uncapacitated k-median problem and the uncapacitated facility location problem. (For the k-median problem, our algorithms require a constant-factor blowup in the parameter k.) This local search heuristic was first proposed several decades ago, and has been shown to exhibit good practical performance in empirical studies. We also extend the above results to obtain constant-factor approximation bounds for the metric versions of capacitated k-median and facility location problems.

1 Introduction

Suppose that you plan to build a new chain of widget stores in a given city, and you have identified potential store sites in a number of different neighborhoods. Further assume that the demand for widgets in each neighborhood of the city is known. If you want to build exactly k stores, where should you locate them in order to minimize the average traveling distance of your customers? If instead you are willing to build any number of stores, and the cost of building a store at each potential site is known, where should you build stores in order to minimize the sum of the total construction cost and the average traveling distance of your customers?

The preceding questions are examples of *facility location problems*. Facility location problems model applications in diverse fields including public policy (e.g., locating fire stations in a city), telecommunications (e.g., locating base stations in wireless networks), and information retrieval (e.g., clustering of documents), and have been studied extensively over the past three decades. For more information on the applications of facility location problems, we refer the reader to [15, 16].

This paper is concerned with metric versions of the following specific facility location problems: (i) the uncapacitated k-median problem, (ii) the uncapacitated facility location problem, (iii) the capacitated k-median problem with unsplittable demands, (iv) the capacitated facility location problem with unsplittable demands, (v) the capacitated k-median problem with splittable demands, and (vi) the capacitated facility location problem with splittable demands. Formal definitions of these problems, all of which are known to be NP-hard, are given in Section 2.

Informally, the uncapacitated k-median (resp., facility location) problem corresponds to the first (resp., second) of the two questions stated above, where we assume that there is no upper bound on the demand that can be satisfied by a given store, and hence the traveling distance of a customer is just the distance from that customer's neighborhood to the nearest store. In the capacitated versions of these problems, there is an upper bound on the demand that can be satisfied by a given store, and we need to exhibit an assignment of customers to stores that minimizes the average traveling distance while respecting the capacity bound of each store. In assigning the demand of customers to stores, there are two natural variations to consider: (i) the demand of a customer must be met by a single store (unsplittable demands), and (ii) the demand of a customer may be divided across any number of stores (splittable demands).

The input to a facility location problem includes a distance matrix specifying the traveling distance from node i to node j, for all i and j. (The nodes correspond to the neighborhoods in our widget chain example.) In the *metric version* of a facility location problem, it is assumed that these distances are nonnegative, symmetric, and satisfy the triangle inequality. All of our results are for metric facility location problems.

The local search heuristic for facility location problems, described more formally in Section 3, is extremely straightforward. The idea is to start with any feasible solution (set of stores) and then to iteratively improve the solution by repeatedly moving to the best "neighboring" feasible solution, where one solution is a neighbor of another if it can be obtained by either adding a facility (store), deleting a facility, or changing the location of a facility. This heuristic was proposed by Kuehn and Hamburger [12], and was subsequently shown to exhibit good practical performance in empirical studies (see, e.g., [5, 19]).

For the k-median problems we consider (facility location variants (i), (iii), and (v) above), we define an (a, b)-*approximation algorithm* as a polynomial-time algorithm that computes a solution using at most bk facilities and with cost at most a times the cost of an optimal solution using at most k facilities. For facility location variants (ii), (iv), and (vi), we define an a-*approximation algorithm* as a polynomial-time algorithm that computes a solution with cost at most a times optimal. Using this terminology, and letting n denote the number of nodes in the input instance, we now summarize the previous approximation results known for the facility location problems considered in this paper: (i) Hochbaum [7] showed that a simple greedy algorithm is an $O(\log n)$-approximation algorithm for the general (i.e., not restricted to metric distances) uncapacitated facility location problem; (ii) Lin and Vitter [14] gave a $(1+\varepsilon, (1+1/\varepsilon)(\ln n+1))$-approximation algorithm for the general uncapacitated k-median problem (their algorithm can also be adapted to match Hochbaum's result); (iii) Lin and Vitter [13] gave a $(2(1+\varepsilon), 1+(1/\varepsilon))$-approximation algorithm for the metric k-median problem; (iv) Shmoys et al. [18] gave a 3.16-approximation algorithm for the metric uncapacitated facility location problem, a 7-approximation algorithm for the metric capacitated facility location problem with splittable demands that requires a blowup of 7/2 in the capacity of each facility, and a 9-approximation algorithm for the metric capacitated facility location problem with unsplittable demands that requires a blowup of 4 in the capacity of each facility. More recently, the result of [18] for the metric uncapacitated facility location problem has been improved by Guha and Khuller [6] and Chudak [3], who give 2.41-approximation and 1.74-approximation algorithms, respectively. Guha and Khuller [6] also show that there is no polynomial-time 7/6-approximation algorithm for the metric uncapacitated facility location problem unless P = NP. (It is also shown in [6] that there is no polynomial-time 1.46-approximation algorithm unless NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$.)

The approximation algorithms of Lin and Vitter [13, 14] and Shmoys et al. [18] are based on careful rounding of fractional solutions to linear programming relaxations. Thus, these algorithms are substantially more sophisticated than the local search heuristic analyzed in this paper. (Note that known polynomial-time algorithms for linear programming [9, 10] are quite complicated.) Nevertheless, we prove that the local search heuristic yields simple algorithms for the uncapacitated k-median and the uncapacitated facility location problems with approximation bounds matching (to within constant factors) the corresponding results in (iii) and (iv) above. In particular, for the metric uncapacitated k-median problem, we prove that for any constant $\varepsilon > 0$, local search yields (a, b)-approximation bounds with $(a, b) = (1+\varepsilon, 3+5/\varepsilon)$ and $(a, b) = (1+5/\varepsilon, 3+\varepsilon)$. For the metric uncapacitated facility location problem, we prove that for any constant $\varepsilon > 0$, local search yields a $(5+\varepsilon)$-approximation algorithm.

Our results for the capacitated facility location problems are similar. For the metric capacitated k-median problem with splittable demands, we prove that for any constant $\varepsilon > 0$,

Problem	*Bound*	*Reference*
metric k-median	$(O(\log k \log\log k), 1)$	[2]
	$(1+\varepsilon, (1+1/\varepsilon)(\ln n + 1))$	[14]
	$(2(1+\varepsilon), 1+(1/\varepsilon))$	[13]
	$(1+\varepsilon, 3+5/\varepsilon)$	this paper
	$(1+5/\varepsilon, 3+\varepsilon)$	this paper
Euclidean k-median	$(1+\varepsilon, 1)$	[1]
general facility location	$O(\log n)$	[7]
metric facility location	≈ 3.16	[18]
	≈ 2.41	[6]
	≈ 1.74	[3]
	$5+\varepsilon$	this paper

Table 1: Approximation bounds for uncapacitated facility location problems.

local search yields (a,b)-approximation bounds with $(a,b) = (1+\varepsilon, 5+5/\varepsilon)$ and $(a,b) = (1+5/\varepsilon, 5+\varepsilon)$ with no blowup in the capacity of each facility. For the metric capacitated facility location problems with splittable demands, we prove that for any constant $\varepsilon > 0$, local search yields an $(8+\varepsilon)$-approximation algorithm with no blowup in the capacity of each facility. Using the reduction of [18] that is based on solving a special case of the generalized assignment problem [17], the above results for the capacitated problems with splittable demands imply constant-factor approximation bounds for the capacitated problems with unsplittable demands assuming a factor of 2 blowup in the capacity of each facility.

For the k-median problem, one shortcoming of the approximation bounds that we have discussed thus far is that while the optimal solution uses at most k facilities, the number of facilities used in the approximate solutions may exceed k by a constant factor. In recent work, Charikar et al. [2] and Arora et al. [1] have obtained approximation algorithms for certain k-median problems which do not incur a blowup in the number of facilities. In particular, Charikar et al. [2] give an $O(\log k \log\log k)$-approximation algorithm for the metric uncapacitated k-median problem, while Arora et al. [1] give a polynomial-time approximation scheme for the special case of the metric uncapacitated k-median problem in which the distances are Euclidean.

The approximation bounds discussed above are summarized in Tables 1 and 2. We remark that the analysis presented in this paper is somewhat sharper than that provided in the preliminary version [11]; as a result, we have obtained constant-factor improvements in all of our bounds.

The facility location problems that we consider in this paper are closely related to a number of other optimization problems (e.g., the k-center problem) that have been studied in the computer science and operations research literature. We refer the reader to [4, 8] for more information about other problems related to facility location.

The rest of the paper is organized as follows. Section 2 formally defines the six facility

Problem	*Bound*	*Capacity Blowup*	*Reference*
metric k-median, splittable demands	$(1+\varepsilon, 5+5/\varepsilon)$	none	this paper
	$(1+5/\varepsilon, 5+\varepsilon)$	none	this paper
metric facility location, splittable demands	7	7/2	[18]
	$8+\varepsilon$	none	this paper
metric k-median, unsplittable demands	$(1+\varepsilon, 5+5/\varepsilon)$	2	this paper
	$(1+5/\varepsilon, 5+\varepsilon)$	2	this paper
metric facility location, unsplittable demands	9	4	[18]
	$16+\varepsilon$	2	this paper

Table 2: Approximation bounds for capacitated facility location problems.

location problems considered in this paper. Section 3 defines the local search heuristic that we analyze. Section 3.1 explains how to find an initial feasible solution. Sections 5 and 6 analyze the uncapacitated k-median and facility location problems, respectively. Sections 8 and 9 analyze the capacitated k-median problem and the capacitated facility location problem with splittable demands, respectively. Section 10 extends these results to capacitated problems with unsplittable demands. Finally, Appendices A and B present some proofs that are deferred in earlier sections.

2 Facility Location Problems

Let $N = \{1, \ldots, n\}$ be a set of locations and $F \subseteq N$ be a set of locations at which we may open a facility. Each location j in N has a demand d_j that must be shipped to j. For any two locations i and j, let c_{ij} denote the cost of shipping a unit of demand from i to j. We assume that the costs are nonnegative, symmetric, and satisfy the triangle inequality. In each problem studied in this paper, we wish to determine a set of open facilities and an assignment of locations to open facilities such that a given objective function is minimized.

In the *uncapacitated k-median problem*, we seek a set of at most k open facilities and an assignment of locations to open facilities such that the shipping cost of the solution is minimized, where the shipping cost associated with a set S of open facilities and an assignment $\sigma : N \mapsto S$ is given by $\sum_{j \in N} d_j c_{j\sigma(j)}$. Given a set S of open facilities, an assignment that minimizes the shipping cost may be obtained by assigning each location j in N to the closest open facility in S. Thus, any solution to the uncapacitated k-median problem is completely characterized by the set of open facilities. For any set S of open facilities, let $C_s(S)$ denote the shipping cost obtained by assigning each location to its closest open facility.

In the *uncapacitated facility location problem*, for each location i in F, we are given a nonnegative cost f_i of opening a facility at i. The problem is to determine a set of open facilities and an assignment of locations to facilities such that the sum of the cost of opening the facilities and the shipping cost is minimized. (The shipping cost is defined as in the uncapacitated k-median problem.) As in the uncapacitated k-median problem, given a set S of open facilities, an assignment that minimizes the total cost is to assign each location j in N to the closest open facility in S. Therefore, any solution to this problem is characterized by the set of open facilities. Given any solution S of open facilities, we let $C_s(S)$, $C_f(S)$, and $C(S)$ denote the shipping cost, facility cost, and the total cost, respectively, obtained by assigning each location to its closest open facility. (Formally, $C_f(S) = \sum_{i \in S} f_i$ and $C(S) = C_s(S) + C_f(S)$.)

We now define *capacitated* variants of the above two problems in which there is a bound M on the total demand that can be shipped from any facility. The capacitated k-median (resp., facility location) problem is defined in the same manner as the uncapacitated k-median (resp., facility location) problem except that any solution must satisfy the additional constraint imposed by the capacity M. That is, a solution given by a set S of open facilities and an assignment σ is feasible only if the capacity constraints are respected at all facilities in S. Two natural variants arise here: (i) *splittable demands*, where the demand of each location can be split across more than one facility, and (ii) *unsplittable demands*, where the demand of each location has to be shipped from a single facility. For the capacitated problems with splittable demands, given a set S of open facilities, an assignment is given by a function $\sigma : N \times S \mapsto \mathbf{R}$, where $\sigma(i, j)$ denotes the amount of demand shipped to location i from facility j. For the capacitated problems with unsplittable demands, an assignment is given by a function $\sigma : N \mapsto S$, as in the uncapacitated problems.

In the uncapacitated problems, given a set of open facilities, an optimal assignment is obtained by simply assigning each location to its closest open facility. In a capacitated problem, however, such an assignment may violate the capacity constraint. Fortunately, for the capacitated problems with splittable demands, given a set S of open facilities, an optimal assignment can be computed in polynomial time by solving an appropriately defined instance of the transportation problem [18]. Thus, for the capacitated variants with splittable demands, any solution is completely characterized by the set S of open facilities. For the capacitated problems with unsplittable demands, however, it is NP-hard to compute an optimal assignment for a given set S of open facilities. Therefore, we require a solution to one of the capacitated problems with unsplittable demands to specify a feasible assignment together with the set of open facilities.

In the k-median problems, note that the total cost for a solution S is same as the shipping cost for S. Hence we use $C(S)$ and $C_s(S)$ interchangeably for the k-median problems.

3 The Local Search Paradigm

All of the algorithms considered in this paper are based on the following framework. We start with an arbitrary feasible solution and repeatedly improve the solution by performing a local search step. We show that within a polynomial number of local search steps, we arrive at a solution achieving the desired approximation factor. The problem of finding an initial feasible solution is discussed in Section 3.1. We now describe the local search step.

Let us consider the uncapacitated problems and the capacitated problems with splittable demands. Recall that any solution to these problems is completely characterized by the associated set S of open facilities. For the k-median problem, we define the *neighborhood* of S as $\{T \subseteq F : |S - T| = |T - S| = 1\}$. For the facility location problem, we define the neighborhood of S as $\{T \subseteq F : |S - T| \leq 1 \text{ and } |T - S| \leq 1\}$. Given a current solution corresponding to a set S of open facilities, the local search step sets the new solution to be a minimum-cost set T in the neighborhood of S. Since the neighborhood contains $O(n^2)$ solutions, a local search step can be performed in polynomial time.

The main motivation behind using the local search step is the following key property of the neighborhood of any solution, which is proved in Theorems 5.1, 6.1, 8.1, and 9.1 for the uncapacitated k-median problem, the uncapacitated facility location problem, the capacitated k-median problem with splittable demands, and the capacitated facility location problem with splittable demands respectively. Let S^* be an optimal solution. Then, for a sufficiently large constant ε, given any solution S such that $C(S) > (1+\varepsilon)C(S^*)$ (for the k-median problems, we also require that $|S| = (1+\alpha)k$ for some sufficiently large constant α), there exists a solution T in the neighborhood of S such that $C(T) \leq C(S)(1 - 1/p(n))$, where $p(n)$ is a fixed polynomial in n. Thus, if we start with a solution S_0 and perform $p(n)\log(C(S_0)/C(S^*))$ local search steps, we will arrive at a solution that has cost at most $(1+\varepsilon)C(S^*)$. Since $\log(C(S_0))$ is polynomial in the input size, the number of local search steps involved is also polynomial in the input size. Moreover, each local search step takes polynomial time. We thus obtain a polynomial-time algorithm achieving an approximation factor of $(1+\varepsilon)$.

3.1 Finding an Initial Feasible Solution

In this subsection we show how to obtain an initial feasible solution for the uncapacitated problems and the capacitated problems with splittable demands. The task of selecting a valid set of open facilities is trivial. For the facility location problems, we select an arbitrary subset S of F. For the k-median problems, we require that $|S| = (1+\alpha)k$ for some sufficiently large constant α. In the extreme case where the total number of available facilities, $|F|$, is less than $(1+\alpha)k$, our algorithm trivially returns F as the solution. Hence throughout this paper, we assume without loss of generality that $|F|$ is at least $(1+\alpha)k$. Given this assumption, we obtain an initial feasible solution for the k-median problems by selecting an arbitrary set S of $(1+\alpha)k$ facilities.

As mentioned in Section 2, the set S of open facilities chosen above completely specifies the initial feasible solution. This is because we can obtain an optimal assignment for the initial solution by simply assigning each location to its closest open facility in the case of the uncapacitated problems, and by solving an appropriately defined instance of the transportation problem in the case of the capacitated problems with splittable demands.

4 Preliminaries for Uncapacitated Problems

In this section we introduce some notations and definitions that are useful for our analysis of the uncapacitated k-median problem in Section 5 and the uncapacitated facility location problem in Section 6.

Given a set S of open facilities and an assignment σ for S, let $N_i(S,\sigma)$ denote the set of locations assigned to a facility i in S. A location j is called a *client* of facility i, under σ, if σ assigns j to i. Let $D_i(S,\sigma)$ denote the total demand shipped from a facility i in S under assignment σ. For notational convenience, the parameters S and/or σ may be dropped from the expressions $N_i(S,\sigma)$ and $D_i(S,\sigma)$ when there is no ambiguity.

Recall that an optimal assignment for a set of open facilities S assigns every client to its closest open facility. Let S^* (resp., S) denote an optimal (resp., current) set of open facilities for the problem instance under consideration, and let σ^* (resp., σ) denote an optimal assignment for S^* (resp., S). Given S, S^*, σ, and σ^*, we define g_i and g_i^* (for each facility i in S) as follows: g_i denotes the shipping cost paid by the clients of i, that is, $g_i = \sum_{j\in N_i(S,\sigma)} d_j c_{ji}$. Correspondingly, g_i^* denotes the shipping cost paid by the same set of clients (i.e., $N_i(S,\sigma)$) in the optimal solution, that is, $g_i^* = \sum_{j\in N_i(S,\sigma)} d_j c_{j\sigma^*(j)}$. Note that $C_s(S) = \sum_{i\in S} g_i$ and $C_s(S^*) = \sum_{i\in S} g_i^*$.

5 Uncapacitated k-Median

Throughout this section, we assume that $p(n)$ is a polynomial in n and that α and β are positive constants satisfying the following condition:

$$(1+\beta)\left(1-\frac{2}{\alpha}-\frac{k}{p(n)}\right) \geq 1+\frac{2}{\alpha}. \tag{1}$$

One possible choice, for example, is $\alpha = 4$, $\beta = 3$, and $p(n) = 8k$. The main result of this section is that, given a solution S of $(1+\alpha)k$ facilities with cost greater than $(1+\beta)C(S^*)$, we can perform a swapping of facilities to get another solution with the same size but significantly reduced cost.

Theorem 5.1 (Swapping Facilities) *Let S be any subset of F such that $|S| = (1+\alpha)k$ and $C(S) > (1+\beta)C(S^*)$. Then there exist u in S and v in F such that $C(S)-C(S+v-u) \geq \frac{C(S)}{p(n)}$.*

We prove the theorem in two stages. First, in Lemma 5.2 we show that we can add a facility to S and get a significant reduction in the cost. Then, in Lemma 5.3, we show that we can drop a facility from S without increasing the cost too much. Combining these two lemmas then yields the above theorem.

Lemma 5.2 (Adding a Facility) *Let S be any subset of F such that $C(S) > C(S^*)$. Then there exists v in F such that $C(S) - C(S+v) \geq \frac{C(S)-C(S^*)}{k}$.*

Proof: Recall that σ and σ^* denote the optimal assignments for S and S^*, respectively. We have

$$C(S) - C(S^*) \;=\; \sum_{j \in N} d_j (c_{j\sigma(j)} - c_{j\sigma^*(j)}) = \sum_{i \in S^*} \sum_{j \in N_i(S^*)} d_j (c_{j\sigma(j)} - c_{ji}).$$

By a simple averaging argument, it follows that there exists v in S^* for which

$$\sum_{j \in N_v(S^*)} d_j (c_{j\sigma(j)} - c_{jv}) \geq \frac{C(S) - C(S^*)}{k}.$$

Now consider the set $S + v$ along with the assignment σ' defined as follows: Set $\sigma'(j)$ to v if j belongs to $N_v(S^*)$, and to $\sigma(j)$ otherwise. Using the fact that $C(S+v, \sigma') \geq C(S+v)$, we obtain

$$C(S) - C(S+v) \;\geq\; C(S) - C(S+v, \sigma') = \sum_{j \in N_v(S^*)} d_j (c_{j\sigma(j)} - c_{jv}) \geq \frac{C(S) - C(S^*)}{k}.$$

■

Lemma 5.3 (Dropping a Facility) *Let S be any subset of F such that $|S| = (1+\alpha)k$. Then there exists u in S such that $C(S-u) - C(S) \leq 2 \cdot \frac{C(S)+C(S^*)}{\alpha k}$.*

Before proceeding to prove this lemma, we first classify the facilities in S as "primary" or "secondary" as follows. For each facility i^* in S^*, consider those facilities of S that are assigned to i^* by σ^* and pick the closest one among these, breaking ties arbitrarily. The closest facility thus chosen is said to be the *primary* facility of i^*. Let S' denote the set of all primary facilities. Facilities which are not primary are said to be *secondary*. For each secondary facility i in S, we define the associated primary facility as the primary facility of $\sigma^*(i)$. Note that the number of secondary facilities is at least $|S| - |S^*|$, which is αk. We will show that one of the secondary facilities can be dropped from S without incurring a "large" increase in the shipping cost. As a first step in this direction, the following claim gives an upper bound on the increase in the shipping cost if a client of a secondary facility i is reassigned to the primary facility associated with i.

Claim 5.4 *Consider a secondary facility i in S, and let i' be the primary facility associated with i. Then for all j in N, we have $c_{ji'} - c_{ji} \leq 2(c_{ji} + c_{j\sigma^*(j)})$.*

Proof: Let i^* denote $\sigma^*(i)$, the facility in S^* that is closest to i. Using the triangle inequality, we upper bound $c_{ji'}$ as follows:

$$c_{ji'} \;\leq\; c_{ji} + c_{ii^*} + c_{i^*i'} \leq c_{ji} + 2c_{ii^*} \leq c_{ji} + 2c_{i\sigma^*(j)} \leq c_{ji} + 2c_{ij} + 2c_{j\sigma^*(j)}.$$

(The second step uses the fact that i' is the primary associated with i^*, i.e., $c_{i^*i'} \leq c_{ii^*}$. The third step holds since i^* is the closest to i among all facilities in S^*. The fourth step uses the triangle inequality.) The claim now follows. ■

Claim 5.5 *Consider a secondary facility i in S and let i' be the primary facility associated with i. If we drop the facility i from S and reassign all clients of i to i', then the increase in the shipping cost is at most $2(g_i + g_i^*)$.*

Proof: It is easy to verify that the increase in shipping cost due to this reassignment is $\sum_{j \in N_i(S)} d_j(c_{ji'} - c_{ji})$. Using the upper bound on $c_{ji'} - c_{ji}$ given by Claim 5.4, and then plugging in the definitions of g_i and g_i^*, completes the proof. ∎

The following claim is needed only for the proof of Theorem 6.1 in Section 6, but is in the same spirit as the previous claim.

Claim 5.6 *Consider any facility i in S and let i^* denote $\sigma^*(i)$. If we drop facility i from S and reassign all clients of i to i^*, then the increase in the shipping cost is at most $g_i + g_i^*$.*

Proof: For every client j of i, we can upper bound $c_{ji^*} - c_{ji}$ by $c_{ji} + c_{j\sigma^*(j)}$ using the same approach as in Claim 5.4. The remainder of the proof is similar to that of Claim 5.5. ∎

Proof of Lemma 5.3: Recall that S' is the set of primary facilities of S. Using the definitions of g_i and g_i^*, we have

$$\sum_{i \in S - S'} (g_i + g_i^*) \leq \sum_{i \in S} (g_i + g_i^*) = C_s(S) + C_s(S^*).$$

By an averaging argument, it follows that there exists a secondary facility v for which $g_v + g_v^*$ is at most $\frac{C_s(S)+C_s(S^*)}{\alpha k}$ (since $|S - S'| \geq \alpha k$). Applying Claim 5.5 for this facility v, it follows that v satisfies the requirements of this dropping lemma. ∎

Proof of Theorem 5.1: We first apply Lemma 5.3 to drop a facility u from S. We then apply Lemma 5.2 to add a facility v to $S - u$. Combining the two operations, we have

$$C(S) - C(S - u + v) \;\; \geq \;\; \frac{C(S-u) - C(S^*)}{k} - 2 \cdot \frac{C(S) + C(S^*)}{\alpha k}.$$

It now suffices to show that the term on the right side is at least $\frac{C(S)}{p(n)}$. Since $C(S-u) \geq C(S)$, it suffices to show that

$$C(S)\left(1 - \frac{2}{\alpha} - \frac{k}{p(n)}\right) \geq C(S^*)\left(1 + \frac{2}{\alpha}\right).$$

Since $C(S) \geq (1+\beta)C(S^*)$ and α, β, and $p(n)$ satisfy Equation (1), this inequality always holds. This completes the proof of the theorem. ∎

The corollary below is related to the question of how small the constants α and β can be. Note that any choice of α and β subject to the constraint of Equation (1) is valid. It is shown below that β can be made arbitrarily small, while on the other hand, α can be made arbitrarily close to 2.

Corollary 5.7 *For any constant $\varepsilon > 0$, there is a constant c such that the following can be computed in polynomial time using the local search heuristic: (i) a solution S with $C(S) \leq (1+\varepsilon)C(S^*)$ and $|S| \leq (1+c)k$, and (ii) a solution S with $|S| \leq (3+\varepsilon)k$ and $C(S) \leq (1+c)C(S^*)$.*

Proof: It can be verified that the following choices of the constants satisfy the constraint of Equation (1): (i) $\beta = \varepsilon$, $c = \alpha = \frac{5}{\varepsilon} + 2$, and $p(n) = \alpha k(1+\varepsilon)$, and (ii) $\alpha = 2+\varepsilon$, $c = \beta = 1 + \frac{5}{\varepsilon}$, and $p(n) = \frac{5\alpha k}{\varepsilon}$.

The first part of the corollary follows by applying Theorem 5.1 with the first set of constants and using the local search algorithm described in Section 3. For the second part of the corollary, we use the second set of constants. ■

The following corollary addresses the issue of making α arbitrarily small. The local improvement method requires α to satisfy the constraint of Equation (1) and hence it has to be above 2. However if we postprocess the solution from the local improvement phase by dropping facilities greedily, we can achieve arbitrarily small α while keeping the cost within a constant factor of optimal.

Corollary 5.8 *For any constant $\varepsilon > 0$, there is a constant c such that a solution S with $|S| \leq (1+\varepsilon)k$ and $C(S) \leq (1+c)C(S^*)$ can be computed in polynomial time using the local search heuristic.*

Proof: Suppose we have a solution S with $|S| = (1+\delta)k$ and $C(S) \leq (1+\mu)C(S^*)$. For example, by part (ii) of Corollary 5.7, we can choose $\delta = 2+\varepsilon'$ and $\mu = 1 + \frac{5}{\varepsilon'}$ for any $\varepsilon' > 0$.

Lemma 5.3 implies that there exists a facility u in S that can be dropped without increasing the cost by more than $\frac{2C(S)+2C(S^*)}{\delta k}$, which is at most $\frac{4C(S)}{\delta k}$ assuming $C(S^*) \leq C(S)$. Therefore, the cost of $S - u$ is at most $C(S)(1 + \frac{4}{\delta k})$. We repeat this dropping process until the number of facilities reduces to $(1+\varepsilon)k$. Let $\hat{S}$ be the set of facilities remaining at the end of this process. Since the $(t+1)$th facility dropped increases the cost by at most a multiplicative factor of $(1 + \frac{4}{\delta k - t})$, it follows that

$$\begin{aligned} C(\hat{S}) &\leq C(S)\left[\left(1+\frac{4}{\delta k}\right)\left(1+\frac{4}{\delta k - 1}\right)\cdots\left(1+\frac{4}{\varepsilon k+1}\right)\right] \\ &= C(S)\frac{(\delta k+4)(\delta k+3)(\delta k+2)(\delta k+1)}{(\varepsilon k+4)(\varepsilon k+3)(\varepsilon k+2)(\varepsilon k+1)} \\ &\leq C(S)\left(\frac{\delta}{\varepsilon}\right)^4. \end{aligned}$$

(The second step follows by expanding and canceling out the common terms in the numerator and the denominator. The third step uses the fact that $\frac{\delta k+i}{\varepsilon k+i} \leq \frac{\delta}{\varepsilon}$, whenever $i \geq 0$ and $\delta \geq \varepsilon$.) Thus $\hat{S}$ is a solution with $|\hat{S}| = (1+\varepsilon)k$, and if δ is chosen to be $2+\varepsilon'$ (for any $\varepsilon' > 0$) we have

$$C(\hat{S}) \leq C(S^*)\left(2+\frac{5}{\varepsilon'}\right)\left(\frac{2+\varepsilon'}{\varepsilon}\right)^4.$$

The corollary follows if we set the constant c to $\left(2+\frac{5}{\varepsilon'}\right)\left(\frac{2+\varepsilon'}{\varepsilon}\right)^4 - 1$, for any $\varepsilon' > 0$.

■

6 Uncapacitated Facility Location

The main result of this section is Theorem 6.1, which shows that a local search step gives a significant improvement in the cost of the current solution if the cost of the current solution is sufficiently larger than the optimal cost. Throughout this section, we assume that $p(n)$ is a sufficiently large polynomial in n.

Theorem 6.1 *Let S be any subset of F such that $C(S) > (5+\varepsilon)C(S^*)$, for some constant $\varepsilon > 0$. Then there exists $T \subseteq F$ such that $|S-T| \le 1$, $|T-S| \le 1$, and $C(S)-C(T) \ge \frac{C(S)}{p(n)}$.*

Corollary 6.2 *For any constant $\varepsilon > 0$, the local search heuristic yields a $(5+\varepsilon)$-approximate solution in polynomial time.* ■

We prove the theorem in two stages. Firstly, in Lemma 6.3, we show that if the shipping cost of S is sufficiently large then we can add a facility to S and obtain a significant improvement in the cost. Second, in Lemma 6.4, we show that if the facility cost of S is sufficiently large then we can either drop a facility from S or substitute a facility in S to obtain a significant improvement in the cost. If none of adding, dropping, or substituting gives a good improvement in the cost, then we combine these two lemmas to obtain a contradictory upper bound on the total cost of S.

Lemma 6.3 (Adding a Facility) *Let S be any subset of F such that $C_s(S) \ge C(S^*) + \frac{nC(S)}{p(n)}$. Then there exists v in F such that $C(S) - C(S+v) \ge \frac{C(S)}{p(n)}$.*

Proof: It suffices to show that for any set $S \subseteq F$, there exists v in F, such that $C(S) - C(S+v) \ge \frac{C_s(S)-C(S^*)}{n}$. The quantity $C_s(S) - C(S^*)$ can be expanded as follows:

$$\begin{aligned} C_s(S) - C_s(S^*) - C_f(S^*) &= -C_f(S^*) + \sum_{j\in N} d_j(c_{j\sigma(j)} - c_{j\sigma^*(j)}) \\ &= \sum_{i\in S^*} -f_i + \sum_{i\in S^*}\sum_{j\in N_i(S^*)} d_j(c_{j\sigma(j)} - c_{ji}) \\ &= \sum_{i\in S^*}\left(-f_i + \sum_{j\in N_i(S^*)} d_j(c_{j\sigma(j)} - c_{ji})\right). \end{aligned}$$

By a simple averaging argument, it follows that there exists a facility v in S^* such that

$$-f_v + \sum_{j\in N_v(S^*)} d_j(c_{j\sigma(j)} - c_{jv}) \ge \frac{C_s(S) - C(S^*)}{n}. \tag{2}$$

Now consider the solution $S+v$ along with the assignment σ' defined as follows: If j is in $N_v(S^*)$, then set $\sigma'(j)$ to v; otherwise set $\sigma'(j)$ to $\sigma(j)$. Since $C(S+v, \sigma')$ is at least $C(S+v)$, Equation (2) implies that

$$C(S) - C(S+v) \;\geq\; C(S) - C(S+v, \sigma') = -f_v + \sum_{j \in N_v(S^*)} d_j(c_{j\sigma(j)} - c_{jv}) \geq \frac{C_s(S) - C(S^*)}{n},$$

completing the proof of the lemma. ■

Lemma 6.4 (Dropping/Swapping Facilities) *Let S be any subset of F. If no facility in S can be dropped or substituted (by some facility not in S) to improve the cost by at least $\frac{C(S)}{p(n)}$, then the following upper bound on the facility cost of S holds:*

$$C_f(S)\left(1 - \frac{n^2}{p(n)}\right) < 2C(S^*) + 2C_s(S) + \frac{C(S)}{n}.$$

We start by developing some useful tools for proving this lemma. A facility i in S is called *cheap* if $f_i \leq \frac{C(S)}{n^2}$, and *expensive* otherwise. Let S_C denote the set of cheap facilities in $S - S^*$. Each facility in S is also classified as *primary* or *secondary* in the same fashion as in Section 5. We let S_{EP} (resp., S_{ES}) denote the set of expensive facilities in $S - S^* - S_C$ that are classified as primary (resp., secondary). Thus $S \cap S^*$, S_C, S_{EP}, and S_{ES} partition the set S. Note that for each i in S_{EP}, the facility $i^* = \sigma^*(i)$ cannot be in S; if i^* belonged to S, then i^* would have been designated as the primary and i would not be a primary.

We start by constructing a set of *candidate operations*, one for each facility in S, as follows:

1. For i in $S \cap S^*$, do nothing.
2. For i in S_C, do nothing.
3. For i in S_{ES}, drop i and reassign the clients of i to the primary facility i' associated with i. Note that i' is already in S.
4. For i in S_{EP}, substitute i with $i^* = \sigma^*(i)$, and reassign all clients of i to i^*. Recall that i^* does not belong to S.

A candidate operation is called *good* if it reduces the cost of the solution by at least $\frac{C(S)}{p(n)}$, and *bad* otherwise. By the hypothesis of Lemma 6.4, we can assume that all of the candidate operations are *bad*. Our main tools for proving Lemma 6.4 are Claims 6.5 and 6.6, which establish the following: If the candidate operation for a facility i is bad, then the facility cost of i is "small".

We first consider S_C, the set of cheap facilities. Since the total number of facilities is at most n and since the cost of each cheap facility is at most $C(S)/n^2$, it trivially follows that

$$C_f(S_C) \leq \frac{C(S)}{n}. \tag{3}$$

Claim 6.5 *If the candidate operations for all of the expensive secondary facilities are bad, then*

$$C_f(S_{ES})\left(1-\frac{n^2}{p(n)}\right) \le \sum_{i\in S_{ES}} 2(g_i+g_i^*).$$

Proof: Consider an expensive secondary facility i in S_{ES}. The decrease in the facility cost caused by the associated candidate operation is f_i and the increase in the shipping cost, by Claim 5.5, is at most $2(g_i+g_i^*)$. So the net decrease in the cost is at least $f_i - 2(g_i+g_i^*)$. If this quantity is greater than $\frac{n^2 f_i}{p(n)}$, then since i is expensive, this candidate operation will be good. Therefore, if this candidate operation is not good, the following upper bound on the facility cost of i holds: $f_i(1-\frac{n^2}{p(n)}) < 2(g_i+g_i^*)$. Summing over all of the expensive secondary facilities yields the required upper bound on the facility cost of S_{ES}. ■

Claim 6.6 *If the candidate operations for all of the expensive primary facilities are bad, then*

$$C_f(S_{EP})\left(1-\frac{n^2}{p(n)}\right) \le C_f(S^*-S)+\sum_{i\in S_{EP}} (g_i+g_i^*).$$

Proof: Consider an expensive primary facility i in S_{EP}. The decrease in the facility cost caused by its candidate operation is $f_i - f_{i^*}$ and the increase in the shipping cost, by Claim 5.6, is at most $g_i+g_i^*$. So the net decrease in the cost is at least $f_i-(g_i+g_i^*+f_{i^*})$. If this quantity is greater than $\frac{n^2 f_i}{p(n)}$, then since i is expensive, this candidate operation is good. Therefore, if this candidate operation is not good, the following upper bound on the facility cost of i holds: $f_i(1-\frac{n^2}{p(n)}) < g_i+g_i^*+f_{i^*}$. Summing over all of the expensive primary facilities yields

$$C_f(S_{EP})\left(1-\frac{n^2}{p(n)}\right) \;<\; \sum_{i\in S_{EP}}(g_i+g_i^*+f_{i^*}) < C_f(S^*-S)+\sum_{i\in S_{EP}}(g_i+g_i^*).$$

(The second step holds since $\sigma^*(i)$ belongs to S^*-S for all i in S_{EP} and $\sigma^*(i)\ne\sigma^*(j)$ for all i and j in S_{EP} such that $i\ne j$. Hence, each facility of S^*-S occurs at most once in the summation.) ■

Proof of Lemma 6.4: Using the upper bounds from Equation (3), Claim 6.5 and Claim 6.6, and adding $C_f(S\cap S^*)$ to both sides, we obtain

$$\begin{aligned}
C_f(S)\left(1-\frac{n^2}{p(n)}\right) &< \frac{C(S)}{n}+C_f(S^*)+2\cdot\sum_{i\in S_{EP}\cup S_{ES}}(g_i+g_i^*)\\
&< \frac{C(S)}{n}+C_f(S^*)+2C_s(S)+2C_s(S^*)\\
&< \frac{C(S)}{n}+2C_s(S)+2C(S^*).
\end{aligned}$$

(The second step follows from the definition of g_i and g_i^*, and the fact that the sets S_{EP} and S_{ES} are disjoint.) ■

Proof of Theorem 6.1: If no facility can be added to S to get a cost improvement of at least $\frac{C(S)}{p(n)}$, then by Lemma 6.3, the shipping cost of S is upper-bounded as follows: $C_s(S) < C(S^*) + \frac{nC(S)}{p(n)}$. Similarly, if no facility of S can be dropped or substituted to get a cost improvement of at least $\frac{C(S)}{p(n)}$, then by Lemma 6.4 we have the following upper bound on the facility cost of S:

$$C_f(S)\left(1-\frac{n^2}{p(n)}\right) \quad < \quad 2C(S^*)+2C_s(S)+\frac{C(S)}{n} < 4C(S^*)+nC(S)\left(\frac{2}{p(n)}+\frac{1}{n^2}\right).$$

(The second step follow from the upper bound on $C_s(S)$.) Adding the upper bounds on the shipping cost and the facility cost, we have

$$C(S)\left(1-\frac{n^2}{p(n)}\right) < 5C(S^*)+nC(S)\left(\frac{3}{p(n)}+\frac{1}{n^2}\right).$$

Rearranging, we obtain

$$C(S)\left(1-\frac{n^2}{p(n)}-\frac{3n}{p(n)}-\frac{1}{n^2}\right) < 5C(S^*).$$

If $p(n)$ is chosen to be sufficiently large, then this contradicts the hypothesis (of Theorem 6.1) that $C(S) > (5+\varepsilon)C(S^*)$. Specifically, we need

$$(5+\varepsilon)\left(1-\frac{n^2}{p(n)}-\frac{3n}{p(n)}-\frac{1}{n^2}\right) > 5.$$

This completes the proof of the theorem. ■

7 Preliminaries for Capacitated Problems with Splittable Demands

In this section we introduce some notations and definitions that are useful in our analysis of the capacitated k-median problem with splittable demands (see Section 8) and the capacitated facility location problem with splittable demands (see Section 9).

Recall that for the capacitated problems with splittable demands, given a set S of open facilities, an assignment is given by a function $\sigma : N \times S \mapsto \mathbf{R}$, where $\sigma(i,j)$ denotes the amount of demand shipped to location i from facility j. Given S and an assignment σ, let $D_v(S,\sigma)$ denote the amount of demand shipped from a facility v in S under the assignment σ, that is, $D_v(S,\sigma) = \sum_{i \in N} \sigma(i,v)$. For notational convenience, we sometimes drop the parameters S and/or σ when there is no ambiguity.

Also recall that given a set S of open facilities, the corresponding optimal splittable assignment σ can be computed in polynomial time by a reduction to the transportation problem [18]. Hence, the solution to the capacitated problem with splittable demands is entirely characterized by the set S of open facilities.

We let S^* (resp., S) denote an optimal (resp., current) solution for a given instance of one of the capacitated problems with splittable demands, and let σ^* (resp., σ) denote an optimal assignment for S^* (resp., S). Recall that M denotes the upper bound on the capacity of any facility.

8 Capacitated k-Median with Splittable Demands

In this section we consider the capacitated k-median problem with splittable demands and show in Theorem 8.1 that, as in the uncapacitated case, each step of the local search heuristic yields a significant improvement in cost whenever the current cost is sufficiently higher than the optimal cost. Throughout this section, we assume that $p(n)$ is a polynomial in n and that α and β are positive constants satisfying the following constraint,

$$(1+\beta)\left(1-\frac{2}{\alpha}-\frac{k}{p(n)}\right) \geq 1+\frac{2}{\alpha}. \tag{4}$$

Note that this constraint is the same as that of Equation (1). Hence, as before, one possible choice for the parameters is $\alpha = 4$, $\beta = 3$, and $p(n) = 8k$.

Theorem 8.1 (Swapping Facilities) *Let S be any subset of F such that $|S| = (3+\alpha)k$ and $C(S) > (1+\beta)C(S^*)$. Then there exist u in S and v in F such that $C(S)-C(S+v-u) \geq \frac{C(S)}{p(n)}$.*

As in the uncapacitated case, we prove this theorem in two stages. First, in Lemma 8.2 we show that we can add a facility to S and get a significant reduction in the cost. Due to the capacity constraints, the proof of this lemma is more involved than in the uncapacitated case (Lemma 5.2). Appendix A gives the proof of this lemma after developing the required technical machinery. Then, in Lemma 8.3, we show that we can drop a facility from S without increasing the cost too much. The final step of proving the above theorem using these two lemmas is the same as in the uncapacitated case (Theorem 5.1).

Lemma 8.2 (Adding a Facility) *Let S be any subset of F such that $C(S) > C(S^*)$. Then there exists v in F such that $C(S) - C(S+v) \geq \frac{C(S)-C(S^*)}{k}$.*

Proof: Deferred to Appendix A. ■

Lemma 8.3 (Dropping a Facility) *Let S be any subset of F such that $|S| = (3+\alpha)k$. Then there exists u in S such that $C(S-u) - C(S) \leq 2 \cdot \frac{C(S)+C(S^*)}{\alpha k}$.*

The simple approach used in the uncapacitated case (Lemma 5.3) does not suffice for the proof of Lemma 8.3. The main reason is that the operation of dropping a facility i and assigning all of its clients to another facility i' may violate the capacity constraint at i'. To get around this hurdle, we use the notion of "heavy" and "light" facilities. A facility i in S is called *heavy* if $D_i(S, \sigma)$ is greater than $M/2$, and *light* otherwise. Note that dropping a light facility i and reassigning the demand $D_i(S, \sigma)$ to another light facility does not violate the capacity constraints. The number of heavy facilities can be at most $2k$, since the total demand of all clients cannot exceed kM. Therefore, the number of light facilities in S is at least $(1+\alpha)k$. Let $\hat{S}$ denote the set of light facilities in S. Our strategy is to focus on $\hat{S}$ and proceed as in the uncapacitated case. Though the fact that each client's demand might be split across several facilities introduces minor differences, the basic idea remains the same as before.

Let us define a function $\hat{\sigma} : N \mapsto S^*$ that maps each location i to the closest facility in S^*, breaking ties arbitrarily. Using $\hat{\sigma}$, we classify the light facilities in $\hat{S}$ as "primary" or "secondary", as follows: For each facility i^* in S^*, consider those light facilities of $\hat{S}$ that are assigned to i^* by $\hat{\sigma}$ and pick the closest one among these, breaking ties arbitrarily. The closest light facility thus chosen is said to be the *primary* facility of i^*. Let S' denote the set of all light facilities that are designated as primary. Light facilities of $\hat{S}$ that are not primary are said to be *secondary*. For each secondary light facility i in $\hat{S}$, we define the primary facility of $\hat{\sigma}(i)$ as the associated primary light facility of i. Note that the number of secondary light facilities is at least $|\hat{S}| - |S^*|$, which is at least αk. We will show that one of the secondary light facilities can be dropped from S without incurring a "large" increase in the shipping cost.

Claim 8.4 *Consider a secondary light facility i in $\hat{S}$, and let i' be the primary light facility associated with i. Then for all j in N, we have $c_{ji'} - c_{ji} \leq 2(c_{ji} + c_{j\hat{\sigma}(j)})$.*

Proof: By the same reasoning as in the proof of Claim 5.4, we obtain:

$$c_{ji'} \leq c_{ji} + c_{ii^*} + c_{i^*i'} \leq c_{ji} + 2c_{ii^*} \leq c_{ji} + 2c_{i\hat{\sigma}(j)} \leq c_{ji} + 2c_{ij} + 2c_{j\hat{\sigma}(j)}.$$

■

Let Δ_i denote the increase incurred in the shipping cost when a secondary light facility i is dropped and the demand $D_i(S, \sigma)$ is reassigned to the primary light facility associated with i. (Note that this reassignment does not violate any capacity constraints.)

Claim 8.5 *For any secondary light facility i, we have $\Delta_i \leq \sum_{j \in N} 2\sigma(j, i)(c_{ji} + c_{j\hat{\sigma}(j)})$.*

Proof: Let i' denote the primary light facility associated with i. The reassignment of demand $D_i(S, \sigma)$ to i' causes $\sigma(j, i)$ units of demand that was originally shipped from i to a client j to now be shipped from i' to j. Hence the increase Δ_i in shipping cost is equal to $\sum_{j \in N} \sigma(j, i)(c_{ji'} - c_{ji})$. Applying the upper bound of Claim 8.4, the result follows. ■

Proof of Lemma 8.3: It suffices to show that there exists a secondary light facility i for which $\Delta_i \leq 2 \cdot \frac{C_s(S) + C_s(S^*)}{\alpha k}$. Recall that $\hat{S} - S'$ is the set of secondary light facilities, and that $|\hat{S} - S'| \geq \alpha k$. We have

$$\begin{aligned}
\sum_{i \in \hat{S} - S'} \Delta_i &\leq 2 \cdot \sum_{i \in \hat{S} - S'} \sum_{j \in N} \sigma(j,i)(c_{ji} + c_{j\hat{\sigma}(j)}) \\
&\leq 2 \cdot \sum_{i \in S} \sum_{j \in N} \sigma(j,i)(c_{ji} + c_{j\hat{\sigma}(j)}) \\
&= 2 \cdot C_s(S) + 2 \cdot \sum_{j \in N} \sum_{i \in S} \sigma(j,i) c_{j\hat{\sigma}(j)} \\
&= 2C_s(S) + 2 \cdot \sum_{j \in N} d_j c_{j\hat{\sigma}(j)} \\
&\leq 2C_s(S) + 2C_s(S^*).
\end{aligned}$$

(The first step uses Claim 8.5. The fourth step uses the fact that $\sum_{i \in S} \sigma(j,i) = d_j$, the demand at j. The last step uses the fact that $\hat{\sigma}(j)$ is the closest facility to j among all facilities in S^*.) By a simple averaging argument, it follows that the required secondary light facility exists. ∎

Proof of Theorem 8.1: Similar to the proof of Theorem 5.1 for the uncapacitated case. ∎

We now turn to the questions of how small the constants α and β can be. Recall that the constraint of Equation (4) to be satisfied by these constants is the same as in the uncapacitated case. Hence, we obtain the following corollaries, which are analogous to Corollaries 5.7 and 5.8 in the uncapacitated case. The only difference is an additive term of $2k$ in the number of facilities used.

Corollary 8.6 *For any constant $\varepsilon > 0$, there is a constant c such that the following can be computed in polynomial time using the local search heuristic: (i) a solution S with $C(S) \leq (1+\varepsilon)C(S^*)$ and $|S| \leq (1+c)k$, and (ii) a solution S with $|S| \leq (5+\varepsilon)k$ and $C(S) \leq (1+c)C(S^*)$.*

Proof: Similar to the proof of Corollary 5.7 for the uncapacitated case. ∎

Corollary 8.7 *For any constant $\varepsilon > 0$, there is a constant c such that a solution S with $|S| \leq (3+\varepsilon)k$ and $C(S) \leq (1+c)C(S^*)$ can be computed in polynomial time.*

Proof: Similar to the proof of Corollary 5.8 for the uncapacitated case. ∎

9 Capacitated Facility Location with Splittable Demands

In this section we consider the capacitated facility location problem with splittable demands and show in Theorem 9.1 that, as in the uncapacitated case, each step of the local search heuristic yields a significant improvement in cost whenever the current cost is sufficiently higher than the optimal cost. Throughout this section, we assume that $p(n)$ is a sufficiently large polynomial in n.

Theorem 9.1 (Swapping Facilities) *Let S be any subset of F such that $C(S) > (8+\varepsilon)C(S^*)$. Then there exists $T \subseteq F$ such that $|S-T| \leq 1$, $|T-S| \leq 1$, and $C(S) - C(T) \geq \frac{C(S)}{p(n)}$.*

Corollary 9.2 *For any constant $\varepsilon > 0$, the local search heuristic yields a $(8+\varepsilon)$-approximate solution in polynomial time.* ∎

We prove the theorem in two stages, as in the uncapacitated case. First, in Lemma 9.3, we show that if the shipping cost of S is sufficiently large, then we can add a facility to S and obtain a significant improvement in the cost. Due to the capacity constraints, the proof of this lemma is more involved than in the uncapacitated case (Lemma 6.3). Appendix A gives the proof of this lemma after developing the required technical machinery. Second, in Lemma 9.4, we show that if the facility cost of S is sufficiently large then we can either drop a facility from S or substitute a facility in S to obtain a significant improvement in the cost. If none of adding, dropping, or substituting gives a significant improvement in the cost, then we combine these two lemmas to obtain a contradictory upper bound on the total cost of S.

Lemma 9.3 (Adding a Facility) *Let S be any subset of F such that $C_s(S) \geq C(S^*) + \frac{nC(S)}{p(n)}$. Then there exists a v in F such that $C(S) - C(S+v) \geq \frac{C(S)}{p(n)}$.*

Proof: Deferred to Appendix A. ∎

Lemma 9.4 (Dropping/Swapping Facilities) *Let S be any subset of F. If no facility of S can be dropped or substituted (by a facility not in S) to improve the cost by at least $\frac{C(S)}{p(n)}$, then the following upper bound on the facility cost of S holds:*

$$C_f(S)\left(1 - \frac{n^2}{p(n)}\right) < 5C(S^*) + 2C_s(S) + \frac{C(S)}{n}.$$

Before proceeding to prove Lemma 9.4, we first show that Theorem 9.1 follows from Lemmas 9.3 and 9.4.

Proof of Theorem 9.1: This is similar to the proof of Theorem 6.1, except that we use Lemmas 9.3 and 9.4 instead of Lemmas 6.3 and 6.4. In this case, we obtain

$$C(S)\left(1 - \frac{n^2}{p(n)} - \frac{3n}{p(n)} - \frac{1}{n^2}\right) < 8C(S^*),$$

which is a contradiction if $p(n)$ is chosen sufficiently large. ∎

It remains to prove Lemma 9.4. Due to the capacity constraint, the simple approach used in the uncapacitated case (Lemma 6.4) does not suffice here. In particular, some of the candidate operations may be invalid because reassigning the demand associated with a facility i to another facility j may violate the capacity constraint at j. We overcome this difficulty by studying a problem that we call the auxiliary allocation problem, the solution to which leads to a construction of suitable candidate operations. More specifically, the solution to the auxiliary allocation problem yields a valid reassignment of the demand associated with a facility that is being dropped or substituted. The auxiliary allocation problem is defined in Section 9.1. Section 9.2 contains the design and the analysis of the candidate operations. Finally, Section 9.3 completes the proof of Lemma 9.4.

9.1 The Auxiliary Allocation Problem

Given S, S^*, σ, and the original distance metric c, the *auxiliary allocation problem* R is constructed as follows:

- The nodes of $S - S^*$ are the *requestors.*
- The nodes of S^* are the *servers.*
- Each requestor node v in $S - S^*$ has a demand of $D_v(S, \sigma)$.
- Each server node v in $S \cap S^*$ has a capacity of $M - D_v(S, \sigma)$.
- Each server node v in $S^* - S$ has a capacity of M.
- The distance function between the nodes is given by c.

(For ease of notation, we drop σ and use $D_v(S)$ for $D_v(S, \sigma)$ throughout this section.) We seek an *allocation* $\beta : (S - S^*) \times S^* \mapsto \mathbf{R}$, that satisfies the demands at all of the requestor nodes without violating the capacity constraints at the server nodes. This allocation β will be used to guide the reassignment of the demand $D_v(S)$ when a facility v in $S - S^*$ is being considered for dropping or substituting. (Note that an allocation is conceptually the same as a splittable assignment. An allocation and an assignment differ only in their domains. While the domain of an assignment is the set of all clients, the domain of an allocation is a subset of the facilities.)

For a requestor node v in $S - S^*$, let $q_\beta(v)$ denote the cost of satisfying the demand at v, that is, $\sum_{j \in S^*} \beta(v, j) c_{vj}$. Let q_β denote the overall cost of the allocation β, defined as $\sum_{v \in S - S^*} q_\beta(v)$. The following lemma shows that there is an auxiliary allocation that is not too expensive.

Lemma 9.5 *There exists an allocation β for the auxiliary allocation problem R with cost q_β at most $C_s(S) + C_s(S^*)$.*

Proof: The proof starts with the given splittable assignments σ and σ^* and constructs β. The detailed proof is deferred to Appendix B. ■

For any requestor node v, we define the servers j in S^* for which $\beta(v, j)$ is nonzero as the *β-suppliers* of v. If every requestor node v in $S - S^*$ has at most one *β-supplier* in $S^* - S$, then β is said to be *refined.* Note that a refined allocation β allows requestor nodes to have multiple β-suppliers in $S \cap S^*$.

For the purpose of constructing the candidate operations, a refined allocation is more useful than an unrefined allocation. To see this, recall that when a facility v in $S - S^*$

is being dropped or substituted, our plan is to reassign the demand $D_v(S)$ among the β-suppliers of v. Since we are allowed to add only one new facility to $S - \{v\}$, it is essential that at most one β-supplier of v be outside S. This condition is satisfied only if β is refined.

The lemma below guarantees the existence of an "inexpensive" refined allocation $\hat{\beta}$ provided that we relax the auxiliary allocation problem by increasing the capacity of each server node i^* in $S^* - S$ from M to $2M$. We use $\hat{R}$ to denote this relaxed auxiliary allocation problem. (Remark: This blowup in the capacities is for the purposes of the analysis only.)

Lemma 9.6 *There exists a refined auxiliary allocation $\hat{\beta}$ for the relaxed auxiliary allocation problem, $\hat{R}$, with cost $q_{\hat{\beta}}$ at most $C_s(S) + C_s(S^*)$.*

Proof: The desired claim follows from Lemma 9.5 and a result of Shmoys and Tardos [17] on the generalized assignment problem. The detailed proof is deferred to Appendix B. ■

For notational convenience we drop the parameter S from $D_v(S)$ and write D_v in what follows. Let D'_v (resp., D''_v) denote the amount of demand shipped to v from servers in $S \cap S^*$ (resp., $S^* - S$) under the refined allocation $\hat{\beta}$. If D''_v is nonzero, then there is a unique $\hat{\beta}$-supplier in $S^* - S$ that we refer to as the *$\hat{\beta}$-foreign-supplier* of v. For the sake of uniformity, we extend this notion even to those requestor nodes v for which D''_v is zero. Though such requestor nodes do not have any $\hat{\beta}$-suppliers in $S^* - S$, we pick an arbitrary facility from $S^* - S$ and call it the $\hat{\beta}$-foreign-supplier of v. Let $Q(i^*)$ denote the set of requestor nodes for which i^* is the (unique) $\hat{\beta}$-foreign-supplier. Note that the sets $Q(i^*)$, for i^* in $S^* - S$, induce a partition of $S - S^*$.

9.2 Candidate Operations

We partition S into five classes and define a candidate operation for the facilities of each class. As in the uncapacitated case, a facility i in S is called *cheap* if $f_i \leq C(S)/n^2$, and *expensive* otherwise. Moreover, a facility i in $S - S^*$ is called *light* if $D''_i \leq M/2$, and *heavy* otherwise. Let S_C denote the set of cheap facilities in $S - S^*$. Let S_{EL} (resp., S_{EH}) denote the set of expensive facilities in $S - S^* - S_C$ that are light (resp., heavy).

We partition the set S_{EL} of expensive light facilities into *primary* and *secondary* facilities as follows: Using a criterion to be described later, a unique facility is chosen from each nonempty set $Q(i^*) \cap S_{EL}$ and is designated as the primary facility associated with i^*. The remaining facilities in each such nonempty set $Q(i^*) \cap S_{EL}$ are called secondary. The precise criterion for choosing the primary facility is not used until Claim 9.9 and is deferred to that point.

We let S_{ELP} (resp., S_{ELS}) denote the set of expensive light facilities that are classified as primary (resp., secondary). Thus, $S \cap S^*$, S_C, S_{EH}, S_{ELP}, and S_{ELS} form a partition of S. We now define *candidate operations* for facilities in each of these subsets, as follows:

1. For i in $S \cap S^*$, do nothing.
2. For i in S_C, do nothing.

3. For i in S_{EH}, substitute i with i^*, the $\hat{\beta}$-foreign-supplier of i. The demand D_i is reassigned among the $\hat{\beta}$-suppliers of i.
4. For i in S_{ELP}, substitute i with i^*, the $\hat{\beta}$-foreign-supplier of i. The demand D_i is reassigned among the $\hat{\beta}$-suppliers of i.
5. For i in S_{ELS}, drop i and let j be the primary facility associated with the $\hat{\beta}$-foreign-supplier of i. The demand D_i is reassigned as follows. First, D'_i is reassigned among the $\hat{\beta}$-suppliers of i in $S \cap S^*$. Second, D''_i is reassigned among j and the $\hat{\beta}$-suppliers of j in $S \cap S^*$; more specifically, a demand of up to $M - D_j$ is reassigned to j, and the remaining demand $(D''_i - M + D_j)$, if any, is reassigned among the $\hat{\beta}$-suppliers of j in $S \cap S^*$.

We first need to ensure that each candidate operation is valid, i.e., the capacity constraints are not violated when an individual candidate operation is performed. (We note that each candidate operation is performed on the solution (S, σ).) The candidate operations for facilities in $S \cap S^*$ and S_C are trivially valid. The validity of a candidate operation for a facility i in $S_{EH} \cup S_{ELP}$ follows from the following two observations: (i) the reassignment respects the capacity constraint at each facility j in $S \cap S^*$ since the capacity of j in the relaxed auxiliary allocation problem is equal to the unused capacity of j in σ, and (ii) the demand reassigned to the $\hat{\beta}$-foreign-supplier of i is at most D_i, which is at most M. Finally, we consider a candidate operation for a facility i in S_{ELS}. The reassignment of D'_i for i in S_{ELS} respects the capacity constraints since it is performed only among facilities in $S \cap S^*$, and for each facility in $S \cap S^*$, its capacity in the auxiliary allocation problem equals its unused capacity in σ. But for the reassignment of D''_i, we need to ensure that there is sufficient remaining capacity at j and the $\hat{\beta}$-suppliers of j in $S \cap S^*$. This is shown in the proof of Claim 9.9. We thus conclude that each candidate operation is valid.

We call a candidate operation *good* if it reduces the cost of the solution by at least $\frac{C(S)}{p(n)}$, and *bad* otherwise. By the hypothesis of the Lemma 9.4, it follows that all of the candidate operations are bad. Claims 9.7, 9.8, and 9.9 show that if the candidate operation for a facility is bad, then its facility cost is "small". The proof of Lemma 9.4 then follows by combining these claims.

Claim 9.7 *If the candidate operations for all of the expensive heavy facilities are bad, then*

$$C_f(S_{EH})\left(1 - \frac{n^2}{p(n)}\right) \leq 4C_f(S^* - S) + \sum_{i \in S_{EH}} q_{\hat{\beta}}(i).$$

Proof: Consider an expensive heavy facility i in S_{EH}. The decrease in the facility cost caused by its candidate operation is $f_i - f_{i^*}$, and by the triangle inequality, the increase in the shipping cost is at most $q_{\hat{\beta}}(i)$. So the net decrease in cost is at least $f_i - (q_{\hat{\beta}}(i) + f_{i^*})$. If this quantity is at least $\frac{n^2 f_i}{p(n)}$, then since i is expensive, this candidate operation is good. Therefore, if this candidate operation is bad, the following upper bound on the facility cost of i holds: $f_i(1 - \frac{n^2}{p(n)}) < f_{i^*} + q_{\hat{\beta}}(i)$.

Note that each i^* in S^*-S can be the candidate substitute for at most four heavy facilities, because under the refined auxiliary allocation $\hat{\beta}$, i^* can supply a demand of at most $2M$, and each heavy facility receives at least $M/2$ units from its unique supplier. Hence, summing over all of the expensive heavy facilities yields the required upper bound on $C_f(S_{EH})$. ■

Claim 9.8 *If the candidate operations for all the expensive light primary facilities are bad, then*

$$C_f(S_{ELP})\left(1-\frac{n^2}{p(n)}\right) \leq C_f(S^*-S) + \sum_{i\in S_{ELP}} q_{\hat{\beta}}(i).$$

Proof: The net cost decrease caused by the substitution of an expensive light primary facility i in S_{ELP} is $f_i - (q_{\hat{\beta}}(i) + f_{i^*})$. If this substitution is bad then, by proceeding as in the previous claim, we obtain the following upper bound on the facility cost of i: $f_i(1-\frac{n^2}{p(n)}) < f_{i^*} + q_{\hat{\beta}}(i)$.

Summing over all of the facilities of S_{ELP}, and using the fact that each i^* in S^*-S has at most one associated primary facility, we obtain the required upper bound on $C_f(S_{ELP})$. ■

Claim 9.9 *If the candidate operations for all of the expensive light secondary facilities are bad, then*

$$C_f(S_{ELS})\left(1-\frac{n^2}{p(n)}\right) \leq \sum_{i\in S_{ELS}} 2q_{\hat{\beta}}(i).$$

Proof: In the paragraphs preceding Claim 9.7, we discussed partitioning the set S_{EL} into two sets S_{ELP} and S_{ELS}, but left the precise definition of the partition unspecified. Here we first specify this partition and then make use of Claim 9.10 below, which implies that if the candidate operation for a facility i in S_{ELS} is bad, then $f_i(1-\frac{n^2}{p(n)}) \leq 2q_{\hat{\beta}}(i)$. Summing over all expensive light secondary facilities then yields the desired upper bound on $C_f(S_{ELS})$.

For each i^* in S^*-S such that $Q(i^*)\cap S_{EL}$ is nonempty, the primary facility associated with i^* is defined as the facility j in $Q(i^*)\cap S_{EL}$ that minimizes $c_{ji^*}+\theta_j$, breaking ties arbitrarily. The parameter θ_j is defined as the average cost per unit demand for servicing an additional demand of $M/2$ units at j, using the remaining capacity (in the current solution S,σ) at facility j and at the $\hat{\beta}$-suppliers of j in $S\cap S^*$. (Remark: If there were no capacity constraints, then θ_j would be zero and the criterion for picking the primary facility would degenerate to that used in the uncapacitated case.)

For the parameter θ_j to be well-defined, we need to ensure that the remaining capacity at facility j and at the $\hat{\beta}$-suppliers of j in $S\cap S^*$ is at least $M/2$. To see that this condition is satisfied, we first observe that the remaining capacity at j is $M-D_j$. Then, we observe that the remaining capacity at the $\hat{\beta}$-suppliers of j in $S\cap S^*$ is at least D'_j, since D'_j is the amount of demand shipped from $\hat{\beta}$-suppliers of j in $S\cap S^*$ to the requestor node j in the auxiliary allocation problem. Adding these two bounds, and using the fact that $D_j - D'_j = D''_j \leq M/2$ (since j is light), we find that the remaining capacity is at least $M/2$. Hence θ_j is well-defined. ■

Claim 9.10 *The net cost decrease caused by the candidate operation for an expensive light secondary facility i in S_{ELS} is at least $f_i - 2q_{\hat{\beta}}(i)$.*

Proof: Let i^* be the $\hat{\beta}$-foreign-supplier of i and let j be the primary facility associated with i^*. Let $q_1(i)$ denote the cost of shipping the demand D'_i from the $\hat{\beta}$-suppliers of i in $S \cap S^*$, and let $q_2(i)$ denote the cost of shipping the demand D''_i from i^*. Thus $q_{\hat{\beta}}(i) = q_1(i) + q_2(i)$.

When i is dropped, the demand D'_i is reassigned to the $\hat{\beta}$-suppliers of i in $S \cap S^*$. This increases the shipping cost by at most $q_1(i)$. It now suffices to bound the shipping cost increase due to the reassignment of D''_i by $q_1(i) + 2q_2(i)$.

We know that the candidate operation of i reassigns the demand D''_i to j and the $\hat{\beta}$-suppliers of j in $S \cap S^*$. By the definition of θ_j, the cost of servicing an additional demand of D''_i originating at j is at most $D''_i\theta_j$ (Recall that $D''_i \leq M/2$ since i is light). Therefore, the increase in the shipping cost due to this reassignment of D''_i is at most

$$\begin{aligned} D''_i(c_{ij} + \theta_j) &\leq D''_i(c_{ii^*} + c_{ji^*} + \theta_j) \\ &\leq D''_i(c_{ii^*} + c_{ii^*} + \theta_i) \\ &\leq 2q_2(i) + \frac{M}{2} \cdot \theta_i \\ &\leq 2q_2(i) + q_1(i). \end{aligned}$$

(The second step uses the fact that j, and not i, was chosen as the primary for i^*; and hence $c_{ji^*} + \theta_j \leq c_{ii^*} + \theta_i$. The third step follows since $q_2(i) = D''_i c_{ii^*}$ and $D''_i \leq M/2$. The fourth step needs some elaboration. The goal here is to prove that $q_1(i)$ is an upper bound on $\frac{M}{2}\theta_i$, which is the cost of servicing an additional demand of $M/2$ units at i using the remaining capacities at i and the $\hat{\beta}$-suppliers of i. Note that the remaining capacity at i is $M - (D'_i + D''_i)$, which is at least $\frac{M}{2} - D'_i$ since $D''_i \leq M/2$. Therefore, $\frac{M}{2}\theta_i$ is at most the cost of satisfying D'_i units of demand at the $\hat{\beta}$-suppliers of i, which is at most $q_1(i)$.) This completes the proof of the claim. ■

9.3 Proof of Lemma 9.4

Using Claims 9.7, 9.8, and 9.9, and the inequality $C_f(S_C) \leq C(S)/n$, we obtain

$$\begin{aligned} C_f(S)\left(1 - \frac{n^2}{p(n)}\right) &= (C_f(S \cap S^*) + C_f(S_C) + C_f(S_{EH}) + C_f(S_{ELP}) + C_f(S_{ELS}))\left(1 - \frac{n^2}{p(n)}\right) \\ &\leq C_f(S \cap S^*) + \frac{C(S)}{p(n)} + 5C_f(S^* - S) + \sum_{i \in S - S^*} 2q_{\hat{\beta}}(i) \\ &\leq \frac{C(S)}{p(n)} + 2q_{\hat{\beta}} + 5C_f(S^*) \\ &\leq \frac{C(S)}{p(n)} + 2C_s(S) + 2C_s(S^*) + 5C_f(S^*) \end{aligned}$$

$$\leq \frac{C(S)}{p(n)} + 2C_s(S) + 5C(S^*).$$

(The third step follows from the definition of $q_{\hat{\beta}}$. The fourth step follows from Lemma 9.6.) This completes the proof of the lemma.

10 Capacitated Problems with Unsplittable Demands

In this section we consider the capacitated problems with unsplittable demands. We obtain constant-factor approximation algorithms for these problems by combining our results for the capacitated problems with splittable demands with a result of Shmoys and Tardos on the generalized assignment problem. In [17], Shmoys and Tardos show how to obtain an unsplittable assignment from a splittable assignment. Their result was presented in the context of job scheduling. Here we give an adaptation of their result to our context.

Theorem 10.1 (Generalized Assignment Problem [17]) *Let S be a set of open facilities and N be a set of clients, and let d_i denote the demand at client i. Suppose there is a splittable assignment from N to S with shipping cost C that assigns at most M units of demand to any facility. Then there exists a polynomial-time computable unsplittable assignment from N to S with shipping cost at most C that assigns at most $M + \max_{i \in N} d_i$ units of demand to any facility.*

We now show how to obtain constant-factor approximation algorithms for the capacitated problems with unsplittable demands. We first relax the unsplittability requirement, and use the local search heuristic to solve the relaxed problem. By Corollary 8.6 (resp., Corollary 9.2), it follows that the splittable solution (S, σ) thus obtained is within a constant factor of the optimal for the capacitated k-median problem (resp., facility location problem) with splittable demands. We then use Theorem 10.1 to obtain an unsplittable assignment σ' for S with shipping cost at most that of the splittable assignment σ for S. However, Theorem 10.1 requires the capacities of all facilities to be increased by a factor of at most two. If we assume, as in [18], that a blowup in the capacity incurs a proportionate increase in the facility cost, then the facility cost for S in the unsplittable case is at most twice that in the splittable case. Therefore, the total cost of the unsplittable solution (S, σ') is at most twice the total cost of the splittable solution (S, σ). Thus, the cost of (S, σ') is within a constant factor of the optimal unsplittable solution. Theorems 10.2 and 10.3 below summarize our results for the capacitated problems with unsplittable demands.

Theorem 10.2 (Capacitated k-median problem with unsplittable demands) *For any constant $\varepsilon > 0$, there is a constant c such that the following can be computed in polynomial time, assuming a factor two blowup in the capacity of each facility: (i) a solution S with $C(S) \leq (1+\varepsilon)C(S^*)$ and $|S| \leq (1+c)k$, and (ii) a solution S with $|S| \leq (5+\varepsilon)k$ and $C(S) \leq (1+c)C(S^*)$.* ■

Theorem 10.3 (Capacitated facility location with unsplittable demands) *For any constant $\varepsilon > 0$, the local search heuristic yields a $(16+\varepsilon)$-approximate solution in polynomial time, assuming a factor two blowup in the capacity of each facility.* ■

References

[1] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 106–113, May 1998.

[2] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 106–113, May 1998.

[3] F. A. Chudak. Improved approximation algorithms for the uncapacitated facility location problem. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization*, June 1998. To appear.

[4] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. Mirchandani and R. Francis, editors, *Discrete Location Theory*, pages 119–171. Wiley, New York, NY, 1990.

[5] G. Diehr. An algorithm for the p-median problem. Technical Report No. 191, Western Management Science Institute, UCLA, June 1972.

[6] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 649–657, January 1998.

[7] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22:148–162, 1982.

[8] D. S. Hochbaum. Various notions of approximations: Good, better, best, and more. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 346–398. PWS Publishing Company, Boston, MA, 1995.

[9] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[10] L. G. Khachiyan. A polynomial algorithm for linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.

[11] M. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local improvement heuristic for facility location problems. Manuscript submitted to *SODA '98*, July 1997.

[12] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9:643–666, 1963.

[13] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44:245–249, 1992.

[14] 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*, pages 771–782, May 1992.

[15] R. F. Love, J. G. Morris, and G. O. Wesolowsky. *Facilities Location: Models and Methods*. North-Holland, New York, NY, 1988.

[16] P. Mirchandani and R. Francis. *Discrete Location Theory*. Wiley, New York, NY, 1990.

[17] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993.

[18] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 265–274, May 1997.

[19] M. B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.

A Capacitated Problems with Splittable Demands: Adding a Facility

In this section we prove Lemmas 8.2 and 9.3, which concern the addition of a new facility to a current solution S. In particular, these lemmas address the changes in the assignment and the total cost when a new facility is added. Recall that for a set S of open facilities, $C_s(S), C_f(S)$, and $C(S)$ denote the shipping cost, facility cost, and the total cost, respectively. For the k-median problem we have $C(S) = C_s(S)$, while for the facility location problem we have $C(S) = C_s(S) + C_f(S)$.

Given a set S of open facilities, the definitions of an assignment of clients to facilities in S and its shipping cost are identical for both the k-median problem as well as the facility location problem. Hence all of the claims made about assignments and shipping costs in this section hold for both the k-median problem as well as the facility location problem.

By the hypothesis of Lemma 8.2 (resp., Lemma 9.3), it follows that $C_s(S)$ is greater than $C_s(S^*)$, and hence that the set $S^* - S$ is nonempty. Let ℓ denote the size of $S^* - S$ and let $u_1, \ldots, u_\ell$ be the facilities in $S^* - S$. We will now show that there exists a facility u_i in $S^* - S$ that satisfies the requirement of Lemma 8.2 (resp., Lemma 9.3).

We start by developing some notation and useful technical machinery. Let σ and σ^* be any optimal (splittable) assignments for S and S^*, respectively. In Lemma A.1 below, we use the assignments σ and σ^* to partition the demand of each client into smaller entities, which we refer to as *chunks*. The properties of this partitioning, which are enumerated in Lemma A.1, are then used to construct a "difference graph" G that captures the differences in the assignments σ and σ^*. We defer the proof of Lemma A.1 to later in this section.

Lemma A.1 (Splitting into Chunks) *Let σ and σ^* be any assignments (not necessarily optimal) for S and S^* respectively. Then for each client j, there exists a partition of the demand d_j into a finite number of chunks, each of which has an associated size, such that the following four properties hold:*

A1. *Each chunk is shipped from a unique facility i in S under assignment σ.*

A2. *Each chunk is shipped from a unique facility i^* in S^* under assignment σ^*.*

A3. *For each facility i in S, the total size of all chunks of d_j shipped from i under assignment σ is $\sigma(j, i)$.*

A4. *For each facility i^* in S^*, the total size of all chunks of d_j shipped from i^* under assignment σ^* is $\sigma^*(j, i^*)$.*

We use d_j^i to denote the ith chunk of d_j. We note that while Lemma A.1 states that each chunk is shipped from a unique facility under σ (resp., σ^*), it does not preclude two distinct chunks d_j^i and d_j^k associated with the same client to be shipped from the same facility under σ (resp., σ^*).

Given the set τ of all chunks, taken over all of the clients, we construct a directed (multi)graph $G = \langle V, A\rangle$ in which we associate a label, a cost, and a size with each arc. The node set V is $S \cup S^*$. The arc set A is in one-to-one correspondence with the set τ, and is constructed as follows: For each chunk d_j^t that is shipped from facilities i and i^* in σ and σ^*, respectively, we add a directed arc $\hat{a}$ from i to i^*. This arc $\hat{a}$ is labeled by the corresponding chunk d_j^t. The size of $\hat{a}$, denoted $Size(\hat{a})$, is set to the size of the chunk d_j^t. The cost of $\hat{a}$, denoted $Cost(\hat{a})$, is set to $c_{ji} - c_{ji^*}$. The directed graph G thus constructed is called the *difference graph*.

Note that the difference graph G may contain cycles, and may also contain multiple arcs between the same pair of nodes. Furthermore, there are no outgoing arcs for nodes in $S^* - S$, and no incoming arcs for nodes in $S - S^*$. The cost of the difference graph G, denoted $Cost(G)$, is defined as $\sum_{\hat{a}\in A} Size(\hat{a})Cost(\hat{a})$. In Lemma A.2, we show that $Cost(G)$ is equal to the difference between the shipping costs of S and S^*.

Lemma A.2 $Cost(G) = C_s(S) - C_s(S^*)$.

Proof: Consider any client j and a particular chunk d_j^t of j. Let the arc corresponding to d_j^t be $\hat{a}$. Note that $Size(\hat{a})Cost(\hat{a})$ is the difference between the shipping cost of the chunk d_j^t in assignment σ and the shipping cost for the same chunk in assignment σ^*. By summing $Size(\hat{a})Cost(\hat{a})$ over all arcs $\hat{a}$ corresponding to chunks of j, we obtain the difference between the shipping cost for client j in σ and the shipping cost for the same client in σ^*. Summing these differences over all of the clients yields the desired claim. ∎

For notational convenience, throughout this section we interpret $D_v(S)$ as zero whenever v does not belong to S. (Recall that for v in S, $D_v(S)$ denotes the total demand that is shipped from facility v in solution S.)

Lemma A.3 *For each v in $S \cup S^*$, the total size of all arcs coming into v is $D_v(S^*)$, and the total size of all arcs going out of v is $D_v(S)$.*

Proof: Consider any client j. From property A3 (resp., property A4) of Lemma A.1, it follows that the total size of all arcs that are labeled by chunks of d_j and that go out of (resp., come in to) v is $\sigma(j,v)$ (resp., $\sigma^*(j,v)$). Summing over all of the clients yields the required result. ■

We now decompose the difference graph G into a set $\mathcal{C}$ of directed cycles and a set $\mathcal{P}$ of directed paths that satisfy some useful properties. Before describing this decomposition, which is obtained in Lemma A.4, we need some additional terminology. A node v in $S \cup S^*$ for which the total size of all arcs coming into v is greater than the total size of all arcs going out of v is called a *surplus* node, and the quantity $D_v(S^*) - D_v(S)$ is called the *excess* at v.

Lemma A.4 (Decomposition of G) *There exists a decomposition of the difference graph G into a set of cycles $\mathcal{C}$ and a set of paths $\mathcal{P}$, with an associated positive size for each path and each cycle, such that:*

- B1. *For each arc $\hat{a}$ in G, the sum of the sizes of all paths and cycles that contain the arc $\hat{a}$ is equal to the size of arc $\hat{a}$ in G.*
- B2. *All paths terminate at a surplus node.*
- B3. *For each surplus node v, the sum of the sizes of all paths terminating at v is equal to the excess at v.*

We defer the proof of Lemma A.4 to later in this section. We note that in the decomposition described in Lemma A.4, an arc $\hat{a}$ may occur in multiple paths and cycles but the size of $\hat{a}$ is partitioned among the sizes of all of the paths and the cycles containing $\hat{a}$.

We define the cost of a path or cycle P to be $\mathit{Size}(P) \cdot \sum_{\hat{a} \in P} \mathit{Cost}(\hat{a})$. (We note that since the cost of an arc may be negative, the cost of a path or cycle may also be negative.) We define the cost $\mathit{Cost}(\mathcal{C})$ (resp., $\mathit{Cost}(\mathcal{P})$) of $\mathcal{C}$ (resp., $\mathcal{P}$) to be the sum of the costs of all the cycles (resp., paths) in $\mathcal{C}$ (resp., $\mathcal{P}$).

Lemma A.5 *Let $\mathcal{C}$ and $\mathcal{P}$ be the set of cycles and paths, respectively, obtained in the decomposition of G. Then $\mathit{Cost}(\mathcal{C}) + \mathit{Cost}(\mathcal{P}) = \mathit{Cost}(G)$.*

Proof: Consider any arc $\hat{a}$ in G. From property B1 of Lemma A.4, it follows that the total contribution from arc $\hat{a}$ to the cost of $\mathcal{C}$ and $\mathcal{P}$ equals $\mathit{Size}(\hat{a})\mathit{Cost}(\hat{a})$. Summing over all of the arcs in G, the lemma follows. ■

We partition the set of paths $\mathcal{P}$ into sets $\mathcal{P}_0, \ldots, \mathcal{P}_\ell$ on the basis of the terminating nodes of the paths. Let $\mathcal{P}_i$, for $1 \le i \le \ell$, denote the set of paths in $\mathcal{P}$ with terminating node

u_i. (Recall that $\{u_i : 1 \leq i \leq \ell\}$ equals $S^* - S$.) Let $\mathcal{P}_0$ denote the set of paths in $\mathcal{P}$ that terminate at a node in $S \cap S^*$. To see that the sets $\mathcal{P}_0, \ldots, \mathcal{P}_\ell$ indeed form a partition of $\mathcal{P}$, note that: (i) every path in $\mathcal{P}$ terminates at a node in S^*, and (ii) if $i \neq j$, then the terminating nodes of a path in $\mathcal{P}_i$ and a path in $\mathcal{P}_j$ are distinct.

We define $Cost(\mathcal{P}_i)$ to be the sum of the costs of all the paths in $\mathcal{P}_i$. From Lemma A.5, it follows that

$$Cost(\mathcal{C}) + \sum_{i=0}^{\ell} Cost(\mathcal{P}_i) = Cost(G) = C_s(S) - C_s(S^*). \tag{5}$$

We will prove Lemma 8.2 (resp., Lemma 9.3) by showing that there exists an i, $1 \leq i \leq \ell$, such that the cost of the solution $S + u_i$ satisfies the upper bound stated in the lemma. We determine u_i, along with an associated assignment, by modifying the original assignment σ through a sequence of "augmentation" operations.

Given an assignment σ', an *augmentation* along a path or cycle P in $\mathcal{C} \cup \mathcal{P}$ yields a new assignment σ'' that is defined as follows. We initially set σ'' to σ'. Then, for each arc (u, v) on P with label d_j^t, we decrease $\sigma''(u, j)$ by $Size(P)$ and increase $\sigma''(v, j)$ by $Size(P)$. In order to prove that the assignment σ'' is valid, we need to ensure that for all i and j, $0 \leq \sigma''(i, j) \leq M$. By the definition of σ'', it is easy to see that if σ' is a valid assignment, then σ'' is also valid if the following two properties hold:

C1. For each arc (u, v) with label d_j^t on P, at least $Size(P)$ units of d_j^t is shipped from facility u to client j in assignment σ'.

C2. If P is a path, then the remaining capacity of the terminating node of P is at least $Size(P)$ in the current assignment σ'.

Property C1 ensures that the amount of demand $Size(P)$ associated with client j that is being taken away from a given facility i under assignment σ'' is indeed assigned to i under assignment σ'. Thus, property C1 implies that for all i and j, $\sigma''(i, j)$ is nonnegative. It remains to ensure that σ'' does not violate the capacity constraints at any of the facilities. Note that for all nodes v on P except the terminating node, the total demand shipped from v in assignment σ'' is the same as that shipped in assignment σ', and hence σ'' does not violate the capacity constraints at any of these internal nodes. Property C2 ensures that σ'' does not violate the capacity constraint even at the terminating node of P.

Lemma A.6, whose proof is deferred to later in this section, establishes the validity of an assignment obtained from σ by any sequence of augmentations on distinct paths and cycles drawn from $\mathcal{C} \cup \mathcal{P}$.

Lemma A.6 (Validity of Assignments) *Let σ' be an assignment obtained from σ by a sequence of augmentations along distinct paths and cycles drawn from $\mathcal{C} \cup \mathcal{P}$. Then, σ' is valid.*

Given a new assignment σ'' that is obtained from a valid assignment σ' by augmenting along a path P in $\mathcal{P}$, it is easy to see that the shipping cost of the new assignment σ'' equals $C_s(\sigma') - Cost(P)$. In Lemma A.7, we determine the cost of an assignment obtained from σ by augmenting along paths in $\mathcal{P}_i$.

Lemma A.7 *For each u_i in $S^* - S$, there exists an assignment for $S + u_i$ with a shipping cost of $C_s(S) - Cost(\mathcal{P}_i)$.*

Proof: We start with the original assignment σ, and augment along the paths in the set $\mathcal{P}_i$ one at a time in any order. By Lemma A.6, it follows that the assignment obtained after the sequence of augmentations is valid.

Since an augmentation along a path P reduces the shipping cost of the assignment by $Cost(P)$, it follows that all of the augmentations along the paths in $\mathcal{P}_i$ reduce the shipping cost by a total of $Cost(\mathcal{P}_i)$. Therefore, the shipping cost of the final assignment is $C_s(S) - Cost(\mathcal{P}_i)$. Moreover, since σ uses only the facilities in S, and since all paths of $\mathcal{P}_i$ terminate at u_i, the final assignment uses only the facilities in $S + u_i$. This completes the proof. ■

Lemma A.8 *Each path or cycle in $\mathcal{C} \cup \mathcal{P}_0$ has a nonpositive cost.*

Proof: First, note that each node of any path or cycle in $\mathcal{C} \cup \mathcal{P}_0$ belongs to S. Now let P be a path or a cycle in $\mathcal{C} \cup \mathcal{P}_0$ with positive cost, if one exists. By starting with σ and augmenting along P, we obtain a new assignment for S with a shipping cost that is strictly less than the shipping cost of the current assignment σ. This contradicts the optimality of σ. ■

We are now ready to prove Lemmas 8.2 and 9.3.

Proof of Lemma 8.2: From Equation (5) and Lemma A.8, it follows that $\sum_{i=1}^{\ell} Cost(\mathcal{P}_i) \geq C_s(S) - C_s(S^*)$. By averaging, there exists an i such that $Cost(\mathcal{P}_i) \geq \frac{C_s(S) - C_s(S^*)}{\ell}$. Applying Lemma A.7 with this value of i, we obtain $C_s(S + u_i) = C_s(S) - Cost(\mathcal{P}_i)$. Combining these bounds, we have

$$\begin{aligned} C(S) - C(S + u_i) &= C_s(S) - C_s(S + u_i) \\ &= Cost(\mathcal{P}_i) \\ &\geq \frac{C_s(S) - C_s(S^*)}{\ell} \\ &\geq \frac{C_s(S) - C_s(S^*)}{k}, \end{aligned}$$

since $\ell \leq k$. This completes the proof of the lemma. ■

Proof of Lemma 9.3: We are given that $C_s(S) - C(S^*) \geq nC(S)/p(n)$. From Equation (5) and Lemma A.8, it follows that $\sum_{i=1}^{\ell} Cost(\mathcal{P}_i) \geq C_s(S) - C_s(S^*)$. Subtracting $C_f(S^*)$ from both sides, we get

$$\sum_{i=1}^{\ell} (Cost(\mathcal{P}_i) - f_{u_i}) \geq C_s(S) - C_s(S^*) - C_f(S^*).$$

By averaging, there exists an i such that $Cost(\mathcal{P}_i) - f_{u_i} \geq \frac{C_s(S) - C_s(S^*) - C_f(S^*)}{\ell}$. Applying Lemma A.7 with this value of i, we obtain $C_s(S + u_i) = C_s(S) - Cost(\mathcal{P}_i)$. Combining these bounds, we have

$$\begin{aligned} C(S) - C(S+u_i) &= C_s(S) - C_s(S+u_i) - f_{u_i} \\ &= Cost(\mathcal{P}_i) - f_{u_i} \\ &\geq \frac{C_s(S) - C_s(S^*) - C_f(S^*)}{\ell} \\ &\geq \frac{C_s(S) - C(S^*)}{n} \\ &\geq \frac{C(S)}{p(n)}, \end{aligned}$$

since $C_s(S) - C(S^*) \geq nC(S)/p(n)$ by assumption. This completes the proof of the lemma. ■

Proof of Lemma A.1: Fix a client j. The assignment σ splits the demand d_j among the facilities in S. Thus, for each facility i in S that satisfies $\sigma(j,i) > 0$, σ defines a chunk of size $\sigma(j,i)$ that is shipped from i. Let X denote the set of these chunks. While X satisfies properties A1 and A3, X may not satisfy properties A2 and A4. We now further partition each chunk in X into smaller chunks and obtain the desired set Y of chunks satisfying all of the four properties.

Consider any chunk x in X of size s. Let p be the facility that ships x in σ. For each i^* in S^* such that $\sigma^*(j,i^*) > 0$, we include a chunk x_{i^*} of size $s\sigma^*(j,i^*)/d_j$ in Y. We stipulate that chunk x_{i^*} is shipped from p in σ and from i^* in σ^*. Since Y is obtained by separately dividing each chunk of X into smaller chunks, Y satisfies properties A1 and A3. Moreover, by construction, Y satisfies properties A2 and A4. ■

Proof of Lemma A.4: The following algorithm finds the required decomposition. Initialize G' to G. Repeat the following procedure until G' has no arcs.

1. If G' has cycles, pick an arbitrary cycle P. Otherwise, let v be an arbitrary node with no incoming arcs and at least one outgoing arc, and let P be a maximal directed path originating at v.
2. Find the smallest remaining size, say s, among all arcs on P. The size of P is set to s.
3. Add P to $\mathcal{P}$ or $\mathcal{C}$ depending on whether it is a path or a cycle.
4. Update G' by decreasing the size of each arc that occurred on P by s. Drop arcs with zero size.

We first note that in each iteration of the above procedure, at least one arc of G is deleted. Since Lemma A.1 implies that the number of edges in G is finite, the algorithm terminates.

We now show that the decomposition obtained satisfies the properties stated in the lemma. Property B1 follows from construction. For properties B2 and B3, consider a node v such that there exists a path in $\mathcal{P}$ that terminates at v. This implies that during some iteration, the node v becomes a sink node. Before that iteration, every execution of step 5 reduces the total size of arcs coming into v by the same amount as the reduction in the total size of arcs going out of v. Hence at the instant when v becomes a sink node, the total size of all arcs coming into v is $D_v(S^*) - D_v(S)$. Since some arcs coming into v still exist, the quantity $D_v(S^*) - D_v(S)$ is positive and hence v is a surplus node. This establishes property B2. Moreover, the total size of all the paths that terminate at v is $D_v(S^*) - D_v(S)$, establishing property B3. ■

Proof of Lemma A.6: We establish the lemma by induction on the sequence of augmentations performed. Let $\mathcal{P}' \subseteq \mathcal{C} \cup \mathcal{P}$ denote the set of paths and cycles along which the augmentations were performed to obtain σ'. In our inductive proof, we will also establish the following claims: (i) for every arc (u, v) in G with associated label d_j^t, the amount of chunk d_j^t that is shipped from facility u to client j in σ' equals the sum of the sizes of all paths and cycles that intersect (u, v) and are not in $\mathcal{P}'$, and (ii) the excess at a surplus node v under σ' equals the sum of the sizes of all paths that terminate at v and are not in $\mathcal{P}'$.

For the induction basis, we consider an empty sequence. Since σ is valid, the claim of the lemma holds trivially. Moreover, claim (i) follows from property B1 of Lemma A.4, while claim (ii) follows from property B3 of Lemma A.4.

As our induction hypothesis, we are given a valid assignment σ' that has been obtained from σ by a nonempty sequence of augmentations. For the induction step, we will establish claims (i) and (ii). The validity of σ'' follows from claims (i) and (ii), since claim (i) implies property C1 and claim (ii) implies property C3.

We first establish claim (i). Let $\hat{a} = (u, v)$ be any arc on P with label d_j^t. By claim (i) of the induction hypothesis, the amount of chunk d_j^t that is shipped from facility u in σ' equals the sum of the sizes of all paths and cycles that intersect (u, v) and are not in $\mathcal{P}'$. Since augmenting along path P reduces this amount by $\mathit{Size}(P)$, claim (i) of the induction step follows.

We now establish claim (ii). If P is a path, then by property B2, P terminates at a surplus node v. By claim (ii) of the induction hypothesis, the excess at a surplus node v under σ' equals the sum of the sizes of all paths that terminate at v and are not in $\mathcal{P}'$. Since augmenting along path P reduces this excess by $\mathit{Size}(P)$, claim (ii) of the induction step follows. This completes the proof of the lemma. ■

B Capacitated Problems with Splittable Demands: The Allocation Problem

In this section we prove Lemmas 9.5 and 9.6, which are related to the allocation problem. We first consider Lemma 9.5, which states that the allocation problem R defined in Section 9 has an allocation β with cost at most $C_s(S^*) + C_s(S)$. The allocation β is constructed from σ and σ^*, the optimal assignments for S and S^*, respectively. The proof uses some of the techniques that were developed in Appendix A.

We start by constructing another difference graph G' that is identical to the difference graph G constructed in Appendix A, except that the costs on the arcs are different: While the cost of an arc $\hat{a}$ is set to $c_{ji} - c_{ji^*}$ in G, we set it to $c_{ji} + c_{ji^*}$ in G'. The cost of the new difference graph G' is defined as before, namely, $\sum_{\hat{a} \in A} Size(\hat{a}) Cost(\hat{a})$. Since the arc costs are different, the cost of G' differs from that of G. Lemma B.1 below determines the cost of G'. Since G and G' are identical except for the arc costs, the cost-independent Lemmas A.3, A.4, and A.5 proved for G also hold for G' and are listed as Lemmas B.3, B.4, and B.5, respectively.

Lemma B.1 (Cost of G') $Cost(G') = C_s(S) + C_s(S^*)$.

Proof: Similar to the proof of Lemma A.2. ■

Corollary B.2 (Cost of $\mathcal{P}$) $Cost(\mathcal{P}) \leq C_s(S) + C_s(S^*)$.

Proof: Since Lemma A.5 holds for G', it follows that $Cost(\mathcal{C}) + Cost(\mathcal{P}) = C_s(S) + C_s(S^*)$. Because all of the arc costs in G' are positive, it follows that $Cost(\mathcal{C}) \geq 0$ and hence the corollary holds. ■

Lemma B.3 *Same as Lemma A.3, but with G replaced by G'.* ■

Lemma B.4 (Decomposition of G') *Same as Lemma A.4, but with G replaced by G'.* ■

Lemma B.5 *Same as Lemma A.5, but with G replaced by G'.* ■

Proof of Lemma 9.5: Using $\mathcal{P}$, we construct the allocation β as follows: For each i in $S - S^*$ and j in $S \cup S^*$, let $\mathcal{P}_{ij}$ denote the set of paths in $\mathcal{P}$ that start at i and terminate at j. Then we set $\beta(i,j) = \sum_{P \in \mathcal{P}_{ij}} Size(P)$. We need to ensure that β is a function from $(S - S^*) \times S^*$ to $\mathbf{R}$. For this it suffices to show that the terminating nodes of all paths P in $\mathcal{P}$ belong to S^*. By Lemma B.3, all nodes with an incoming arc in G' belong to S^*, since all other nodes v have $D_v(S^*) = 0$. Hence the nodes of S^* are the only nodes at which a path P in $\mathcal{P}$ can terminate.

Claim B.6 below establishes that β is a valid allocation. Then Claim B.7 establishes the required upper bound on the cost of β. ■

Claim B.6 *The allocation β, constructed as above, is a valid solution to the allocation problem R.*

Proof: We first show that the demands of all requestor nodes are satisfied. Consider an arbitrary requestor node v in $S - S^*$. By Lemma B.3, the total size of all arcs going out of v in G' is $D_v(S)$. Moreover, since $D_v(S^*)$ is zero, there are no arcs in G' coming into v. Hence the node v can only occur as the starting node of a path in $\mathcal{P}$; v cannot occur as a non-starting node in a path and it cannot occur in a cycle. It follows from property B1 of

Lemma B.4 that the sum of the sizes of all the paths in $\mathcal{P}$ starting at v is $D_v(S)$. Therefore, by the construction of β, the total demand shipped to v is

$$\sum_{j \in S^*} \beta(v, j) = \sum_{j \in S^*} \sum_{P \in \mathcal{P}_{vj}} \mathit{Size}(P) = \sum_{P \in \mathcal{P}} \mathit{Size}(P) = D_v(S).$$

We now show that the capacity constraints are not violated at any of the server nodes. Consider an arbitrary server node v in S^*. By the construction of β, the total demand shipped from v under β (i.e., $\sum_{i \in S - S^*} \beta(i, v)$) is at most the sum of the sizes of all paths in $\mathcal{P}$ terminating at v. By properties B2 and B3 of Lemma B.4, the sum of the sizes of all paths terminating at v is equal to the excess of v, which is $D_v(S^*) - D_v(S)$. Since $D_v(S^*) \leq M$, it follows that the total demand shipped from v under β is at most the capacity of v. Thus no capacity constraints are violated. This establishes that β is a valid allocation. ■

Claim B.7 *The cost of the allocation β is at most $C_s(S) + C_s(S^*)$.*

Proof: Consider any arc $\hat{a}$ in G'. Let the endpoints of $\hat{a}$ be u and u^*, and let the chunk corresponding to $\hat{a}$ be d_y^t, where y is a client. Recall that $\mathit{Cost}(\hat{a})$ is $c_{uy} + c_{yu^*}$. By the triangle inequality, this is at least c_{uu^*}. For notational convenience, we use $c_{\hat{a}}$ to denote c_{uu^*}. Now consider any path P in $\mathcal{P}$ with starting and terminating nodes i and j. Then $\sum_{\hat{a} \in P} \mathit{Cost}(\hat{a}) \geq \sum_{\hat{a} \in P} c_{\hat{a}} \geq c_{ij}$ by the triangle inequality. Thus,

$$\mathit{Cost}(P) = \mathit{Size}(P) \cdot \sum_{\hat{a} \in P} \mathit{Cost}(\hat{a}) \geq \mathit{Size}(P) c_{ij}.$$

Summing over all paths P starting at i and terminating at j, we obtain $\beta(i, j) c_{ij} \leq \sum_{P \in \mathcal{P}_{ij}} \mathit{Cost}(P)$. Then, summing over all i and j, it follows that the cost of allocation β is at most $\mathit{Cost}(\mathcal{P})$. The claim then follows from Corollary B.2. ■

We now turn to the proof of Lemma 9.6, which establishes the existence of a refined allocation for the relaxed allocation problem.

Proof of Lemma 9.6: Let β be the (unrefined) allocation provided by Lemma 9.5 for the allocation problem R. Using R and β, we first construct a restricted allocation problem R', which involves the servers in $S^* - S$ only. We then apply a result of Shmoys and Tardos (Theorem 10.1) to R' to obtain an unsplittable allocation for R', which is then combined with β to construct the required refined allocation $\hat{\beta}$ for R. (An allocation α is said to be unsplittable if each requestor has at most one supplier.)

Before proceeding, we need some notation to split the demands and shipping costs into two parts, one for servers in $S \cap S^*$, and the other for servers in $S^* - S$. Throughout this proof, for notational convenience, we use D_v to denote $D_v(S)$. Let D'_v (resp., D''_v) denote the amount of demand shipped to a requestor node v from servers in $S \cap S^*$ (resp., $S^* - S$) under the allocation β. That is, D'_v equals $\sum_{j \in S \cap S^*} \beta(v, j)$ and D''_v equals $\sum_{j \in S^* - S} \beta(v, j)$. Note that $D'_v + D''_v = D_v$. Similarly, let $q_{1\beta}$ (resp., $q_{2\beta}$) denote the cost of shipping the demand from servers in $S \cap S^*$ (resp., $S^* - S$). More precisely, $q_{1\beta} = \sum_{i \in S - S^*} \sum_{j \in S \cap S^*} \beta(i, j) c_{ij}$ and $q_{2\beta} = \sum_{i \in S - S^*} \sum_{j \in S^* - S} \beta(i, j) c_{ij}$. Clearly, $q_{1\beta} + q_{2\beta} = q_\beta$.

The precise formulation of R' is as follows:

- The nodes of $S - S^*$ are the requestors.
- The nodes of $S^* - S$ are the servers.
- Each requestor node v in $S - S^*$ has a demand of D''_v .
- Each server node v in $S^* - S$ has a capacity of M.
- The distance function between the nodes is given by c.

Consider the allocation β' obtained by restricting β to servers in $S^* - S$. Thus, for each i in $S - S^*$ and j in $S^* - S$, set $\beta'(i,j) = \beta(i,j)$. The allocation β' is clearly valid for R' and has cost $q_{2\beta}$. Theorem 10.1 then implies that there exists an unsplittable allocation β'' of cost at most $q_{2\beta}$ that is feasible for R' provided that the capacity constraints on the servers are relaxed to $M + \max_{i \in S-S^*} D''_i$, which is at most $2M$. Since β'' is an unsplittable allocation, it ensures that each requestor node has at most one supplier in $S^* - S$. Using β'' and β, we construct a refined allocation $\hat{\beta}$ for $\hat{R}$ as follows:

- For each i in $S - S^*$ and j in $S^* - S$, set $\hat{\beta}(i,j) = \beta''(i,j)$.
- For each i in $S - S^*$ and j in $S^* \cap S$, set $\hat{\beta}(i,j) = \beta(i,j)$.

It is straightforward to verify that the above allocation $\hat{\beta}$ is a valid refined allocation for $\hat{R}$. Moreover, the cost of $\hat{\beta}$ is equal to the cost of β'' plus $q_{1\beta}$. Hence it is upper-bounded by q_β, which in turn is at most $C_s(S) + C_s(S^*)$, by Lemma 9.5. This completes the proof. ■