A constant-factor approximation algorithm for the k-median problem

(Extended Abstract)

Moses Charikar[*] Sudipto Guha[†] Éva Tardos[‡] David B. Shmoys[§]

Abstract

We present the first constant-factor approximation algorithm for the metric k-median problem. The k-median problem is one of the most well-studied clustering problems, i.e., those problems in which the aim is to partition a given set of points into clusters so that the points within a cluster are relatively close with respect to some measure. For the metric k-median problem, we are given n points in a metric space. We select k of these to be cluster centers, and then assign each point to its closest selected center. If point j is assigned to a center i, the cost incurred is proportional to the distance between i and j. The goal is to select the k centers that minimize the sum of the assignment costs. We give a $6\frac{2}{3}$-approximation algorithm for this problem. This improves upon the best previously known result of $O(\log k \log\log k)$, which was obtained by refining and derandomizing a randomized $O(\log n \log\log n)$-approximation algorithm of Bartal.

We also give constant factor approximation algorithms for several natural extensions of the problem.

[*]moses@cs.stanford.edu. Stanford University, Stanford, CA 94305. Research supported by the Pierre and Christine Lamond Fellowship, NSF Grant IIS-9811904 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

[†]sudipto@cs.stanford.edu. Stanford University, Stanford, CA 94305. Research Supported by IBM Cooperative Fellowship, NSF Grant IIS-9811904 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

[‡]eva@cs.cornell.edu. Cornell University, Ithaca, NY 14853. Research partially supported by NSF grants CCR-9700163 ONR grants N00014-98-1-0589 and N00014-96-1-0050.

[§]shmoys@cs.cornell.edu. Cornell University, Ithaca, NY 14853. Research partially supported by NSF grants CCR-9700029 and DMS-9505155 and ONR grant N00014-96-1-00500.

1 Introduction

For the *metric k-median problem*, we are given n points in a metric space. We must select k of these to be cluster centers, and then assign each input point j to the selected center that is closest to it. If location j is assigned to a center i, we incur a cost proportional to the distance between i and j. The goal is to select the k centers so as to minimize the sum of the assignment costs. We give a $6\frac{2}{3}$-approximation algorithm for this problem, that is, a polynomial-time algorithm that finds a feasible solution of objective function value within a factor of $6\frac{2}{3}$ of the optimum.

Lin & Vitter [18] considered the k-median problem with arbitrary assignment costs, and gave a polynomial-time algorithm that finds, for any $\epsilon > 0$, a solution for which the objective function value is within a factor of $1+\epsilon$ of the optimum, but is infeasible: it opens $(1+1/\epsilon)(\ln n + 1)k$ cluster centers. Lin & Vitter also provided evidence that this result is best possible via a reduction from the set cover problem.

Consequently, it is quite natural to consider special cases. The problem is solvable in polynomial time on trees [14, 22]. However, for general metric spaces, the problem is NP-hard to solve exactly. Arora, Raghavan & Rao [1] give a polynomial-time approximation scheme for the k-median problem with 2-dimensional Euclidean inputs.

We study the case of general metric k-center, assuming that the input points are located in a metric space, or in other words, to assume that the assignment costs are symmetric and satisfy the triangle inequality. Lin & Vitter [17] also gave a polynomial-time algorithm for the metric k-median problem that, for any $\epsilon > 0$, finds a solution of cost no more than $2(1+\epsilon)$ times the optimum, while using at most $(1+1/\epsilon)k$ cluster centers. The first non-trivial approximation algorithm that produces a feasible solution (i.e., one that uses at most k centers) is due to Bartal [3, 4]. By combining his result on the approximation of any metric by tree metrics with the fact that the k-median problem can be solved optimally in a tree metric, Bartal gave a randomized $O(\log n \log\log n)$-approximation algorithm for the k-median problem. This algorithm was subsequently derandomized and refined to yield an $O(\log k \log\log k)$-approximation algorithm by Charikar, Chekuri, Goel, & Guha [5].

Approximation algorithms have been studied for a variety of clustering problems. The *k-center problem* is the min-max analogue of the k-median problem: one opens centers at k locations out of n, so as to minimize the maximum distance that an unselected location is from its nearest center. Hochbaum & Shmoys [13] and subsequently Dyer & Frieze [10] gave 2-approximation algorithms for the metric case problem (which is best possible unless $\mathcal{P} = \mathcal{NP}$), and also gave extensions for weighted variants. Gonzalez [11] considered the variant in which the objective is to minimize maximum distance between a pair of points in the same cluster, and independently gave a 2-approximation algorithm (which is also best possible).

The k-median problem is closely related to the *uncapacitated facility location problem*, which is a central problem in the Operations Research literature (see, e.g., the survey of Cornuéjols, Nemhauser, & Wolsey [9]). In this problem, each location has a cost f_i, the cost of opening a center (or facility) at location i. There is no restriction on the number of facilities that can be opened, but instead the goal is to minimize the total cost: the cost of the selected facilities plus the sum of the assignment cost of each location to its closest open facility, where we assume that the cost of assigning location j to facility i is proportional to the distance between i and j. Shmoys, Tardos, & Aardal [20] use the techniques of Lin & Vitter [17] to give the first constant-factor approximation algorithm for the metric uncapacitated facility location problem. The quality of approximation has been improved in a sequence of papers [12, 6, 7]. The best result currently known is a $(1 + 2/e)$-approximation algorithm due to Chudak & Shmoys [6, 7]. Guha & Khuller [12] have shown two hardness of approximation results as well: the problem is MAX SNP-hard, and even more surprisingly, for any $\rho < 1.463$, no ρ-approximation algorithm exists unless unless $\mathcal{NP} \subseteq DTIME(n^{O(\log\log n)})$. Sviridenko [21] independently observed the former hardness result, and also subsequently strengthened the latter to depend only on the assumption that $\mathcal{P} \neq \mathcal{NP}$. All of these algorithms rely on solving a natural linear programming relaxation of the problem and rounding the optimal fractional solution. Korupolu, Plaxton, & Rajaraman [16] analyze variants of simple local search algorithms for several clustering problems and show, for example, that for any $\epsilon > 0$, this leads to a $(5 + \epsilon)$-approximation algorithm for the uncapacitated facility location problem.

Our algorithms are based on the approach of solving a natural linear programming relaxation of the problem, and rounding the optimal fractional solution. The linear programming relaxation is analogous to the relaxation used by the approximation algorithms for the facility location problem, and has also been studied for the k-median problem on trees [24, 23].

We obtain our result by combining the filtering technique of Lin & Vitter [18] with a more sophisticated method for selecting which centers to open. The filtering technique of Lin & Vitter [18] guarantees that the cost of the solution does not exceed the LP optimum by too much by making sure that in the integer solution, each location pays an assignment cost not too much more than the corresponding part of the cost of the optimal fractional solution. This kind of location-by-location guarantee is not possible for the k-median problem: some locations will necessarily pay a significantly greater assignment cost in the integer solution than in the fractional solution. However, we show how to select the centers so that there is only a small constant increase in the average assignment cost.

In Section 8 we give constant factor approximation algorithms for several natural extensions of the problem. We consider versions where centers also have costs, where centers have limited capacity, and where the distance function satisfies a relaxed triangle inequality.

2 The k-median problem

It is more natural to state our algorithm in terms of a slight generalization of the usual k-median problem, which we described in the introduction. We shall let the input consist of a set of locations N and a bound k, where there is a specified assignment cost c_{ij} between each pair of points $i, j \in N$. In addition, we shall also be given a demand d_j for each location $j \in N$; this demand can either be viewed as a weight that indicates the importance of the location, or as specifying the number of clients present at that location. The usual statement of the k-median problem corresponds to the special case in which $d_j = 1$, for each $j \in N$. Note that this generalization also allows the possibility that there are locations with no demand (i.e., where $d_j = 0$), but can still be selected as centers.

We shall assume that the assignment costs are nonnegative, symmetric, and satisfy the triangle inequality: that is, $c_{ij} = c_{ji}$ for each $i, j \in N$, and $c_{ij} + c_{jk} \geq c_{ik}$ for each $i, j, k \in N$. The problem is to select k of the locations as centers, and assign each location in N to one of the k selected centers so as to minimize the total weighted assignment cost incurred.

The k-median problem can be stated as the following integer program, where the 0-1 variable y_i, $i \in N$, indicates if the location i is selected as a center, and the 0-1 variable x_{ij}, $i,j \in N$, indicates if location j is assigned to the center at i:

$$\text{minimize} \sum_{i,j\in N} d_j c_{ij} x_{ij} \qquad (1)$$

subject to

$$\begin{aligned}
\sum_{i\in N} x_{ij} &= 1, && \text{for each } j \in N, && (2)\\
x_{ij} &\leq y_i, && \text{for each } i,j \in N, && (3)\\
\sum_{i\in N} y_i &\leq k, && && (4)\\
x_{ij} &\in \{0,1\}, && \text{for each } i,j \in N, && (5)\\
y_i &\in \{0,1\}, && \text{for each } i \in N. && (6)
\end{aligned}$$

The constraints (2) ensure that each location $j \in N$ is assigned to some center $i \in N$, the constraints (3) ensure that whenever a location j is assigned to a center i, then a center must have been opened at i, and (4) ensures that at most k centers are open.

2.1 Outline of the method

Consider the linear programming relaxation to the integer program (1)–(6), where the 0-1 constraints (5) and (6) are replaced, respectively, with

$$\begin{aligned}
x_{ij} &\geq 0, && \text{for each } i,j \in N, && (7)\\
y_i &\geq 0, && \text{for each } i \in N. && (8)
\end{aligned}$$

Let $(\bar{x},\bar{y})$ denote a feasible solution to the LP relaxation and let $\bar{C}_{LP}$ denote its objective function value. For each location $j \in N$, let $\bar{C}_j$ denote the cost incurred by this fractional solution for assigning (one client at) location j, i.e.,

$$\bar{C}_j = \sum_{i\in N} c_{ij}\bar{x}_{ij} \qquad \text{for each } j \in N. \qquad (9)$$

Note that $\bar{C}_{LP} = \sum_{j\in N} d_j \bar{C}_j$. We will round the fractional solution $(\bar{x},\bar{y})$ to a feasible solution to the integer program of cost at most $6\frac{2}{3}\bar{C}_{LP}$. The outline of our approximation algorithm is as follows.

Step 1: First we simplify the problem instance by consolidating nearby locations. We will not change the linear programming solution $(\bar{x},\bar{y})$ but modify only the demands. Let d' denote the new set of demands. This modification will have the following properties:

- The modification will not increase the cost of the fractional solution $(\bar{x},\bar{y})$:

$$\bar{C}'_{LP} = \sum_{i,j\in N} d'_j c_{ij}\bar{x}_{ij} \leq \sum_{i,j\in N} d_j c_{ij}\bar{x}_{ij} = \bar{C}_{LP}.$$

- In the resulting instance, all locations with positive demand will be far from each other:

$$c_{ij} > 4\max(\bar{C}_i,\bar{C}_j) \qquad (10)$$
$$\text{for each } i,j \in N \text{ such that } d'_i > 0 \text{ and } d'_j > 0.$$

- Each feasible integer solution for the modified instance with demands d'_j, $j \in N$, can be converted to a feasible integer solution for the original instance, at an added cost of at most $4\bar{C}_{LP}$.

Let $N' = \{j : d'_j > 0\}$ denote the set of locations with positive demand. We will show that this modified instance is simpler, e.g., $|N'| \leq 2k$.

Step 2: Next, we simplify the structure of the solution by consolidating nearby fractional centers. First, we modify the solution $(\bar{x},\bar{y})$ to obtain a new solution (x',y') such that

$$\begin{aligned}
y'_j &= 0, && \text{for each } j \in N \text{ such that } d'_j = 0, && (11)\\
y'_i &\geq \frac{1}{2}, && \text{for each } i \in N \text{ such that } d'_i > 0. && (12)
\end{aligned}$$

We will refer to such a solution as a $\frac{1}{2}$*-restricted solution*. Further, the cost of the $\frac{1}{2}$-restricted solution produced is at most $2\bar{C}_{LP}$.

We further modify the $\frac{1}{2}$-restricted solution, without increasing its cost, to obtain an $\{\frac{1}{2},1\}$-solution $(\hat{x},\hat{y})$: that is, $\hat{y}_i = 0$ for each $i \notin N'$ and $\hat{y}_i$ is either $\frac{1}{2}$ or 1 for each $i \in N'$.

Step 3: Finally, we show how to convert a feasible $\{\frac{1}{2},1\}$-integral solution to the linear programming relaxation to a feasible integral solution of cost at most $\frac{4}{3}$ times the cost of the $\{\frac{1}{2},1\}$-integral solution.

Using these steps, we obtain the following theorem.

Theorem 1 *For the metric k-median problem, the outlined method yields a $6\frac{2}{3}$-approximation algorithm.*

Proof: Suppose that the feasible solution $(\bar{x},\bar{y})$ to be rounded is an optimal solution to the linear program (1)–(4) and (7)–(8). Clearly, $\bar{C}_{LP}$ is a lower bound on the optimal value of the k-median problem. Steps 1–3 above create a feasible integer solution to the modified input of cost at most $2\frac{2}{3}\bar{C}_{LP}$. This can then be converted to an integer solution to the original instance while adding at most $4\bar{C}_{LP}$ to the cost. This results in a feasible integer solution of cost at most $6\frac{2}{3}\bar{C}_{LP}$, i.e., at most $6\frac{2}{3}$ times the minimum possible. □

3 Step 1: Consolidating locations

Recall that $\bar{C}_j = \sum_{i\in N} c_{ij}\bar{x}_{ij}$ is the cost that the linear program pays for assigning one unit of demand at location j. The goal of the first step of the method is to consolidate demands at nearby locations. More formally, we want to guarantee that for all pairs of locations i,j both with positive demand, $c_{ij} > 4\max(\bar{C}_i, \bar{C}_j)$. The modified instance of the location problem is obtained without changing the LP solution $(\bar{x}, \bar{y})$. For some locations j, we will simply "move" the demand d_j to a nearby location $j' \in N$, which is no more than $4\bar{C}_j$ away from location j.

Assume for notational simplicity that $N = \{1, \dots, n\}$, and the locations are indexed in increasing order of $\bar{C}_j$, i.e., $\bar{C}_1 \leq \ldots \leq \bar{C}_n$. We create the modified instance as follows. Start with $d'_j \leftarrow d_j$, for each locations $j \in N$.

- Consider the locations $j = 1, \dots, n$ in this order. When considering location j, check if there is another location $i < j$ such that $d'_i > 0$ and $c_{ij} \leq 4\bar{C}_j$. If there is such a location, then add the demand of location j to one such location i: select some i for which $c_{ij} \leq 4\bar{C}_j$, and set

$$d'_i \leftarrow d'_i + d'_j; \tag{13}$$
$$d'_j \leftarrow 0. \tag{14}$$

Let d' denote the resulting demands, whereas d denotes the original demands. Let N' denote the set of locations with $d'_j > 0$, i.e., $N' = \{j \in N : d'_j > 0\}$. Note that $(\bar{x}, \bar{y})$, the feasible solution to the LP relaxation with the original demands, is also a feasible solution for the modified input. The next lemma follows directly from the algorithm.

Lemma 2 *Locations $i, j \in N'$ satisfy (10).*

Lemma 3 *The cost of the fractional solution $(\bar{x}, \bar{y})$ for the input with modified demands is at most its cost for the original input, that is,*

$$\sum_{i,j \in N} d'_j c_{ij} \bar{x}_{ij} \leq \sum_{i,j \in N} d_j c_{ij} \bar{x}_{ij}. \tag{15}$$

Proof: The LP objective function value (1) of the solution $(\bar{x}, \bar{y})$, for any demands $\bar{d}$, can be written as $\sum_{j \in N} \bar{d}_j \bar{C}_j$. The lemma now follows directly, since d' is obtained by moving demand from a location j to a location i with $\bar{C}_i \leq \bar{C}_j$. □

Lemma 4 *For any feasible integer solution (x', y') for the modified input with demands d', there is a feasible integer solution for the original input of cost at most $4\bar{C}_{LP}$ more than the cost of (x', y') with demands d'.*

Proof: Consider a location $j \in N$ that has its demand moved to j' in the modified input. Each location j was moved by at most $4\bar{C}_j$, and so we have that $c_{j'j} \leq 4\bar{C}_j$. If j is assigned to the center to which j' is assigned by x', then, by the triangle inequality, restoring j to its original position increases its unit assignment cost by at most $4\bar{C}_j$. Summing over all j, we obtain the desired bound. □

4 Step 2: Consolidating centers

In this section, we consider the problem with demands d'_j for $j \in N'$ as defined in the previous section, i.e., we assume that the modified demands satisfy (10).

We simplify the structure of the solution by consolidating nearby fractional centers. First, we modify the solution $(\bar{x}, \bar{y})$ to obtain a new solution (x', y') of cost at most $2\bar{C}_{LP}$ such that $y'_i = 0$ for each $i \notin N'$ and $y'_i \geq \frac{1}{2}$ for each $i \in N'$.

We further modify the $\frac{1}{2}$-restricted solution to obtain, without increasing its cost, a $\{\frac{1}{2}, 1\}$-solution $(\hat{x}, \hat{y})$: that is, $\hat{y}_i = 0$ for each $i \notin N'$ and $\hat{y}_i$ is either $\frac{1}{2}$ or 1 for each $i \in N'$.

The first step will be accomplished using the filtering technique of Lin & Vitter [18]. The main observation behind this technique is the following lemma, which shows that each location j has at least half of its demand assigned to relatively "nearby" partially open centers: for each $j \in N$, there is a total of at least 1/2 of an open center within a radius of $2\bar{C}_j$.

Lemma 5 *For any feasible fractional solution $(\bar{x}, \bar{y})$, $\sum_{i:c_{ij} \leq 2\bar{C}_j} \bar{y}_i \geq \frac{1}{2}$ for each $j \in N$.*

Proof: Recall that $\bar{C}_j = \sum_{i \in N} c_{ij} \bar{x}_{ij}$. In any weighted average, less than half of the total weight can be given to values more than twice the average; that is,

$$\sum_{i:c_{ij} > 2\bar{C}_j} \bar{x}_{ij} < \frac{1}{2}. \tag{16}$$

Since $\sum_{i \in N} \bar{x}_{ij} = 1$ and $\bar{x}_{ij} \leq \bar{y}_i$ for each $i \in N$,

$$\sum_{i:c_{ij} \leq 2\bar{C}_j} \bar{y}_i \geq \sum_{i:c_{ij} \leq 2\bar{C}_j} \bar{x}_{ij} \geq \frac{1}{2}. \tag{17}$$

This proves the lemma. □

Now we are ready to prove the following result.

Theorem 6 *There is a $\frac{1}{2}$-restricted solution (x', y') of cost at most $2\bar{C}_{LP}$.*

Proof: We will modify the solution $(\bar{x}, \bar{y})$ by moving each fractional center to the location in N' closest it. We will prove that this produces a $\frac{1}{2}$-restricted solution of cost at most twice the cost of the solution $(\bar{x}, \bar{y})$.

We start by setting $x' \leftarrow \bar{x}$ and $y' \leftarrow \bar{y}$. First consider only the constraints (11). For each location $i \in N$ at which there is a partially open center (i.e., $y'_i > 0$) and yet $d'_i = 0$, we completely close the fractional center at location i, and move it to the closest location $j \in N'$. The fractional center at i is moved to j by setting

$$y'_j \leftarrow \min(1, y'_i + y'_j); \tag{18}$$
$$y'_i \leftarrow 0. \tag{19}$$

When we move the fractional center at i to location j, then we also change the assignments accordingly; that is, for each location $j' \in N$, we set

$$x'_{jj'} \leftarrow x'_{jj'} + x'_{ij'}; \quad (20)$$
$$x'_{ij'} \leftarrow 0. \quad (21)$$

First observe that these changes result in a feasible linear programming solution that also satisfies the constraints (11). We will worry about the constraints (12) later.

Next we show that these changes at most double the cost of the solution. Consider some location j' with positive demand $d'_{j'}$. For any fractional center i that is moved, say, to location j, consider the corresponding change in the assignment cost of j'. The fraction $x_{ij'}$ previously assigned to i is now assigned to j instead. By the triangle inequality, we get that $c_{jj'} \le c_{ij} + c_{ij'}$. By the fact that the fractional center at i is moved to j, and not to j', we know that $c_{ij} \le c_{ij'}$. Hence, $c_{jj'} \le 2c_{ij'}$. By considering all demand points j' and all fractional centers i that are moved, we see that this claim implies that the cost of (x', y') is at most twice the cost of $(\bar{x}, \bar{y})$.

Finally, we must show that the new solution satisfies the constraints (12). Note that by Step 1, any two locations with positive demand are far apart, i.e., the demands d' satisfy (10). This implies that for each location j with positive demand, all partially opened centers within a $2\bar{C}_j$ radius of j must be closer to j than to any other location in N'; hence all of these will be moved to j. Lemma 5 implies that, the sum of the fractional center values on the locations within this radius of j is at least $\frac{1}{2}$. All of these fractional centers are moved to j, and hence we see that $y'_j \ge 1/2$, as required. □

Note that the existence of a $\frac{1}{2}$-restricted solution for the modified instance implies that there are at most $2k$ locations with positive modified demand. If we open a center at each such location, we would get a solution (for the modified input) of cost 0. This implies that the original input has a solution using at most $2k$ centers that costs at most $4\bar{C}_{LP}$.

Next, we obtain a $\{\frac{1}{2}, 1\}$-solution for the modified instance. We begin by examining the structure of any $\frac{1}{2}$-restricted solution. For any $\frac{1}{2}$-restricted solution with fractional center values y, it is easy to express the corresponding optimal fractional assignment x. Consider a location $j \in N'$. The optimal linear programming solution will have $x_{jj} = y_j$, i.e., each location in N' will use its own partially open center to the maximum extent possible. Since $y_j \ge 1/2$, this leaves less than half of j left to be assigned, and so, for any $k \in N'$, we can set $x_{kj} = 1 - x_{jj} (= 1 - y_j)$, and be sure that the resulting solution is feasible (i.e., $x_{kj} \le 1/2 \le y_k$). The best alternative is to let k be the location in N' that is closest to j (other than j itself). For each $j \in N'$, let $s(j)$ be the closest location to j in N' (other than j), where ties are broken by choosing the location with smallest index. We have just proved the following lemma.

Lemma 7 *The minimum cost of a $\frac{1}{2}$-restricted solution (x', y') is $\sum_{j \in N'} d'_j (1 - y'_j) c_{s(j)j}$.* □

This lemma implies that we can view the cost of a $\frac{1}{2}$-restricted solution (x', y') as a function solely of the values y'. Using this lemma, we are ready to prove the following theorem.

Theorem 8 *For any $\frac{1}{2}$-restricted solution (x', y'), there exists a $\{\frac{1}{2}, 1\}$-integral solution of no greater cost.*

Proof: By Lemma 7, the cost of the $\frac{1}{2}$-restricted solution (x', y') can be expressed as

$$\sum_{j \in N'} d'_j c_{s(j)j} - \sum_{j \in N'} d'_j c_{s(j)j} y'_j. \quad (22)$$

We will construct a $\frac{1}{2}$-restricted solution whose cost is minimum possible. The first term of (22) is a constant independent of y'. The second term is maximized if the values y'_j are as large as possible for locations $j \in N'$ with the largest values $d'_j c_{s(j)j}$. The $\frac{1}{2}$-restricted solution of minimum cost is obtained as follows. Let $n' = |N'|$. Sort the locations $j \in N'$ in decreasing order of their weight $d'_j c_{s(j)j}$. Set $\hat{y}_j = 1$ for the first $2k - n'$ locations, and set $\hat{y}_j = 1/2$ for the remaining $2(n' - k)$ locations. The solution $(\hat{x}, \hat{y})$ is a $\{\frac{1}{2}, 1\}$-integral solution. By the above discussion, the cost of $(\hat{x}, \hat{y})$ is at most the cost of the $\frac{1}{2}$-restricted solution (x', y'). □

Thus, Step 2 produces a $\{\frac{1}{2}, 1\}$-integral solution $\hat{y}_j$, $j \in N'$, so that each location $j \in N'$ is assigned to itself and at most one other location $s(j)$. Of course, if $\hat{y}_j = 1$, then there is no need to consider $s(j)$ at all; we will adopt the convention that if $\hat{y}_j = 1$, then $s(j) = j$.

5 Step 3: Rounding a $\{\frac{1}{2}, 1\}$-solution to an integral one

Finally, we show how to convert the $\{\frac{1}{2}, 1\}$-solution obtained in the previous section to an integral solution. We will first show a simple method that increases the cost incurred by at most a factor of 2. Then we outline how to convert to an integer solution while increasing the cost incurred by at most a factor of $\frac{4}{3}$.

Consider the $\{\frac{1}{2}, 1\}$-solution $(\hat{x}, \hat{y})$ to the modified instance. Recall the structure of any optimal $\{\frac{1}{2}, 1\}$-restricted solution. For each location $j \in N'$ the optimal assignment is to set $\hat{x}_{jj} = \hat{y}_j$, and if $\hat{y}_j < 1$ then set $\hat{x}_{kj} = 1 - \hat{y}_j$ for the closest other location $k = s(j)$ in N'. We use these $(j, s(j))$ pairs as edges to obtain a collection of trees spanning the vertices in N'. We then use the collection of trees to obtain an integral solution.

More precisely, we build a collection of trees as follows. For each node $i \in N'$ with $\hat{y}_i = \frac{1}{2}$, draw a directed edge from i to $s(i)$. By the definition of $s(j)$, each component of this graph contains cycles of length at most 2; in fact, each component has at most 1 cycle, which then corresponds to the closest pair of nodes in it. For each cycle, we choose one of the two vertices as a root and delete the directed edge from the root to the other vertex. This will result in a collection of rooted trees. We say that $s(i)$ is the parent of i, if we have a directed edge from i to $s(i)$.

We first describe a simple rounding scheme that produces an integral solution of cost at most twice the cost of the $\{\frac{1}{2}, 1\}$-solution.

Define the level of any node to be the number of edges on the path from the node to the root.

The locations of the centers are chosen as follows. We build centers at all nodes i such that $\hat{y}_i = 1$. We divide the nodes $\{i | \hat{y}_i = \frac{1}{2}\}$ into two subsets corresponding to the odd and even levels and build centers at nodes in the smaller of the two subsets. This ensures that the number of centers built is at most $\sum \hat{y}_i \leq k$.

It is easy to argue this rounding at most doubles the cost of the solution. Each node i with $\hat{y}_i = 1$ gets a center and its contribution to the cost of the solution is 0. For each node i with $\hat{y}_i = \frac{1}{2}$, either i or $s(i)$ is chosen as a center (since one is at an odd level and the other is at an even level). Hence the contribution of i to the cost of the solution is at most $d'_i c_{s(i)i}$ which is twice its contribution to the cost of the $\{\frac{1}{2}, 1\}$-solution.

Thus, the cost of the integral solution produced is at most twice the cost of the $\{\frac{1}{2}, 1\}$-integral solution. If we substitute this bound into the proof of Theorem 1, we get an 8-approximation algorithm for the k-median problem.

Next we describe a better procedure for rounding the $\{\frac{1}{2}, 1\}$-solution. Consider the new instance defined on vertices of N', where the demands are the same as in the modified instance produced by Step 1, and the underlying metric space is the collection of trees that we build in Step 3. This change in the metric space keeps the length of the tree edges unchanged, increases the distance of some pairs of locations, but the value of the $\{\frac{1}{2}, 1\}$-solution $(\hat{x}, \hat{y})$ remains unchanged. The k-median problem can be solved optimally when the metric space is a tree [22] and the dynamic programming algorithm is easily modified to work on a collection of trees. Now, we can ignore the $\hat{y}_i$ values and solve the instance on this collection of trees optimally.

To prove that the resulting solution is close to optimal on the original metric space we need to show that there is an integral solution with cost at most a factor of 4/3 times the cost of the $\{\frac{1}{2}, 1\}$-solution. The previous analysis guarantees the existence of an integral solution for the collection of trees with cost at most twice the cost of the $\{\frac{1}{2}, 1\}$-solution. In fact, for the k-median problem on a tree, the integrality gap (with respect to the optimal LP solution) was known to be $2 - \frac{2}{k+1}$ [24, 23]. In Section 7, we will prove an integrality gap of $\frac{4}{3}$ for the $\{\frac{1}{2}, 1\}$-solution. This implies that the optimal solution for the new instance defined by the collection of trees is of cost at most $\frac{4}{3}$ times the cost of the $\{\frac{1}{2}, 1\}$-integral solution.

At this point we will make a slight detour and review the algorithm, and then return to the upper bound of the integrality gap of $\{\frac{1}{2}, 1\}$-solutions.

6 The algorithm in retrospect

Reflecting on the algorithm and the analysis we presented, it turns out that the underlying algorithm is much simpler than it might appear from our description. The algorithm can be viewed as building a collection of trees from the LP solution and then solving the problem optimally on this collection. Here is the complete algorithm stated slightly differently.

Step 1′: Solve the LP and perform (a modified version of) Step 1 as in Section 3. Here, instead of modifying the demands, we build a collection of stars on the set of locations N. We have one star in the collection for every location $j \in N'$; this star consists of edges from j to each location i such that the demand of i was moved to j in the original Step 1.

Step 2′: Obtain a collection of trees by connecting each vertex $j \in N'$ to its nearest neighbor $s(j) \in N'$ (ties broken as before).

Step 3′: Solve the modified instance consisting of the original demands where the underlying metric is the collection of trees produced in Step 2.

The 'algorithm' presented in Steps 1-3 proves that the integrality gap of Steps 1′-3′ is at most $6\frac{2}{3}$.

7 Bounding the integrality gap of a $\{\frac{1}{2}, 1\}$-solution

We will show that there exists an integral solution of cost at most $\frac{4}{3}$ times that of the $\{\frac{1}{2}, 1\}$-integral solution. This is tight, as shown by the example of a 4 node tree with 0 demand at the root, and demand 1 at each of the 3 children, and $k = 2$.

The basic idea is to produce a probability distribution on integral solutions of expected cost at most $\frac{4}{3}$ times the cost of the $\{\frac{1}{2}, 1\}$-solution.

We will prune the set of trees in the forest in two phases to construct trees with at most 3 levels. The distribution will be defined on these small level trees.

In the earlier proof that the integrality gap is at most 2, a node i either has an open center, or its nearest neighbor $s(i)$ has one. In this section, we will relax this condition and instead require that if neither i nor $s(i)$ has an open center, then $s(s(i))$ must have one. However, that might create a problem when $i \neq s(i)$ and $s(s(i)) = i$, i.e., for pairs of nodes that are closest to each other. In this case, we require that either i or $s(i)$ has an open center.

For each node in the tree, its parent is the location in N' closest to it. Thus, the distance from $s(i)$ to $s(s(i)$ is at most the distance from i to $s(i)$. It would be desirable that the distance from i to $s(s(i))$ is not twice the distance to $s(i)$, which is possible in the worst case. We will ensure that for each node i, either $s(s(i))$ will be within $\frac{3}{2}$ times the distance to $s(i)$, or there is a center open at i or $s(i)$. The first phase of the tree pruning is designed to guarantee this property. The idea is that if c_{ij}, the distance from i to the closest child j of i, is more than half of the distance from i to $s(i)$, then we delete the edge between i and $s(i)$ from our tree, and make i the root of a new tree. We must be careful with the order in which we consider the locations for such deletions to avoid creating single node components.

Creating small level trees :

Lemma 9 *We can break the trees in the forest and create a collection of trees Γ_1 such that the trees of Γ_1 span the nodes of the original trees, and every tree $T_i \in \Gamma_1$ with root i satisfies the following properties*

- *P1: T_i has at least two vertices.*
- *P2: If $\hat{y}_i = \frac{1}{2}$ then the nearest child of i is within a distance $2c_{s(i)i}$.*
- *P3: For location $j \neq i$ that is included in the tree T_i the distance to its parent is $s(i)$.*
- *P4: Along a path from a leaf to the root i, the lengths of the edges decrease by at least a factor of 2.*

Proof: Consider a tree T in the forest. Recall the level of a node is the number of edges required to reach the root from the node. The subtree rooted at a vertex v represents the tree restricted to the descendants of the node v. The notation $T_1 - T_2$ denotes the tree T_1 restricted to all the vertices not in T_2.

We will visit the nodes of each tree by visiting all descendants of a node before visiting the node, and visiting the nearest child of the root as the last node before visiting the root. Consider one of the original trees T. We start by setting $T' = T$, and we will modify T' as we visit the nodes.

Assume we are visiting a node i that is neither the root, nor the closest child of the root. Let T_i denote the subtree of T' rooted at i. If node i has a child, and the distance from i to its nearest child j is less than twice the distance to its parent $s(i)$, i.e., $c_{ij} \leq 2c_{s(i)i}$, then prune the tree T' by removing the subtree T_i rooted at node i, and adding T_i to the collection Γ_1.

If i is the next-to-last node in the order and $\hat{y}_{s(i)} = \frac{1}{2}$. In this case $s(s(i)) = i$. We make $s(i)$ a child of i and add this modified subtree T_i' rooted at i to Γ_1.

If we are visiting the root of T (all other nodes in the tree have already be visited), then the current tree T' is rooted at i, and we add this tree to Γ_1.

In the full version of the paper we show that this construction satisfies the properties claimed. □

Next we will further prune the trees in the collection Γ_1 so that the resulting trees have at most three levels (including the root). Consider the following steps:

(a) Consider a tree $T_i \in \Gamma_1$. Remove all the edges between the levels $2p+1$ and $2p+2$ for all $p \geq 0$. (Edges between levels $2p$ and $2p+1$ are undisturbed.)

(b) The above will result into a decomposition into trees having at most two levels (stars). For the singleton nodes created in this process (nodes that are in level $2p+2$ and have no children), attach them to their parent in T_i. Represent this collection by $\Gamma_2(T_i)$.

After these two steps are performed on the entire forest of trees, let Γ be the collection of trees of at most 3 levels formed in this fashion. $\Gamma = \bigcup_T \Gamma_2(T)$.

Distribution over small trees : We will construct a probability distribution over integral solutions for the trees in Γ such that the expected cost is small compared to the cost of the $\{\frac{1}{2}, 1\}$-integral solution. We will need to introduce one more definition.

A tree $T_i \in \Gamma$ is defined to be an **even** tree if $\sum \hat{y}_j$ over vertices j in this tree is integral; otherwise it is defined to be an **odd** tree.

An even tree T_i with $\sum \hat{y}_v = p$ will be assigned p centers. Clearly the number of odd trees will be even. We will randomly choose half of the odd trees to be 1-trees. The others will be 0-trees. A tree T_i with $\sum \hat{y}_j = p + \frac{1}{2}$, for some integer p, will have $p+1$ centers if it is chosen to be a 1-tree, otherwise have p centers. It is immediate that we open exactly k centers.

There are two cases, depending on whether the root i of a tree has $\hat{y}_i$ equal to 1 or $\frac{1}{2}$. We present the former, and simpler, case first. Throughout this discussion, we will choose a subset of k elements from a given set, where the subset is chosen uniformly at random among all such subsets; we shall say that we then *uniformly choose* k elements. The selections for different trees are made independently.

Case 1: Root i of a tree has $\hat{y}_i = 1$. Consider the following distribution to select centers:

1. If the root has $2r$ descendants uniformly choose r of them to build a center.
2. If the root has $2r+1$ descendants, uniformly choose $r+1$ of them if the tree is a 1-tree, and uniformly choose r of them otherwise.

Lemma 10 *For Case 1, the expected distance of each node i to its nearest center is at most $\frac{2}{3}c_{s(i)i}$.*

Proof: Consider any node i that is not the root. First, we observe that for each node i, the probability that a center is opened at i is $\frac{1}{2}$. This is immediate if the tree is an even tree. For an odd tree, we get this by averaging the probabilities given that the tree is a $0-$tree or a $1-$tree.

Now consider a node i at level 1. If i does not have a center, then the distance to its nearest center, the root, is at most $c_{is(i)}$; otherwise it has a center at distance 0. Thus, the expected distance is $\frac{1}{2}c_{is(i)}$.

Finally, let i be a node at level 2. As before, node i is selected as a center with probability $\frac{1}{2}$. Assume node i is not selected; then the closest center is at distance at most $c_{is(i)}$ if its parent $s(i)$ is selected, and at distance at most $\frac{3}{2}c_{is(i)}$ if neither i nor $s(i)$ are selected. To finish the proof we compute the conditional expectation that $s(i)$ is selected assuming that i is not selected. □

Case 2: Root i of a tree has $\hat{y}_i = \frac{1}{2}$. Before describing the distribution, we outline its desired properties, as follows:

D1: If the root is not chosen, then all nodes in level 1 of the tree must be chosen.

D2: The root of any tree in Γ will be chosen with probability at least $\frac{2}{3}$.

D3: Each node in level 1 must be chosen with probability at least $\frac{1}{3}$.

D4: Each node in level 2 must be chosen with probability at least $\frac{1}{2}$.

The next lemma explains why property $D1$ is desired.

Lemma 11 *If property $D1$ is true, if a root node i in tree $T_i \in \Gamma$ is not chosen then it will have a center within a distance at most $2c_{is(i)}$.*

Proof: By property $D1$, if the root i of a tree T_i is not chosen, then its nearest child is chosen. This implies the lemma if a tree with root i was either an original tree, or was created by the first phase of the pruning process. If the tree was created in the second phase of the pruning process than we use an ancestor of the closest node $s(i)$ (which is in a separate subtree) to prove the lemma. □

The root i is at level 0. If both the number of level-1 nodes is odd and the number of level-2 nodes is even then we consider the following distribution:

1. The distribution will be the product of these two distributions: the nodes in level 0 (root) and level 1 will have a distribution and independently, the level-2 nodes will have a distribution. We will describe the distribution over level-2 nodes first.
2. If there are $2r$ level-2 nodes, uniformly choose r to open as centers. If there are $2r+1$ nodes in level 2, (and so the number of level-1 nodes is odd, and this is an odd tree), uniformly choose $r+1$ of them if this is a 1-tree, and uniformly choose r of them if it is 0-tree.
3. If there is a single level-1 node, then open a center at the root with probability $\frac{2}{3}$ and at the child with probability $\frac{1}{3}$.
4. If the number of level-1 nodes is $2t-1$ ($t > 1$), open a center at the root and uniformly choose $t-1$ level-1 vertices as centers.
5. If the number of level-1 nodes is 2, (and so the number of level-2 nodes is even and this is an odd tree), then, if the tree is a 1-tree, build a center at both level-1 nodes with probability $\frac{2}{3}$, and build a center at the root and any one of the level-1 nodes with probability $\frac{1}{3}$. If the tree is a 0-tree, build a center at the root.
6. If the number of level-1 nodes is $2t$ ($t > 1$), open a center at the root and uniformly choose t level-1 nodes if the tree is a 1-tree. Otherwise, open a center at the root and uniformly choose $t-1$ level-1 nodes as centers.

If there are an even number of level-1 nodes and an odd number of level-2 nodes, we consider a different distribution:

1. If there are 2 level-1 nodes and $2r+1$ level-2 nodes, then with probability $\frac{2}{3}$ open centers at the root and $r+1$ uniformly chosen level-2 nodes. With probability $\frac{1}{3}$, open centers at the 2 level-1 nodes and r uniformly chosen level-2 nodes.
2. If there are $2t$ level-1 nodes ($t > 1$) and $2r+1$ level-2 nodes, always open a center at the root. With probability $\frac{1}{2}$, uniformly choose $t-1$ level-1 nodes and, independently, uniformly choose $r+1$ level-2 nodes. With probability $\frac{1}{2}$, uniformly choose t level-1 nodes and, independently, uniformly choose r level-2 nodes.

Lemma 12 *The above distribution satisfies the four properties $D1$-$D4$, and by Lemma 11, the expected distance to the nearest center for each node j is at most $\frac{2}{3}c_{s(j)j}$.*

Proof: Omitted. □

Since the cost of the nodes in the $\{\frac{1}{2}, 1\}$-solution is $\sum_i \frac{1}{2}d'_i c_{s(i)i}$ and the expected cost of the above distribution is at most $\sum_i \frac{2}{3}d'_i c_{s(i)i}$, we can conclude the following,

Theorem 13 *The $\{\frac{1}{2}, 1\}$-solution has an integrality gap of at most $\frac{4}{3}$.*

8 Extensions

In this section we describe the extensions of the k-median problem that we considered, and give our results for each. The full details of the algorithms, and their proofs are deferred to the full version of the paper.

8.1 The k-median problem with center costs

In this subsection we consider a common generalization of the uncapacitated facility location problem and the k-median problem: we add to the former problem the additional constraint that at most k facilities may be opened. This problem has also been studied extensively in the OR literature [9]. For the metric case of this problem, we give a 9.8-approximation algorithm, the first approximation algorithm for this problem with a constant performance guarantee.

The input to this problem consists of the locations N, the parameter k, metric assignment costs c_{ij}, for each $i, j \in N$, and a center cost f_i, for each $i \in N$. The problem is to select at most k of the locations as centers, and assign each location in N to one of the selected centers so as to minimize the total cost of the selected centers plus the total assignment cost incurred. The problem can be stated as an integer program using the same variables and constraints as the k-median problem; that is, minimize $\sum_{i,j \in N} d_j c_{ij} x_{ij} + \sum_{i \in N} f_i y_i$ subject to the constraints (2)-(6).

We can modify our k-median algorithm to obtain an approximation algorithm for this problem. As before, we start by considering the linear programming relaxation to the integer program. We round the fractional solution $(\bar{x}, \bar{y})$ to a feasible solution to the integer program following the same outline as in Section 2, where the main difference is that we have to pay extra attention to opening centers that are relatively cheap.

Theorem 14 *The k-median problem with center costs can be approximated within a factor of 9.8.*

8.2 The capacitated k-median problem

We consider the *capacitated* version of the k-median problem which is similar to the uncapacitated problem with an additional parameter M and the additional constraint that the total demand serviced by any center is at most M. There are two variants of the problem with capacities. In the capacitated problem with *unsplittable demand*, each location must be assigned to a single center as before. In the capacitated problem with *splittable demand*, each location may split its demand across multiple centers. The constant-factor approximation algorithms for both the uncapacitated facility location problem and the k-center problem can be extended to the capacitated versions of these problems [20, 8, 2, 15].

We show that a modified version of the algorithm for the uncapacitated case gives a constant factor approximation for the capacitated problem with a constant factor blowup in the capacity.

This blowup in capacity is inevitable if we use the cost of the LP solution as a lower bound and insist on a solution with at most k centers. It is possible to construct examples which demonstrate that in order to obtain an integral solution whose cost is within a constant factor of the cost of the LP solution, we must either increase the capacity by a factor of at least 2 or increase the number of centers by a factor of at least 2.

We begin by solving the LP for the k-median problem with the following additional constraint that restricts the load on every center to be at most M:

$$\sum_{j \in N} d_j x_{ij} \le M y_i \quad \text{for each } i \in N.$$

We first identify a set of locations N' as in Step 1 of the algorithm for the k-median problem. However, demands are not modified as before.

In the uncapacitated problem, centers were built only at locations in N'. We cannot do this in the capacitated case, as this would not satisfy the capacity constraints. Instead, for every location i we consider the set of fractional centers assigned to i and redistribute the total fractional center value to the location i and the locations closest to it so that every positive fractional center has value at least a $\frac{1}{2}$.

Theorem 15 *The capacitated k-median problem with splittable demands can be approximated within a factor of* 16 *with at most factor* 3 *blowup in capacity.*

The algorithm for the splittable case can be combined with a result of Shmoys and Tardos [19] to yield an algorithm for the unsplittable case.

Theorem 16 *The capacitated k-median problem with unsplittable demands can be approximated within a factor of* 16 *with at most factor* 4 *blowup in the capacity.*

8.3 Distance with relaxed triangle inequality

We can extend our algorithm and analysis to give a constant factor approximation for the problem where the objective is to minimize the sum of the squares of the distances of vertices to their nearest centers. The distance measure satisfies the property $c_{ik} \le \delta(c_{ij} + c_{jk})$. It follows from the fact that wherever triangle inequality is used in the proof of the approximation factor, a relaxed version of the triangle inequality can be used with a loss in the approximation guarantee.

Acknowledgments This work is a combined version of two earlier papers, by Charikar & Guha, and by Tardos & Shmoys, which independently obtained nearly identical results. We would like to thank David Williamson for helping to facilitate the merger, and Chandra Chekuri, Fabian Chudak, Ashish Goel, and Rajeev Motwani for several useful discussions at the starting point of this research. Part of this work was done while the first two authors were visiting IBM T. J. Watson Research Center in Yorktown Heights.

References

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

[2] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15:385–415, 1993.

[3] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[4] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[5] 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 114–123, 1998.

[6] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture Notes in Computer Science*, pages 180–194, Berlin, 1998. Springer.

[7] F. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. Unpublished manuscript, 1998.

[8] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the capacitated facility location problem. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages S875–S876, 1999.

[9] 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. John Wiley and Sons, Inc., New York, 1990.

[10] M. E. Dyer and A. M. Frieze. A simple heuristic for the p-center problem. *Oper. Res. Lett.*, 3:285–288, 1985.

[11] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[12] 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, 1998.

[13] D. S. Hochbaum and D. B. Shmoys. A best possible approximation algorithm for the k-center problem. *Math. Oper. Res.*, 10:180–184, 1985.

[14] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems, Part II: p-medians. *SIAM J. Appl. Math.*, 37:539–560, 1979.

[15] S. Khuller and Y. J. Sussmann. The capacitated k-center problem. In *Proceedings of the 4th Annual European Symposium on Algorithms, Lecture Notes in Computer Science 1136*, pages 152–166, Berlin, 1996. Springer.

[16] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1–10, 1998.

[17] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inform. Proc. Lett.*, 44:245–249, 1992.

[18] 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, 1992.

[19] D. B. Shmoys and E. Tardos. An Approximation Algorithm for the Generalized Assignment Problem. *Math. Programming*, 62:461–474, 1993.

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

[21] M. Sviridenko. Personal communication, July, 1998.

[22] A. Tamir. An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs. *Oper. Res. Lett.*, 19:59–94, 1996.

[23] S. de Vries, M. Posner, and R. Vohra. The K-median Problem on a Tree. *Working paper*, Ohio State University, Oct. 1998.

[24] J. Ward, R. T. Wong, P. Lemke, and A. Oudjit. Properties of the tree K-median linear programming relaxation. Unpublished manuscript, 1994.