Approximation Algorithms for Metric Facility Location and k-Median Problems Using the Primal-Dual Schema and Lagrangian Relaxation

Kamal Jain[*] *Vijay V. Vazirani*[*]

Abstract

We present approximation algorithms for the metric uncapacitated facility location problem and the metric k-median problem achieving guarantees of 3 and 6 respectively. The distinguishing feature of our algorithms is their low running time: $O(m \log m)$ and $O(m \log m(L + \log(n)))$ respectively, where n and m are the total number of vertices and edges in the underlying complete bipartite graph on cities and facilities. The main algorithmic ideas are a new extension of the primal-dual schema and the use of Lagrangian relaxation to derive approximation algorithms.

1 Introduction

Given costs for opening facilities and costs for connecting cities to facilities, the uncapacitated facility location problem seeks a minimum cost solution that connects each city to an open facility. Clearly, this problem is applicable to a number of industrial situations. For a modern day application, consider the problem of locating proxy servers on the web. For this reason it has occupied a central place in operations research since the early 60's [3, 28, 30, 37, 38], and has been studied from the perspectives of worst case analysis, probabilistic analysis, polyhedral combinatorics and empirical heuristics (see [11, 34]). In the last few years, there has been renewed interest in tackling this problem, this time from the perspective of approximation algorithms [2, 13, 24, 29, 32, 36]. In this paper, we carry this further by developing an approximation algorithm based on the primal-dual schema. We further use this algorithm as a subroutine to solve a related problem, the k-median problem. The latter problem differs in that there are no costs for opening facilities, instead a number k is specified, which is an upper bound on the number of facilities that can be opened. The two algorithms achieve approximation guarantees of 3 and 6 respectively.

Both of our algorithms work only for the metric case, i.e., when the connecting costs satisfy the triangle inequality; both problems are **NP**-hard for this case as well. If the connection costs are unrestricted, approximating either problem is as hard as approximating set cover, and therefore cannot be done better than $O(\log n)$ factor, unless $\mathbf{NP} \subseteq \tilde{\mathbf{P}}$. For the first problem, this is straightforward to see, and for the second, this is established by Lin and Vitter [31].

[*] Address: College of Computing, Georgia Institute of Technology, Atlanta, GA 30332–0280, E-mail: {kjain,vazirani}@cc.gatech.edu. Research supported by NSF Grant CCR-9820896.

The distinguishing feature of our algorithms is their low running time: $O(m \log m)$ and $O(m \log m(L + \log(n)))$ respectively, where n and m are the total number of vertices and edges in the underlying complete bipartite graph on cities and facilities ($n = n_c + n_f$ and $m = n_c \times n_f$, where n_c and n_f are the number of cities and facilities) and L is the number of bits needed to represent a connecting cost. In particular, the running time of the first algorithm is dominated by the time taken to sort the connecting costs of edges. It is worth pointing out that our facility location algorithm is also suitable for distributed computation.

The first constant factor algorithm for the metric uncapacitated facility location problem was given by Shmoys, Tardos and Aardal [36], improving on Hochbaum's bound of $O(\log n)$ [25] (see [32] for another $O(\log n)$ factor algorithm). Their approximation guarantee was 3.16. After some improvements [24, 12], the current best factor is $(1 + 2/e)$, due to Chudak and Shmoys [13]. The drawback of these algorithms, based on LP-rounding, is that they need to solve large linear programs, and so have prohibitive running times for most applications. A different approach was recently used by Korupolu, Plaxton and Rajaraman [29]. They showed that a well known local search heuristic achieves an approximation guarantee of $(5 + \epsilon)$, for any $\epsilon > 0$. However, even this algorithm has a high running time of $O(n^6 \log n/\epsilon)$. Regarding hardness results, the work of [24, 39] establishes that a better factor than 1.463 is not possible, unless $\mathbf{NP} \subseteq \tilde{\mathbf{P}}$.

Researchers have felt that the primal-dual schema should be adaptable in interesting ways to the combinatorial structure of individual problems, and that its full potential has not yet been realized in the area of approximation algorithms. Our work substantiates this belief. We extend the scope of this schema in the following way: All primal-dual approximation algorithms obtained so far [1, 5, 22, 41, 21, 35, 26, 20, 23] work with a pair of covering and packing linear programs, i.e., a primal-dual pair of LP's such that all components of the constraint matrix, objective function vector and right hand side vector are non-negative. This includes, for instance, [41, 21] in which the overall LP-relaxation does have negative coefficients; however, the problem is decomposed into phases, and the relaxation used in each phase is a covering program. On the other hand, our algorithm works with primal and dual programs that do have negative coefficients.

Despite this added complexity, our algorithm has a particularly simple description: Each city j keeps raising its dual variable, α_j, until it gets connected to an open facility. All other primal and dual variables simply respond to this change, trying to maintain feasibility or satisfying complementary slackness conditions. For the latter, we give a new mechanism as well.

Until the work of [35] (which relaxed the dual program itself), all approximation algorithms based on the primal-dual schema used the mechanism formalized in [41]. In the first phase, an integral primal solution is found, satisfying the primal complementary slackness conditions; however, this solution may have redundancies. In the second phase, a minimal solution is extracted, typically via a reverse delete procedure, and in the process, dual complementary slackness conditions get satisfied with a relaxation factor. The final algorithm has this factor as its approximation guarantee.

Our first phase is similar. In the second phase, we first ensure that *all* complementary slackness conditions are satisfied; however, the primal solution may be infeasible. The solution is augmented – this time the primal conditions need to be relaxed by a factor of 3, which is also the approximation guarantee of the algorithm.

The k-median problem also has numerous applications, especially in the context of clustering, and has also been extensively studied. In recent years, the problem has found new clustering applications in the area of data mining (see [6]).

A non-trivial approximation algorithm for this problem eluded researchers for many years. The breakthrough was made by Bartal [4], who gave a factor $O(\log n \log \log n)$ algorithm using a probabilistic approximation of metric spaces by tree metrics. After a slight improvement to a factor of $O(\log k \log\log k)$ [7], a constant factor algorithm was recently obtained by Charikar, Guha, Tardos and Shmoys [9] using a technique of Lin and Vitter [32]. Their algorithm has an approximation guarantee of $6\frac{2}{3}$; however, it has the same drawback since it uses LP-rounding. Their algorithm uses several ideas from the constant factor algorithms obtained for the facility location problem, thus making one wonder if there is a deeper connection between the two problems.

In this paper, we establish such a connection between the the two problems: that a Lagrangian relaxation of the k-median problem is the facility location problem. This enables us to use our algorithm for the facility location problem as a subroutine to solve the k-median problem. The Lagrangian relaxation technique has been used implicitly in the past by Garg [18] to obtain a factor 3 algorithm for the k-MST problem. In this paper, we make its use transparent. We also abstract our ideas into a general method for deriving approximation algorithms using this technique.

These ideas also help solve a common generalization of the two problems – in which facilities have costs, and in addition, there is an upper bound on the number of facilities that can be opened. We give a factor 6 approximation algorithm for this problem as well; the previous bound was 9.8 [9].

The *capacitated* facility location problem, in which each facility i can serve at most u_i cities, has no non-trivial approximation algorithms. Part of the problem is that all LP-relaxations known for this problem have unbounded integrality gap (see [36]). In Section 5 we give a factor 4 approximation algorithm for the variant in which each facility can be opened an unbounded number of times; if facility i is opened y_i times, it can serve at most $u_i y_i$ cities. A special case of this version, in which the capacities of all the facilities are assumed to be equal, is solved with factor 3 in [14], again using LP-rounding. The special case of uniform capacities is solved within a factor of 5, using at most one extra copy of a facility at each location in [15].

Building on ideas presented in this paper, Charikar and Guha [8] have obtained the following improved results: a factor 1.853 algorithm for the facility location problem and a factor 4 algorithm for the k-median problem, both with running times of $O(n^3)$.

2 The metric uncapacitated facility location problem

The *uncapacitated facility location problem* seeks a minimum cost way of connecting cities to open facilities. It can be stated formally as follows: Let G be a bipartite graph with bipartition (F, C), where F is the set of *facilities* and C is the set of *cities*. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to (opened) facility i. The problem is to find a subset $I \subseteq F$ of facilities that should be opened, and a function $\phi : C \rightarrow I$ assigning cities to open facilities in such a way that the total cost of opening facilities and connecting cities to open facilities is minimized. We will consider the *metric* version of this problem, i.e., the c_{ij}'s satisfy the triangle inequality.

We will adopt the following notation: $n_c = |C|$ and $n_f = |F|$. The total number of vertices $n_c + n_f = n$ and the total number of edges $n_c \times n_f = m$.

Consider the following integer program for this problem, due to Balinski [3]. In this program, y_i is an indicator variable denoting whether facility i is open, and x_{ij} is an indicator variable denoting whether city j is connected to the facility i. The first constraint ensures that each city is connected to at least one facility, and the second ensures that this facility must be open.

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i \\ \text{subject to} \quad & \forall j\in C: \ \sum_{i\in F} x_{ij} \geq 1 \\ & \forall i\in F, j\in C: \ y_i - x_{ij} \geq 0 \\ & \forall i\in F, j\in C: \ x_{ij} \in \{0,1\} \\ & \forall i\in F: \ y_i \in \{0,1\} \end{aligned} \tag{1}$$

The LP-relaxation of this program is:

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i \\ \text{subject to} \quad & \forall j\in C: \ \sum_{i\in F} x_{ij} \geq 1 \\ & \forall i\in F, j\in C: \ y_i - x_{ij} \geq 0 \\ & \forall i\in F, j\in C: \ x_{ij} \geq 0 \\ & \forall i\in F: \ y_i \geq 0 \end{aligned} \tag{2}$$

The dual program is:

$$\begin{aligned} \text{maximize} \quad & \sum_{j\in C} \alpha_j \\ \text{subject to} \quad & \forall i\in F, j\in C: \ \alpha_j - \beta_{ij} \leq c_{ij} \\ & \forall i\in F: \ \sum_{j\in C} \beta_{ij} \leq f_i \\ & \forall j\in C: \ \alpha_j \geq 0 \\ & \forall i\in F, j\in C: \ \beta_{ij} \geq 0 \end{aligned} \tag{3}$$

2.1 Relaxing primal complementary slackness conditions

Our algorithm is based on the primal-dual schema. As stated in the introduction, instead of the usual mechanism of relaxing dual complementary slackness conditions, we relax the primal conditions. Before showing how this is done, let us give the reader some feel for how the dual variables "pay" for a primal solution by considering the following simple setting: suppose LP (2) has an optimal solution that is integral, say $I \subseteq F$ and $\phi : C \rightarrow I$. Thus, under this solution, $y_i = 1$ iff $i \in I$, and $x_{ij} = 1$ iff $i = \phi(j)$.

Let $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ denote an optimal dual solution. The reader can verify that primal and dual complementary slackness conditions imply the following facts:

- Each open facility is fully paid for, i.e., if $i \in I$, then

$$\sum_{j:\ \phi(j)=i} \beta_{ij} = f_i.$$

- Suppose city j is connected to facility i, i.e., $\phi(j) = i$. Then, j does not contribute for opening any facility besides i, i.e., $\beta_{i'j} = 0$ if $i' \neq i$. Furthermore, $\alpha_j - \beta_{ij} = c_{ij}$. So, we can think of α_j as the total price paid by city j; of this, c_{ij} goes towards the use of edge (i, j), and β_{ij} is the contribution of j towards opening facility i.

Suppose the primal complementary slackness conditions were relaxed as follows, while maintaining the dual conditions:

$$\forall j \in C: \quad (1/3)c_{\phi(j)j} \ \leq \ \alpha_j - \beta_{\phi(j)j} \leq c_{\phi(j)j},$$

and

$$\forall i \in I: \quad (1/3)f_i \ \leq \sum_{j:\ \phi(j)=i} \beta_{ij} \ \leq \ f_i.$$

Then, the cost of the (integral) solution found would be within thrice the dual found, thus leading to a factor 3 approximation algorithm. However, we would like to obtain the stronger inequality stated in Theorem 7, in which the dual pays at least one-third the connection cost, but must pay completely for opening facilities. This stronger inequality will be needed in order to use this algorithm to solve the k-median problem.

For this reason, we will relax the primal conditions as follows. The cities are partitioned into two sets, *directly connected* and *indirectly connected*. Only directly connected cities will pay for opening facilities, i.e., β_{ij} can be non-zero only if j is a directly connected city and $i = \phi(j)$. For an indirectly connected city j, the primal condition is relaxed as follows:

$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$

All other primal conditions are maintained, i.e., for a directly connected city j,

$$\alpha_j - \beta_{\phi(j)j} = c_{\phi(j)j},$$

and for each open facility i,

$$\sum_{j:\ \phi(j)=i} \beta_{ij} = f_i.$$

2.2 The algorithm

Our algorithm consists of two phases. In Phase 1, the algorithm operates in a primal-dual fashion. It finds a dual feasible solution, and also determines a set of tight edges and temporarily open facilities, F_t. Phase 2 consists of choosing a subset I of F_t to open, and finding a mapping, ϕ, from cities to I.

Algorithm 1

Phase 1

We would like to find as large a dual solution as possible. This motivates the following underlying process for dealing with the non-covering-packing pair of LP's. Each city j keeps raising its dual variable, α_j, until it gets connected to an open facility. All other primal and dual variables simply respond to this change, trying to maintain feasibility or satisfying complementary slackness conditions.

A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0. Initially, each city is defined to be *unconnected.* Throughout this phase, the algorithm raises the dual variable α_j for each unconnected city j uniformly at unit rate, i.e., α_j will grow by 1 in unit time. When $\alpha_j = c_{ij}$ for some edge (i, j), the algorithm will declare this edge to be *tight.* Henceforth, dual variable β_{ij} will be raised uniformly, thus ensuring that the first constraint in LP (3) is not violated. β_{ij} goes towards paying for facility i. Each edge (i, j) such that $\beta_{ij} > 0$ is declared *special.*

Facility i is said to be *paid for* if $\sum_j \beta_{ij} = f_i$. If so, the algorithm declares this facility *temporarily open.* Furthermore, all unconnected cities having tight edges to this facility are declared *connected* and facility i is declared the *connecting witness* for each of these cities. (Notice that the dual variables α_j of these cities are not raised anymore.) In the future, as soon as an unconnected city j gets a tight edge to i, j will also be declared connected and i will be declared the connecting witness for j (notice that $\beta_{ij} = 0$, and so edge (i, j) is not special). When all cities are connected, the first phase terminates. If several events happen simultaneously, the algorithm executes them in arbitrary order.

Remark 2 At the end of Phase 1, a city may have paid towards temporarily opening several facilities. However, we want to ensure that a city pay for only the facility that it is eventually connected to. This is ensured in Phase 2, which chooses a subset of temporarily open facilities for opening permanently.

Phase 2

Let F_t denote the set of temporarily open facilities and T denote the subgraph of G consisting of all special edges. Let T^2 denote the graph that has edge (u, v) iff there is a path of length at most 2 between u and v in T, and let H be the subgraph of T^2 induced on F_t. Find any maximal independent set in H, say I. All facilities in the set I are declared *open.*

For city j, define $\mathcal{F}_j = \{i \in F_t \mid (i, j) \text{ is special}\}$. Since I is an independent set, at most one of the facilities in $\mathcal{F}_t$ is opened. If there is a facility $i \in \mathcal{F}_j$ that is opened, then set $\phi(j) = i$, and declare city j *directly connected.* Otherwise, consider tight edge (i', j) such that i' was the connecting witness for j. If $i' \in I$, again set $\phi(j) = i'$ and declare city j *directly connected* (notice that in this case $\beta_{i'j} = 0$). In the remaining case that $i' \notin I$, let i be any neighbor of i' in graph H such that $i \in I$. Set $\phi(j) = i$ and declare city j *indirectly connected.*

I and ϕ define a primal integral solution: $x_{ij} = 1$ iff $\phi(j) = i$, and $y_i = 1$ iff $i \in I$. The values of α_j and β_{ij} obtained at the end of Phase 1 form a dual feasible solution.

2.3 Analysis

We will show how the dual variables α_j's pay for the primal costs of opening facilities and connecting cities to facilities. Denote by α_j^f and α_j^e the contributions of city j to these two costs respectively; $\alpha_j = \alpha_j^f + \alpha_j^e$. If j is indirectly connected, then $\alpha_j^f = 0$ and $\alpha_j^e = \alpha_j$. If j is directly connected, then the following must hold:

$$\alpha_j = c_{ij} + \beta_{ij},$$

where $i = \phi(j)$. Now, let $\alpha_j^f = \beta_{ij}$ and $\alpha_j^e = c_{ij}$.

Lemma 3 *Let $i \in I$. Then,*

$$\sum_{j:\, \phi(j)=i} \alpha_j^f = f_i.$$

Proof : Since i is temporarily open at the end of Phase 1, it is completely paid for, i.e.,

$$\sum_{j:\, (i,j) \text{ is special}} \beta_{ij} = f_i.$$

The critical observation is that each city j that has contributed to f_i must be directly connected to i. For each such city, $\alpha_j^f = \beta_{ij}$. Any other city j' that is connected to facility i must satisfy $\alpha_{j'}^f = 0$. The lemma follows. □

Corollary 4 $\sum_{i \in I} f_i = \sum_{j \in C} \alpha_j^f$.

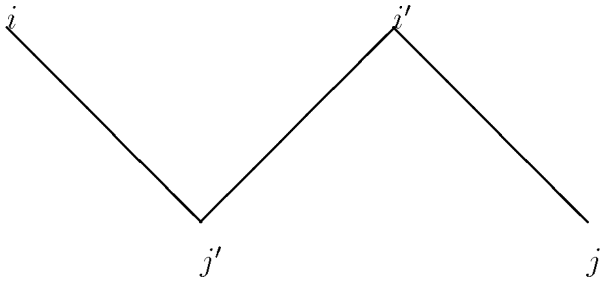
Recall that α_j^f was defined to be 0 for indirectly connected cities. So, only the directly connected cities pay for the cost of opening facilities.

Lemma 5 *For an indirectly connected city j, $c_{ij} \le 3\alpha_j^e$, where $i = \phi(j)$.*

Proof : Let i' be the connecting witness for city j. Since j is indirectly connected to i, (i, i') must be an edge in H. In turn, there must be a city, say j', such that (i, j') and (i', j') are both special edges. Let t_1 and t_2 be the times at which i and i' were declared temporarily open during Phase 1.

Since edge (i', j) is tight, $\alpha_j \ge c_{i'j}$. We will show that $\alpha_j \ge c_{ij'}$ and $\alpha_j \ge c_{i'j'}$. Then, the lemma will follow by using the triangle inequality.

Since edges (i', j') and (i, j') are tight, $\alpha_{j'} \ge c_{ij'}$ and $\alpha_{j'} \ge c_{i'j'}$. Since both these edges are special, they must both have gone tight before either i or i' is declared temporarily open. Consider the time $\min(t_1, t_2)$. Clearly, $\alpha_{j'}$ cannot be growing beyond this time. Therefore, $\alpha_{j'} \le \min(t_1, t_2)$. Finally, since i' is the connecting witness for j, $\alpha_j \ge t_2$. Therefore, $\alpha_j \ge \alpha_{j'}$, and the required inequalities follow. □



Remark 6 If instead of picking all special edges in T, all tight edges were picked, then Lemma 5 does not hold. However, if the facilities in H are ordered in the order in which they were temporarily opened, and I is picked to be the lexicographically first maximal independent set, then Lemma 5 holds again.

Theorem 7 *The primal and dual solutions constructed by the algorithm satisfy:*

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} + 3\sum_{i\in F} f_i y_i \le 3\sum_{j\in C}\alpha_j.$$

Proof : For a directly connected city j, $c_{ij} = \alpha_j^e \le 3\alpha_j^e$, where $\phi(j) = i$. Combining with Lemma 5 we get

$$\sum_{i\in F, j\in C} c_{ij}x_{ij} \le 3\sum_{j\in C}\alpha_j^e.$$

Adding this to the following inequality obtained from Corollary 4 gives the theorem:

$$3\sum_{i\in I} f_i \le 3\sum_{j\in C}\alpha_j^f.$$

□

2.4 Running time

Sort all the edges by increasing cost – this gives the order and the times at which edges go tight. For each facility, i, we maintain the the number of cities that are currently contributing towards it, and the *anticipated time*, t_i, at which it would be completely paid for if no other event happens on the way. Initially all t_i's are infinite, and each facility has 0 cities contributing to it. The t_i's are maintained in a binary heap so we can update each one and find the current minimum in $O(\log n_f)$ time. Two types of events happen, and they lead to the following updates.

- An edge (i, j) goes tight.
  - If facility i is not temporarily open, then it gets one more city contributing towards its cost. The amount contributed towards its cost at the current time can be easily computed. Therefore, the anticipated time for facility i to go be paid for can be recomputed in constant time. The heap can be updated in $O(\log n_f)$ time.
  - If facility i is already temporarily open, city j is declared connected, and α_j is not raised anymore. For each facility i' that was counting j as a contributor, we need to decrease the number of contributors by 1, and recompute the anticipated time at which it gets paid for.
- Facility i is completely paid for. In this event, i will be declared temporarily open, and all cities contributing to i will be declared connected. For each of these city, we will execute the second case of the previous event, i.e., update facilities that they were contributing towards.

The next theorem follows by observing that each edge (i, j) will be considered at most twice. First, when it goes tight. Second, when city j is declared connected. For each consideration of this edge, we will do $O(\log n_f)$ work.

Theorem 8 *Algorithm 1 achieves an approximation factor of 3 for the facility location problem, and has a running time of $O(m \log m)$.*

2.5 Tight example

The following infinite family of examples shows that the analysis of our algorithm is tight: The graph has n cities, $1, 2, \ldots, n$ and two facilities 1 and 2. For each city j, $c_{2j} = 1$. $c_{11} = 1$ and all other c_{ij}'s follows from the tight triangle inequalities. f_1 and f_2 are ϵ and $(n+1)\epsilon$ respectively, for a small number ϵ.

The optimal solution is to open facility 2 and connect all cities to it, at a total cost of $(n+1)\epsilon + n$. Algorithm 1 will however open facility 1 and connect all cities to it, at a total cost of $\epsilon + 1 + 3(n-1)$.

2.6 Extension to arbitrary demands

A small extension to Algorithm 1 enables it to handle the following generalization to arbitrary demands. For each city j, a non-negative demand d_j is specified; any open facility can serve this demand. The cost of serving this demand via facility i is $c_{ij}d_j$.

The only change to IP (1) and LP (2) is that in the objective function, $c_{ij}x_{ij}$ is replaced by $c_{ij}d_jx_{ij}$. This changes the first constraint in the dual (3) to

$$\forall i \in F,\ j \in C:\ \alpha_j - \beta_{ij} \le c_{ij}d_j.$$

The only change to Algorithm 1 is that for each city j, α_j is raised at rate d_j. Notice that because of the change in the first constraint in the dual, edge (i,j) still goes tight at time c_{ij}. However, once (i,j) goes tight, β_{ij} will be increasing at rate d_j, and so facility i may get opened earlier than in the unit demands case.

An easy way to see that this modification works is to reduce to the unit demands case by making d_j copies of city j. The change proposed above to Algorithm 1 is more general, since it works even if d_j is non-integral, and even if it is exponentially large.

3 The metric k-median problem

The k-median problem differs from the facility location problem in two respects: there is no cost for opening facilities, and there is an upper bound, k, on the number of facilities that can be opened; k is not fixed, it is supplied as part of the input. Once again, we will assume that the edge costs satisfy the triangle inequality.

The power of primal-dual algorithms lies in efficiently making "judicious" local improvements. On the other hand, the constraint that at most k facilities be opened is a global constraint – one that is not easy to satisfy through such an algorithm. We observe that the Lagrangian relaxation of the k-median problem is the facility location problem. This enables us to replace this global constraint by a penalty for opening each facility.

Following is an integer program for the k-median problem. The indicator variables y_i and x_{ij} play the same role as in (1).

$$\text{minimize} \quad \sum_{i \in F, j \in C} c_{ij}x_{ij} \qquad (4)$$

$$
\begin{array}{ll}
\text{subject to} & \forall j \in C: \ \sum_{i\in F} x_{ij} \geq 1 \\
& \forall i \in F, j \in C: \ y_i - x_{ij} \geq 0 \\
& \sum_{i\in F} -y_i \geq -k \\
& \forall i \in F, j \in C: \ x_{ij} \in \{0,1\} \\
& \forall i \in F: \ y_i \in \{0,1\}
\end{array}
$$

The LP-relaxation of this program is:

$$
\begin{array}{ll}
\text{minimize} & \sum_{i\in F, j\in C} c_{ij}x_{ij} \\
\text{subject to} & \forall j \in C: \ \sum_{i\in F} x_{ij} \geq 1 \\
& \forall i \in F, j \in C: \ y_i - x_{ij} \geq 0 \\
& \sum_{i\in F} -y_i \geq -k \\
& \forall i \in F, j \in C: \ x_{ij} \geq 0 \\
& \forall i \in F: \ y_i \geq 0
\end{array}
\tag{5}
$$

The dual program is:

$$
\begin{array}{ll}
\text{maximize} & \sum_{j\in C} \alpha_j - zk \\
\text{subject to} & \forall i \in F, j \in C: \ \alpha_j - \beta_{ij} \leq c_{ij} \\
& \forall i \in F: \ \sum_{j\in C} \beta_{ij} \leq z \\
& \forall j \in C: \ \alpha_j \geq 0 \\
& \forall i \in F, j \in C: \ \beta_{ij} \geq 0 \\
& z \geq 0
\end{array}
\tag{6}
$$

3.1 The high level idea

The similarity in the linear programs of the two problems is exploited as follows: Take an instance of the k-median problem, assign a cost of z for opening each facility, and find optimal solutions to LP (2) and LP (3), say $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ respectively. By the strong duality theorem,

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} + \sum_{i\in F} zy_i \ = \ \sum_{j\in C} \alpha_j.$$

Now, suppose that the primal solution $(\boldsymbol{x}, \boldsymbol{y})$ happens to open exactly k facilities (fractionally), i.e., $\sum_i y_i = k$. Then, we claim that $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta}, z)$ are optimal solutions to LP (5) and LP (6) respectively.

Feasibility is easy to check. Optimality follows by substituting $\sum_i y_i = k$ in the above equality, and rearranging terms to show that the primal and dual solutions achieve the same objective function value:

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} \;=\; \sum_{j\in C} \alpha_j - zk.$$

Let's use this idea, together with Algorithm 1 and Theorem 7, to obtain a "good" integral solution to LP (5). Suppose with a cost of z for opening each facility, Algorithm 1 happens to find solutions $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where the primal solution opens exactly k facilities. By Theorem 7,

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} + 3zk \;\leq\; 3\sum_{j\in C} \alpha_j.$$

Now, observe that $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta}, z)$ are primal (integral) and dual feasible solutions to the k-median problem satisfying

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} \;\leq\; 3(\sum_{j\in C} \alpha_j - zk).$$

Therefore, $(\boldsymbol{x}, \boldsymbol{y})$ is a solution to the k-median problem within thrice the optimal.

Notice that proof of factor 3 given above would not work if less than k facilities were opened; if more than k facilities are opened, the solution is infeasible for the k-median problem. The remaining problem is to find a value of z so that *exactly* k facilities are opened. Several ideas are required for this. The first is the following principle from economics: taxation is an effective way of controlling the amount of goods coming across the border – raising tariffs will reduce in-flow and vice versa. In a similar manner, raising z should reduce the number of facilities opened and vice versa.

It is natural now to seek a modification to Algorithm 1 that can find a value of z so that exactly k facilities get opened. This would lead to a factor 3 approximation algorithm. We don't know if this is possible. Instead, we present the following strategy which leads to a factor 6 algorithm. For the rest of the discussion, assume that we never encountered a run of the algorithm which resulted in exactly k facilities being opened.

Clearly, when $z = 0$, the algorithm will open all facilities, and when z is very large, it will open only one facility. The later value of z can be picked to be nc_{max}, where c_{max} is the length of the longest edge. We will conduct a binary search on the interval $[0, nc_{\text{max}}]$ to find z_2 and z_1 for which the algorithm opens $k_2 > k$ and $k_1 < k$ facilities respectively, and furthermore, $z_1 - z_2 \leq (c_{\text{min}}/12n_c^2)$, where c_{min} is the length of the shortest non-zero edge. Let $(\boldsymbol{x}^s, \boldsymbol{y}^s)$ and $(\boldsymbol{x}^l, \boldsymbol{y}^l)$ be the two primal solutions found, with $\sum_{i\in F} y_i^s = k_1$ and $\sum_{i\in F} y_i^l = k_2$ (the superscripts s and l denote "small" and "large" respectively). Further, let (α^s, β^s) and (α^l, β^l) be the corresponding dual solutions found.

Let $(\boldsymbol{x}, \boldsymbol{y}) = a(\boldsymbol{x}^s, \boldsymbol{y}^s) + b(\boldsymbol{x}^l, \boldsymbol{y}^l)$ be a convex combination of these two solutions, with $ak_1 + bk_2 = k$; under these conditions, $a = (k_2 - k)/(k_2 - k_1)$ and $b = (k - k_1)/(k_2 - k_1)$. Since $(\boldsymbol{x}, \boldsymbol{y})$ is a feasible (fractional) solution to the facility location problem that opens exactly k facilities, it is also a feasible (fractional) solution to the k-median problem. In this solution, each city is connected to at most two facilities.

Lemma 9 *The cost of $(\boldsymbol{x}, \boldsymbol{y})$ is within a factor of $(3 + 1/n_c)$ of the cost of an optimal fractional solution to the k-median problem.*

Proof : By Theorem 7 we have:

$$\sum_{i\in F,\, j\in C} c_{ij}x^s_{ij} \;\leq\; 3(\sum_{j\in C}\alpha^s_j - z_1k_1),$$

and

$$\sum_{i\in F,\, j\in C} c_{ij}x^l_{ij} \;\leq\; 3(\sum_{j\in C}\alpha^l_j - z_2k_2).$$

Since $z_1 > z_2$, $(\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l)$ is a feasible dual solution to the facility location problem even if the cost of facilities is z_1. We would like to replace z_2 by z_1 in the second inequality, at the expense of the increased factor. This is achieved using the upper bound on $z_1 - z_2$, and the fact that $\sum_{i\in F,\, j\in C} c_{ij}x^l_{ij} \;\geq\; c_{\min}$. We get:

$$\sum_{i\in F,\, j\in C} c_{ij}x^l_{ij} \;\leq\; (3+\frac{1}{n_c})(\sum_{j\in C}\alpha^l_j - z_1k_2).$$

Multiplying this inequality by b and the first inequality by a and adding, we get

$$\sum_{i\in F,\, j\in C} c_{ij}x_{ij} \;\leq\; (3+\frac{1}{n_c})(\sum_{j\in C}\alpha_j - z_1k),$$

where $\boldsymbol{\alpha} = a\boldsymbol{\alpha}^s + b\boldsymbol{\alpha}^l$. Let $\boldsymbol{\beta} = a\boldsymbol{\beta}^s + b\boldsymbol{\beta}^l$. Observe that $(\boldsymbol{\alpha}, \boldsymbol{\beta}, z_1)$ is a feasible solution to the dual of the k-median problem. The lemma follows. □

In Section 3.2 we give a randomized rounding procedure that obtains an integral solution to the k-median problem from $(\boldsymbol{x}, \boldsymbol{y})$, with an increase in cost by a small factor. In Section 3.3 we derandomize this procedure.

3.2 Randomized rounding

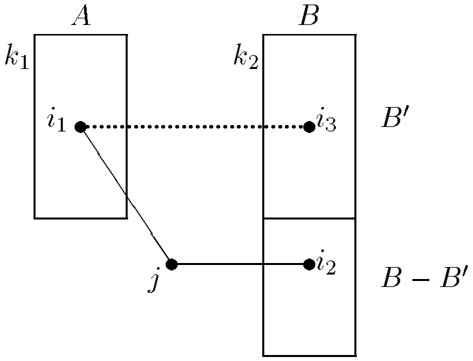
We give a randomized rounding procedure that produces an integral solution to the k-median problem from $(\boldsymbol{x}, \boldsymbol{y})$. In the process, it increases the cost by a multiplicative factor of $1 + \max(a, b)$.

Let A and B be the sets of facilities opened in the two solutions, $|A| = k_1$ and $|B| = k_2$. For each facility in A, find the closest facility in B – these facilities are not required to be distinct. Let $B' \subset B$ be these facilities. If $|B'| < k_1$, arbitrarily include additional facilities from $B - B'$ into B' until $|B'| = k_1$.

With probability a, open all facilities in A, and with probability $b = 1 - a$, open all facilities in B'. In addition, a set of cardinality $k - k_1$ is picked randomly from $B - B'$ and facilities in this set are opened. Notice that each facility in $B - B'$ has a probability of b of being opened. Let I be the set of facilities opened, $|I| = k$.

The function $\phi : C \to I$ is defined as follows. Consider city j, and suppose that it is connected to $i_1 \in A$ and $i_2 \in B$ in the two solutions. If $i_2 \in B'$, then one of i_1 and i_2 is opened by the procedure given above, i_1 with probability a and i_2 with probability b. City j is connected to the open facility.

If $i_2 \in B - B'$ (see figure 3.2), let $i_3 \in B'$ be the facility in B that is closest to i_1. City j is connected to i_2, if it is open. Else, it is connected to i_1, if it is open. If neither i_2 or i_1 is open, then j is connected to i_3.



Denote by $\text{cost}(j)$ the connection cost for city j in the fractional solution $(\boldsymbol{x}, \boldsymbol{y})$; $\text{cost}(j) = ac_{i_1 j} + bc_{i_2 j}$.

Lemma 10 *The expected connection cost for city j in the integral solution, $E[c_{\phi(j)j}] \leq (1+\max(a,b))\text{cost}(j)$. Moreover, $E[c_{\phi(j)j}]$ can be efficiently computed.*

Proof : If $i_2 \in B'$, $E[c_{\phi(j)j}] = ac_{i_1 j} + bc_{i_2 j} = \text{cost}(j)$. Consider the second case, that $i_2 \notin B'$. Now, i_2 is open with probability b. The probability that i_2 is not open and i_1 is open is $(1-b)a = a^2$, and the probability that both i_2 and i_1 are not open is $(1-b)(1-a) = ab$. This gives

$$E[c_{\phi(j)j}] \leq bc_{i_2 j} + a^2 c_{i_1 j} + abc_{i_3 j}.$$

Since i_3 is the facility in B that is closest to i_1, $c_{i_1 i_3} \leq c_{i_1 i_2} \leq c_{i_1 j} + c_{i_2 j}$, where the second inequality follows from the triangle inequality. Again, by the triangle inequality, $c_{i_3 j} \leq c_{i_1 j} + c_{i_1 i_3} \leq 2c_{i_1 j} + c_{i_2 j}$. Therefore,

$$E[c_{\phi(j)j}] \leq bc_{i_2 j} + a^2 c_{i_1 j} + ab(2c_{i_1 j} + c_{i_2 j}).$$

Now, $a^2 c_{i_1 j} + abc_{i_1 j} = ac_{i_1 j}$. Therefore,

$$E[c_{\phi(j)j}] \leq (ac_{i_1 j} + bc_{i_2 j}) + ab(c_{i_1 j} + c_{i_2 j}) \leq (ac_{i_1 j} + bc_{i_2 j})(1 + \max(a,b)).$$

Clearly, in both cases, $E[c_{\phi(j)j}]$ is easy to compute. □

Let $(\boldsymbol{x}^k, \boldsymbol{y}^k)$ denote the integral solution obtained to the k-median problem by this randomized rounding procedure. Then,

Lemma 11 $$E[\sum_{i \in F,\, j \in C} c_{ij} x^k_{ij}] \leq (1 + \max(a,b))(\sum_{i \in F,\, j \in C} c_{ij} x_{ij}),$$
and moreover, the expected cost of the solution found can be computed efficiently.

3.3 Derandomization

Derandomization follows in a straightforward manner using the method of conditional expectation. First, the algorithm opens the set A with probability a, and the set B' with probability $b = 1 - a$. Pick A, and compute the expected value if $k - k_1$ facilities are randomly chosen from $B - B'$. Next, do the same by picking B' instead of A. Choose to open the set that gives the smaller expectation.

Second, the algorithm opens a random subset of $k - k_1$ facilities from $B - B'$. For a choice $D \subset B - B'$, $|D| \leq k - k_1$, denote by $E[D, B - (B' \cup D)]$ the expected cost of the solution if all facilities in D and additionally $k - k_1 - |D|$ facilities are randomly opened from $B - (B' \cup D)$. Since each facility of $B - (B' \cup D)$ is equally likely to be opened, we get

$$E[D, B - (B' \cup D)] = \frac{1}{|B - (B' \cup D)|} \sum_{i \in B - (B' \cup D)} E[D \cup \{i\}, B - (B' \cup D \cup \{i\})].$$

This implies that there is an i such that

$$E[D \cup \{i\}, B - (B' \cup D \cup \{i\})] \leq E[B', B - (B' \cup D)].$$

Choose such an i and replace D by $D \cup \{i\}$. Notice that the computation of $E[D \cup \{i\}, B - (B' \cup D \cup \{i\})]$ can be done as in Lemma 11.

3.4 Running time

It is easy to see that $a \leq 1 - 1/n_c$ (this happens for $k_1 = k - 1$ and $k_2 = n_c$) and $b \leq 1 - 1/k$ (this happens for $k_1 = 1$ and $k_2 = k + 1$). Therefore, $1 + \max(a, b) \leq 2 - 1/n_c$. Altogether, the approximation guarantee is $(2 - 1/n_c)(3 + 1/n_c) < 6$. Using the method of conditional probabilities, this procedure can be derandomized, as in Section 3.3. The binary search will make $O(\log_2(n^3 c_{\max}/c_{\min})) = O(L + \log n)$ probes. The running time for each probe is dominated by the time taken to run Algorithm 1; randomized rounding takes $O(n)$ time and derandomization takes $O(m)$ time. Hence we get

Theorem 12 *The algorithm given above achieves an approximation factor of 6 for the k-median problem, and has a running time of $O(m \log m(L + \log(n)))$.*

The running time of the algorithm can also be made strongly polynomial by standard method of discretizing the costs to integers of magnitude $O(poly)$.

3.5 Tight example

We do not have a tight example of factor 6 for the complete k-median algorithm. However, we give below an infinite family of instances which show that the analysis of the randomized rounding procedure cannot be improved.

The two solutions $(\boldsymbol{x}^s, \boldsymbol{y}^s)$ and $(\boldsymbol{x}^l, \boldsymbol{y}^l)$ open one facility, f_0, and $k + 1$ facilities, $f_1, \ldots, f_{k+1}$ respectively. The distance between f_0 and any other f_i is 1, and that between two facilities in the second set is 2. All n cities are at a distance of 1 from f_0, and at a distance of ϵ from f_{k+1}. The rest of the distances are given by the triangle inequality. The convex combination is constructed with $a = 1/k$ and $b = 1 - 1/k$.

Now, the cost of the convex combination is $an + b\epsilon n$. Suppose the algorithm picks f_1 as the closest neighbor of f_0. Now, the expected cost of the solutions produced by the randomized rounding procedure is $n(b\epsilon + a^2 + ab(2+\epsilon))$. Letting ϵ tend to 0, the cost of the convex combination is essentially na, and that of the rounded solution is $na(1+b)$.

3.6 A Lagrangian relaxation technique for approximation algorithms

Lagrangian relaxation is a fundamental technique in combinatorial optimization. In this section, we will abstract the ideas developed above to give one method of using this technique to derive approximation algorithms. This method does not require the constraints of the problem to be linear, and in fact we will present it in a very general setting.

Let P_1 be the following optimization problem:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}) && (7)\\ &\text{subject to} && \mathbf{P}(\boldsymbol{x}) \\ & && g(\boldsymbol{x}) = k \end{aligned}$$

where f and g are arbitrary real valued functions, $\mathbf{P}$ is an arbitrary predicate, and k is a constant. Let OPT_1 denote the optimal value of this problem, and let $\boldsymbol{a}$ be the value of $\boldsymbol{x}$ at which the optimum is attained.

The Lagrangian relaxation technique consists of relaxing certain constraints by moving them into the objective function, together with associated Lagrange multipliers. We will use this technique to relax the constraint $g(\boldsymbol{x}) = k$. Let z be the Lagrange multiplier. Now, for any value of z,

$$\min_{x:\mathbf{P}(\boldsymbol{x})} f(\boldsymbol{x}) + z(g(\boldsymbol{x}) - k)$$

is a lower bound on OPT_1. To see this notice that substituting $\boldsymbol{x} = \boldsymbol{a}$ in the above expression gives OPT_1. Therefore,

$$\max_{z} \min_{x:\mathbf{P}(\boldsymbol{x})} f(\boldsymbol{x}) + z(g(\boldsymbol{x}) - k) \tag{8}$$

is also a lower bound on OPT_1. Let L be the value of this expression.. Let us rewrite this expression as

$$\max_{z} \left[\left(\min_{x:\mathbf{P}(\boldsymbol{x})} f(\boldsymbol{x}) + zg(\boldsymbol{x}) \right) - zk \right].$$

Now, for each value of z, consider the following optimization problem, which we will call $P_2(z)$:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}) + zg(\boldsymbol{x}) && (9)\\ &\text{subject to} && \mathbf{P}(\boldsymbol{x}) \\ & && && (10) \end{aligned}$$

Let $\mathrm{OPT}_2(z)$ denote the optimum value of this problem. We will show how to derive an approximation algorithm for problem P_1 using an approximation algorithm for problem P_2.

Let $\mathcal{A}$ be an approximation algorithm which, for each z, finds a solution $\boldsymbol{x}$ satisfying

$$f(\boldsymbol{x}) + \alpha z g(\boldsymbol{x}) \leq \alpha \mathrm{OPT}_2(z)$$

for some constant $\alpha \geq 1$. Notice that we have multiplied one term on the left hand side by α as well, and so this is stronger than an α factor approximation algorithm for problem P_2. It must pick a solution so $zg(\boldsymbol{x})$ is completely paid for by $\mathrm{OPT}_2(z)$.

Theorem 13 *Suppose there exists approximation algorithm $\mathcal{A}$ defined above. Suppose further that there is a polynomial time procedure $\mathcal{R}$ that uses $\mathcal{A}$ as a subroutine and finds a value of z for which the solution found by $\mathcal{A}$ satisfies $g(\boldsymbol{x}) = k$. Then, there is an α factor approximation algorithm for problem P_1.*

Proof : By the premise, we can find in polynomial time a value of z and a solution $\boldsymbol{x}$ such that

$$f(\boldsymbol{x}) + \alpha z g(\boldsymbol{x}) \leq \alpha \mathrm{OPT}_2(z) \text{ and } g(\boldsymbol{x}) = k.$$

Substituting, we get

$$f(\boldsymbol{x}) \leq \alpha(\mathrm{OPT}_2(z) - zk).$$

The important observation is that for any value of z,

$$\mathrm{OPT}_2(z) - zk \leq L,$$

since L was defined to be the optimal value of expression (8). Therefore, $f(\boldsymbol{x}) \leq \alpha L$. Since L is a lower bound on OPT_1, we get $f(\boldsymbol{x}) \leq \alpha \mathrm{OPT}_1$. Since $g(\boldsymbol{x}) = k$, $\boldsymbol{x}$ is a feasible solution as well to problem P_1. □

Procedure $\mathcal{R}$ will be problem dependent. For instance, for the k-MST problem, after getting the two solutions, Garg [19] uses additional structural properties to obtain a tree containing exactly k vertices.

For the k-median problem presented above, this involved doing a binary search to find two very close values of z for which g attains values k_1 and k_2 with $k_1 < k < k_2$, taking a convex combination of these solutions and doing a randomized rounding to get an integral solution with a further slight loss in the approximation factor. Other than the last step of randomized rounding, the remaining steps apply to any problem with linear constraints.

For instance, consider the following variant of the k-median problem. Instead of being specified a bound k on the number of facilities to be opened, we are specified the cost of opening each facility and an upper bound allowed for opening facilities. Subject to this constraint, the problem is to minimize the total connection cost. For this problem, we do not know how to carry out the last step of randomized rounding, and leave this as an open problem.

Another interesting phenonmenon, which we call *decoupling*, can happen when we take Lagrangian relaxation. Suppose we have two kind of facilities, hospitals and schools. Suppose the total number of hospitals and schools we can open is at most k (in practice, this might be the result of a budget constraint) so that each city is connected to one hospital and one school. This problem can be thought of as two facility location problems *coupled* with a k-median kind of constraint. If we take its Lagrangian relaxation we get rid of k-median kind of constraint and get two independent instances of the facility location problem, which can be solved separately.

4 A common generalization of the two problems

Consider the uncapacitated facility location problem with the additional constraint that at most k facilities can be opened. This is a common generalization of the two problems solved in this paper – if k is made n_f, we get the first problem and if the facility costs are set to zero, we get the second problem.

The techniques of this paper yield a factor 6 algorithm for this generalization as well. The high level idea is as follows. Using the Lagrangian relaxation technique, we will first remove the restriction that at most k facilities be opened, and instead set the cost of opening each facility i to $f_i + z$. Now, binary search on z will yield two values of z, close to each other, for which Algorithm 1 opens $k_1 < k$ and $k_2 > k$ facilities respectively. An appropriate convex combination of these two solutions gives a fractional solution that opens exactly k facilities, with the additional property that each city is connected to at most two facilities. The cost of this solution is within thrice the cost of an optimal fractional solution. Notice that the randomized rounding procedure it ensures that the expected cost of opening facilities in the rounded solution is the same as the cost of opening facilities in the convex combination. Finally, the derandomization procedure can also be carried out in this setting.

Theorem 14 *There is a factor 6 approximation algorithm for common generalization of uncapacitated facility location and k-median problems in which facilities have costs and at most k of them can be opened.*

5 Dealing with capacities

We consider the following variant of the capacitated metric facility location problem. Each facility can be opened an unbounded number of times; if facility i is opened y_i times, it can serve at most $u_i y_i$ cities. The LP-relaxation of this problem has the following extra constraint:

$$\forall i \in F: \; u_i y_i - \sum_{j \in C} x_{ij} \geq 0.$$

Let the dual variable corresponding to this constraint be γ_i. Then, the dual program is:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{j \in C} \alpha_j \\ \text{subject to} & \forall i \in F, j \in C: \; \alpha_j - \beta_{ij} - \gamma_i \leq c_{ij} \\ & \forall i \in F: \; u_i \gamma_i + \displaystyle\sum_{j \in C} \beta_{ij} \leq f_i \\ & \forall j \in C: \; \alpha_j \geq 0 \\ & \forall i \in F: \; \gamma_i \geq 0 \\ & \forall i \in F, j \in C: \; \beta_{ij} \geq 0 \end{array} \tag{11}$$

For each facility i, let us fix $\gamma_i = \frac{3f_i}{4u_i}$. This step enables us to get rid of the variables γ_i from LP (11), and the resulting linear program is again the dual of an uncapacitated facility location problem. The primal program for this modified dual is:

$$\begin{aligned}\text{minimize} \quad & \sum_{i\in F, j\in C} (c_{ij} + \frac{3f_i}{4u_i})x_{ij} + \sum_{i \in F} \frac{f_i}{4} Y_i \qquad (12)\\ \text{subject to} \quad & \forall j \in C: \sum_{i\in F} x_{ij} \geq 1 \\ & \forall i \in F, j \in C: \ Y_i - x_{ij} \geq 0 \\ & \forall i \in F, j \in C: \ x_{ij} \geq 0 \\ & \forall i \in F: \ Y_i \geq 0\end{aligned}$$

It is easy to see that $c_{ij} + \frac{3f_i}{4u_i}$ still satisfies the triangle inequality. Using Algorithm 1, we can now find a 0/1 integral solution to this LP satisfying

$$\sum_{i\in F, j\in C} (c_{ij} + \frac{3f_i}{4u_i})x_{ij} + 3\sum_{i\in F} \frac{f_i}{4} Y_i \leq 3 \sum_{j\in C} \alpha_j,$$

by Theorem 7. Now, our solution to the capacitated problem is: x_{ij}'s are as in this solution, and $y_i = \lceil \frac{\sum_{j\in C} x_{ij}}{u_i} \rceil$. This gives the following relationship between y_i and Y_i:

$$y_i \leq Y_i + \frac{\sum_{j\in C} x_{ij}}{u_i}.$$

Using this relationship and the above inequality we get:

$$\sum_{i\in F, j\in C} c_{ij}x_{ij} + \frac{3}{4} \sum_{i\in F} f_i y_i \leq 3 \sum_{j\in C} \alpha_j.$$

This implies

$$\sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i \leq 4 \sum_{j\in C} \alpha_j,$$

thereby giving an approximation guarantee of factor 4.

Remark 15 Generalizations of the problems considered in Sections 4 and 5 to the case of arbitrary demands for cities can also be solved within the factors given above, using ideas from Section 2.6.

6 l_2^2 clustering

Our k-median algorithm extends, in a fairly straightforward manner, to obtaining a constant factor algorithm for the problem of l_2^2 clustering. This holds even for the case that the number of clusters and the dimension of the space are arbitrary – a case for which constant factor algorithms were not observed before. However, we note that such a result follows from previous constant factor k-median algorithms as well. The result below should be considered preliminary – the factor obtained is too high. See [16] for a factor 2 algorithm for the case that k is fixed.

Given a set of n points $S = \{v_1, \ldots, v_n\}$ in d-dimensional space and a number k, the problem is to find a minimum cost k-clustering, i.e., to find k points, called *centers*, $f_1, \ldots, f_k$, so as to minimize the sum of squares of distances from each point v_i to its closest center. This naturally defines a partitioning of the n points into k clusters.

Suppose points $v_1, \ldots v_t$ form one of these clusters with center f_1. Define the *centroid* of $v_1, \ldots v_t$ to be $c = (v_1 + \cdots + v_t)/t$. It is well known that

$$\sum_{i=1}^{t} ||v_i - f_1||^2 = \sum_{i=1}^{t} ||v_i - c||^2 + t||f_1 - c||^2,$$

where $||u - v||$ denotes the square of the Euclidean distance between points u and v. So, each center must be the centroid of its cluster. Therefore, this problem can be stated as a k-median problem. The cities are the n given points, $v_1, \ldots, v_n$, and the facilities are the centroid of each subset of points. The cost of connecting a city to a facility is the square of the Euclidean distance between them. Since there are exponentially many facilities, the corresponding LP is exponential sized, and we do not know how to deal efficiently with it.

One way of getting around this difficulty is to choose centers from the given points only. Suppose the closest point from c to $v_1, \ldots v_t$ is v_1, say. Then, using the above equality, we get

$$\sum_{i=1}^{t} ||v_i - v_1||^2 \leq \sum_{i=1}^{t} ||v_i - c||^2 + t||v_1 - c||^2 \leq 2\sum_{i=1}^{t} ||v_i - c||^2.$$

Therefore, the cost of the optimal clustering with the given points as centers is within a factor of 2 of the optimal clustering on centroids. The former problem can be expressed as a polynomial sized k-median LP, and its Lagrangian relaxation as a polynomial sized facility location LP. Our facility location algorithm solves the Lagrangian relaxation with a factor of 9. The reason for the larger factor is that edge costs do not satisfy the triangle inequality. Instead, the statement of Lemma 5 needs to be modified to $c_{ij} \leq 9\alpha_j^e$. For the same reason, the factor for randomized rounding also increases to 6. This gives an overall factor of $2 \times 9 \times 6 = 108$ for l_2^2-clustering.

7 Discussion

A large fraction of the theory of approximation algorithms, as we know it today, is built around linear programming, which provides two main algorithm design techniques: rounding and the primal-dual schema. Both techniques have yielded algorithms with good approximation guarantees, often achieving the integrality gap of the relaxation being used. However, with respect to the running times of the algorithms derived, the two methods differ widely. Rounding resorts to the "big hammer" approach of solving the linear program and therefore leads to inefficient algorithms. On the other hand, the primal-dual schema leaves enough room to exploit the special combinatorial structure of individual problems and has therefore lead to efficient algorithms. Once the algorithm is obtained, typically the scaffolding of linear programming can be completely dispensed with to obtain a purely combinatorial algorithm. As was done in this paper, it seems worthwhile examining various algorithms derived using rounding, to see if efficient combinatorial algorithms achieving the same factors can be obtained.

Besides the objective measure of running time, another aspect in which primal-dual algorithms are superior to rounding based algorithms is the ease with which the core algorithmic idea can be modified, generalized and adapted to special circumstances or variants of the original problem. In this respect, our algorithm has met special success. Besides the various generalizations covered in this paper, the core idea has been used to solve:

- A *fault tolerant* version of the facility location problem, in which we are given a connectivity requirement r_j with each city j, specifying the number of open facilities city j should be connected to [27].
- The *prize collecting* version of both facility location and k-median problems. In this version, we are not required to connect each city to an open facility; however, there is a specified penalty which we have to pay if a city is not connected [10].
- The *outlier* version of the facility location problem, in which we are specified a number T, and are required to connect only T cities to open facilities [10]. [10] reduce this problem to the prize collecting version by using the Lagrangian relaxation technique.
- The *on-line median* problem, in which k is not prespecified and is chosen on-line [33].

It is instructive to compare the current status of primal-dual approximation algorithms with the (mature) status of exact primal-dual algorithms. In the latter setting, only one underlying mechanism is used: iteratively ensuring all complementary slackness conditions. On termination, an optimal (integral) solution to the LP is obtained. In the former setting, we are not seeking an optimal solution to the LP (since the LP may not have any optimal integral solutions), and so there is a need to introduce a further relaxation. Relaxing complementary slackness conditions (which itself can be carried out in more than one way) is only one of the possibilities (see [35] for an alternative mechanism). Another point of difference is that in the exact setting, more sophisticated dual growth algorithms have been given, e.g. [17]. In the approximation setting, other than [35], all primal-dual algorithms use a simple greedy dual growth algorithm.

So far, the primal-dual schema has been used for obtaining good integral solutions to an LP-relaxation. However, it seems powerful enough for the following more general scenario: when the **NP**-hard problem is captured not through an integer program, but in some other manner, and there is an LP that provides a relaxation of the problem. In this setting, the primal-dual schema will try to find solutions that are feasible for the original **NP**-hard problem, and are near-optimal in quality. This open problem was first mentioned in [40].

In Section 3.6 we have stated our Lagrangian relaxation technique in a very general setting in which the constraints of the problem are provided by arbitrary predicates. This includes, for instance, the possibility of non-linear constraints. It will be interesting to see if this technique finds applications in non-linear settings. It will also be interesting to derive an approximation algorithm for a problem in which there are two global constraints, via the Lagrangian relaxation technique, for instance, the outlier k-median problem, in which we are specified the number of facilities that can be opened and the number of cities that need to be connected.

At a more detailed level, the issue of modifying Algorithm 1 so it opens exactly k facilities deserves some thought – this is a possible avenue for improving the factor for the k-median problem. It would be nice to improve the running time of the facility location algorithm in case the metric is specified as the closure of a sparse graph, rather than a complete bipartite graph. Another question is to obtain a non-trivial approximation algorithm for the capacitated facility location problem.

8 Acknowledgement

We wish to thank Pete Veinott for interesting discussions. Thanks to David Williamson sharing his insights about the Lagrangian relaxation technique and for pointing out that Garg's k-MST algorithm can be viewed as the use of the Lagrangian relaxation technique. Thanks also to Moses Charikar for suggesting a simplification to our facility location algorithm.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. on Computing*, 24:440-456, 1995.

[2] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems.*Proc. 30th ACM Symp. on Theory of Computing*, 106-113, 1998.

[3] M. L. Balinksi. On finding integer solutions to linear programs. *Proc. IBM Scientific Computing Symp. on Combinatorial Problems*, 225-248. IBM, 1966.

[4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proc. 37th IEEE Symp. on Foundation of Computer Science*, 184-193, 1996.

[5] R. Bar-Yehuda and S. Even. A linear time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198-203, 1981.

[6] P. S. Bradley, U. M. Fayyad, and O. L. Mangasarian. Mathematical programming for data mining: formulations and Challenges. Microsoft Technical Report, January 1998.

[7] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group Steiner trees and k-median. *Proc. 30th ACM Symp. on Theory of Computing*, 114-123, 1998.

[8] M. Charikar, and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. *Proc. 40th Annual Symposium on Foundations of Computer Science*, 378-388, 1999.

[9] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Proc. 31st ACM Symp. on Theory of Computing*, 1-10, 1999.

[10] M. Charikar, S. Khuller, D.M. Mount, and G. Narshimhan. Algorithms for facility location problems with outliers. em Proc. 12th ACM-SIAM Annual Symposium on Discrete Algorithms, 642-651 2001.

[11] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. Mirchandani and R. Francis, eds., *Discrete Location Theory*, 119-171. John Wiley and Sons, New York, 1990.

[12] F. Chudak. Improved approximation algorithms for uncapacitated facility location. In R.E. Bixby, E.A. Boyd and R.Z. Rios-Mercado, eds., *Integer Programming and Combinatorial Optimization*, Springer LNCS Vol. 1412, 180-194, 1998.

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

[14] F. Chudak and D. Shmoys. Improved approximation algorithms for the capacitated facility location problem. *Proc. 10th ACM-SIAM Symp. on Discrete Algorithms*, S875-S876, 1999.

[15] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Proc. Integer Programming and Combinatorial Optimization*, 1999.

[16] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs. *Proc. 10th Symp. on Discrete Algorithms*, 291-299, 1999.

[17] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards B*, 69B:125-130, 1965.

[18] N. Garg. A 3-approximation for the minimum tree spanning k-vertices. *Proc. 37th IEEE Symp. on Foundation of Computer Science*, 302-309, 1996.

[19] N. Garg. Private communication, 1999.

[20] N. Garg, V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow in multicut in trees. *J. Algorithms*, 18, 3-20, 1997.

[21] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. Williamson. Improved approximation algorithms for network design problems. *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms*, 223-232, 1994.

[22] M. X. Goemans, D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal of Computing*, 24:296-317, 1995.

[23] M.X. Goemans, D.P.Williamson, The Primal-Dual Method for Approximation Algorithms and its Application to Network Design Problems, in *Approximation Algorithms for NP-hard Problems*, D. Hochbaum, Ed., 144-191, PWS Publishing, Boston, MA, 1997.

[24] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Proc. 9th ACM-SIAM Symp. on Discrete Algorithms*, 649-657, 1998.

[25] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Math. Programming*, 22:148-162, 1982.

[26] K. Jain, Ion Măndoiu, Vijay V. Vazirani, and David P. Willamson. A primal-dual schema based approximation algorithm for the element connectivity problem. *Proc. 10th Symp. on Discrete Algorithms*, 484-489, 1999.

[27] K. Jain and V.V. Vazirani. An Approximation Algorithm for the Fault Tolerant Metric Facility Location Problem. *Proc. 3rd APPROX Conference*, Springer-Verlag, LNCS 1671, 2000.

[28] L. Kaufman, M. vanden Eede, and P. Hansen. A plant and warehouse location problem. *Operational Research Quarterly*, 28:547-557, 1977.

[29] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Proc. 9th ACM-SIAM Symp. on Discrete Algorithms*, 1-10, 1998.

[30] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Sci.,* 9:643-666, 1963.

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

[32] J.-H. Lin and J. S. Vitter. ϵ-approximation with minimum packing constraint violation. *Proc. 24th ACM Symp. on Theory of Computing,*771-782, 1992.

[33] R.R. Mettu and C.G. Plaxton. The online median problem. *Proc. 41st IEEE Symp. on Foundation of Computer Science*, 339-348, 2000.

[34] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1990.

[35] S. Rajagopalan and V. V. Vazirani, On the Bidirected Cut Relaxation for the Metric Steiner Tree Problem. *Proc. 10th ACM-SIAM Symp. on Discrete Algorithms*, 742-751, (1999).

[36] D. B. Shmoys, E. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. *Proc. 29th ACM Symp. on Theory of Computing*, 265-274, 1997.

[37] J. F. Stollsteimer. *The effect of technical change and output expansion on the optimum number, size and location of pear marketing facilities in a California pear producing region.* PhD thesis, University of California at Berkeley, Berkeley, California, 1961.

[38] J. F. Stollsteimer. A working model for plant numbers and locations. *J. Farm Econom.*, 45:631-645, 1963.

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

[40] V. V. Vazirani. Primal-Dual Schema Based Approximation Algorithms. *Proc. 1st Annual International Conference, COCOON*, 650-652, 1995.

[41] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:435-454, December 1995.