The Integrality Gap of Capacitated Facility Location

Zoë Abrams[1] Adam Meyerson [2]
Kamesh Munagala[3] Serge Plotkin [4]

December 2002
CMU-CS-02-199

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We consider the facility location problem with *hard non-uniform* capacities. We examine the natural integer programming formulation of this problem. We show that for every constant factor blowup in capacities, the integrality gap of the LP relaxation is a constant. We present a smooth trade-off for the cost versus the blowup in capacities. Non-uniform capacities make the problem significantly more difficult than the case involving uniform capacities, leading us to develop new rounding techniques.

[1]Department of Computer Science, Stanford University. Email: za@cs.stanford.edu

[2]Aladdin Project, Carnegie-Mellon University. Research supported by NSF grant CCR-0122581 and ARO grant DAAG55-98-1-0170. Email: adam@cs.cmu.edu

[3]Department of Computer Science, Stanford University. Supported by ONR N00014-98-1-0589. Email: kamesh@cs.stanford.edu

[4]Department of Computer Science, Stanford University. Supported by ARO Grants DAAG55-98-1-0170 and ONR Grant N00014-98-1-0589. Email: plotkin@cs.stanford.edu

Keywords: algorithms, linear program rounding, approximation, facility location

1 Introduction

We study the facility location problem, where we are given demands in a metric space which have to be satisfied by opening a set of facilities. The decision is where to locate of the facilities, and the objective is to optimize the cost of the facilities we open, and the total distance we have to ship the demand. This problem arises naturally in application such as the placement of warehouses [6] and caches on the web [13, 2, 15]. It also arises as a subroutine in solving several network design problems [8, 11, 9].

We consider the variant of this problem where each facility has a *hard* upper bound on the amount of demand it can serve. This is a natural assumption in many situations. For example, if a facility is a supermarket in a chain, we may not want overcrowding of any particular store. A similar argument can be made for web caches.

Our Results: We present a constant factor approximation for this problem by rounding the linear relaxation of the natural integer program. We show an integrality gap of 9.76 for the linear relaxation, while blowing up the capacities by a factor of 5. In fact, the integrality gap is a constant for *every* constant factor blowup in capacities.

Our algorithm relaxes the capacities by a constant factor. This is unavoidable, as the linear relaxation has an unbounded integrality gap otherwise [18]. We present smooth trade-off results for the cost of the solution versus the slack in capacity constraints.

Previous Results: Pál et al [16] present a 8.53 approximation for capacitated facility location using combinatorial local search, without blowing up the capacities. Since our goal is to work with a linear programming lower bound rather than with the optimal integer solution, our results are not comparable to [16].

Our Techniques: Non-uniform capacities make the application of standard rounding techniques from [14, 18] and related works difficult. There is no obvious *locality* in the fractional solution that can be exploited. Our rounding scheme exploits structure in the fractional solution by rounding in phases, where in each phase, we open one particular facility (chosen according to the natural greedy rule of cheapest cost per unit capacity). The problem with the greedy approach is that if some facility has a huge capacity, it looks very attractive to open, while there may be no way of sending demand comparable to its capacity there. The main idea then is to define *virtual capacity* of a facility in terms of the demand in its neighborhood, and use these capacities in computation instead of the real ones. Since the virtual capacities keep changing as we re-route demand, to make the analysis tractable, we need to define an *auxiliary* linear program which remains feasible as we open facilities.

Related Problems: Uncapacitated facility location is MAX-SNP hard [7], and has several constant factor approximations, using local search [3, 12, 1], linear program rounding [14, 18] and primal-dual approach [10], to mention a few. In addition, there has been work on capacitated facility location where either the capacities are the same on all facilities [12, 5], or where we are allowed multiple copies of a facility at a location [4, 10].

There has also been some previous work on facility location with lower bounds [11, 8, 9]. We consider both upper and lower bounds in our formulation.

2 Preliminaries

We are given a set J of demands in a metric space with distances $d(i, j)$ between points i and j. We are given a set of feasible centers I, each with a capacity (U_i) and a cost (c_i). Our goal is to open some subset of facilities in I and assign the demand points J to open facilities without violating the capacity constraints, such that the sum of the cost of facilities opened plus the total distance we send demand points is minimized.

We will assume that all the demand points carry one unit of demand. If this is not the case, we can apply the Generalized Assignment rounding scheme from [17] at the end of the algorithm, and lose a small additional constant. The details can be found in [18], and are therefore omitted.

3 Integer Program Formulation

We can write the following integer program for this problem. Here, y_i denotes whether facility i is open, and f_{ij} denotes the assignment of demand j to facility i.

$$\text{Minimize} \sum_{i \in I} c_i y_i + \sum_{j \in J} \sum_{i \in I} f_{ij} d(i,j)$$

$$\begin{array}{rcll} \sum_{i \in I} f_{ij} & = & 1 & \forall j \in J \\ \sum_{j \in J} f_{ij} & \leq & U_i \cdot y_i & \forall i \in I \\ f_{ij} & \leq & y_i & \forall i,j \\ f_{ij}, y_i & \in & \{0,1\} & \forall i,j \end{array}$$

Scaling: Once we have solved the linear relaxation of the integer program, we can compute for each demand point j a value $D(j) = \sum_{i \in I} f_{ij} d(i,j)$. This is the average distance this demand travels. We will perform filtering as in [14], setting $f_{ij} = 0$ if $d(i,j) > \frac{4}{3} D(j)$. This removes distances which are much further than the mean. We scale up the other f_{ij} by a factor of 4 so that the sum is still unity. This forces us to increase the values of all capacities by 4.
Auxiliary Linear Program: Scaling enables us to create a new linear program for the problem, which has a feasible solution with revised capacities for facilities being $U_i' = 4U_i$. The cost of this feasible solution is at most the original cost because the distance demand is sent decreases. The integer program can be written as follows:

$$\text{Minimize} \sum_{i \in I} c_i \cdot y_i + \sum_{j \in J} a_j \cdot D(j)$$

$$\begin{array}{rcll} \sum_{i \in I} f_{ij} & = & a_j & \forall j \in J \\ \sum_{j \in J} f_{ij} & \leq & U_i' \cdot y_i & \forall i \in I \\ f_{ij} & \leq & y_i & \forall i,j \\ d(i,j) > 2D(j) & \Rightarrow & f_{ij} = 0 & \forall i,j \\ f_{ij}, y_i & \in & [0,1] & \forall i,j \end{array}$$

We do not solve this integer program. We simply observe that the fractional solution to the original program is feasible for the linear relaxation of this program. We will use this new program only to maintain feasibility at all times. Initially we have $a_j = 1$ for all $j \in J$, but these values will decrease as the algorithm progresses.

4 Rounding the Fractional Solution

The rounding proceeds in phases, in each of which we concentrate the fractional values y_i to which the facilities are open. We repeat as long as there is some fractional y_i.

Defining Neighborhoods: For each facility $i \in I$ we compute $AVG(i) = (\sum_{j\in J} f_{ij} D(j))/(\sum_{j\in J} f_{ij})$. This is the weighted average of $D(j)$ values seen at facility i, or in other words, it is the average distance traveled by demands in reaching this facility. We observe that the total cost of the linear program is equal to $\sum_{i\in I}(c_i y_i + AVG(i)\sum_{j\in J} f_{ij})$. In the rounding scheme below, we will send demand j to facility i iff $d(i,j) \le c \cdot AVG(i)$, where c is a constant which we will determine later. We define the neighborhood $B(i)$ as $B(i) = \{j : d(i,j) \le \frac{4}{3}D(j) \le \frac{8}{3}AVG(i)\}$.

The basic rounding idea is to open facilities in order of their cost per unit capacity. We have to be careful, however, in how we define the capacity. A facility could have a huge capacity, but there could be very little demand in its neighbourhood to send there. We therefore need a notion of virtual capacity, which is the smaller of the real capacity and the total demand in the neighbourhood.

Virtual Capacities: Set $U_i' = \min(U_i', \sum_{j\in B(i)} 2)$. As mentioned above, this is just setting the capacity to the minimum of the real capacity and the total demand in the neighbourhood which can be sent to it.

Cheapest Facility: We select $i^* \in I$ which minimizes $r(i^*) = AVG(i^*) + (c_{i^*}/U'_{i^*})$. The value $r(i)$ is the cost per unit demand we pay (in terms of both the facility cost and routing cost) at this facility location.

Proportional Reroute: For all facilities i' s.t. $d(i^*, i') \le \frac{16}{3}AVG(i^*)$, we close i' and send demand routed to it to i^*. We do this until i^* reaches capacity, at which point we declare it open. The rerouting proceeds *proportionally*, that is, we remove the same ϵ fraction of flow from all demands sent to i' and send that fraction to i. We update y_i' appropriately, as $y_{i'} \leftarrow y_{i'} - \epsilon$. Note that decreasing the value $y_{i'}$ still keeps the linear program feasible.

Updating Assignments: Forall $j \in J$, we set $a_j = a_j - f_{i^*j}$. This represents the fraction of demand j which is still assigned to fractionally open facilities.

Removing Small Assignments: For some $\beta > 0$ to be chosen later, if $a_j < \beta$, we eliminate j from the demand set, and increase its current assignments to $\frac{1}{1-\beta}$.

Cleanup: If for any i, $\sum_{j\in J} f_{ij} = 0$, set $y_i = 0$. We remove i^* from the set of feasible centers. We will open i^* in the solution.

This completes one phase of the algorithm. We update the virtual capacities for facilities which are still fractionally open[1] and proceed.

5 Analysis

We will need to prove that the second linear program remains feasible at the end of every phase, and that the total cost is diminishing. When the algorithm completes and we open all the selected i^*, we need to make sure that the total cost is not too large.

Lemma 1 *The second linear program is feasible at the end of each iteration.*

[1] These values could decrease due to assignment of demands to other facilities.

Proof: The only place where we might become infeasible is in the modification of the capacity values. Suppose that the capacity of i was reduced in step one. We observe that $U'_i = \sum_{j \in B(i)} 2$. For any j with $f_{ij} > 0$ we must have $d(i,j) \leq \frac{4}{3}D(j)$ because of the filtering step. Because of the definition of $AVG(i)$, we know that at least half the demand flowing to i in the fractional solution has $D(j) \leq 2AVG(i)$ (applying Markov's inequality). So $2\sum_{j \in B(i)} f_{ij} \geq \sum_{j \in J} f_{ij}$. Noticing that $f_{ij} \leq y_i$ we can deduce that $U'_i \geq 2\sum_{j \in B(i)} f_{ij}/y_i \geq \sum_{j \in J} f_{ij}/y_i$ from which it follows that $\sum_{j \in J} f_{ij} \leq U'_i y_i$ and the equation remains satisfied.

We continue by rerouting flow to i^* and removing i^*, and the linear program is feasible on the remaining demands and facilities. ■

Lemma 2 *After merging nearby facilities into i^*, we have $\sum_j f_{i^*j} \geq \frac{\beta}{2}U'_{i^*}$.*

Proof: Consider some demand $j \in B(i^*)$. Suppose this demand is also sent fractionally to i'. We must have $d(i',j) \leq \frac{4}{3}D(j) \leq \frac{8}{3}AVG(i^*)$ because of filtering. It follows that $d(i',i^*) \leq \frac{16}{3}AVG(i^*)$ because of the triangle inequality. So after merging, either we have i^* full or else this demand point j has $f_{i^*j} = a_j$. Let us suppose i^* isn't full. It follows that the total demand accumulated at i^* must be at least $\sum_{j \in B(i^*)} a_j \geq \frac{\beta}{2}U'_{i^*}$ and the lemma follows. ■

We now look at the auxiliary linear program at the end of the phase, and try to compute its cost. We claim that the cost reduces. Note that we have eliminated i^* from the problem.

Lemma 3 *At the moment that i^* is removed from consideration, the total cost is reduced by at least $\sum_{j \in J} f_{i^*j}D(j) + \frac{\beta c_{i^*}}{2}$.*

Proof: The $c_{i^*}y_{i^*}$ part comes from no longer paying for i^* (which is no longer a feasible center). The value of a_j has also been reduced for each j, and this gives rise to the first part of the reduction in cost. ■

Lemma 4 *The process of absorbing facilities into i^* and then eliminating i^* reduces the cost by at least $(\sum_{j \in J} f_{i^*j})(AVG(i^*) + \frac{c_{i^*}}{U'_{i^*}})$.*

Proof: Suppose that during the absorbing process, i^* absorbs $\Delta(i)$ demand from facility i. This means that the cost due to i has been reduced by at least $\Delta(i)(AVG(i) + \frac{c_i}{U'_i})$. So the total reduction in cost may be expressed by $\sum_i \Delta(i)(AVG(i) + \frac{c_i}{U'_i})$. We selected i^* to minimize $AVG(i) + \frac{c_i}{U'_i}$, so it follows that the total reduction in cost is at least $\sum_i \Delta(i)(AVG(i^*) + \frac{c_{i^*}}{U'_{i^*}})$. Of course, the sum of $\Delta(i)$ over all i (including i^* itself) is exactly the total demand at i^* when it is removed from the set of feasible centers. The lemma follows. ■

Lemma 5 *When the algorithm terminates, the cost paid to open i^* and send points there is at most $c_{i^*} + \sum_{j \in J} f_{i^*j}(\frac{4}{3}D(j) + \frac{16}{3}AVG(i^*))$.*

Proof: Consider a point j which is sent to i^* in our solution. What is the maximum possible value of $d(i^*,j)$? The filtered LP solution sent j to some facility i', and this facility i' was absorbed by i^*. Filtering guarantees that $d(i',j) \leq \frac{4}{3}D(j)$ and the absorbing process guarantees that $d(i',i^*) \leq \frac{16}{3}AVG(i^*)$, so we conclude that $d(i^*,j) \leq \frac{4}{3}D(j) + \frac{16}{3}AVG(i^*)$. Summing over j and adding the facility cost of i^* gives the result claimed. ■

Theorem 1 *When the algorithm terminates, we can open facilities and send the points to facilities without exceeding capacities by more than a factor of* 5.28*, and our total cost does not exceed* 8.8 *times the original LP cost.*

Proof: When the algorithm terminates, we can multiply all f_{ij} for opened facilities by $\frac{1}{1-\beta}$ and send every point somewhere; for $\beta = 0.24$, this exceeds capacities by at worst a factor of 1.32, multiplied by the factor of 4 created in the filtering step.

Each pass through the loop reduces the LP cost by some value R. In the end we will have to pay some A for the facility opened (and to send points there). The lemmas above bound the values of A and R.

In particular we observe that $A \leq \frac{4}{3}R + \frac{16}{3}R = \frac{20R}{3}$. So in reducing the LP cost to zero, we pay at most $\frac{20}{3}$ times the LP cost. This is then increased by 1.32 in the rounding of a_j values, yielding the theorem claimed. ■

5.1 Cost Versus Capacity Tradeoff

There is a tradeoff between the approximation on cost and the blowup in capacities. Consider variables such that α is our filtering factor on demand points (*i.e.*, $d(i,j) \leq \alpha D(j)$), β is the lower bound on a_j (*i.e.*, $\beta \leq a_j \leq 1$), and γ is our filtering factor on facilities (*i.e.*, $D(j) \leq \gamma AVG(i)$). We make the following adjustments to the algorithm:

1. For all $i \in I$ set $U'_i = \min(U'_i, \frac{\gamma}{\gamma-1} \sum_{j: d(i,j) \leq \alpha D(j) \leq \alpha\gamma AVG(i)} 1)$.

2. Consider closing centers i' which have $d(i^*, i') \leq 2\alpha\gamma AVG(i^*)$ one by one.

Theorem 2 *The integrality gap of the linear program is bounded by* $\frac{1}{1-\beta}\max(2\alpha\gamma + \alpha, \frac{\gamma}{\beta(\gamma-1)})$*, while blowing up the capacities by a factor of at most* $\frac{\alpha}{(1-\beta)(\alpha-1)}$.

References

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for the k-median and facility location problems. *Proc. 33rd ACM STOC*, 2001.

[2] I. Baev and R. Rajaraman. Approximation algorithms for data placement in arbitrary networks. *Proceedings of 12th ACM-SIAM SODA*, 2001.

[3] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. *Proceedings of 40th IEEE FOCS*, 1999.

[4] F. Chudak and D. Shmoys. Improved approximation algorithms for the capacitated facility location problem. *Proceedings of 10th ACM-SIAM SODA*, 1999.

[5] F. Chudak and D. Williamson. Improved approximation algorithms for capacitated facility location problems. *Proceedings of 7th IPCO Conference*, 1999.

[6] G. Cornuejols, M. Fisher, and G. Nemhauser. On the uncapacitated location problem. *Annals of Discrete Math.*, 1:163–178, 1977.

[7] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Proceedings of 9th ACM-SIAM SODA*, 1998.

[8] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. *Proceedings of 41st IEEE FOCS*, 2000.

[9] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flow. *Proc. 33rd ACM STOC*, 2001.

[10] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. *Proceedings of 40th IEEE FOCS*, 1999.

[11] D. Karger and M. Minkoff. Building steiner trees with incomplete global knowledge. *Proceedings of 41st IEEE FOCS*, 2000.

[12] M. Korupolu, G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Proceedings of 9th ACM-SIAM SODA*, 1998.

[13] B. Li, M. Golin, G. Italiano, X. Deng, and K. Sohraby. On the optimal placement of web proxies in the internet. *Proceedings of INFOCOM*, 1999.

[14] J.-H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violations. *Proceedings of 24th ACM STOC*, 1992.

[15] A. Meyerson, K. Munagala, and S. Plotkin. Web caching using access statistics. *Proceedings of 12th ACM-SIAM SODA*, 2001.

[16] M. Pál, É. Tardos, and T. Wexler. Facility location with non-uniform hard capacities. *Proceedings of 42nd IEEE FOCS*, 2001.

[17] D. B. Shmoys and É. Tardos. Scheduling unrelated machines with costs. *Proceedings of 4th ACM-SIAM SODA*, pages 448–454, 1993.

[18] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. *Proceedings of 29th ACM STOC*, 1997.