On the Integrality Gap of Capacitated Facility Location

Zoë Abrams[*] Adam Meyerson[†] Kamesh Munagala[‡] Serge Plotkin [§]

Abstract

We consider the facility location problem with hard non-uniform upper and lower bounds on the amount of demand that can be routed to any facility. We examine the natural integer programming formulation of this problem. First, for the version with just the upper bounds (*i.e.*, with the lower bounds being zero), 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. Secondly, we show how to incorporate lower bounds into *any* approximation algorithm for the version with just the upper bounds. Non-uniform capacities make the problem significantly more difficult than the case involving uniform capacities.

[*]Department of Computer Science, Stanford CA 94305. Email: `za@cs.stanford.edu`.

[†]Department of Computer Science, Stanford University CA 94305. Supported by ARO DAAG-55-97-1-0221. Email: `awm@cs.stanford.edu`.

[‡]Department of Computer Science, Stanford University CA 94305. Supported by ONR N00014-98-1-0589. Email: `kamesh@cs.stanford.edu`.

[§]Supported by ARO Grants DAAG55-98-1-0170 and ONR Grant N00014-98-1-0589. Department of Computer Science, Stanford University CA 94305. Email: `plotkin@cs.stanford.edu`.

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 *hard* upper and lower bounds 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. On the other hand, the store is profitable only if there is a certain minimum number of customers. This implies a lower bound on the amount of demand any open facility needs to serve. A similar argument can be made about web caches – it does not make sense to administer and maintain a cache which receives few hits, while on the other hand, we do not want congestion at a cache.
Our Results: We present a constant factor approximation for this problem by rounding the linear relaxation of the natural integer program. Our results are two-fold:

1. We first begin by ignoring the lower bounds, and focussing on the IP for the version with just upper bounds. 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.

2. We then show how to incorporate lower bounds at a small loss in the approximation ratio into *any* algorithm with upper bounds.

Our algorithms relax the upper and lower bounds by constant factors. 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 facility location with only the upper bound constraints using local search, without blowing up the capacities. We use an entirely different linear programming based approach. Though our approach needs to blow up the capacities by a constant factor, it has the advantage of showing a constant integrality gap. We feel this will have applications in cases where this algorithm is required as a subroutine in some bigger network design problem [2, 15, 9, 8, 11]. In the analysis of algorithms for these problems, it is often simple to construct fractional solutions to the sub-problems from the optimal solution, and therefore, a good integrality gap is essential.
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 a natural greedy rule). We simply re-adjust the primal variables to maintain a feasible solution to an auxiliary linear program, which is based on setting capacities to reflect existing neighboring demand. We then carefully choose an accounting scheme based on the adjustment to the variables we made in this phase.
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 Non-uniform Capacitated Facility Location

In this section, we present an algorithm for the version of the problem without the lower bounds. We will incorporate lower bounds in the next section.

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. This assumption can be removed by applying the Generalized Assignment rounding scheme from [17] at the end of the algorithm. The details can be found in [18], and are therefore omitted.

2.1 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.

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

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

Once we have solved the linear relaxation of the integer program, we can compute for each demand point j a value $D(j) = \sum_{i \in I} f_{ij} d(i,j)$. We will perform filtering as in [14], setting $f_{ij} = 0$ if $d(i,j) > \frac{4}{3} D(j)$ and subsequently multiplying the values of all capacities and allocations by 4. This enables us to create a new linear program for the problem, which has a feasible solution with all capacities increased by a factor of 4. Revised capacities for facilities are $U_i' = 4U_i$. 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 change as the algorithm progresses, always maintaining that $\beta \leq a_j \leq 1$. The exact value of β will be chosen later.

2.2 Rounding the Fractional Solution

For each $i \in I$ we can 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. We observe that the total cost of the linear program is equal to $\sum_{i \in I}(c_i y_i + AVG(i) \sum_{j \in J} f_{ij})$. One phase of rounding scheme is summarized in Algorithm 1. We repeat as long as there is some fractional y_i.

Algorithm 1 One Phase of the Rounding Scheme

for all $i \in I$ **do**
    Set $U_i' = \min(U_i', \sum_{j: d(i,j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i)} 2)$.
end for
Select $i^* \in I$ which minimizes the value of $AVG(i^*) + (c_{i^*}/U_{i^*}')$.
for all i' s.t. $d(i^*, i') \leq \frac{16}{3} AVG(i^*)$ **do**
    while $\sum_j f_{i'j} > 0$ and $\sum_j f_{i^*j} < U_{i^*}$ **do**
        Close i'.
        Reroute the demand from i' to i^* absorbing every flow at an equal fraction, updating the $y_{i'}$ appropriately.
    end while
end for
for all j **do**
    $a_j = a_j - f_{i^*j}$
    If $a_j < \beta$, eliminate j from the demand set.
end for
If for any i, $\sum_{j \in J} f_{ij} = 0$, set $y_i = 0$.
Remove i^* from the set of feasible centers. We will open i^* in the solution.

We will need to prove that the second linear program remains feasible and 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 2.1. *The second linear program is feasible at the end of each iteration.*

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 $U_i' = \sum_{j: d(i,j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i)} 2$. 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 $2 \sum_{j: d(i,j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i)} f_{ij} \geq \sum_{j \in J} f_{ij}$. Noticing that $f_{ij} \leq y_i$ we can deduce that $U_i' \geq 2 \sum_{j: d(i,j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i)} f_{ij}/y_i \geq \sum_{j \in J} f_{ij}/y_i$ 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 (possibly illegally) to i^* and removing i^*, and the linear program is feasible on the remaining demands and facilities. □

Lemma 2.2. *After merging nearby facilities into i^*, we have $\sum_j f_{i^*j} \geq \frac{\beta}{2} U_{i^*}'$.*

Proof. Consider some demand j which has $d(i^*, j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i^*)$. 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 $\sum_{j: d(i^*, j) \leq \frac{4}{3} D(j) \leq \frac{8}{3} AVG(i^*)} a_j \geq \frac{\beta}{2} U_{i^*}'$ and the lemma follows. □

Lemma 2.3. *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 2.4. *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 2.5. *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 2.6. *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. □

2.3 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 2.7. *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 $\frac{\alpha}{(1-\beta)(\alpha-1)}$.*

3 Incorporating Lower Bounds

In this section, we assume a facility i has a lower bound L_i on the amount of demand it needs to satisfy. This is in addition to the conditions from the previous section although the particular algorithm used is irrelevant. Our rounding scheme is similar to the load balanced facility location rounding scheme from [11, 8], but the non-uniformity again creates complications for which we need new ideas.

As in [11, 8], we solve the capacitated version of facility location, assigning each facility a new cost $f_i' = f_i + S_i$ where S_i is the travel cost of sending L_i closest demand to facility i. This at most doubles the cost of our solution. We now modify the solution so that every open facility i serves at least $\frac{L_i}{4}$ demand.

3.1 Algorithm

The algorithm is described in Algorithm 2.

Algorithm 2 Algorithm to satisfy lower bounds

1: Select a facility i serving less than $\frac{L_i}{4}$ amount of demand.
2: We will compute a set of close facilities M. Consider demand in S_i that is not served by i in the current solution. This quantity is at least $\frac{3L_i}{4}$. Take the closest demand points constituting half this demand. Call this set of demand points P. Construct M to include the facilities other than i that serve these demand points in the current solution. Let D_i be the amount of demand currently being served by the facility i.
3: Compute $T = \sum_{i' \in M} U_{i'} - D_{i'}$.
4: If $T \geq D_i$, close i and reroute its demand to the facilities $i' \in M$ saturating closer facilities first.
5: Else, if $T < D_i$, reroute demand from $i' \in M$ to i in order of the closest i' facilities without decreasing demand below $\frac{L_{i'}}{4}$ for any facility i'.
6: If there exists a facility that serves less than $\frac{L_i}{4}$, goto Step 1.

3.2 Analysis

We will need to prove that either step 4 or step 5 is feasible at each iteration and that the cost of the solution does not increase.

Lemma 3.1. *Step 4 does not increase the cost of the solution.*

Proof. Since we are paying f_i' for facility i, we are paying S_i more than is actually being used in opening the facility. We can consider this S_i to be the amount available to be spent on additional routing to meet the lower bound for i.

The additional cost of routing at most $\frac{L_i}{4}$ amount of demand from i to points $j \in P$ cannot exceed $\frac{S_i}{2}$. This is because P is the closer routing destination for half of $\frac{3L_i}{4}$ demand and therefore routing the other $\frac{2}{8}$ must be at least the cost of routing the same amount from P. The distance $\forall j \in P, \forall i' \in M$ $d(i', j)$ must be at most $d(i, j)$. If $d(i', j) > d(i, j)$ demand at j would have been routed to i instead of i', since the only constraint causing demand to travel further than its closest open facility is the capacity constraint which is not an impediment in this situation since the lower bound at i is not filled. Therefore, the cost of traveling from $j \in P$ to $i' \in M$ cannot exceed $\frac{S_i}{2}$ and the total routing cost for closing facility i is less than or equal to S_i, which is the cost available for rerouting. $\square$

Lemma 3.2. *Step 5 does not increase the cost of the solution.*

Proof. We showed in the above proof that the cost of routing $\frac{L_i}{4}$ demand from i through $j \in P$ to $i' \in M$ is at most S_i. Routing the same amount in the opposite direction must also be at most S_i. □

Lemma 3.3. *It is possible to do one or the other option in Step 2 while mainting a feasible solution that remains within the specified bounds of each facility.*

Proof. We will prove by contradiction. The amount of additional demand that facilities in M can handle without exceeding the upper bound of any facility is $\sum_{i' \in M} U_{i'} - D_{i'}$. The total amount of demand that can be removed from facilities in M without decreasing demand below $\frac{1}{4}$ the lower bound of any facility is $\sum_{i' \in M} D_{i'} - \frac{L_{i'}}{4}$.

If Lemma 3.3 is false, then we cannot route D_i into facilities in M without overflowing the upper bounds and we cannot route $\frac{L_i}{4} - D_i$ away without decreasing demand below some i' lower bound yielding the following equations:

$$\sum_{i' \in M} U_{i'} - D_{i'} < D_i$$

$$\sum_{i' \in M} D_{i'} - \frac{U_{i'}}{4} < \sum_{i' \in M} D_{i'} - \frac{L_{i'}}{4} < \frac{L_i}{4} - D_i$$

Adding them together, $\sum_{i' \in M} U_{i'} < \frac{L_i}{3}$. But since the facilities in M serve at least $\frac{2L_i}{8}$ by definition, we know $\sum_{i' \in M} U_{i'} \geq \frac{2L_i}{8}$ and we have reached a contradiction. □

Theorem 3.4. *The rounding scheme described in Algorithm 2 preserves the cost of the solution, while ensuring that open facility i receives at least $\frac{L_i}{4}$ demand.*

3.3 Performance Gains With Additional Assumptions

In some cases we can assume that our capacities are at least a constant factor above our lower bounds. In these situations, we can improve the slack in the lower bounds.

Replacing the assumed fraction $\frac{1}{4}$ in Section 3 with the variable β and assuming that for every facility, the upper bound exceeds the lower bound by at least a factor α, we obtain the following equation:

$$\frac{\beta^2}{\alpha} - (3 + \frac{1}{\alpha})\beta + 1 = 0$$

In order to reach the contradiction in Lemma 3.3, β must be smaller than the solution to the above equation. For $\alpha = 1$, we have $\beta = 2 - \sqrt{3}$.

Solving for general problem with both lower and upper bounds, the previous section guarantees the capacities are approximated by a factor of 5, so that $\alpha = 5$. We can now guarantee $\beta = 0.32$. As α approaches infinity, β approaches $\frac{1}{3}$. This is consistent with the $\frac{1}{3}$ approximation on the lower bounds in [11, 8], which solves the problem without the upper bounds.

4 Conclusions

An immediate open question is to try and extend the local search framework in [16] to work for the general problem with lower bounds. The difficulty is that the local swap operation now involves solving the maximum constrained partition problem, which is very hard to approximate [19] unless we are willing to relax the capacity constraints by a constant factor.

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.

[19] D. Zuckerman. NP-complete problems have a version that's hard to approximate. *Proc. Eight Ann. Structure in Complexity Theory Conf., IEEE Computer Society*, pages 305–312, 1993.