Improved Approximation for Universal Facility Location

Naveen Garg * Rohit Khandekar * Vinayaka Pandit †

Abstract

The Universal Facility Location problem (UniFL) is a generalized formulation which contains several variants of facility location including capacitated facility location (1-CFL) as its special cases. We present a $6+\epsilon$ approximation for the UniFL problem, thus improving the $8+\epsilon$ approximation given by Mahdian and Pal. Our result bridges the existing gap between the UniFL problem and the 1-CFL problem.

1 Preliminaries

The Universal Facility Location (UniFL) problem was introduced by Hajiaghayi et al. [1]. In the UniFL problem, we are given a set of facilities F, a set of clients C, and a distance metric $\{c_{ij} : i, j \in F \cup C\}$. Each client $j \in C$ is associated with an integer demand $d_j \geq 0$. For each $i \in F$, the *facility cost* of i is given by a non-decreasing, continuous function $f_i(u_i)$ of the total capacity u_i installed at i. The goal is to install capacities u_i at every facility $i \in F$ and assign all the demands to the facilities such that each facility i serves at most u_i units of the demand. The demand of a client can be split among multiple facilities. The *service cost* of assigning a unit demand of a client $j \in C$ to a facility $i \in F$ is given by c_{ji}. The objective is to minimize the sum of the total facility cost and the total service cost.

Mahdian and Pál [2] give $8+\epsilon$ approximation for UniFL using a local search algorithm. Zhang et al. [3] improve their analysis to give $7+\epsilon$ approximation. Zhang et al. [4] also give $6+\epsilon$ approximation for the capacitated facility location problem (1-CFL). We generalize their result to UniFL with a new local search operation called `Double_Pivot`. We give a local search algorithm which outputs a local optimum with cost at most 6 times the cost of the optimum. This yields $6+\epsilon$ approximation algorithm. Our local search algorithm starts with an arbitrary solution and outputs a local optimum solution with respect to the following operations. The first two operations were introduced by Mahdian and Pál [2].

1. `add`(s, δ): Increase the capacity at $s \in F$ by $\delta \geq 0$. The change in cost can be computed exactly and the operation is performed if it saves cost.
2. `Single_Pivot`(s, Δ): The vector $\Delta \in \Re^{|F|}$ indicates the increase in the installed capacity at each facility. A facility $i \in F$ is said to *shrink* if $\Delta_i < 0$, and *grow* if $\Delta_i > 0$. Each shrinking facility i sends $|\Delta_i|$ units of its demand to the pivot $s \in F$. Each growing facility i receives Δ_i units of demand from s. For a valid operation, we assume $\sum_{i\in F} \Delta_i = 0$. The increase in cost of the solution is at most $\sum_{i\in F}(f_i(u_i + \Delta_i) - f_i(u_i) + c_{si}|\Delta_i|)$. We perform the operation if the upper bound is negative.
3. `Double_Pivot`$(s_1, s_2, \Delta^1, \Delta^2)$: It is similar to performing two `Single_Pivot` operations at a time. The vector $\Delta^1 \in \Re^{|F|}$ specifies the rerouting of demand through s_1 and $\Delta^2 \in \Re^{|F|}$ specifies the rerouting of demand through s_2. The overall increase in the capacities is given by $\Delta^1 + \Delta^2$. We assume $\sum_{i\in F} \Delta_i^1 = \sum_{i \in F} \Delta_i^2 = 0$. The increase in cost of the solution is now at most $\sum_{i\in F}(f_i(u_i + \Delta_i^1 + \Delta_i^2) - f_i(u_i) + c_{s_1 i}|\Delta_i^1| + c_{s_2 i}|\Delta_i^2|)$. We perform the operation if the upper bound is negative.

Mahdian and Pál [2] show how to use dynamic programming to find a `Single_Pivot` operation with the minimum upper bound on the increase in the cost. The `Double_Pivot` can be implemented using similar ideas.

2 Analysis

Let S and S^* denote a local optimum solution and a global optimum solution respectively. Let u_i and u_i^* denote the capacities installed at $i \in F$ in S and S^* respectively. Let $C_s(S)$ and $C_f(S)$ denote the service and the facility costs of S, and $C(S) = C_s(S) + C_f(S)$ denote the total cost of S. Similarly define $C_s(S^*), C_f(S^*)$, and $C(S^*)$. Using the fact that S is locally optimum with respect to the `add` operation, Mahdian and Pál [2] proved the following upper bound on the service cost of S.

LEMMA 2.1. $C_s(S) \leq C_f(S^*) + C_s(S^*)$.

The rest of the paper is devoted to bounding $C_f(S)$. Let the facilities $F_+ = \{i \in F \mid u_i - u_i^* > 0\}$ be called *sources* and the facilities $F_- = \{i \in F \mid u_i - u_i^* < 0\}$ be called *sinks*. Consider the transshipment problem

*Dept. CSE, Indian Institute of Technology, Delhi.
†IBM India Research Laboratory, New Delhi.

in which each $i \in F_+$ sources a ow of $u_i - u_i^*$ and each $i \in F_-$ sinks a ow of $|u_i - u_i^*|$. Let the cost of routing a unit demand from a facility s to a facility t be c_{st}. Mahdian and Pál [2] proved the minimum cost of the transshipment to be at most $C_s(S) + C_s(S^*)$. Furthermore, the support graph of the minimum cost transshipment is a forest with edges going between F_+ and F_-. We define operations based on this forest.

Let y denote the the optimal transshipment with the property described above. Thus, $y(s,t)$ denotes the ow between $s \in F_+$ and $t \in F_-$. For $t \in F_-$, let $y(\cdot,t) = \sum_{s \in F_+} y(s,t)$ and for $s \in F_+$, let $y(s,\cdot) = \sum_{t \in F_-} y(s,t)$. Similarly, for $A \subset F_-$, let $y(\cdot,A) = \sum_{t \in A} y(\cdot,t)$ and for $B \subset F_+$, let $y(B,\cdot) = \sum_{s \in B} y(s,\cdot)$.

Root each tree $\mathcal{T}$ in the forest at some node $r \in F_-$. For a node v, let $K(v)$ denote the set of its children. Thus $K(t) \subseteq F_+$ if $t \in F_-$ and $K(s) \subseteq F_-$ if $s \in F_+$. We borrow the following notation from [2] and [4]. Consider a node $t \in F_-$. We call t *weak* if $\sum_{s \in K(t)} y(s,t) > y(\cdot,t)/2$ and *strong* otherwise. We call $s \in K(t)$ *heavy* if $y(s,t) > y(\cdot,t)/2$ and *light* otherwise. Note that there can be at most one heavy node in $K(t)$. A light node $s \in K(t)$ is called *dominant* if $y(s,t) \geq y(s,\cdot)/2$ and *non-dominant* otherwise. We denote the set of dominant nodes in $K(t)$ by $\mathtt{Dom}(t)$ and the set of non-dominant nodes in $K(t)$ by $\mathtt{NDom}(t)$. For $s \in \mathtt{NDom}(t)$, let $W(s)$ denote the set of weak children of s and let $Rem(s) = \max\{y(s,t) - \sum_{t' \in W(s)} y(s,t'), 0\}$. If $|\mathtt{NDom}(t)| = k$, we re-index the facilities in $\mathtt{NDom}(t)$ as $s_1, s_2, \ldots, s_k$ such that $Rem(s_1) \leq Rem(s_2) \leq \ldots \leq Rem(s_k)$.

Consider a tree $\mathcal{T}$ in the forest. For $t \in F_-$ which is a non-leaf node in $\mathcal{T}$, let $\mathcal{T}_t$ denote the subtree rooted at t containing all the children and grand-children of t. Thus $\mathcal{T}_t$ is of depth at most two. For each such $\mathcal{T}_t$ in the forest, we consider the following operations.

1. Consider the operations $\mathtt{Single\_Pivot}(s_i, \Delta)$ for $s_i \in \mathtt{NDom}(t)$ such that $i = 1, \ldots, k-1$ where $k = |\mathtt{NDom}(t)|$. We send $y(s_i,\cdot)$ units of ow out of s_i as follows: $2y(s_i,t')$ to all $t' \in W(s_i)$ (this is feasible as t' is weak), $y(s_i,t')$ to all $t' \in K(s_i) \setminus W(s_i)$, and $Rem(s_i)$ to the facilities in $K(s_{i+1}) \setminus W(s_{i+1})$. It is straightforward to set the vector Δ. The indexing of the facilities in $\mathtt{NDom}(t)$ ensures that ow routed across every edge e in $\mathcal{T}_t$ is at most $2y(e)$, i.e., two times the ow across e in the transshipment y.

2A. **Case when t is strong.** As t is strong, there is no heavy node in $K(t)$. Consider the operation $\mathtt{Double\_Pivot}(t, s_k, \Delta^1, \Delta^2)$. We send the ow $y(\mathtt{Dom}(t),\cdot)$ from the nodes in $\mathtt{Dom}(t)$ to t and a ow of $y(s_k,t)$ from s_k to t. We set the vector Δ^1 accordingly. For each $t' \in K(s_k)$, we send a ow of $y(s_k,t')$ from s_k to t' and set Δ^2 accordingly.

2B. **Case when t is weak and there is a heavy node $s_0 \in K(t)$.** Consider $\mathtt{Single\_Pivot}(s_0, \Delta)$. For each $t' \in K(s_0)$, we send a ow of $y(s_0,t')$ from s_0 to t' and a ow of $y(s_0,t)$ from s_0 to t. The vector Δ is set accordingly. Furthermore for the facilities in $\mathtt{Dom}(t)$ and s_k, consider $\mathtt{Double\_Pivot}(t, s_k, \Delta^1, \Delta^2)$ same as the one explained in the Case 2A above.

2C. **Case when t is weak and there is no heavy node in $K(t)$.** Zhang et al. [4] show that there exists $\gamma_1 \in \mathtt{Dom}(t)$ such that the set $\mathtt{Dom}(t) \setminus \{\gamma_1\}$ can be partitioned into $\mathtt{Dom}_1$ and $\mathtt{Dom}_2$ satisfying $y(\gamma_1,t) + y(\mathtt{Dom}_i,\cdot) \leq y(\cdot,t)$ for $i = 1,2$ where $\gamma_2 = s_k$. We consider two operations $\mathtt{Double\_Pivot}(t, \gamma_i, \Delta_i^1, \Delta_i^2)$ for $i = 1,2$. Here the vector Δ_i^1 is set such that the ow $y(\mathtt{Dom}_i,\cdot)$ is sent from the nodes in $\mathtt{Dom}_i$ to t and Δ_i^2 is set such that the ow $y(\gamma_i,t)$ is sent from γ_i to t and $y(\gamma_i,t')$ is sent to each $t' \in K(\gamma_i)$. The definition of dominant nodes can be used to show that the operations are feasible.

After rerouting of the ow in above operations, if the capacity of a facility $s \in F_+$ reduces to u_s^*, we say that s is *closed*. If on the other hand, the capacity of a facility $t \in F_-$ is increased to at most u_t^*, we say that t is *opened*.

As shown by Zhang et al. [4] it is easy to verify that

LEMMA 2.2. *The operations considered above are such that each facility $s \in F_+$ is closed exactly once, each facility $t \in F_-$ is opened at most three times, and the flow across every edge in e in the forest is at most $2y(e)$.*

As proved in [2, 4], Lemmas 2.2 and 2.1 imply,

LEMMA 2.3. $C_f(S) \leq 3C_f(S^*) + 2(C_s(S) + C_s(S^*)) \leq 5C_f(S^*) + 4C_s(S^*)$.

Lemmas 2.1 and 2.3 together imply that $C(S) \leq 6C_f(S^*) + 5C_s(S^*)$, yielding a $6 + \epsilon$ approximation algorithm. Standard scaling techniques can be used to improve the ratio to $3 + 2\sqrt{2} + \epsilon < 5.83 + \epsilon$.

References

[1] M. Hajiaghayi, M. Mahdian, and V. Mirrokni, *The facility location problem with general cost functions*, Networks, 42(2003), pp. 42–47.

[2] M. Mahdian and M. Pál, *Universal facility location*, in Proceedings of ESA (2003), pp. 409–422.

[3] J. Zhang, B. Chen, and Y. Ye, *Personal communication.*

[4] J. Zhang, B. Chen, and Y. Ye, *Multi-exchange local search algorithm for capacitated facility location problem*, in Proceedings of IPCO (2004), pp. 219–233.