A Fast Algorithm for Facility Location Problem

Wenhao Shu
School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
E-mail: shuwenhao@126.com

Abstract—**In this paper, we present an approximation algorithm achieving approximation guarantee of 2 for the classical uncapacitated facility location problem. The distinguishing feature of our designed algorithm is the overall low running time of $O(m\log m)$, where $m = n_c \times n_f$, n_c and n_f are the number of cities and facilities. Though the approximation factor is 1.61 in Ref.[1], whereas the running time is $O(n^3)$, where $n=n_c+n_f$. Compared with the approximation algorithm in Ref.[1], the advantage of our algorithm is it has more applications with lower running time.**

Index Terms—**facility location problem, approximation algorithm, dual fitting, linear programming**

I. Introduction

Uncapacitated facility location problem (UFL) [17] is a classical problem that has been widely studied in operations research, such as strategically choosing placement of routers and caches in the network design in order to minimize the total cost under satisfying all clients' demands [18,19]. Many facility location models are built around the UFL problem that has been proven NP-hard. There has been renewed interest in tackling this problem from the perspective of approximation algorithms. Correspondingly, a couple of different techniques are used to analyze approximation ratio for the UFL problem and its variants, there are several main algorithm design techniques: linear programming rounding [13], primal-dual schema [11], greedy method [12], local search technique [10] and combinations of these methods with cost scaling and greedy post processing.

In the (uncapacitated) facility location problem, we are given a finite set of clients C that require a certain service. To provide such a service, we need to open a potential subset of facilities F. For each facility $i \in F$, opening it costs a non-negative number f_i; and for each facility-city pair (i, j), serving a client $j \in C$ by facility i costs c_{ji}. We assume that the c_{ji}'s satisfies the triangle inequality. The distance function is metric, which means that it satisfies the following triangle inequality: $d_{ij} + d_{ij'} + d_{i'j'} \geq d_{i'j}$, for all i, $i' \in F$ and j, $j' \in C$. That is, connecting i' to j directly is cheaper than taking the detour via j' and i. Given such a problem, the task is to open a subset of the facilities in F, and connect each city to open facilities, such that the total cost of opening facilities and connecting cities to open facilities is minimized. Afterwards, researchers extend the UFL problem by proposing Fault-Tolerant Facility Location problem [14]. The fault-tolerant facility location is a generalization of the UFL problem where each client has independent fault-tolerance requirement on connectivity. In many setting, it is essential to provide safeguards against failures by designing fault-tolerant solutions. For example, in a distributed network, we want to place caches and assign data requests to caches so as to be resistant against caches becoming unavailable due to node or link failures. At present, a practical extension of the fault-tolerant facility location problem is provided [15]. The differences of the fault-tolerant facility location problem is that a set of sites equipped with number of facilities as resources, and a set of cities with set R as corresponding connection requirements. This model provides optimal resource allocation between enterprises and clients in today's cloud platforms, which becomes essential in many contemporary applications [18, 24, 25,27].

In addition, some other extension versions of the uncapacitated facility location problem are considered. Its variants of the problem have resulted in much progress in recent years. There is often a capacitated version of UFL. Each facility i can be assigned to serve at most μ_i cities, that indicates the capacity of this facility, i.e., the number of cities it can serve, where μ_i is a positive integer, if facility i is opened y_i times, it can serve at most $\mu_i y_i$ cities. And we also consider the k-median problem which is similar to the facility location problem [28,29]. Two differences from the facility location problem are considered: there is no cost for opening facilities, and there is an upper bound k on the number of facilities, where k is an input parameter that is not fixed. The k-median problem has numerous applications, especially in the context of clustering. This problem has found new clustering applications in the area of data mining. Clearly, some variants of the facility location problem are applicable to a number of industrial situations.

In this paper, we consider the uncapacitated facility location problem from the view of applications, though the approximation factor of the approximation algorithm

Manuscript received September 2, 2012.

in the literature [1] for the UFL problem is low, our designed algorithm has a particularly lower running time. The differences of our algorithm are as follows: at first, we sort all the facility costs and connection costs, and then each time we choose one facility to open according to a set of cities at the first rank of connection costs with the most cost-effectiveness, and connect this set of cities to the opened facility. Similarly, a set of cities at the second rank of connection costs do the same events, and so on, until all unconnected cities have connected to some opened facilities. At the same time, the corresponding dual algorithm also has a simple description below: the algorithm starts at time 0, at this time, all cities at the first rank of connection costs increase the total cost until two following events occur. One is that for some closed facility i, when it receives from cities at the first rank is equal to the cost of opening i, we open facility i, and connect these cities to i. The other is that for some unconnected city j but some facility i has already opened, when the total cost of j is equal to the connection cost between j and i, we connect city j to facility i. Then until all cities at the last rank have been connected the algorithm terminates.

This paper is organized as follows. Section II reviews some related works about the facility location problem. In Section Ⅲ, we provide a fast algorithm for the UFL problem. In addition, a case study is illustrated to explain the proposed algorithm. In Section Ⅳ, we use dual fitting to analyze the approximation factor of the designed algorithm. Section Ⅴ presents conclusions and outlines our further research trends.

II. Related Works

The UFL problem has attracted great interest of researchers in both theoretical computer science and operations research. We now briefly review past work on UFL and related problems. The first constant factor algorithm for the metric uncapacitated facility location problem was given by Shmoys, Tardos and Aardal [3] based on the filtering and rounding technique of Lin and Vitter [20]. Their approximation guarantee was 3.16. LP rounding was among the very first techniques that yield non-trivial approximation guarantees for metric facility location. Using the randomized rounding technique, Chudak and Shmoys [4] gave an algorithm with approximation ratio $(1+2/e)\approx 1.736$, which is a significant improvement on the previously approximation guarantees. Recently, Byrka [5] used the idea of scaling-up fractional solution and techniques from [6] to obtain another LP-rounding algorithm with approximation ratio of 1.5. The currently best known approximation ratio is achieved by the 1.488-approximation algorithm of Li [7]. The drawback of these algorithms based on LP-rounding is that they have prohibitive running times for most applications because they need to solve large linear programs. Charikar and Guha obtained 1.853-apprximation and 1.728-approximation algorithms [21] by using primal-dual theory and greedy augmentation. Chudak and Sviridenko improve the approximation ratio to 1.736 [4] and 1.582 [23] also by rounding an optimal fractional solution to a linear program. These results are achieved through linear programming and rounding. Regarding hardness results, Guha and Khuller [2] proved that it is impossible to get an approximation factor 1.463 for the metric facility location problem, unless $NP\subseteq DTIME[n^{O(\log\log n)}]$. A different approach local search was used by Korupolu, Plaxton and Rajarama [8]. Local search techniques have been very popular as heuristics for hard combinatorial optimization problems. They analyzed a well known local search heuristic and showed that it achieves an approximation guarantee of $(5+\varepsilon)$. However, the algorithm has a fairly high running time of $O(n^6\log n/\varepsilon)$. At the same time, Jain and Vazirani [16, 3] substantiates the primal-dual schema has its full potential to realize in the area of approximation algorithms. And their algorithm runs a quasi-linear time whose running time is $O(m\log m)$. Another greedy approach is used to improve the solution: iteratively open facility at a time if it improves the cost of the solution. And it also showed that greedy improvement can be used as a post-processing step to improve the approximation guarantee of certain facility location algorithms. In paper [2], a simple greedy heuristic with the algorithm by Shmoys, Tardos and Aardal [3] can be used to obtain an approximation guarantee of 2.408. In paper [1], an improvement of the greedy algorithm of [9] is presented for the facility location problem, achieving an approximation factor of 1.61 with the running time $O(n^3)$. Sometimes, it is hard to carry out the algorithm with high running time in real-world tasks. In this paper, our contribution is to reduce the time complexity of solving the facility location problem.

Ⅲ. Fast algorithm for facility location problem

In this section, we first recall the uncapacitated facility location problem as integer linear programming. Then we design an algorithm with running time $O(m\log m)$, where $m=n_c\times n_f$, n_c and n_f are the number of cities and facilities. And a case study illustrates the feasibility and validity of the proposed algorithm. Finally, we use dual fitting technique to analyze the approximation factor of the proposed algorithm.

A. Problem Formulation

The Uncapacitated Facility Location (UFL) problem seeks a minimum cost way of connecting cities to open facilities, which can be formulated by the following integer linear program. This problem has occupied a central place in operations research since the early 60's, and has been studied from the perspectives of worst case analysis, probabilistic analysis, polyhedral combinatorics and empirical heuristics.

$$\text{Minimize} \sum_{i\in F,\, j\in C} c_{ji}x_{ji} + \sum_{i\in F} f_i y_i$$

Subjected to $\forall j \in C : \sum_{i \in F} x_{ji} = 1$ (1)

$\forall i \in F, j \in C : y_i - x_{ji} \geq 0$ (2)

$\forall i \in F : y_i \in \{0,1\}$ (3)

$\forall i \in F, j \in C : x_{ji} \in \{0,1\}$ (4)

In the above formulation, y_i is an indicator variable denoting whether facility i is open, and x_{ji} is an indicator variable denoting whether city j is connected to 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. The objective is to open a subset of the facilities in F, and connect each city to an open facility so that the total cost is minimized. To reduce the time complexity of solving the facility location problem, our designed algorithm is developed as follows.

B. The Algorithm

Our designed algorithm is shown in the following. At first, we sort all the facility costs and connection costs, and then each time we choose one facility to open according to a set of cities at the ranks of connection costs with the most cost-effectiveness, and connect the set of cities to the opened facility. The idea of cost effectiveness essentially stems from a similar notion in the greedy algorithm for the set cover problem; the cost-effectiveness is measured as the ratio of the cost incurred to the number of new cities served. In addition, delete the connected cities from the set of cities then remove the cost-effectiveness of the chosen facility, and update the minimum cost-effectiveness to a certain facility. The most cost-effective set can be found either by using direct computation, or by using the dual program of the linear programming formulation for the problem. This process terminates until all cities have been connected. Note that once a city gets connected to a facility, its budget remains constant and it cannot revoke its contribution to a facility, so it can never get connected to another facility with a higher connection cost. And a facility can be chosen again after being opened, but its opening cost is counted only once since we set f_i to zero after the first time the facility is picked by the algorithm. As far as cities are concerned, every city j is removed from C, when connected to an open facility, and is not taken into consideration again.

Algorithm: A fast algorithm for facility location problem

Input: $\forall i \in F$, $j \in C$,opening cost of facility f_i ,connection cost c_{ji} between i and j.

Set $U = C$, $\forall i, j, y_i = 0, x_{ji} = 0, t = 1$.

Output: $\forall i, j, y_i, x_{ji}$.

Step1: $\forall i \in F$, Sort the opening cost f_i of all facilities;

Step2: $\forall i \in F$, $j \in C$, Sort the connection cost c_{ji} of all facility-city pairs;

Step 3: While $U \neq \varnothing$, do:

Step3.1: For the cities at the rank t of connection costs, count the number of connections to each facility i denoting as k ,and calculate the cost-effectiveness $r_{it} = \dfrac{f_i + \sum c_{ji}}{k}$ (Note that if facility i has already been opened, then consider the cost-effectiveness $r_{it} = \dfrac{\sum c_{ji}}{k}$); choose the most cost-effectiveness of the facility to open from the set of cost-effectiveness R, set $y_i = 1$ when satisfies the condition $\min_{i \in F} r_{it}$, and set $f_i = 0$;if two facilities have the same cost-effectiveness, we choose one facility to open according to the number of connection;

Step3.2: Delete a set of cities that correspond the most cost-effectiveness to a certain facility i, i.e. $U = U - \{j\}$, and remove the cost-effectiveness from the set of cost-effectiveness R;

Step3.3: Update the most cost-effectiveness that facility i corresponds to;// Reserve the lowest connection cost between cities and facility each time.

Step3.4: If all cities get connected to an open facility, the algorithm terminates; else $t++$, turns to Step 3.1;

The time complexity of the proposed algorithm is analyzed as follows: the time of Step 1 for sorting all the facility costs in ascender order is $O(n_f \log n_f)$, where n_f is the number of facilities; the time of Step 2 for sorting all the connection costs in ascender order is $O(n_c \times n_f log(n_c \times n_f))$, where n_c is the number of cities; the total events of Step 3 is completed at most $O(n_c \times n_f)$. So the time complexity of the proposed algorithm is $O(m \log m)$, where $m = n_c \times n_f$. Therefore the algorithm can be finished efficiently in polynominal time.

C. Case study

Assume we are given a set of facilities F consists of five facilities, and the set of cities C consists of seven cities, for each facility $i \in F$, the facilities cost $f_i (i = 1, 2, \cdots, 5)$ and connection cost $c_{ji} (j = 1, 2, \cdots, 7)$ are shown below, respectively. All clients in C require a certain service, to provide such a service, we need to open a subset of facilities F, such that the total cost of opening facilities and connecting cities to open facilities is minimized. Here we are given all the sorted facilities costs and connection costs. Note that c_{ji} denotes the connection cost between the j-th city and i-th facility.

$f_4 = 5$, $f_1 = 6$, $f_5 = 8$, $f_2 = 10$, $f_3 = 12$

$c_{12} = 2, c_{11} = 4, c_{13} = 5, c_{15} = 6, c_{14} = 8$

$c_{22} = 2, c_{21} = 3, c_{23} = 6, c_{24} = 7, c_{25} = 9$

$c_{33} = 1, c_{34} = 4, c_{31} = 6, c_{35} = 7, c_{32} = 8$

$c_{43} = 2, c_{44} = 3, c_{45} = 4, c_{41} = 5, c_{42} = 7$

$c_{55} = 3, c_{52} = 4, c_{51} = 7, c_{54} = 9, c_{53} = 10$

$c_{61} = 1, c_{64} = 3, c_{65} = 6, c_{63} = 8, c_{62} = 9$

$c_{74} = 3, c_{72} = 5, c_{75} = 7, c_{73} = 8, c_{71} = 12$

For the given case, we use the above algorithm to calculate which facilities to open and all the cities connect to which opened facilities. Due to the above facilities costs and the connection costs have been ordered. According to Step 3, do the first loop, we have $c_{12} = 2$, $c_{22} = 2$, $f_2 = 10$; $c_{33} = 1$, $c_{43} = 2$, $f_3 = 12$; $c_{55} = 3$, $f_5 = 8$; $c_{61} = 1$, $f_1 = 6$; $c_{74} = 3$, $f_4 = 5$. The cost-effectiveness of the second, third, fifth, first and fourth facility is computed as:

$$r_{21} = \frac{f_2 + c_{12} + c_{22}}{2} = \frac{10+2+2}{2} = 7,$$

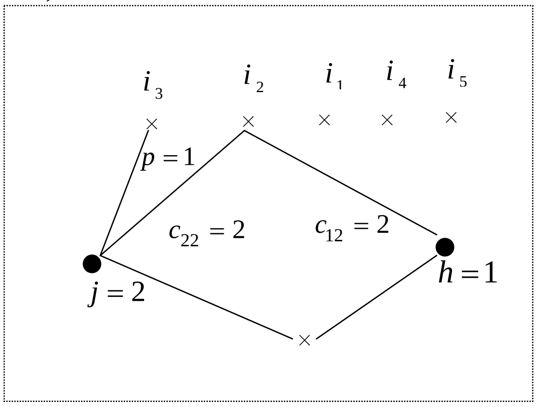
$$r_{31} = \frac{f_3 + c_{33} + c_{43}}{2} = \frac{12+1+2}{2} = 7.5,$$

$$r_{51} = \frac{f_5 + c_{55}}{1} = \frac{8+3}{1} = 11,$$

$$r_{11} = \frac{f_1 + c_{61}}{1} = \frac{6+1}{1} = 7,$$

$$r_{41} = \frac{f_4 + c_{74}}{1} = \frac{5+3}{1} = 8.$$

Where r_{i1} represents the cost-effectiveness for the i -th facility at the first rank of connection costs. During this process, the cost-effectiveness of r_{21} and r_{11} are the same, but the second facility is to open, because the number of connected cities is more than that of the first facility. In addition,



(a) $p = 1$

Figure 1. The p-th of contributions to open the second facility

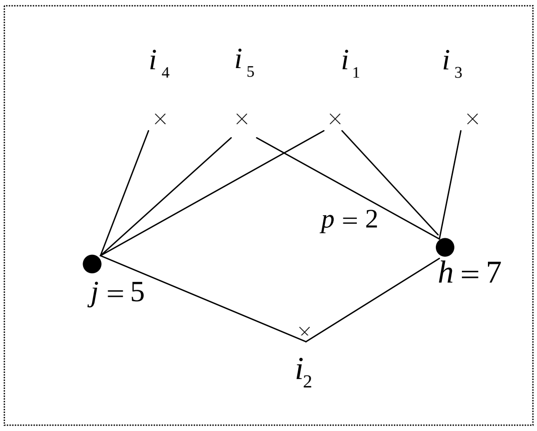
we connect the first and second city to the second facility, then delete them from the set of cities and remove the cost-effectiveness r_{21} from the set of cost-effectiveness. As shown in Figure 1, the first and second cities have minimum cost-effectiveness to connect the second facility.

For the second loop, we have $c_{34} = 4, c_{44} = 3, c_{64} = 3$, $f_4 = 5; c_{52} = 4, c_{72} = 5$, $f_2 = 0$. The cost-effectiveness of the fourth and second facility is computed as:

$$r_{42} = \frac{f_4 + c_{34} + c_{44} + c_{64}}{3} = \frac{5+3+3+4}{3} = 5,$$

$$r_{22} = \frac{f_2' + c_{52} + c_{72}}{2} = \frac{0+5+4}{2} = 4.5.$$

During this process, the cost-effectiveness of r_{22} is minimum, we connect the fifth and seventh city to the second facility, then delete them from the set of cities and remove the cost-effectiveness r_{22} from the set of cost-effectiveness. In addition, we update the minimum cost-effectiveness r_{42} for the fourth facility. As shown in Figure 1, the fifth city and seventh city have enough minimum cost-effectiveness to connect the second facility, and the cost of the second facility is 0, because at the first phase, the second facility is open.



(b) $p = 2$

Figure 2. The p-th of connections to the second facility

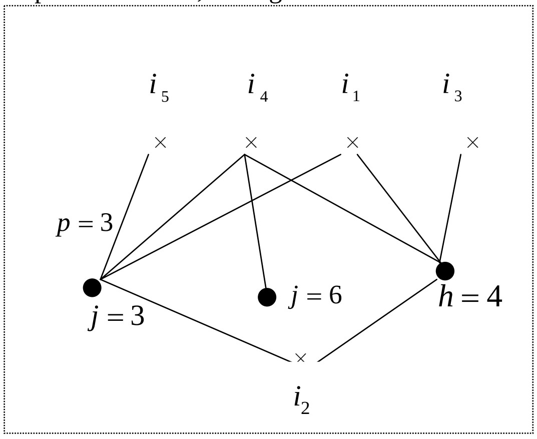
For the third loop, we have $c_{31} = 6$, $f_1 = 6$; $c_{45} = 4$, $c_{65} = 6$, $f_5 = 8$. The cost-effectiveness of the first and fifth facility is computed as:

$$r_{13} = \frac{f_1 + c_{31}}{1} = \frac{6+6}{1} = 12,$$

$$r_{53} = \frac{f_5 + c_{45} + c_{65}}{2} = \frac{4+6+8}{2} = 9.$$

During this process, because the preserved cost-effectiveness of r_{42} is minimum, we open the fourth facility, then connect the third, fourth and sixth city to the fourth facility as shown in Figure 3 and delete them from the set of cities and remove the cost-effectiveness r_{42}

from the set of cost-effectiveness. In addition, we update the minimum cost-effectiveness r_{53} for the fifth facility. Because all unconnected cities have been connected to some opened facilities, the algorithm terminates.



(b) $p=3$

Figure 3. The p-th of contributions to open the fourth facility

The outputs of the algorithm: y_2, $y_4=1$, that is the second and fourth facility are open; x_{12}, x_{22}, x_{52}, x_{72}, x_{34}, x_{44}, $x_{64}=1$, that is the first, second, fifth, and seventh city are connected to the second facility, and the third, fourth, and sixth city are connected to the fourth facility.

Ⅳ. Analysis of the Proposed Algorithm

After designing the algorithm for the UFL problem, a case study shows the feasibility and validity of the proposed algorithm. We will use dual fitting to analyze the approximation factor of above algorithm. The method of dual fitting can be described as follows, assuming a minimization problem: the basic algorithm is combinatorial--in the case of set cover it is in fact a simple greedy algorithm. Using the linear programming relaxation of the problem and its dual, one first interprets the combinatorial algorithm as a primal-dual-type algorithm--an algorithm that is iteratively making primal and dual updates. Strictly speaking, this is not a primal-dual algorithm, since the dual solution computed is, in general, infeasible. However, one shows that the primal integral solution found by the algorithm is fully paid for by the dual computed. By fully paid for we mean that the objective function value of the primal solution is bounded by that of the dual. The main step in the analysis consists of dividing the dual by a suitable factor, say r, and showing that the shrunk dual is feasible, i.e., it fits into the given instance. The shrunk dual is then a lower bound on OPT, and r is the approximation guarantee of the algorithm.

Different from prima-dual schema [16, 22], dual fitting relax the feasibility of the dual solution. Assume the dual solution become feasible after shrunk by a factor, then this factor is the approximation factor of the algorithm. The technique that we used in this paper seems to be useful tool for analyzing heuristic and local search algorithms. At first, by relaxing the integrality constraints of above integer programming shown in Section Ⅲ, we get the following linear program.

$$\text{Minimize } \sum_{i\in F, j\in C} c_{ji}x_{ji} + \sum_{i\in F} f_i y_i$$

$$\text{Subjected to } \forall j\in C: \sum_{i\in F} x_{ji} = 1 \quad (1)$$

$$\forall i\in F, j\in C: y_i - x_{ji} \ge 0 \quad (2)$$

$$\forall i\in F: y_i \ge 0 \quad (3)$$

$$\forall i\in F, j\in C: x_{ji} \ge 0 \quad (4)$$

Recall the dual program of the uncapacitated facility location problem, in which v_j and w_{ij} are dual variables. The dual program can also be used to prove the approximation factor of the algorithm. Similarly, we will use the LP-formulation of facility location to analyze our algorithm. As we will see, the dual formulation of the problem helps us to understand the nature of the problem and the proposed algorithm.

$$\max mize \sum_{j\in C} v_j$$

$$\text{Subject to } \sum_{j\in C} w_{ji} \le f_i, \forall i\in F \quad (1)$$

$$v_j - w_{ji} \le c_{ij}, \ \forall i\in F, \ \forall j\in C \quad (2)$$

$$w_{ji} \ge 0, \ \forall i\in F, \ \forall j\in C. \quad (3)$$

Based on the complementary slackness condition, a very nice interpretation to the dual can be given.

Each open facility is fully paid for, i.e., if $i\in F$, then $\sum_{j\in C} w_{ji} = f_i$. This interpretation is that for a facility i to be opened, the cities have to pay the entire cost f_i. City j is connected to facility i, i.e, $v_j - w_{ji} = c_{ij}$. v_j can be interpreted as the total cost by city j is willing to pay, and w_{ji} can be interpreted as the amount that j is willing to contribute for opening facility i.

Here we think of v_j as the total of cost paid by city j (the cost of opening facility and connecting city to facility), the algorithm is described as follows:

```
Step1: At the beginning $t=0$, let all facilities be
       unopened, and set $\hat{f}_i = 2f_i$, and let each city
       be unconnected, and set $\alpha_j = 0$;
Step2: While $U \ne \varnothing$, do:
       For each city $j\in C$ in each rank of connection
       costs, raise $\alpha_j$ for each unconnected city $j$
       uniformly at unite rate until $\sum_{j\in C}(\alpha_j - c_{ji})$
```

$= \hat{f}_i$,if facility i is opened, raise α_j until $\alpha_j = c_{ji}$;

Step2.1: Some unopened facility i receive enough paid to open itself, that is $\sum_{j\in C}(\alpha_j - c_{ji})$. $= \hat{f}_i$ In this case, set $y_i = 1$ and if the cities satisfy $\alpha_j \geq c_{ji}$, set $x_{ji} = 1$,then delete the cities that have been connected to open facility from each rank of connection costs;

Step 2.2: For some city $j \in C$, its cost is enough is connect an open facility i which is not connected before, i.e., $\alpha_j = c_{ji}$. In this case, set $x_{ji} = 1$,then delete the cities from each rank of connection costs;

Step 3: If all cities get connected to an open facility, the algorithm terminates; else turns to Step 2.

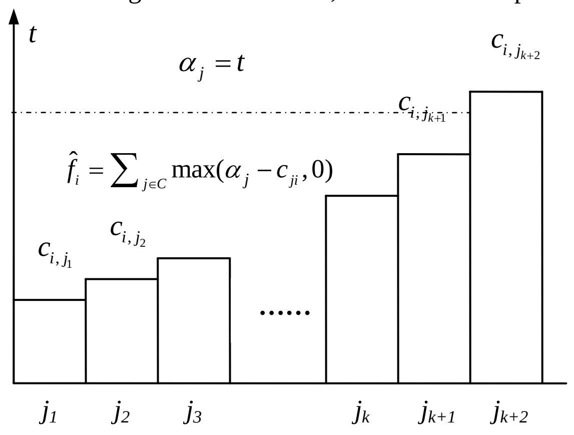


Figure 4. Cost Offers for Opening a Facility

In above algorithm, each facility has a list of cities that are ordered according to their connection costs to the facility. As showed in Figure 4, the most cost-effectiveness will consist of a facility and a set, containing the first k cities in this order. Crucial to this algorithm is a subselection step, which ensures that every facility and the clients that connect to it are adequately accounted for in the dual-fitting analysis. We describe the above algorithm very briefly. The graph can be assumed to a complete bipartite graph between the facilities and the demand nodes. This algorithm maintains dual values for each demand node v_j and each edge w_{ji}. Initially all these variables are set to zero. The algorithm grows the dual variables corresponding to the demand nodes. At various times some of the dual variables v_j stop growing, to maintain feasibility of the dual solution.

To evaluate the performance guarantee of the algorithm, we can get the following Theorem 1 to analyze the approximation factor of this facility location problem. An approximation algorithm with a factor of ρ is a polynomial time algorithm that produces a solution whose total cost of opening facilities and connecting cities to open facilities is at most ρ times the optimal solution cost. The proposed algorithm achieves an approximation factor of 2 for the UFL problem.

Theorem 1. The above algorithm achieves an approximation factor of 2 for the facility location problem.

Proof: For above algorithm, we can easily obtain that $\sum_{j\in C}\alpha_j = \sum_{j\in C, i\in F'} c_{ji}) + \sum_{i\in F'} \hat{f}_i$,where F' is the set of facilities opened by the algorithm. Because in each rank of connection costs, we delete the cities that have been connected some open facilities such that $\sum_{j\in C} w_{ji} > f_i$,which does not satisfy the conditions of dual problem $\sum_{j\in C} w_{ji} \leq f_i$. To get dual feasible solutions for the dual linear program, set $\hat{f}_i = 2f_i$, so that (v, w) is in feasible for the dual problem, where $v_j = \alpha_j / 2$, $w_{ji} = \max(v_j - c_{ji}, 0)$. The total cost of facilities costs and connection costs is $\sum_{j\in C, i\in F'} c_{ji} + \sum_{i\in F'} f_i \leq \sum_{j\in C, i\in F'} c_{ji} + 2\sum_{i\in F'} f_i = \sum_{j\in C}\alpha_j = 2\sum_{j\in C} v_j$, by weak duality, $\sum_{j\in C} v_j \leq OPT$, thus $\sum_{j\in C, i\in F'} c_{ji} + \sum_{i\in F'} f_i \leq 2OPT$.

V. CONCLUSION AND FUTURE WORK

The facility location problem has been proved NP-hard. A large fraction of the theory of approximation algorithm, as we know it today, is built around linear programming, which provides some main algorithm design techniques. Some of the techniques have yielded algorithms with good approximation guarantees. However, with respect to the running times of the algorithms derived, the methods differ. The algorithms with low running time but not too high approximation guarantees have more practical applications. In this paper, we provided an algorithm with running time of $O(m \log m)$, achieving approximation guarantee of 2 by the dual fitting technique. The distinguishing feature of our algorithm is the lower running time compared with that of algorithm in Ref. [1]. At the same time, the proposed solution of the UFL problems will provide useful knowledge and experience for further optimal resource allocation strategies on the contemporary cloud platforms. In the future work, we will use this algorithm as a subroutine to solve some related variants of the facility location problem, such as the k-median problem.

ACKNOWLEDGMENTS

This work was supported in part by the Innovation Funds of Excellence Doctor of Beijing Jiaotong University

(2012YJS025), and the Natural Science Foundation of China (61170232 and 61100218).

REFERENCES

[1] K. Jain, M. Mahdian, and A. Saberi, "A new greedy approach for facility location problems," In Proceeding of the 30th Annual ACM symposium on Theory of Computing, New York, USA, pp.731-740, 2002.

[2] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," Journal of Algorithms, vol.31, pp.228-248, 1999.

[3] D. B. Shmoys, E. Tardos, and K. Aardal, "Approximation algorithms for facility location problems," In Proceeding of the 29th ACM Symposium on Theory of Computing, pp. 265-274, 1997.

[4] F. Chudak and D. Shmoys, "Improved approximation algorithms for uncapacitated facility location," In R.E. Bixby, E. A. Boyd and R. Z. Rios-Mercado, eds., Integer Programming and Cominatorial Optimization, Springer LNCS Vol. 1412, pp.180-194, 1998.

[5] J. Byrka, and K. I. Aardal, "An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem," SIAM Journal on Computing, vol.39, no.6, pp. 2212-2231, 2010.

[6] M. Mahdian, Y. Ye, J. Zhang, "Approximation algorithms for metric facility location problems," SIAM Journal on Computing, 2006, vol.36, no.2, pp.411-432, 2006.

[7] S. Li, "A 1.488 approximation algorithm for the uncapacitated facility location problem," Luca Aceto, In Proceedings of ICALP, PartⅡ, Switzerland: Springer, pp. 77-88, 2011.

[8] M. R. Korupolu, and C. G. Plaxton, "Analysis of a Local Search Heuristic for Facility Location Problems," Journal of Algorithms, vol.37, no.1, pp.146-188, 2000.

[9] M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, "A greedy facility location algorithm analyzed using dual-fitting. In Proceeding of 4th International Workshop on Approximation Algorithms for Combinatorial Optimizati-on," pp.127-137, 2001.

[10] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit, "Local search heuristics for k-median and facility location problems," In Proceedings of the 33th ACM Symposium on Theory of Computing, pp.21-29, 2001.

[11] K. Jain, and V. V. Vazirani, "Primal-dual approximation algorithms for metric facility location and k-median problems," In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp.2-13, 1999.

[12] K. Jain, M. Mahdian, E. Saberi, V. V. Vazirani, "Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP," Journal of the ACM, vol.50, no.6, pp.795-824, 2003.

[13] J. Byrka, A. Srinivasan, C. Swamy, "Fault-tolerant facility location: a randomized dependent LP-rounding algorithm," Friedrich Eisenbrand and F. Bruce, In Proceedings of IPCO, Switzerland: Springer, pp.244-257, 2010.

[14] S. Guha, A. Meyerson, and K. Munagala, "Improved Approximation Algorithms for Fault-tolerant Facility Location," In Proceeding of the 12th ACM-SIAM Symposium on Discrete Algorithms, pp.636- 641, 2011.

[15] S. Xu, H. Shen, "The fault-tolerant facility allocation problem," In: Y. Dong, D. Z. Du, Ibarra(eds.), ISAAC LNCS, vol.5878, pp.689-698, 2009.

[16] K. Jain and V. V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation," Journal of the ACM, vol.48, no.2, pp.274-296, 2001.

[17] B. Pitu, R. L. Mirchandani (eds.): Discrete Location Theory. John Wiley, New York, 1990.

[18] Mehdi Eghbal, Naoto Yorino, Yoshifumi Zoka,Comparative Study on the Application of Modern Heuristic Techniques to SVC Placement Problem, Journal of Computers, Vol 4, No 6 (2009), 535-541.

[19] J. B. Ilan, G. Kortsarz, and D. Peleg, "How to allocate network centers," Journal of the Algorithms, vol.15, pp. 385-415, 1993.

[20] J. H. Lin, and J. S. Vitter, " ε -approximations with minimum packing constraint violation," In Proceedings of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, pp.771-782, 1992.

[21] M.Charikar and S. Guha, "Improved combinatorial algorithms for facility location problems," SIAM J.Comput, vol.34, no.4, pp.803-824, 2005.

[22] V. V. Vazirani. "Approximation Algorithms," Springer-Verlag, Berlin, 2001.

[23] M.Sviridenko. " An improved approximation algorithm for the metric uncapacitated facility location problem ". In Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, pp. 240-257, London, UK, 2002.

[24] Yongjun Peng, Zhengbing Hu, Xi Guo, Research on the Evolution Law and Response Capability Based on Resource Allocation Model of Unconventional Emergency, Journal of Computers, Vol 5, No 12 (2010), 1899-1906.

[25] Changbing Jiang, Research on Logistics Network Infrastructure Based on HCA and DEA-PCA Approach, Journal of Computers, Vol 5, No 4 (2010), 533-540.

[26] J. Byrka and K. Aardal, The approximation gap for the metric facility location problem is not yet closed, Oper. Res. Lett., vol.35, pp. 379–384, 2007.

[27] Ye, Y., Zhang, J.: An approximation algorithm for the dynamic facility location problem. Combinatorial Optimization in Communication Networks, Kluwer Academic Publishers, pp. 623-637 (2005).

[28] Xu, G., Xu, J.: An improved approximation algorithm for uncapacitated facility location problem with penalty. Journal of Combinatorial Optimization 17, 424-436 (2008).

[29] Du, D., Lu, R., Xu, D.: A primal-dual approximation algorithm for the facility location problem with submodular penalty. Algorithmica 63, 191-200 (2012).

Wenhao Shu was born in Jiangxi, P. R. China, in 1985. She received the M.S. degree in computer science and technology from Guangxi Normal University, Guangxi, China, in 2011. She is a doctoral student at school of computer and information technology of Beijing Jiaotong University. Her main research interests include Design and Analysis of Algorithms, Granular Computing and Intelligent Information Processing.