Information and Computation 222 (2013) 45–58

Contents lists available at SciVerse ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

A 1.488 approximation algorithm for the uncapacitated facility location problem

Shi Li [1]

Department of Computer Science, Princeton University, Princeton, NJ 08540, USA

ARTICLE INFO

Article history:
Available online 26 October 2012

Keywords:
Approximation
Facility location problem
Theory

ABSTRACT

We present a 1.488-approximation algorithm for the metric uncapacitated facility location (UFL) problem. Previously, the best algorithm was due to Byrka (2007). Byrka proposed an algorithm parametrized by γ and used it with $\gamma \approx 1.6774$. By either running his algorithm or the algorithm proposed by Jain, Mahdian and Saberi (STOC'02), Byrka obtained an algorithm that gives expected approximation ratio 1.5. We show that if γ is randomly selected, the approximation ratio can be improved to 1.488. Our algorithm cuts the gap with the 1.463 approximability lower bound by almost 1/3.

1. Introduction

In this paper, we present an improved approximation algorithm for the (metric) uncapacitated facility location (UFL) problem. In the UFL problem, we are given a set $\mathcal{F}$ of potential facility locations, each location $i \in \mathcal{F}$ with a facility cost f_i, a set $\mathcal{C}$ of clients and a metric d over $\mathcal{F} \cup \mathcal{C}$. The goal is to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ of locations to open facilities, so that the sum of the total facility cost and the connection cost is minimized. The total facility cost is $\sum_{i \in \mathcal{F}'} f_i$, and the connection cost is $\sum_{j \in \mathcal{C}} d(j, i_j)$, where i_j is the closest facility to j in $\mathcal{F}'$.

The UFL problem is NP-hard and has received a lot of attention. In 1982, Hochbaum [5] presented a greedy algorithm for the non-metric UFL with $O(\log n)$-approximation guarantee. Constant factor approximation algorithms are known for the metric UFL. Shmoys, Tardos and Aardal [13] used the filtering technique of Lin and Vitter [11] to give a 3.16-approximation algorithm, which is the first constant factor approximation for the metric UFL problem. After that, a large number of constant factor approximation algorithms were proposed [4,9,3,7,8,12]. The current best known approximation ratio is 1.50, given by Byrka [1].

On the negative side, Guha and Kuller [6] showed that there is no ρ-approximation for the UFL problem if $\rho < 1.463$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{O(\log\log n)})$. Later, Sviridenko [14] strengthened the result by changing the condition to "unless $\mathbf{NP} = \mathbf{P}$". Jain et al. [8] generalized the result to show that no (γ_f, γ_c)-bifactor approximation exists for $\gamma_c < 1 + 2e^{-\gamma_f}$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{O(\log\log n)})$. An algorithm is a (γ_f, γ_c)-approximation algorithm if the solution given by the algorithm has expected total cost at most $\gamma_f F^* + \gamma_c C^*$, where F^* and C^* are respectively the facility and the connection cost of an optimal solution for the linear programming relaxation of the UFL problem, which is described later.

Building on the work of Byrka [1], we give a 1.488-approximation algorithm for the UFL problem. (The preliminary version of this paper appeared in [10].) Byrka presented an algorithm $A_1(\gamma)$ which gives the optimal bifactor approximation $(\gamma, 1 + 2e^{-\gamma})$ for $\gamma \geqslant \gamma_0 \approx 1.6774$. By either running $A_1(\gamma_0)$ or the (1.11, 1.78)-approximation algorithm A_2 proposed by

E-mail address: shili@cs.princeton.edu.

[1] This research was supported by NSF awards MSPA-MCS 0528414, CCF 0832797, and AF 0916218.

0890-5401/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ic.2012.01.007

Jain, Mahdian and Saberi [8], Byrka was able to give a 1.5-approximation algorithm. We show that the approximation ratio can be improved to 1.488 if γ is randomly selected. To be more specific, we show

Theorem 1. *There is a distribution over* $(1,\infty)\cup\{\perp\}$ *such that the following random algorithm for the UFL problem gives a solution whose expected cost is at most* 1.488 *times the cost of the optimal solution*: *we randomly choose a* γ *from the distribution*; *if* $\gamma=\perp$, *return the solution given by* A_2; *otherwise, return the solution given by* $A_1(\gamma)$.

Due to the $(\gamma, 1+2e^{-\gamma}-\epsilon)$-hardness result given by [8], there is a hard instance for the algorithm $A_1(\gamma)$ for every γ. Roughly speaking, we show that a fixed instance cannot be hard for two different γ's. Guided by this fact, we first give a bifactor approximation ratio for $A_1(\gamma)$ that depends on the input instance and then introduce a 0-sum game that characterizes the approximation ratio of our algorithm. The game is between an algorithm designer and an adversary. The algorithm designer plays either $A_1(\gamma)$ for some $\gamma>1$ or A_2, while the adversary plays an input instance for the UFL problem. By giving an explicit (mixed) strategy for the algorithm designer, we show that the value of the game is at most 1.488.

The remaining part of the paper is organized as follows. In Section 2, we review the approximation algorithm $A_1(\gamma)$, $\gamma>1$ in [1], which gives a $(\gamma, 1+2e^{-\gamma})$-bifactor approximation for $\gamma\geqslant\gamma_0\approx 1.67736$, and then we give our algorithm in Section 3.

2. Review of the Algorithm $A_1(\gamma)$ in [1]

In $A_1(\gamma)$, $\gamma>1$, we first solve the following natural linear programming relaxation for the UFL problem.

$$\min \sum_{i\in\mathcal{F}, j\in\mathcal{C}} d(i,j)x_{i,j} + \sum_{i\in\mathcal{F}} f_i y_i \quad \text{s.t.}$$

$$\sum_{i\in\mathcal{F}} x_{i,j} = 1 \quad \forall j\in\mathcal{C} \tag{1}$$

$$x_{i,j} - y_i \leqslant 0 \quad \forall i\in\mathcal{F},\ j\in\mathcal{C} \tag{2}$$

$$x_{i,j}, y_i \geqslant 0 \quad \forall i\in\mathcal{F},\ j\in\mathcal{C} \tag{3}$$

In the integer programming correspondent to the above LP relaxation, we have additional constraint that $x_{i,j}, y_i\in\{0,1\}$ for every $i\in\mathcal{F}$ and $j\in\mathcal{C}$. y_i indicates if the facility i is open and $x_{i,j}$ indicates if the client j is connected to the facility i. Eq. (1) says that the client j must be connected to some facility and inequality (2) says that a client j can be connected to a facility i only if i is open.

If the y-variables are fixed, x-variables can be assigned greedily in the following way. Initially, $x_{i,j}=0$. For each client $j\in\mathcal{C}$, execute the following steps. Sort facilities by their distances to j; then for each facility i in the order, assign $x_{ij}=y_i$ if $\sum_{i'\in\mathcal{F}} x_{i',j}+y_i\leqslant 1$ and $x_{i,j}=1-\sum_{i'} x_{i',j}$ otherwise.

After obtaining a solution (x,y), we modify it by scaling the y-variables up by γ. Let $\bar{y}$ be the scaled vector of y-variables. We reassign x-variables using the above greedy process to obtain a new solution $(\bar{x},\bar{y})$.

Without loss of generality, we can assume that the following conditions hold for every $i\in\mathcal{C}$ and $j\in\mathcal{F}$:

1. $x_{i,j}\in\{0,y_i\}$;
2. $\bar{x}_{i,j}\in\{0,\bar{y}_i\}$;
3. $0<\gamma y_i=\bar{y}_i\leqslant 1$.

Indeed, the above conditions can be guaranteed by splitting facilities. To guarantee the first condition, we split i into 2 co-located facilities i' and i'' and let $x_{i',j}=y_{i'}=x_{i,j}$, $y_{i''}=y_i-x_{i,j}$ and $x_{i'',j}=0$, if we find some facility $i\in\mathcal{F}$ and client $j\in\mathcal{C}$ with $0<x_{i,j}<y_i$. The other x variables associated with i' and i'' can be assigned naturally. We update $\bar{x}$ and $\bar{y}$ variables accordingly. Similarly, we can guarantee the second condition. To guarantee the third condition, we can remove the facilities i with $y_i=0$; we can split a facility i into 2 co-located facilities i' and i'' with $\bar{y}_{i'}=1$ and $\bar{y}_{i''}=\bar{y}_i-1$, if we find some facility $i\in\mathcal{F}$ with $\bar{y}_i>1$.

Definition 2 *(Volume).* For some subset $\mathcal{F}'\subseteq\mathcal{F}$ of facilities, define the *volume* of $\mathcal{F}'$, denoted by $\mathrm{vol}(\mathcal{F}')$, to be the sum of $\bar{y}_i$ over all facilities $i\in\mathcal{F}'$, i.e., $\mathrm{vol}(\mathcal{F}')=\sum_{i\in\mathcal{F}'}\bar{y}_i$.

Definition 3 *(Close and distant facilities).* For a client $j\in\mathcal{C}$, we say a facility i is one of its *close facilities* if $\bar{x}_{i,j}>0$. If $\bar{x}_{i,j}=0$, but $x_{i,j}>0$, then we say i is a *distant facility* of client j. Let $\mathcal{F}^C_j$ and $\mathcal{F}^D_j$ be the sets of close and distant facilities of j, respectively. Let $\mathcal{F}_j=\mathcal{F}^C_j\cup\mathcal{F}^D_j$.

Note that if $\bar{x}_{i,j}>0$, then $x_{i,j}>0$, due to the greedy assignment of x and $\bar{x}$ variables.