Approximation algorithm for $k$-level hard capacitated facility location problem

Donglei Du∗  Xing Wang†  Dachuan Xu‡

Abstract

In this paper, we consider $k$-level hard capacitated facility location problem.

Keywords: Hard capacitated; facility location problem; approximate algorithm.

1 Introduction

In the hard capacitated facility location problem (HCFLP), we are given a set of clients $D$ and sets of facilities $F_1,F_1,...,F_k$. Each facility $i_k \in F_k$ is specified by a cost $f_{i_k}$, which incurred when facility $i_k$ is open, and by a capacity $u_{i_k}$, which is the maximum demand that facility $i$ can serve. The cost between facilities is $c_{i_1,i_2}$. Each client $j \in D$ has a demand $d_j$ that must be served by paths $p=(i_1, i_2,...,i_k)$ along with open facilities $i_1 \in F_1,..., i_k \in F_k$. The objective is to open some facilities $X_t \in F_t$ on each level $t = 1, 2,..., k$ and to connect each demand site $j \in D$ to a path so that cost of opening and connecting is minimized.

2 The $k$-HCFLP Algorithm

In this section, we give the algorithm for $k$-level hard capacitated facility location. It is based on the algorithm of 1-level hard capacitated facility location. In section 2.1, we introduce the algorithm

∗Faculty of Administration, University of New Brunswick, Canada. E-mail address: ddu@unb.ca
†Department of Applied Mathematics, Beijing University of Technology, 100 Pingleyuan, Chaoyang District, Beijing 100022, P.R. China. E-mail address: wangxing198491@emails.bjut.edu.cn.
‡Corresponding author. Department of Applied Mathematics, Beijing University of Technology, 100 Pingleyuan, Chaoyang District, Beijing 100022, P.R. China. E-mail address: xude@bjut.edu.cn.
of 1-level hard capacitated facility location problem used in ([6]). In section 2.2, we introduce our algorithm of \(k\)-level hard capacitated facility location problem.

Let \(P\) denote the set of all paths of length \(k - 1\) connecting a node in \(F_1\) to a node in \(F_k\). For a path \(p = (i_1, i_2, \ldots, i_k)\), let \(c(p) = \sum_{t=2}^{k} c_{i_{t-1}i_t}\). Let \(M\) be an instance of \(k\)-HCFLP and \(SOL\) be a solution of it. Let \(X_t\) be the facilities opened in \(SOL\) for any \(t = 1, 2, \ldots, k\). Let \(X_t(j) \subseteq X_t\) be the facilities opened in \(SOL\) to serve \(j \in D\) for any \(t = 1, 2, \ldots, k\). For our analysis it would be convenient to represent the total cost of any \(SOL\) for \(k\)-HCFLP in the split form \(F_{SOL} + C_{SOL}\), where \(F_{SOL}\) and \(C_{SOL}\) stand for the facility and connection costs, respectively. To break down \(C_{SOL}\) further, for any \(t = 2, \ldots, k\), let \(C_{t}^{SOL}\) denote the total connection cost between open facilities on level \(t - 1\) and open facilities on level \(t\). Hence \(C_{SOL} = \sum_{t=1}^{k} C_{t}^{SOL}\) where \(C_{1}^{SOL}\) stands for the total connection cost between clients and facilities on level 1. Similarly, let \(F_{t}^{SOL}\) denote the total cost to open facilities on level \(t\), and thus \(F_{SOL} = \sum_{t=1}^{k} F_{t}^{SOL}\).

For any instance \(M\) of \(k\)-HCFLP, we define an instance \(M_{k-1}\) of \((k - 1)\)-HCFLP and an instance \(S\) of 1-HCFLP in the following way:

1. \(M_{k-1}\) is obtained from \(M\) by deleting all the facilities at level 1. Thus, in \(M_{k-1}\) the set of facilities lying on level \(r\) is \(F_{r-1}\), and the connection cost between \(j \in D\) and \(i_2 \in F_2\) is

\[
\min_{v \in F_1} \{c_{jv} + c_{vi_2}\} \tag{1}
\]

2. \(S\) is obtained from \(M\) by deleting all facilities at levels greater than 1, an by doubling all the edges costs between \(D\) and \(F_1\).

2.1 The Multiexchange Local Search Algorithm

In the case \(k = 1\), we give the following program:

\[
\begin{align*}
\min & \sum_{j \in D} \sum_{i \in F} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \\
\text{s.t.} & \quad \sum_{i \in F} x_{ij} = 1, \quad \forall j \in D, \\
& \quad x_{ij} \leq y_i, \quad \forall j \in D, i \in F, \\
& \quad \sum_{j \in D} x_{ij} \leq u_i y_i, \quad \forall i \in F, \\
& \quad x_{ij} \geq 0, \quad \forall j \in D, i \in F \\
& \quad y_i \in \{0, 1\}, \quad \forall i \in F
\end{align*}
\]
To solve the instance of 1-HCELP we use the local search algorithm in ([6]). Let FEA be any feasible solution of 1-HCFLP, ZCY is the solution obtained by Multichange Local Search Algorithm in ([6]), from the following two ineualities

\[ F_{ZCY} \leq 5F_{FEA} + 4C_{FEA} \]  \hspace{1cm} (3)
\[ C_{ZCY} \leq F_{FEA} + C_{FEA} \]  \hspace{1cm} (4)

we can obtained the following lemma.

**Lemma 2.1** Let FEA be any feasible solution of 1-HCFLP, ZCY is the solution obtained by Multichange Local Search Algorithm in ([6]). Then,

\[ F_{ZCY} + C_{ZCY} \leq 6F_{FEA} + 5C_{FEA} \]

### 2.2 The k-HCFLP algorithm

Let \( x_{jp} \) denote the part that client \( j \) is assigned to path \( p \). Let \( y_{it} \) be equal to 1 if facility \( i_t \) is opened at level \( t \), and 0 otherwise. We obtain the following programming of the \( k \)-level hard capacitated facility location problem:

\[
\begin{align*}
\text{min} & \quad \sum_{p \in P} \sum_{j \in D} c_{jp}x_{jp} + \sum_{t=1}^{k} \sum_{i_t \in F_t} f_{i_t}y_{i_t} \\
\text{s.t.} & \quad \sum_{p \in P} x_{jp} = 1 \quad \forall j \in D, \\
& \quad \sum_{p:i_t \in P} x_{jp} \leq y_{i_t} \quad \forall j \in D, i_t \in p, t = 1, 2, ..., k \\
& \quad \sum_{j \in D} \sum_{p:i_t \in P} x_{jp} \leq u_{i_t}y_{i_t} \quad \forall i_t \in F_t, t = 1, 2, ..., k \\
& \quad x_{jp} \geq 0, \quad \forall j \in D, p \in P \\
& \quad y_{i_t} \in \{0, 1\}, \quad \forall i_t \in F_t, t = 1, 2, ..., k.
\end{align*}
\]

**Algorithm 2.2**

**Step 1.** Solve \( S \) by local search.

Solve \( S \) by mutiexchange local search algorithm in ([6]), let \( F_1(j) \) denote the subset of \( F_1 \) that each \( j \in D \) is assigned to in the solution of \( S \). And each \( j \in D \) is assigned to \( i_1(j) \in F_1(j) \) with \( \alpha_{ji_1(j)} \).

**Step 2.** Solve \( M_{k-1} \).
Solve $M_{k-1}$ by the $k$-level algorithm, obtain a solution for $M_{k-1}$. Let $P_{(k-1)}(j)$ denote the set of paths that each $j \in D$ is assigned to in the solution of $M_{k-1}$, then $P_{(k-1)}(j) = \{(i_2(j),...,i_k(j))\}$ such that $i_2(j) \in F_2(j),...,i_k(j) \in F_k(j)$. Let $F_l$ denote the subset of $F_1$ that is opened to serve each $j \in D$ in the solution of $M_{k-1}$. Each $j \in D$ is assigned to path $p_{(k-1)}(j) = (i_2(j),...,i_k(j))$ with $\beta_{jP_{(k-1)}(j)}$.

Step 3. Construct a transportation problem for each $j \in D$.

In the transportation problem for each $j \in D$:

- $F_1(j)$ is the set of supply center, $P_{(k-1)}(j)$ is a set of receiving center
- $\alpha_{j,i_1(j)}$ : the supply for each $i_j(j) \in F_1(j)$
- $\beta_{jP_{(k-1)}(j)}$ : the demand for each $p_{(k-1)}(j) \in P_{(k-1)}(j)$
- $c_{i_1(j)p_{(k-1)}(j)}$ : the connection cost between $i_1(j) \in F_1(j)$ and $p_{(k-1)}(j) \in P_{(k-1)}(j)$ equals to $c_{i_1(j)p_{(k-1)}(j)}$.

Solve the transportation problem for each $j \in D$ in the solution $j \in D$ is assigned to $p_{(k-1)}(j) \in P_{(k-1)}(j)$ with $\gamma_{i_1(j),p_{(k-1)}(j)}$.


On the basis of these solutions, we construct a solution for $M$, denoted by ALG, by connecting $j$ to the path $(i_1(j),i_2(j),...,i_k(j))$ with $\gamma_{i_1(j),p_{(k-1)}(j)}$, for each $i_1(j) \in F_1(j)$ and each $p_{(k-1)}(j) \in P_{(k-1)}(j)$. The connection cost from $j$ to the path $(i_1(j),i_2(j),...,i_k(j))$ for each $i_1(j) \in F_1(j)$ and each $p_{(k-1)}(j) \in P_{(k-1)}(j)$ is $c_{j,i_1(j)} + c(p_{(k-1)}(j))$.

3 Analysis

Theorem 3.1 Let $k \geq 2$, for any solution $\text{SOL}$ of $M$, the solution $\text{ALG}$ retrieved by Algorithm (2.2) satisfies

$$F^{\text{ALG}} + C^{\text{ALG}} \leq 6 F^{\text{SOL}} + 5(1+2(k-1))C^{\text{SOL}}$$

Proof: We proceed by induction on $k$. Let $\text{SOL}$ be any solution of $M$. In $\text{SOL}$, each $j \in D$ is assigned to $P(j) = \{(i_1(j),i_2(j),...,i_k(j)) : i_t(j) \subseteq X_t(j), t=1,2,...,k\}$ with $x_{jp(j)}$.

Observe that $\text{SOL}$ induces a solution $\text{SOLS}$ to $S$ by assigning $j$ to $i_1(j) \in X_1(j)$ with $\sum_{p(j) \in P(j), i_1(j) \in p(j)} x_{jp(j)}$. And $\text{SOL}$ induce a solution $\text{SOLS}M_{k-1}$ to $M_{k-1}$ by assigning $j$ to $(i_2(j),...,i_k(j))$ with $\sum_{p(j) \in P(j), i_2(j),...,i_k(j) \in p(j)} x_{jp(j)}$. That is, $j$ in $\text{SOLS}$ is assigned to $(i_1(j) \in X_1(j)$ with connection
cost at most \[\sum_{p(j)\in P(j): i_1(j)\in p(j)} 2c_{ji_1(j)x_jp(j)}\] and \(j\) in \(D\) is assigned to \(j\) to \((i_2(j), \ldots, i_k(j))\) with connection cost at most \[\sum_{i_1(j)\in p_1(j)} c_{jp(j)x_jp(j)}\] from the construction of connection costs. Then, we have

\[
F^{SOLS} = F^{SOL}_1, C^{SOLS} = 2C^{SOL}_1 \\
F^{SOLM_1} = F^{SOL}_2 + \cdots + F^{SOL}_k, C^{SOLS} \leq C^{SOL}_1 + C^{SOL}_2 + \cdots + C^{SOL}_k
\]

(6)

Let \(ALG\) denote the solution of Algorithm and \(ALGS\), \(ALGM_1\) denote the solution of \(S\) and \(M_{k-1}\) respectively in Algorithm.

Recall that the connection cost in \(S\) are doubled from the edge costs between \(D\) and \(F_1\) in \(M\). Hence, by lemma (2.1) and (6), we have

\[
F^{ALGS} + C^{ALGS} \leq 6F^{SOLS} + 5C^{SOLS} \\
\leq 6F^{SOL}_1 + 10C^{SOL}_1
\]

(7)

Assume now that \(k = 2\). In this case \(M_1\) is an instance of 1-HCFLP and by lemma (2.1) and the definition of \(M_1\),

\[
F^{ALGM_1} + C^{ALGM_1} \leq 6F^{SOLS} + 5C^{SOLS} \\
\leq 6F^{SOL}_2 + 10(C^{SOL}_1 + C^{SOL}_2)
\]

(8)

By the construction of \(ALG\) and the triangle inequality

\[
F^{ALG} + C^{ALG} \\
= F^{ALGS} + \frac{1}{2}C^{ALGS} + F^{ALGM_1} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} \sum_{j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} c_{j_1(i_1(j))i_2(j)}r_{j_1(i_1(j))i_2(j)} \\
\leq F^{ALGS} + \frac{1}{2}C^{ALGS} + F^{ALGM_1} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} \sum_{j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} (c_{j_1(i_1(j))} + c_{j_2(i_1(j))}r_{j_1(i_1(j))i_2(j)}) \\
= F^{ALGS} + \frac{1}{2}C^{ALGS} + F^{ALGM_1} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} \sum_{j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} c_{j_1(i_1(j))}r_{j_1(i_1(j))i_2(j)} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} c_{j_2(i_1(j))}r_{j_1(i_1(j))i_2(j)} \\
= F^{ALGS} + \frac{1}{2}C^{ALGS} + F^{ALGM_1} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} c_{j_1(i_1(j))}r_{j_1(i_1(j))i_2(j)} + \sum_{j\in D; j_1(i_1(j))\in F_1(j); i_2(j)\in F_2(j)} c_{j_2(i_1(j))}r_{j_1(i_1(j))i_2(j)} \\
= F^{ALGS} + \frac{1}{2}C^{ALGS} + F^{ALGM_1} + \frac{1}{2}C^{ALGS}
\]

and thus by (7) and (8), we have
Now assume that it is true for each number of levels smaller than \( k \). Then since \( M_{k-1} \) is an instance of \((k-1)\)-HCFLP, and by the definition of \( M_{k-1} \),

\[
F_{\text{ALG}}^{M_{k-1}} + C_{\text{ALG}}^{M_{k-1}} \\
\leq 6F_{\text{SOL}}^1 + 5(1 + 2(k - 2))(C_{1}^{\text{SOL}} + C_{2}^{\text{SOL}}) + 5(1 + 2(k - 2)) \sum_{t=3}^{k} C_{t}^{\text{SOL}}
\]  
(11)

Again, by the construction of \( ALG \) and the triangle inequality,

\[
F_{\text{ALG}}^{ALG} + C_{\text{ALG}}^{ALG} \\
\leq F_{\text{ALG}}^{ALS} + C_{\text{ALG}}^{ALS} + F_{\text{ALG}}^{M_{k-1}} + C_{\text{ALG}}^{M_{k-1}}
\]  
(12)

and thus, by (7) and (11)

\[
F_{\text{ALG}}^{ALG} + C_{\text{ALG}}^{ALG} \\
\leq 6F_{\text{SOL}}^1 + 5(1 + 2(k - 1))(C_{1}^{\text{SOL}} + C_{2}^{\text{SOL}}) + 5(1 + 2(k - 1)) \sum_{t=2}^{k} C_{t}^{\text{SOL}}
\]  
(13)

For any given instance of the \( k \)-HCFLP, if we scale the facility costs \( f_{i} \) by a factor \( \delta = (k - 3) + \sqrt{k^2 + 2k + 5} \) and apply Algorithm 2.2 to solve the scaled instance, by (3) and (4) we have \( \delta F^{ZCY} \leq 5\delta F^{FEA} + 4C^{FEA} \) and \( C^{ZCY} \leq \delta F^{FEA} + C^{FEA} \), that is

\[
F^{ZCY} + C^{ZCY} \leq (5 + \delta)F^{FEA} + (\frac{4}{3} + 1)C^{FEA}.
\]

It is as the proof of Theorem 3.1. We can obtain

\[
F_{\text{ALG}} + C_{\text{ZCY}} \leq (5 + \delta)F_{\text{SOL}} + (\frac{4}{3} + 1)(1 + 2(k - 1))C_{\text{SOL}}
\]

And then, we get the corollaries:

**Corollary 3.2** Let \( k \geq 2 \), for any small constant \( \varepsilon > 0 \), the \( k \)-level algorithm runs in polynomial time with approximation guarantee of \( k + 2 + \sqrt{k^2 + 2k + 5} + \varepsilon \)

**Corollary 3.3** Let \( k = 2 \), for any small constant \( \varepsilon > 0 \), the \( k \)-level algorithm runs in polynomial time with approximation guarantee of \( 4 + \sqrt{13} + \varepsilon \leq 7.61 \)
4 Discussions

Acknowledgements The third author’s research was supported by NSF of China (Grant no. 60773185) and Program for Beijing Excellent Talents.

References


