An Effective Algorithm for the Three-Stage Facility Location Problem on a Tree-Like Network

Edward Kh. Gimadi\textsuperscript{1,2} and Aleksandr S. Shevyakov\textsuperscript{2(2)}

\textsuperscript{1} Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia
\textsuperscript{2} Novosibirsk State University, Novosibirsk, Russia

\begin{flushleft}
\textit{gimadi@math.nsc.ru}  \\
\textit{shevash.97@gmail.com}
\end{flushleft}

\textbf{Abstract.} In this article we consider a three-level facility location problem on a tree-like network under the restriction that the transportation costs for a unit of production from one node to another is equal to the sum of the edges in the path connecting these nodes. As a result we construct an exact algorithm for this problem and prove his complexity equaled $O(nm^6)$, where $n$ is the number of the production demand points and, $m$ is an upper bound on the number of possible facility location sites of each level.

\textbf{Keywords:} Three-level facility location problem · Tree-like network · Polynomial-time algorithm

\section{Introduction}

The class of multistage facility location problems is characterized by existence of several stages of manufacturing where the raw materials are processed before the end-product arrives to the final consumer. Petroleum mining and processing, when the crude from a well first comes to an oil-processing plant and afterwards the petrol arrives to oil stations that are the final consumers in this case, is the classic example of a two-stage facility location problem. It is known that, in the general case, the facility location problem is NP-hard even in the classical one-stage variant \cite{3,4}. A fairly modern overview of the multistage facility location problems is given in \cite{9} (see also addition in \cite{8}).

Therefore, it looks appropriate for the multistage facility location problem to carry out investigations in the following two ways:

1. Search for the special cases of the problem in which constructing exact polynomial time algorithms are possible. In \cite{5,6} for the metric $k$-stage facility location problem on the path, an algorithm of time complexity $O(n^3 \sum_{r=1}^{k} m_r)$

Supported by the program of fundamental scientific researches of the SB RAS, and by the Ministry of Science and Higher Education of the Russian Federation under the 5-100 Excellence Programme.

\copyright Springer Nature Switzerland AG 2021
W. M. P. van der Aalst et al. (Eds.): AIST 2020, CCIS 1357, pp. 267–274, 2021.
https://doi.org/10.1007/978-3-030-71214-3_22
268 E. Kh. Gimadi and A. S. Shevyakov

is developed, where \( n \) is the number of the production demand points and \( m_r \) is the number of possible places for the facilities opening at stage \( r \). The algorithm is polynomial for the constant number of stages. For the metric three-stage problem on the path, in [5,6], algorithms of the time complexity \( \mathcal{O}(nm_1m_2m_3) \), and \( \mathcal{O}(nm^3) \), where \( m \) is an upper bound on the number of possible facility location points at each stage. In [7], the exact algorithm constructed with time complexity \( \mathcal{O}(nm^3) \).

2. Development of approximate algorithms with guaranteed performance. For example, in [1] for the metric \( k \)-stage facility location problem, a combinatorial algorithm was constructed with the preciseness bound 3.27, which improves drastically the previous record equal to 6 [2]. For the cases \( k = 2 \) and \( k = 3 \), the algorithms were designed with bounds 2.4211 and 2.8446 respectively. Later, for the case \( k = 2 \) in [10] with the use of the greedy algorithms techniques, some polynomial algorithm was constructed with the preciseness bound 1.77.

In this article we would like to generalize the result showed in [7] for the three level problem. At the end we build an exact algorithm for this problem and prove his complexity equaled \( O(nm^6) \), where \( n \) is the number of the production demand points and, \( m \) is an upper bound on the number of possible facility location sites of each level.

2 The General Formulation and Definitions

Let \( N = 1, \ldots, n \) be the set of the end-product demands points and \( M_r \subset N \) be the set of all possible facility location points at stage \( r, r = 1, 2, 3 \). The prices \( g_i^r \geq 0 \) are known for locating (opening) a facility of stage \( r \) of point \( i \in M_r \). For every demand point \( j \in N \), the demand volumes \( b_j \geq 0 \) and the transportation fees \( c_{ij} \geq 0 \) related to the delivery of production unit from the point \( i \) to \( j \) are given. For satisfying the demand of \( j \in N \), the necessary volume \( b_j \) of the product must pass the following way: an open facility of the 3rd level \( \rightarrow \) an open facility of the 2nd level \( \rightarrow \) an open facility of level 1 \( \rightarrow \) the point \( j \).

It is assumed that each end-product demand point and each facility point of every stage get the production only from one producer and, moreover, a facility of stage 1 gets the production from a facility of the stage 2 and this one gets the production from a facility of the stage 3. We will say below that \( i \in M_r; r = 1, 2, 3 \), participates in serving the demand point \( j \in N \) if \( i \) is used in the path of the facilities via which \( j \) gets the end-product. Note also that, at the same point, there can be the facilities of stages 1, 2 and 3 at the same time.

The goal is to choose subsets of the facility location points of each stage \( I_r \subset M_r; r = 1, 2, 3; \) and make an assignment of the chosen facilities to the demand points in such a way that the sum fees for opening all chosen facilities and transportation the production of all open facilities to the corresponding demand points would be minimized.

Let us define the general mathematical model of the three-stage location problem. It is necessary to minimize
The nodes of the network correspond to the demand points. The transportation fee matrix \((c_{ij})\) corresponds to an acyclic network \(G = (N, E)\), where \(N = \{1, \ldots, n\}\) and \(E = \{e_k \mid 1 \leq k < n\}\) are the sets of nodes and edges respectively. The nodes of the network correspond to the demand points. The transportation fee \(c_{ij}\) from node i to node j for unit of the product is defined as the sum of the 

\[
\sum_{i \in M_1} g_1^i x_i + \sum_{k \in M_2} g_k^2 y_k + \sum_{l \in M_3} g_l^3 z_l + \sum_{j \in N} b_j \sum_{l \in M_3} \sum_{k \in M_2} \sum_{i \in M_1} (c_{lk} + c_{ki} + c_{ij}) x_{lki} \rightarrow \min
\]

by the variables \(x_i, y_k, z_l, x_{lki}\) under the following restrictions

\[
\sum_{l \in M_3} \sum_{k \in M_2} \sum_{i \in M_1} x_{lki} = 1, \quad j \in N;
\]

\[
\sum_{k \in M_2} \sum_{i \in M_1} x_{lki} \leq z_l, \quad j \in N, \quad l \in M_3;
\]

\[
\sum_{l \in M_3} \sum_{i \in M_1} x_{lki} \leq y_k, \quad j \in N, \quad k \in M_2;
\]

\[
\sum_{l \in M_3} \sum_{k \in M_2} x_{lki} \leq x_i, \quad j \in N, \quad i \in M_1;
\]

where \(x_i, y_k, z_l, i \in M_1, k \in M_2, l \in M_3\) are the choice variables for the facilities of stages 1, 2 and 3 respectively (if \(x_i = 1\) then in the corresponding solution the facility \(i \in M_1\) of stage 1 is open); \(x_{lki}\) \(l \in M_3, k \in M_2, i \in M_1,\) and \(j \in N,\) are the transportation variables that define the facilities participating servicing the demand point \(j\) (if \(x_{lki} = 1\) then in the corresponding solution the demand point \(j\) gets the product from the facility \(i \in M_1\) of stage 1 which, in its turn, gets the product from the facility \(k \in M_2\) of stage 2, which, in its turn, gets the product from the facility \(l \in M_3\) of stage 3).

Define the model in terms of the assignment variables. For this some additional definitions are necessary:

\(\pi^r = (\pi_1^r, \ldots, \pi_n^r)\) is the assignment vector of the \(r\)-th stage facilities, where \(\pi_i^r\) is the index of the point from \(M_r\), where the facility of stage \(r\) servicing the demand point \(j, r = 1, 2, 3; 1 \leq j \leq n\) is located (open);

\(\pi = (\pi^1, \pi^2, \pi^3)\) is the triplet of the assignment vectors also referred to as the solution of the problem;

\(I^r(\pi) = \cup_{j \in N} \{\pi_j^r\}\) is the set of facilities of stage \(r\) which are used (open) in the solution \(\pi\), \(r = 1, 2, 3,\)

The three-stage location problem in terms of the assignment variables can be written in conciser:

\[
\sum_{i \in I^1(n)} g^1_i + \sum_{k \in I^2(\pi)} g^2_k + \sum_{l \in I^3(\pi)} g^3_l + \sum_{j \in N} b_j (c_{\pi_j^3 \pi_j^3} + c_{\pi_j^2 \pi_j^1} + c_{\pi_j^1 j}) \rightarrow \min
\]

In the paper, we consider the three-stage location problem on a tree-like network. More specifically, we consider the problems where the transportation fee matrix \((c_{ij})\) corresponds to an acyclic network \(G = (N, E)\), where \(N = \{1, \ldots, n\}\) and \(E = \{e_k \mid 1 \leq k < n\}\) are the sets of nodes and edges respectively. The nodes of the network correspond to the demand points. The transportation fee \(c_{ij}\) from node i to node j for unit of the product is defined as the sum of the
lengths of the edges in the path connecting these nodes. Note also that $c_{ij} = c_{ji}$ and $c_{ik} \leq c_{ij} + c_{jk}$ for every $i, j, k \in N$.

For each $j \in N$, denote by $N_j$ the set of descendants of node $j$ (the maximum subtree of the initial tree with the root vertex $j$), and by $I^r_j(\pi) = \cup\{\pi^r_k \mid k \in N_j\}$, the set of the $r$-th stage facilities servicing the clients of the set $N_j$ in the assignment $\pi$.

Introduce inductively the special notations $(\mu^2_j(\pi), \mu^3_j(\pi))$: for the root of the tree $j = 1$, let

$$
(\mu^2_1(\pi), \mu^3_1(\pi)) = \arg\min\{c_{1k} + c_{kl} \mid k \in I^2(\pi), l \in I^3(\pi)\}
$$

and, for the node $j, 1 < j \leq n$, let

$$(\mu^2_j(\pi), \mu^3_j(\pi)) = \begin{cases} (\mu^2_i(\pi), \mu^3_i(\pi)), & \text{where } i \text{ is the father of } j, \text{ if } c_{j\mu^2_i(\pi)} + c_{\mu^2_i(\pi)\mu^3_i(\pi)} \\ \arg\min\{c_{jk} + c_{kl} \mid k \in I^2(\pi), l \in I^3(\pi)\}; & \text{otherwise.} \end{cases}
$$

Also introduce the special notations $\zeta_j(\pi)$, for the facility of stage 3 closest to node $j \in N$ used in the assignment $\pi$ (note that this facility must be open): for the root of the tree $j = 1$, let

$$
\zeta_1(\pi) = \arg\min\{c_{1k} \mid k \in I^3(\pi)\}
$$

and, for the node $j, 1 < j \leq n$, let

$$
\zeta_j(\pi) = \begin{cases} \zeta_i(\pi), & \text{where } i \text{ is the father of } j, \text{ if } c_{j\zeta_i(\pi)} = \min\{c_{jk} \mid k \in I^3(\pi)\} \\ \arg\min\{c_{jk} \mid k \in I^3(\pi)\}, & \text{otherwise.} \end{cases}
$$

Sometimes for the convenience we will omit $\pi$ in the notations of $\mu^r_j(\pi), \zeta_j(\pi)$ and simply write $\mu^r, \zeta, r = 2, 3$ if it is clear what assignment is implied.

**Remark 1.** If node $j$ is a son of node $i$ then $\mu^2_j \in N_i \cup \mu^2_i$

**Remark 2.** If $\mu^2_i = \mu^2_j$, then $\mu^3_i = \mu^3_j$. For each $i, j \in N$

**Remark 3.** If node $j$ is a son of node $i$ then $\zeta_j \in N_i \cup \zeta_i$

**Remark 4.** If node $j$ is a son of node $i$ then $\mu^3_j \in N_i \cup \mu^3_i \cup \zeta_i$

Introduce some additional definitions. Let some solution $\pi$ of the three-stage location problem on the network be given. For each node $j$, call the nodes $k \in N_j$ for which

$$
\pi^1_k \notin N_j \cup \{\pi^1_j\}
$$

bad nodes of stage 1. By the bad nodes of stage 2 for $j$ we call the nodes $k \in N_j$ for which

$$
\pi^2_k \notin N_j \cup \{\pi^2_j\} \cup \mu^2_j
$$
And by the bad nodes of stage 3 for \( j \) we call the nodes \( k \in N_j \) for which

\[ \pi^3_k \notin N_j \cup \{ \pi^3_j \} \cup \mu^3_j \cup \zeta_j \]

The nodes for which the opposite inclusions are true are called the good nodes of stage 1 for \( j \), the good nodes of stage 2 for \( j \) and the good nodes of stage 3 for \( j \) respectively. Call node \( k \) bad for \( j \) if it is a bad node of stage 1, 2 or 3 for \( j \). A node \( k \) is called bad if it is bad for some vertex of the tree. A node \( k \) is called good vertex for \( j \) if it is a good node of stage 1, 2 and 3 for \( j \) respectively. Call node \( k \) bad for \( j \) if it is a bad node of stage 1, 2 or 3 for \( j \). A node \( k \) is called bad if it is bad for some vertex of the tree. A node \( k \) is called good if it is good for all vertices of the tree. Denote the number of bad nodes of stages 1, 2 and 3 for a vertex \( j \) by \( \nu^1_j \), \( \nu^2_j \) and \( \nu^3_j \) respectively.

\[ \nu_j(\pi) = \nu^1_j(\pi) + \nu^2_j(\pi) + \nu^3_j(\pi) \]

is called the index of a solution \( \pi \).

### 3 Main Statements

**Lemma 1.** Given an optimal solution \( \pi \), if a node \( k \) is good for \( j \) and \( j \) is good for \( i \) then \( k \) is good for \( i \).

Now we prove the fundamental property of the optimal solutions of the problem. This property will be used for constructing some algorithm for searching optimal solution.

**Theorem 1.** There is an optimal solution of the two-stage location problem on the network in which, for every node \( t \in N \), Properties (I1) are satisfied:

\[ I^1_t(\pi) \subset N_t \cup \{ \pi^1_t \} \]

\[ I^2_t(\pi) \subset N_t \cup \{ \pi^2_t \} \cup \mu^2_t(\pi) \]

\[ I^3_t(\pi) \subset N_t \cup \{ \pi^3_t \} \cup \mu^3_t(\pi) \cup \zeta_t(\pi) \]

Using the definitions above, we can reformulate the statement of the theorem as follows: There is an optimal solution of the three-stage location problem on the network with zero index; i.e., there is an optimal solution \( \pi \) for which \( \nu(\pi) = 0 \).

### 4 Description of the Algorithm

Denote the initial three-stage location problem by \( M = \langle M_1, M_2, M_3; N \rangle \). Consider the family of the subproblems

\[ M_j(i, k, k', l, l', l'') = \{ (M_1, M_2, M_3; N_j \mid \pi^1_j = i, \pi^2_j = k, \mu^3_j(\pi) = k', \pi^3_j = l, \mu^3_j(\pi) = l', \zeta_j(\pi) = l'' \}, \]

\[ i \in M_1, \ k, k' \in M_2, \ l, l', l'' \in M_3, \ 1 \leq j \leq n \}. \]
Let \( F_j(i, k, k', l, l', l'') \) denote the optimal solution of Problem \( M_j(i, k, k', l, l', l'') \) satisfying Property (I1). Introduce the additional notations:

\[
F_j(k, k', l, l', l'') = \min_{i \in M_1} F_j(i, k, k', l, l', l''); \quad F_j(k, l, l', l'') = \min_{k' \in M_2} F_j(k, k', l, l', l''); \\
F_j(l, l', l'') = \min_{k' \in M_2} F_j(k, l, l', l''); \quad F_j(l', l'') = \min_{l' \in M_3} F_j(l, l', l''); \\
F_j(l'') = \min_{l'' \in M_3} F_j(l'').
\]

Note that, by Theorem 1, the optimal solution of the initial problem \( M = \langle M_1, M_2, M_3; N \rangle \) (denote it by \( F^* \)) coincides with \( F_1 \). Let

\[
g_{kk'} = g^1_k + g^2_{kk'}, k \neq k'; \quad g_{kk} = g^3_k; r = 2, 3; \\
g^3_{ll'} = g^3_l + g^3_{ll} + g^3_{ll}, l \neq l', l'' \neq l' \neq l''; \quad g^3_{ll''} = g^3_{ll''}; \\
G_{j(i, k, k', l, l', l'')}^{\beta_u \ldots \beta_v} = \min_{P \setminus \{\beta_u, \ldots, \beta_v\} \subset N_j} F_j(i, k, k', l, l', l''), \quad \{\beta_u, \ldots, \beta_v\} \subset \{i, k, k', l, l'', l''\} \Rightarrow \beta_u \neq \beta_v \text{ if } u \neq v.
\]

Also introduce the set \( \Gamma \), which contains all subsets of \( \{i, k, k', l, l', l''\} \) except all subsets, which contains the element \( i \), but doesn’t contain the element \( k \), as well as all subsets, which contains the element \( k \), but doesn’t contain the element \( i \). Obviously that \( \Gamma \) contains 32 elements.

The recurrence for counting \( F_j(i, k, k', l, l', l'') \) yields

**Theorem 2.** For all \( j \in N, i \in M_1, k, k' \in M_2 \) and \( l, l', l'' \in M_3 \) we have

\[
F_j(i, k, k', l, l', l'') = g^1_i + g^2_{kk'} + g^3_{ll''} + b_2(c_k + c_{ki} + c_{ij}) \\
+ \sum_{t \in S_j} \min \left\{ G^\beta_{j(i, k, k', l, l', l'')}^{\beta_u \ldots \beta_v} - \sum_{\beta \in \beta_u \ldots \beta_v} r_{\beta}^{\beta} \mid (\beta_u, \ldots, \beta_v) \in \Gamma \right\},
\]

where

\[
r_{\beta} = \begin{cases} 
1, & \text{if } \beta = i, \\
2, & \text{if } \beta = k, k', \\
3, & \text{if } \beta = l, l', l''.
\end{cases}
\]

(Here and below, \( S_j \) is the set of the descendants of node \( j \)).

This theorem allows us to construct an exact algorithm solving initial problem.
5 The Algorithms

The recurrences of Theorem 2 imply Algorithm $A$ for solving the problem. It consists of $h$ steps, where $h$ is the height of the initial tree $N$.

Algorithm $A$

**Step** $s$, $1 \leq s < h$. For each node $j$, that is at distance $h - s + 1$ from the root of the tree, and for each vertices $i \in M_1, k, k' \in M_2$ and $l, l' \in M_3$ calculate $F_j(i, k, k', l, l', l'')$ and $\{G^\beta_u, ..., \beta_v | (\beta_u, ..., \beta_v) \in \Gamma\}$

**step** $h$ For each nodes $i \in M_1, k, k' \in M_2, l, l', l'' \in M_3$ we calculate $F_1(i, k, k', l, l', l'')$. After find the value $F_1$ coinciding with the optimal solution of the problem.

Using Algorithm $A$, we can find the optimum value of the aim function of the initial problem. For finding the optimal permutation $\pi$ we need the inverse Algorithm $\tilde{A}$ that also follows from Theorem 2.

**Theorem 3.** The time complexity of Algorithms $A$ and $\tilde{A}$ is at most $O(nm^6)$.

6 Conclusion

The article considered the three-stage facility location problem on an acyclic network. The transportation fee from node i to node j for unit of the product is defined as the sum of the lengths of the edges in the path connecting these nodes.

Using the formulation the problem in terms of the assignment variables and the useful structure properties of optimal solutions, it was possible to construct a polynomial-time exact algorithm for solving the problem using the dynamic programming technique. The time complexity of Algorithm is at most $O(nm^6)$. Thus, the proposed technique solves the problem in linear time.

However, for the further research it would be interesting to improve the achieved estimate of time complexity depending on the number of facilities. In addition, the intriguing question remains about the polynomial time solvability of the $k$-Stage Facility Location Problem on a tree-like networks with arbitrary number of stages.

References

1. Ageev, A., Ye, Y., Zhan, J.: Improved combinatorial approximation algorithms for the k-Level facility location problem. SIAM J. Discrete Math. 18(1), 207–217 (2004)
2. Bumb, A., Kern, W.: A simple dual ascent algorithm for the multilevel facility location problem. In: Goemans, M., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) APPROX/RANDOM-2001. LNCS, vol. 2129, pp. 55–63. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44666-4_10
3. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979)
4. Mirchandani, P., Francis, R. (eds.): Discrete Location Theory. Wiley, New York (1990)
5. Gimad, E.Kh.: Effective algorithms for multistage location problem on a path. Diskret. Anal. Issled. Oper. Ser. 1 2(4), 13–31 (1995). (in Russian)
6. Gimadi, E.Kh.: Exact algorithm for some multi-level location problems on a chain and a tree. In: Operation Research Proceedings of SOR 1996, Braunschweig, Germany, 1996, pp. 72–77. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-60744-8_14
7. Gimadi, E.Kh, Kurochkin, A.A.: An effective algorithm for the two-stage location problem on a tree-like network. J. Appl. Industr. Math. 7(2), 177–186 (2013). https://doi.org/10.1134/S1990478913020063
8. Gimadi, E.Kh., Shamardin, Y.V.: On multi-level network facility location problem. In: CEUR Workshop Proceedings, 2018, vol. 2098, pp. 150–158 (2018). EID: 2-s2.0-85047982363
9. Ortiz-Astorquiza, C., Contreras, I., Laporte, G.: Multi-level facility location problems. Eur. J. Oper. Res. 1–15 (2017)
10. Zhang, J.: Approximating the two-level facility location problem via a quasi-greedy approach. In: Proceedings of 15th ACM-SIAM Symposium on Discrete Algorithms SODA, New Orleans, Louisiana, USA, 11–14 January 2004, pp. 808–817. SIAM, Philadelphia (2004)