

ORIGINAL RESEARCH

Open Access



An algorithm for solving the bi-objective median path-shaped facility on a tree network

Abdallah Aboutahoun^{1,2} and Fatma El-Safty^{3*}

*Correspondence:

f_elsafty87@sci.dmu.edu.eg

³Faculty of Science, Damanhour University, Damanhour, Egypt
Full list of author information is available at the end of the article

Abstract

In this paper, an algorithm for the bi-objective median path (BMP) problem on a tree network is considered. The algorithm is based on the two-phase method which can compute all Pareto solutions for the BMP problem. The first phase is applied to find supported Pareto solutions by solving the uni-objective problem $D(P, \lambda)$. The second phase is used to compute the unsupported Pareto solutions by applying a k -best algorithm which computes the k -best Pareto solutions in order of their objective values.

Keywords: Two-phase method, Pareto solution, Median path, k -best algorithm

2010 Mathematics Subject Classification: 90B10; 90B50; 90B85; 05C82

Introduction

The problem of locating a path on a tree network has been receiving a lot of attention in the literature since it has many applications including the design of pipe lines, public transit line, and railroad lines. Most of papers study the problem of locating a path on a tree which minimizes either the sum of weighted distances from all vertices to the path (median criterion) or minimizes the maximum weighted distance from the vertices to the path (the center criterion). The path which satisfies the first criterion is called a median path [4, 5, 19, 20, 22, 34], and the path which satisfies the second criterion is called a central path [1, 2, 7, 28, 33]. The median criterion is used to reduce the total cost of the service systems. On the other hand, the center criterion is used to serve the clients who are located far away from the facility. Becker et al. [6] studied the problem of finding a path on a tree by considering both the median and the center criterion. They solved two problems, and each one was about finding a path of bounded length which minimizes one criterion with a restriction on the other criterion. They gave $O(n \log^2 n)$ algorithms for both problems.

In real world optimization problems, there is usually more than one objective that has to be optimized, thus leading to multi-objective optimization problems. In general, no single solution exists that optimizes all objective functions simultaneously and they are conflicting. The optimal solution concept is replaced with efficient (Pareto) solution. Several authors studied multi-objective optimization problems (see, e.g., [11, 24, 25, 29]).

The location problem is concerned with the selection of the best set of facility sites on a network to supply a set of costumers. The objective is to locate a number of facilities

which satisfies some criteria. Traditionally, these facilities (single facilities) are located at vertex or along an edge of the network [14, 26]. However, some authors extended the concept of the location problem to include sites which can be represented by subtrees or paths (extensive facilities) of the underlying network. Averbakh and Berman [3] introduced the problem of locating a path on a tree network which minimizes both the center and the median criteria. Tamir et al. [31] described an $O(n \log n)$ algorithm for siting the optimal location of a tree-shaped facility in a tree network, using a combination of the center criterion and the median criterion. Puerto et al. [23] considered the problem introduced in [3] with an additional constraint, namely that the Pareto path must be of bounded length, whereas this article concentrates on the bi-objective median path (BMP) of a tree. The aim of the BMP problem is to find the set of all Pareto solutions. A two-phase approach is proposed to solve such problem. In the first phase, supported Pareto solutions are computed with a parametric median path algorithm. The other Pareto solutions which are unsupported are computed in the second phase using a ranking algorithm applied on restricted areas of the objective space.

Puerto et al. [23] considered the problem of locating a path-shaped facility with bounded length on a doubly weighted tree network. They used a bi-objective model where each vertex is associated with a pair of non-negative weights, namely, the center weight and the median weight. The aim of their work focused on determining all the non-dominated points in the objective space. Their approach can be viewed as an application of the well-known ϵ -constraint approach. They provided an $O(n \log n)$ algorithm to find the set of different ϵ values \mathcal{E} and proved that the cardinality of \mathcal{E} is less than or equal $2n$. Therefore they applied at most $2n$ cent-dian models and each cent-dian model needs $O(n \log n)$ time to be solved. Following the same approach presented in [23] to solve the (BMP) problem leads to generating the set \mathcal{E} with $|\mathcal{E}| \leq n(n+1)/2$. Hence, the restricted models which give the non-dominated points will be nearly run $n(n+1)/2$ times and each one of them needs $O(n \log n)$ time, whereas the proposed approach based on the two-phase technique reduces the number of running times to $3r + r' - 1$, where r is the number of the supported non-dominated points and r' is the number of unsupported non-dominated points. These results can be summarized in Table 1.

In this paper, a new bi-objective median path-shaped facility location problem is studied.

- For this purpose, an algorithm of time complexity $O(n \log n)$ is presented to locate a median path with bounded length of an n -tree network.
- Pareto solutions are categorized into two types, namely, supported and unsupported.
 - An $O(n \log n)$ time algorithm is given to compute the supported Pareto solutions. This algorithm based on the weighted scalarization method.
 - An $O(n \log n)$ time algorithm is given to compute the unsupported Pareto solutions. This algorithm is based on the k -best method.

The remainder of this paper is organized as follows. Related works are presented in “[Related works](#)” section. “[Notations and definitions](#)” section introduces the definitions and the notations of the (BMP) problem. In “[Median path of a tree](#)” section, an $O(n \log n)$ median path algorithm is mentioned. The “[Two-phase method to solve BMP problem](#)” section is devoted to the algorithm that solves the BMP problem. An illustrative example is given in the “[An illustrative example](#)” section. A case study is

Table 1 Summary of results

	Bi-objective location problem Algorithm proposed in [23]	Algorithm proposed in this paper
Model	$\min_{P \in \mathcal{P}} (M^1(P), D^2(P))$, where $M^1(P) = \max_{v_j \in V} w_j^1 d(v_j, P)$, $D^2(P) = \sum_{v_j \in V} w_j^2 d(v_j, P)$	$\min_{P \in \mathcal{P}} (D^1(P), D^2(P))$, where $D^1(P) = \sum_{v_j \in V} w_j^1 d(v_j, P)$, $D^2(P) = \sum_{v_j \in V} w_j^2 d(v_j, P)$
Used approach	ϵ -constraint method	Two-phase method
Subproblems	Cent-dian problem of the form $\min_{P \in \mathcal{P}} D^2(P)$ s.t. $M^1(P) = \epsilon$	For supported Pareto paths \mathcal{P}_{SE} $\min_{P \in \mathcal{P}} \lambda D^1(P) + (1 - \lambda) D^2(P)$ For unsupported Pareto paths $\min_{P \in \mathcal{P} \setminus \mathcal{P}_{SE}} \lambda D^1(P) + (1 - \lambda) D^2(P)$
Complexity	$O(n \log n)$	$O(n \log n)$

stated in the “Case study” section. The final conclusion of this paper is in the “Conclusion” section.

Related works

Although researchers have paid much attention to the location problems, surprisingly, certain generalization of these problems, which take into account various real-life considerations, has not been studied thoroughly. Hamacher et al. [14] addressed the multi-criteria median problem taking into account several weights on the vertex. Ramos et al. [26] studied the problem of locating a single facility on a network in the presence of several median objectives. Colebrook and Sicilia [9] addressed the λ -cent-dian problem on a multi-objective network $G = (V, E)$, $|V| = n$, $|E| = m$. They developed a polynomial approach in $O(m^2 n^2 k^2)$ time that solves this problem for a given value of λ , where k is the number of the objective functions. They also generalized the model to problem such as the multi-objective center problem and the multi-objective median problem. Hassin et al. [15] considered the problem of locating facilities on the vertices of a network to maximize the expected demand service and gave two polynomial time exact algorithms. Also, hubs are particular facilities which serve as mediators through aggregation, classification, and distribution of network flows from the origin to a destination. Kahag et al. [16] modeled a multi-objective hub-location-allocation problem and formulated the problem as a constrained bi-objective optimization model to minimize the total costs as well as minimizing the total system time. Ghezavati and Hosseinifar [12] considered a hub location problem where the weights of the vertices are stochastic. They first proposed a bi-objective location problem with value at risk criterion that optimizes the opportunity lost by maximizing the probability that total demands arriving at a facility to be less than the capacity.

Notations and definitions

Let $T = (V, E)$ be a tree network, with $|V| = n$. Each edge $e = (v, u) \in E$ has a positive length $l(e)$. Let the pair of non-negative weights (w_i^1, w_i^2) associate to each vertex $v_i \in$

V representing two median weights corresponding to two median criteria. Let P be a discrete path in T , that is, its endpoints are restricted to be vertices. The length of P is $L(P) = \sum_{e \in P} l(e)$. Let \mathcal{P} be the set of discrete bounded paths in T , that is, $\mathcal{P} = \{P \subset T : P \text{ is discrete and } L(P) \leq l\}$. For any two vertices $v, u \in V$, the distance $d(v, u)$ between v and u is the length of the path $P_{v,u}$ connecting them. Define the distance between a vertex $v \in V$ and a path $P \in \mathcal{P}$ by $d(v, P) = \min_{u \in P} d(v, u)$. For any path $P \in \mathcal{P}$, the sum of weighted distances from P to all vertices $v \in V$ is $D^r(P) = \sum_{v_i \in V} w_i^r d(v_i, P)$, $r = 1, 2$. A median path P with respect to the first or the second criterion is the path which minimizes $D^r(P)$, $r \in 1, 2$. Let D be a two-dimensional function which maps each path $P \in \mathcal{P}$ into a point in the plane \mathbb{R}^2 such that

$$D : \mathcal{P} \rightarrow \mathbb{R}^2$$

$$P \rightarrow (D^1(P), D^2(P)) \tag{1}$$

In this article, all paths $P \in \mathcal{P}$ which minimize the two-dimensional function D are found, so the bi-objective median path BMP problem can be formulated as

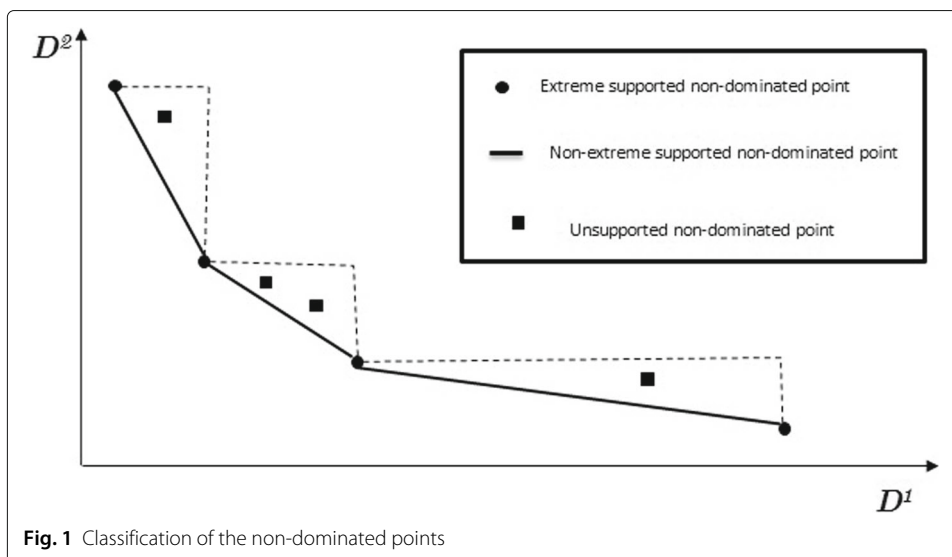
$$\min_{P \in \mathcal{P}} D(P) = \begin{cases} D^1(P) = \sum_{v_j \in V} w_j^1 d(v_j, P) \\ D^2(P) = \sum_{v_j \in V} w_j^2 d(v_j, P) \end{cases} \tag{2}$$

Some necessary concepts from multi-objective optimization [8, 10] will be introduced. The set \mathcal{P} is called the decision (feasible) space and the set $D(\mathcal{P}) = \{(D^1(P), D^2(P)) : P \in \mathcal{P}\}$ is called the objective space.

Definition 1 A feasible path $P \in \mathcal{P}$ is said to be Pareto solution for problem (2) if there is no path $P' \in \mathcal{P}$ such that $D^1(P') \leq D^1(P)$ and $D^2(P') \leq D^2(P)$ with at least one strict inequality. The image $D(P) = (D^1(P), D^2(P))$ is called non-dominated point (outcome).

Let \mathcal{P}_E and \mathcal{D}_N be the set of Pareto solutions and the set of non-dominated points, respectively. The set \mathcal{D}_N can be viewed graphically as depicted in Fig. 1.

The set of Pareto solutions can be divided into two subsets as follows:



- Supported solutions \mathcal{P}_{SE} which are the optimal solutions of the problem:

$$\min_{P \in \mathcal{P}} \lambda D^1(P) + (1 - \lambda) D^2(P), \tag{3}$$

for $0 \leq \lambda \leq 1$. All the supported non-dominated points \mathcal{D}_{SN} are lying on the boundary of the convex hull of $D(\mathcal{P})$.

The supported solutions \mathcal{P}_{SE} are partitioned in two types. Extreme supported solution $P \in \mathcal{P}_{SE}$ whose image $D(P) \in \mathcal{D}_{SN}$ is located on a vertex of the convex hull of $D(\mathcal{P})$ and non-extreme supported solution $P \in \mathcal{P}_{SE}$ whose image $D(P) \in \mathcal{D}_{SN}$ is located on the interior of an edge of the convex hull of $D(\mathcal{P})$ (see Fig. 1).

- Unsupported solutions $\mathcal{P}_{UE} = \mathcal{P}_E \setminus \mathcal{P}_{SE}$ which are not optimal solutions of problem (3) and their image \mathcal{D}_{UN} is lying in the interior of triangle constructed by marking out dominating area of two consecutive extreme supported points (see Fig. 1).

Median path of a tree

Several authors studied the problem of finding a median path [4, 20, 22] in a tree network where each vertex has only one weight. Becker et al. [5] provided an algorithm for a median path with bounded length in weighted tree with time complexity $O(n \log^2 n)$. Tamir et al. [30] gave an $O(n \log n)$ algorithm for locating a median path in weighted tree with length constraint. Here, an $O(n \log n)$ algorithm for finding a median path of a weighted tree with bounded length is mentioned. The proposed algorithm starts with a decomposition of the tree T into sequence of subtrees. This decomposition depends on the definition of a middle vertex of a tree.

Definition 2 [5, 17] *Given a weighted tree T , a middle vertex m of T is a vertex which minimizes the maximum of the number of vertices of the subtrees obtained by removing m .*

Remark 1 *A middle vertex has maximum subtree cardinality less than or equal to $\frac{n}{2}$. Computing the middle vertex requires $O(n)$ time as presented in [13].*

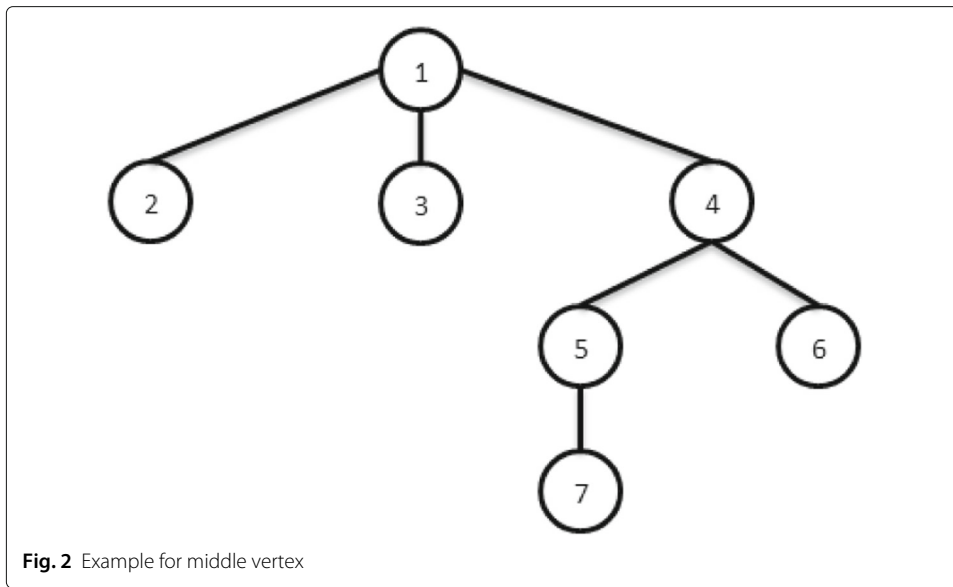
In Fig. 2, there is a tree T consisting of seven vertices. The vertex v_4 is considered as the middle vertex of T since removing v_4 (see Fig. 3) leads to three subtrees with maximum cardinality $3 \leq \frac{7}{2}$.

The median path algorithm

Let $T = (V, E)$ be an n -tree. Each vertex $v \in V$ has only one non-negative weight w_v , and each edge $e \in E$ is assigned to a positive length $l(e)$. In this case, there is only one sum of weighted distances function \mathbb{D} and it is defined as

$$\mathbb{D}(P) = \sum_{v \in V} w_v d(v, P). \tag{4}$$

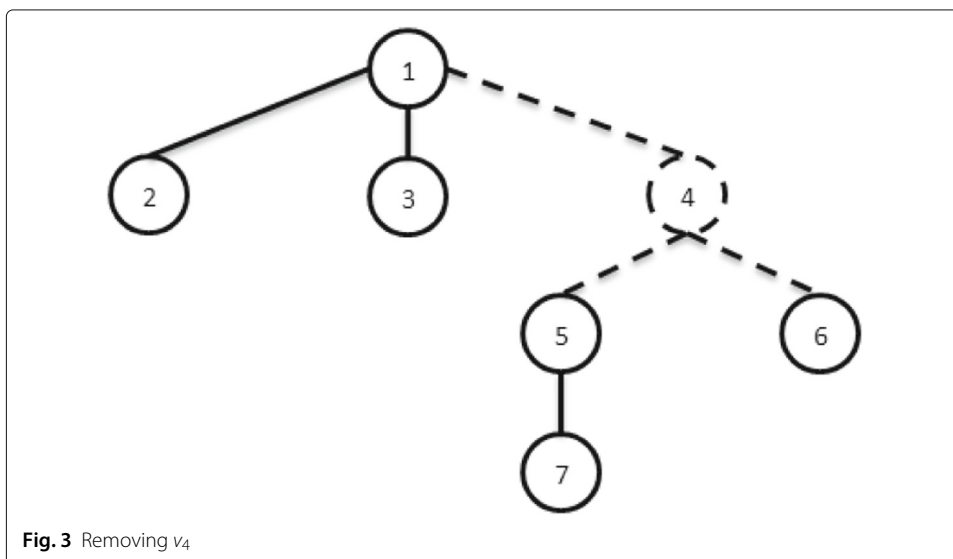
Running a middle decomposition of the tree T leads to sequence of subtrees. For each subtree T' in this decomposition, the distances from its middle, say m' to all other vertices in T' are computed and sorted. This occurs in $O(n \log n)$ total time. Let T be rooted at its middle vertex m , and denote by T_m the rooted tree. Let $f(v)$ be the father of v in T_m , $\text{Son}(v)$ be the set of sons of v . Each vertex $v \in V$ splits the tree T_m into two subtrees, $T_v^B = (V_v^B, E_v^B)$ and $T_v^U = (V_v^U, E_v^U)$, where T_v^B denotes the below subtree of T_m rooted at vertex v and T_v^U denotes the upper subtree of T_m which was induced by $(V - V_v^B) \cup \{v\}$, that is $V_v^B \cup V_v^U = V$ and $V_v^B \cap V_v^U = \{v\}$ (see Fig. 4). The tree T_m can be decomposed

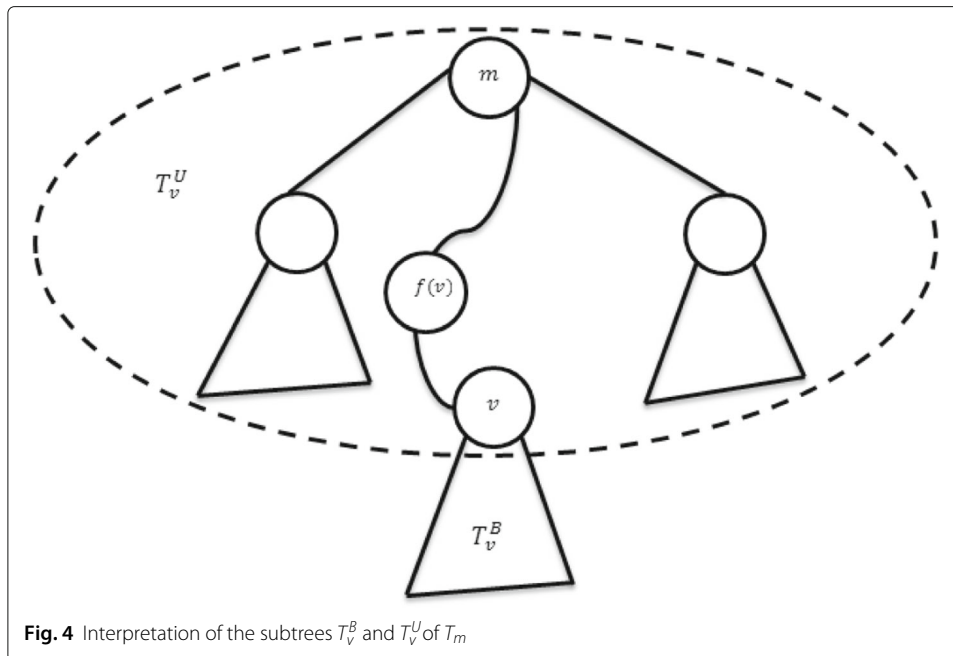


into two subtrees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ such that $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \{m\}$, $E_1 \cup E_2 = E$, and $|V_1| = n_1$, $|V_2| = n_2$. Let $\text{Son}(1, m)$ and $\text{Son}(2, m)$ be the sets of the sons of the middle vertex m in T_1 and T_2 , respectively. Denote by $\text{Sum}_B(v)$ and $\mathbb{D}_B(v)$ the sum of weights of vertices in T_v^B and the sum of weighted distances of the vertices in T_v^B to the vertex v , respectively, see [5]. By scanning from the leaves to the root, $\text{Sum}_B(v)$ and $\mathbb{D}_B(v)$ are computed in $O(n)$ time as follows:

$$\text{Sum}_B(v) = \begin{cases} w_v & \text{if } v \text{ is a leaf in } T_m, \\ w_v + \sum_{u \in \text{Son}(v)} \text{Sum}_B(u) & \text{otherwise.} \end{cases} \quad (5)$$

$$\mathbb{D}_B(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf in } T_m, \\ \sum_{u \in \text{Son}(v)} \mathbb{D}_B(u) + \text{Sum}_B(u)l(v, u) & \text{otherwise.} \end{cases} \quad (6)$$





The following quantities will be needed:

$$\mathbb{D}^r(m) = \sum_{u \in \text{Son}(r,m)} \mathbb{D}_B(u) + \text{Sum}_B(u)l(m, u), \quad r = 1, 2. \tag{7}$$

Definition 3 [5] Given two paths P_{v_i,v_j} and P_{v_j,v_k} with edge disjoint, the distance saving of P_{v_j,v_k} with respect to P_{v_i,v_j} , is the reduction of the sum of weighted distances obtained by adding P_{v_j,v_k} to P_{v_i,v_j} , that is,

$$\text{sav}(P_{v_i,v_j}, P_{v_j,v_k}) = \mathbb{D}(P_{v_i,v_j}) - \mathbb{D}(P_{v_i,v_k}). \tag{8}$$

Remark 2 If the first path consists of only one vertex v_i , write $\text{sav}(v_i, P_{v_i,v_k})$ instead of $\text{sav}(P_{v_i,v_i}, P_{v_i,v_k})$. In linear time, $\text{sav}(v_i, P_{v_i,v_k})$ is computed as follows:

$$\text{sav}(v_i, P_{v_i,v_k}) = \begin{cases} 0 & \text{if } v_k = v_i, \\ \text{sav}(v_i, P_{v_i,f(v_k)}) + \text{Sum}_B(v_k)l(v_k, f(v_k)) & \text{otherwise.} \end{cases} \tag{9}$$

Also, calculate $\forall v_k \in V_r, r = 1, 2$, the following quantities in linear time.

$$\mathbb{D}(r, v_k) = \mathbb{D}^r(m) - \text{sav}(m, P_{m,v_k}), \quad v_k \in T_r. \tag{10}$$

Note that $\mathbb{D}(r, v_k)$ is the sum of weighted distances of all vertices in T_r from the path P_{m,v_k} . Let $\mathcal{V}^1 = (v_{i(1)}, v_{i(2)}, \dots, v_{i(n_1)})$ and $\mathcal{V}^2 = (v_{q(1)}, v_{q(2)}, \dots, v_{q(n_2)})$ be the non-decreasing ordering of the vertices in V_1 and V_2 by their distances from the middle vertex m respectively. The best paths of length at most l which are included in $T_1 \setminus \{m\}$ and $T_2 \setminus \{m\}$ may be found by computing for each vertex $v_k \in \mathcal{V}^1$, the best path of length bounded above by l which has v_k as an endpoint. Starting with $v_{i(n_1)}$, if $d(m, v_{i(n_1)}) > l$, there is no such path. Otherwise, find the largest index $t = t(n_1)$ such that $d(v_{i(n_1)}, v_{q(t)}) \leq l$ and

$d(v_{i(n_1)}, v_{q(t+1)}) > l$. Let $v_Q, Q \in \mathcal{Q} = \{q(1), q(2), \dots, q(t(n_1))\}$ be a vertex which gives $\min_{s=1, \dots, t(n_1)} \mathbb{D}(2, v_{q(s)})$ and put

$$\alpha_{i(n_1)} = \mathbb{D}(1, v_{i(n_1)}) + \mathbb{D}(2, v_Q), \tag{11}$$

where $\alpha_{i(n_1)}$ is the value of the sum of the weighted distances of the best path which has $v_{i(n_1)}$ as an endpoint. Continuing on with $v_{i(n_1-1)}, \dots, v_{i(1)}$, in linear time all terms $\alpha_{i(r)}, r = n_1, \dots, 1$ are evaluated. The value of the median function of the best path of length at most l which passes through the middle vertex m and contains vertices in both $V_1 - \{m\}$ and $V_2 - \{m\}$ is given by

$$\min_{r=n_1, \dots, 1} \alpha_{i(r)}.$$

Suppose that $v_I, I \in \mathcal{I} = \{i(1), i(2), \dots, i(n_1)\}$ is a vertex where $\min_{r=n_1, \dots, 1} \alpha_{i(r)}$ takes place therefore the path P_{v_I, v_Q} is passing through the middle vertex m and minimizing the median function with $\mathbb{D}(P_{v_I, v_Q}) = \min_{r=n_1, \dots, 1} \alpha_{i(r)}$.

To find the best path in T_1 , replace the weight of m by $\sum_{v \in V_2} w_v$, delete all vertices in $V_2 - \{m\}$ from T , and solve the problem in the resulting subtree T_1 and add the constant $\mathbb{D}^2(m) = \sum_{v_t \in V^2} w_t d(v_t, m)$ to the value of the median function. By similar technique, the best path in T_2 can be located. For simplicity, define i' as (to be used later):

$$i' = \begin{cases} 1 & \text{if } i = 2, \\ 2 & \text{if } i = 1. \end{cases} \tag{12}$$

Algorithm 1. Median Path

Input: a weighted tree $T = (V, E)$
Output: a feasible path P^* minimizes the median function with $\mathbb{D}^* = \mathbb{D}(P^*)$
 let $\mathbb{D}^* = \infty, k = 1$, and $\mathbb{D}_{\text{root}}(k) = 0$
 $(P, \alpha) = \text{BestPath}(T, k)$
 set $\mathbb{D} = \alpha + \mathbb{D}_{\text{root}}(k)$
while $\mathbb{D} < \mathbb{D}^*$ **do**
 $P^* = P, \mathbb{D}^* = \mathbb{D}$
 for $i = 1, 2$ **do**
 while $\mathbb{D}^i(m(k)) < \mathbb{D}^*$ **do**
 $w_{m(k)} = \sum_{v \in V_i} w_v$, delete vertices of $V_i - \{m(k)\}$
 set $k = k + 1, \mathbb{D}_{\text{root}}(k) = \mathbb{D}_{\text{root}}(k - 1) + \mathbb{D}^i(m(k - 1))$
 $(P, \alpha) = \text{BestPath}(T_i, k)$ (see 12)
 $\mathbb{D} = \alpha + \mathbb{D}_{\text{root}}(k)$
 endwhile
 endfor
endwhile

Procedure: BestPath(T', k)

Input: a subtree $T' = (V', E')$ of T with n' vertices and associated index k
Output: a feasible path P' passing through middle vertex $m(k)$ of T' whose endpoints lying in T_1 and T_2 and minimizing the median function
 find a middle vertex $m(k)$ of T' and root T' at $m(k)$
 decompose T' into $T_1 = (V'_1, E'_1)$ and $T_2 = (V'_2, E'_2)$ such that $V'_1 \cup V'_2 = V'$, $V'_1 \cap V'_2 = \{m(k)\}$, $E'_1 \cup E'_2 = E'$, and $|V'_1| = n'_1$, $|V'_2| = n'_2$
 sort vertices in V'_1 and V'_2 in non-decreasing order by their distances from $m(k)$ such that $\mathcal{V}'_1 = (v_{i(1)}, v_{i(2)}, \dots, v_{i(n'_1)})$ and $\mathcal{V}'_2 = (v_{q(1)}, v_{q(2)}, \dots, v_{q(n'_2)})$
for $j = n'_1 : 1$ **do**
 find $t(j)$ such that $d(v_{i(j)}, v_{q(t)}) \leq l$ and $d(v_{i(j)}, v_{q(t+1)}) > l$
 let v_Q be the vertex which gives $\min_{q(s) \in \mathcal{Q}} \mathbb{D}(2, v_{q(s)})$
 $\alpha_{i(j)} = \mathbb{D}(1, v_{i(j)}) + \mathbb{D}(2, v_Q)$
endfor
 let v_I be a vertex which gives $\alpha' = \min_{j=1, \dots, n'_1} \alpha_{i(j)}$
 set $P' = P_{v_I, v_Q}$

Theorem 1 (See [30].) *The time complexity of median path algorithm is $O(n \log n)$.*

The k -best median path in a tree

It is known that $|\mathcal{P}| \leq \frac{n}{2}(n + 1)$. The k -best median path $P \in \mathcal{P}$ means the k -path which minimizes the median function with $k = 1, 2, \dots, \frac{n}{2}(n + 1)$. The algorithm presented in the previous subsection finds the first best median path, but it is sometimes needed to locate the second, the third, ..., etc., best median paths of the tree T . Therefore, this algorithm is modified. For the sake of simplicity, it is assumed that the best median path $P^* = P_{v_I, v_Q}$ with endpoints $v_I \in T_1$ and $v_Q \in T_2$. To compute the second best path in T , the first median path $P^* = P_{v_I, v_Q}$ is excluded and the second median path is located by updating the quantities $\alpha_{i(j)}$ in linear time as follows:

$$\alpha_{i(j)} = \begin{cases} \mathbb{D}(1, v_{i(j)}) + \min_{q(s) \in \mathcal{Q} - \{Q\}} \mathbb{D}(2, v_{q(s)}) & \text{if } v_{i(j)} = v_I, \\ \alpha_{i(j)} & \text{otherwise.} \end{cases} \tag{13}$$

Two-phase method to solve BMP problem

The two-phase method is based on computing the supported and unsupported Pareto solutions separately (see [32]). In phase 1, the supported Pareto solutions are computed by solving the weighted sum problem. In phase 2, the unsupported Pareto solutions are computed by an enumerative approach. The search space in phase 2 is restricted due to information obtained from phase 1 since it is restricted to triangles given by two consecutive supported non-dominated points. These triangles are said to be the viable regions.

Phase 1

It is used to find the supported Pareto solutions whose corresponding supported non-dominated points lie on the boundary of the convex hull. The weighted-sum scalarization

is one of the most common methods used for this purpose. In this technique, the individual criteria are combined together so that one can solve a series of single objective problems. Consider the following minimization problem:

$$\min_{P \in \mathcal{P}} \lambda D^1(P) + (1 - \lambda) D^2(P), \tag{14}$$

where $0 \leq \lambda \leq 1$. The problem (14) is equivalent to finding a median path P in the tree T which minimizes the sum of weighted distances function

$$D(P, \lambda) = \sum_{i=1}^n (\lambda w_i^1 + (1 - \lambda) w_i^2) d(v_i, P), \quad 0 \leq \lambda \leq 1. \tag{15}$$

The idea of this phase is based on finding two solutions P_1 and P_2 with $D(P_1) = (x_1, y_1)$, and $D(P_2) = (x_2, y_2)$ obtained by solving the problem (15) for each criterion separately. These solutions must be supported Pareto solutions. If they are identical, then the BMP problem has unique minimum solution. Otherwise another supported Pareto solution P with $D(P) = (x, y)$ between $D(P_1)$ and $D(P_2)$ is computed (if there exists such one).

Algorithm 2. Phase 1

Input: a doubly weighted tree $T = (V, T)$
Output: supported Pareto solutions set \mathcal{P}_{SE}
 $P^1 \leftarrow$ solve (15) with $\lambda = 1, s^1 \leftarrow (D^1(P^1), D^2(P^1))$
 $P^2 \leftarrow$ solve (15) with $\lambda = 0, s^2 \leftarrow (D^1(P^2), D^2(P^2))$
if $s^1 = s^2$ **then**
 $\mathcal{P}_E \leftarrow \{P^1\}$, and $\mathcal{D}_N \leftarrow \{s^1\}$
else
 $P = \text{Uniobjsearch}(P^1, P^2), s \leftarrow (D^1(P), D^2(P))$
 $\mathcal{P}_E \leftarrow \{P^1, P, P^2\}$, and $\mathcal{D}_N \leftarrow \{s^1, s, s^2\}$
endif

Procedure: Uniobjsearch(P^i, P^j)

Input: two supported paths P^i, P^j with $D(P^i) = (x_i, y_i), D(P^j) = (x_j, y_j)$
Output: supported path P^r lying between P^i and P^j
 $P^r \leftarrow$ solve (15) for $\lambda = \frac{y_i - y_j}{(y_i - y_j) + (x_j - x_i)}, s^{(r)} \leftarrow (D^1(P^{(r)}), D^2(P^{(r)}))$
if $P^r \notin \mathcal{P}_E$ **then**
 $\mathcal{P}_E \leftarrow \mathcal{P}_E \cup \{P^r\}$, and $\mathcal{D}_N \leftarrow \mathcal{D}_N \cup \{s^{(r)}\}$
 Uniobjsearch(P^i, P^r)
 Uniobjsearch(P^r, P^j)
end if

Theorem 2 *The running time of Phase 1 algorithm is $O(rn \log n)$, where r is the number of the supported non-dominated points.*

Proof: The running time of Median Path algorithm is $O(n \log n)$. The uniobjsearch procedure can be applied at most $2r$ time, where r is the number of the supported non-dominated points. Hence, the overall time of the phase 1 algorithm is $O(rn \log n)$.

Phase 2

The unsupported non-dominated points lie within triangles, viable regions, formed by two consecutive supported non-dominated points. Phase 2 searches for unsupported non-dominated points separately in each viable region.

Algorithm 3. Phase 2

Input: a doubly weighted tree $T = (V, E)$ and the set \mathcal{D}_{SN}

Output: the set of all Pareto solutions \mathcal{P}_E and the set \mathcal{D}_N

for each two consecutive supported points $s^i = (x', y')$, $s^j = (x'', y'')$ **do**

solve problem (15) with $\lambda = \frac{y' - y''}{(y' - y'') + (x'' - x')}$ and define corresponding viable region

for $k = 1, \dots, \frac{n(n+1)}{2}$ **do**

determine the k^{th} best median path P and let $s = D(P)$

if no more solutions exist in the considered viable region **then**

break

else

$\mathcal{P}_E \leftarrow \mathcal{P}_E \cup \{P\}$, and $\mathcal{D}_N \leftarrow \mathcal{D}_N \cup \{s\}$

endif

endfor

endfor

Theorem 3 *The running time of phase 2 algorithm is $O(Rn \log n)$, where $R = |\mathcal{D}_N|$.*

An illustrative example

The computations of algorithms phase 1 and phase 2 are illustrated by using the following tree given in Fig. 5. The following (BMP) problem is studied:

$$\min_{P \in \mathcal{P}} D(P) = \begin{cases} \sum_{v_j \in V} w_j^1 d(v_j, P), \\ \sum_{v_j \in V} w_j^2 d(v_j, P), \end{cases}$$

where $\mathcal{P} = \{P \subset T : L(P) \leq 26\}$.

It is equivalent to locating the median path in the tree depicted in Fig. 6 with different values of λ . Table 2 shows the use of phase 1 and resulting three supported Pareto solutions. The supported Pareto solutions in order obtained are $P^1 = P_{v_7, v_{11}}$, $P^2 = P_{v_1, v_{12}}$, and $P^3 = P_{v_7, v_{12}}$. Their corresponding supported non-dominated points are $s^1 = (131, 689)$, $s^2 = (219, 350)$, and $s^3 = (169, 474)$. Figure 7 shows the two viable regions sketched by these points.

Table 3 shows the application of phase 2 algorithm to the first viable region (first V.R.) defined by the triangle whose vertices are $s^1 = (131, 689)$, $s^3 = (169, 474)$ and $(169, 689)$. The point $s^4 = (139, 676)$ is located in the first V.R. After that phase 2 is applied to the second viable region (2nd V.R.) which is given by the triangle whose vertices are $s^3 = (139, 474)$, $s^2 = (219, 350)$, and $(219, 474)$. The two unsupported points $s^5 = (177, 461)$ and $s^6 = (194, 447)$ are lying in the 2nd V.R. The unsupported points found by phase 2 are shown in Fig. 7.

Case study

Mahmoud et al. [18] gave a case study as an application to the multi-objective location problem. They presented a methodology based on geographic information system (GIS)

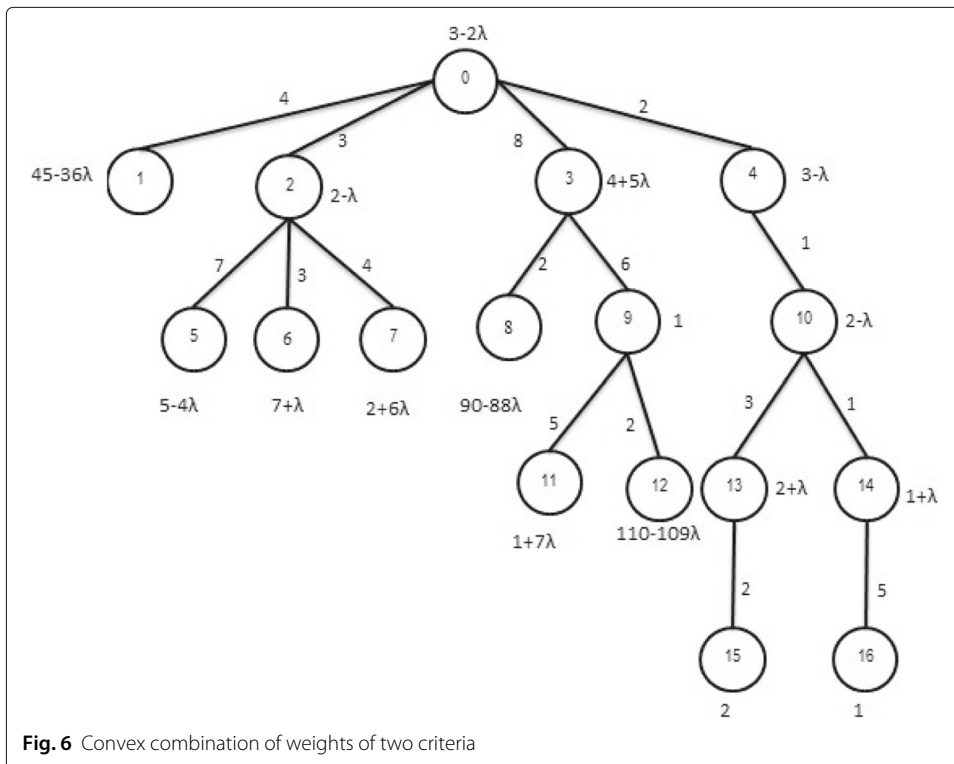
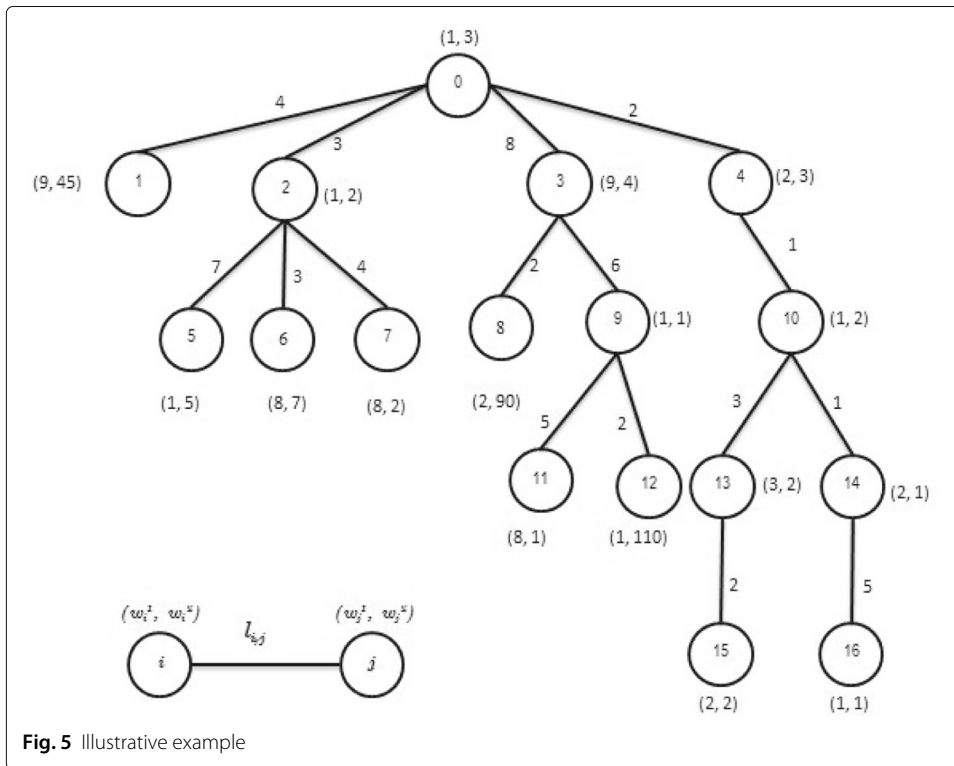


Table 2 Example solutions computed by Phase 1

λ	Path P	$D(P)$	Comment
$\lambda = 1$	$P_{v7,v11}$	$s^1 = (131, 689)$	Obtained from minimizing criterion 1
$\lambda = 0$	$P_{v1,v12}$	$s^2 = (219, 350)$	Obtained from minimizing criterion 2
$\lambda = \frac{339}{427}$	$P_{v7,v12}$	$s^3 = (169, 474)$	Supported point between s^1 and s^2
$\lambda = \frac{215}{253}$	$\begin{cases} P_{v7,v11} \\ P_{v7,v12} \end{cases}$	$\begin{cases} s^1 = (131, 689) \\ s^3 = (169, 474) \end{cases}$	No supported points between s^1 and s^3
$\lambda = \frac{124}{174}$	$\begin{cases} P_{v1,v12} \\ P_{v7,v12} \end{cases}$	$\begin{cases} s^2 = (219, 350) \\ s^3 = (169, 474) \end{cases}$	No supported points between s^3 and s^2

and multi-criteria decision analysis (MCDA) for regionally locating and sizing desalination facilities for domestic water supply. The GIS allows consideration of complex spatial terrain and topographic impacts on locating desalination facilities and MCDA procedure maximizes utility using a 0–1 integer programming model for selecting cities to be served under specified budget allocations. This methodology is applied to a graph with N vertices representing a region with N municipalities. The total number of combinations S up to N municipalities receiving desalinated water supply from various combinations of plants is $S = \sum_{i=1}^N \binom{N}{i}$. The S alternatives are identified as binary variables p_i ($i = 1, \dots, S$), with $\mathcal{M}_i(p_i)$ representing the discrete set of municipalities receiving water from desalination facility alternative i if $p_i = 1$; otherwise, $\mathcal{M}_i(p_i) = \phi$ if $p_i = 0$ and $\mathcal{T}_i(z_i)$ are the sets for each municipality $i = 1, \dots, N$ representing the desalination plant alternatives capable of delivering water to city i , assuming binary variable $z_i = 1$, indicating that city i is delivering desalinated water. They considered the model

$$\min_{x_i} \sum_{j \in \mathcal{M}_i(p_i)} W_j d^2(x_i, \ell_j) \tag{16}$$

$$\max_z \sum_{i=1}^N U_i z_i \tag{17}$$

$$\min C_{\max}, \tag{18}$$

where setting $p_i = 1$ in (16) places plant i at spatial location x_i ,

W_j means the municipal demand at location ℓ_j ,

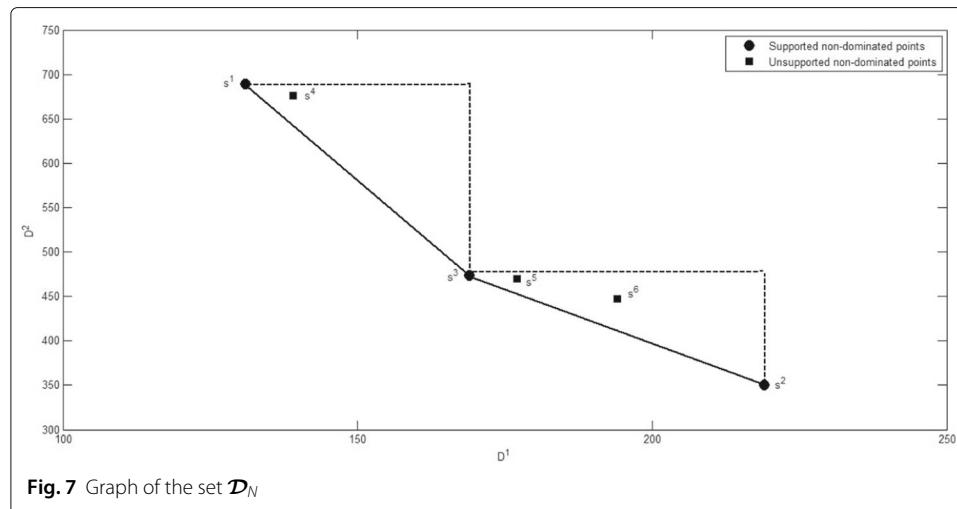


Fig. 7 Graph of the set \mathcal{D}_N

Table 3 Example solutions computed by phase 2

λ	k	Path P	$D(P)$	Comment
$\lambda = \frac{215}{253}$	1	$P_{V_7, V_{12}}$	$s^3 = (169, 474)$	Supported point lies in the first V.R.
		$P_{V_7, V_{11}}$	$s^1 = (131, 689)$	Supported point lies in the first V.R.
	2	$P_{V_6, V_{11}}$	$s^4 = (139, 676)$	Unsupported point lies in the first V.R.
		$P_{V_6, V_{12}}$	$s^5 = (177, 461)$	Unsupported point lies out in the first V.R.
$\lambda = \frac{124}{174}$	3	$P_{V_5, V_{12}}$	$s^6 = (194, 447)$	Unsupported point lies out in the first V.R.
	1	$P_{V_1, V_{12}}$	$s^2 = (219, 350)$	Supported point lies in the second V.R.
	2	$P_{V_6, V_{12}}$	$s^5 = (177, 461)$	Unsupported point lies in the second V.R.
	3	$P_{V_5, V_{12}}$	$s^6 = (194, 447)$	Unsupported point lies out in the second V.R.
	4	$P_{V_2, V_{12}}$	$(201, 482)$	Dominated point lies out in the second V.R.

$d(x_i, \ell_j)$ is the distance between the plant and city j , $j \in \mathcal{M}_i(p_i)$,

$U_i = \sum_{k=1}^R w_k u_{ik}, \forall i = 1, \dots, N$, such that w_k is weight-assigned by a decision-maker to each criterion $k = 1, \dots, R$; and u_{ik} is a score for city i related to criterion k , and

C_{\max} is the least-cost selection of plants that supply water to only the selected cities.

They applied this model to the Northwestern coast of Egypt far from the freshwater sources in the Nile Valley and the Delta to optimally locate and size desalination facilities over the region (see Fig. 8).

Conclusion

The problem of siting special types of subgraphs such as trees and paths of the underlying network is considered as an extensive facility location problem. There are many applications in which several weights per each vertex need to be considered representing demand, importance, population, etc. This paper concentrates on the location problem of a path-shaped facility with bounded length on a doubly weighted tree network. A two-phase method is presented to finding all Pareto solutions.

The proposed problem in this paper is considered as desirable facilities that people like to be nearby. However, there also exist facilities that are undesirable and obnoxious

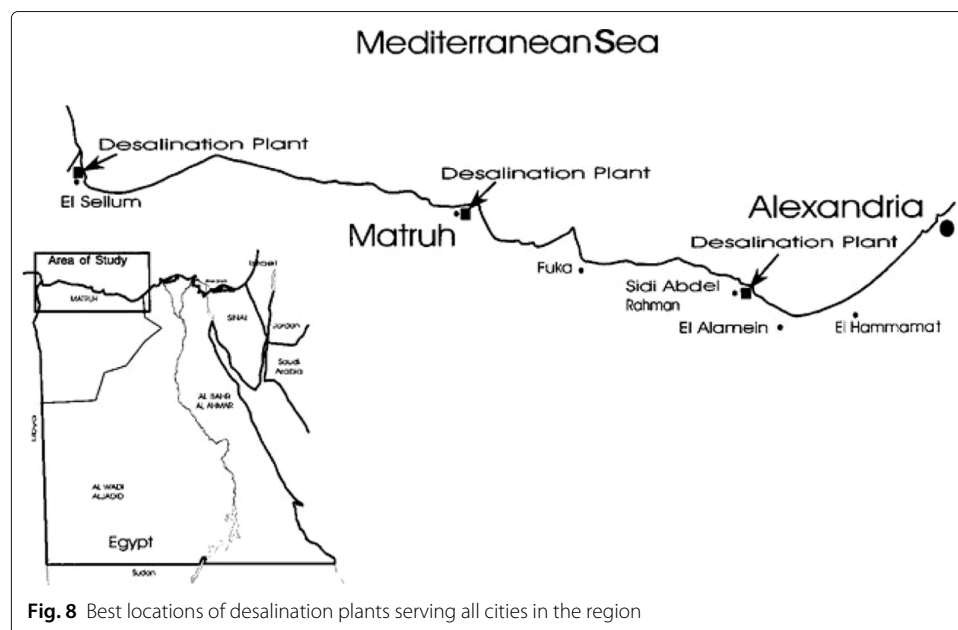


Fig. 8 Best locations of desalination plants serving all cities in the region

such as nuclear reactors, military installations, and chemical plants (see [21, 27]). For the future directions, one aims to study undesirable and obnoxious facilities in a multi-criteria networks.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments and suggestions, which greatly improved the presentation of this paper.

Authors' contributions

AWA suggested and analyzed the algorithms of the considered problem. FE applied the algorithms to an illustrative example and was a major contributor in writing the manuscript. Both authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Zewail City of Science and Technology, 6th of October City, Giza, Egypt. ²Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt. ³Faculty of Science, Damanshour University, Damanshour, Egypt.

Received: 6 March 2019 Accepted: 17 December 2019

Published online: 10 January 2020

References

1. Aboutahoun, A. W., El-Safty, F.: A linear algorithm for a minimax location of a path-shaped facility with a specific length in a weighted tree network. *Int. J. Oper. Res.* **1**. <https://doi.org/10.1504/IJOR.2021.10016889>
2. Alizadeh, B., Etemad, R.: Linear time optimal approaches for reverse obnoxious center location problems on networks. *Optimization*. **65**, 2025–2036 (2016)
3. Averbakh, I., Berman, O.: Algorithms for path medi-centers of a tree. *Comput. Oper. Res.* **26**, 1395–1409 (1999)
4. Becker, R. I.: Inductive algorithms on finite trees. *Quaest. Math.* **13**, 165–181 (1990)
5. Becker, R. I., Chang, Y. I., Lari, I., Scozzari, A., Stochi, G.: Finding the ℓ -core of a tree. *Discrete Appl. Math.* **118**, 25–42 (2002)
6. Becker, R. I., Lari, I., Scozzari, A.: Algorithms for central-median paths with bounded length on trees. *European J. Oper. Res.* **179**, 1208–1220 (2007)
7. Bhattacharya, B., Shi, Q., Tamir, A.: Optimal algorithm for the path/tree shaped facility location problems on trees. *Algorithmica*. **55**, 601–618 (2009)
8. Chankong, V., Haimes, Y. Y.: *Multiobjective decision making: theory and methodology*. Dover publications, New York (2008)
9. Colebrook, M., Sicilia, J.: A polynomial algorithm for the multicriteria cent-dian location problem. *European J. Oper. Res.* **179**, 1008–1024 (2007)
10. Ehrgott, M.: *Multicriteria Optimization*. Springer, Berlin (2005)
11. Ehrgott, M., Gandibleux, X.: Bound sets for biobjective combinatorial optimization problems. *Comput. Oper. Res.* **34**, 2674–2694 (2007)
12. Ghezavati, V., Hosseiniifar, P.: Application of efficient metaheuristics to solve a new bi-objective optimization model for hub facility location problem considering value at risk criterion. *Soft Comput.* **22**, 195–212 (2016)
13. Goldman, A. J.: Optimum center location in simple networks. *Transport. Sci.* **5**, 212–221 (1971)
14. Hamacher, H. W., Labbe, M., Nickel, S.: Multicriteria network location problems with sum objectives. *Networks*. **33**, 79–92 (1999)
15. Hassin, R., Ravi, R., Salman, F. S.: Multiple facility location on a network with linear reliability order of edges. *J. Comb. Optim.* **34**, 931–955 (2017)
16. Kahag, M. R., Niaki, S. T. A., Seifbarghy, M., Zabih, S.: Bi-objective optimization of multi-server intermodel hub-location-allocation problem in congested systems: modeling and solution. *J. Ind. Eng. Int.* **15**, 221–248 (2019)
17. Kariv, O., Hakimi, S. L.: An algorithmic approach to network location problems, Part I, The p -centers. *SIAM J. Appl. Math.* **37**, 513–538 (1979)
18. Mahmoud, M. R., Fahmy, H., Labadie, J. W.: Multicriteria siting and sizing of desalination facilities with geographic information system. *J. Water Res. Pl-ASCE*. **128**, 113–120 (2002)
19. Minioka, E., Patel, N. H.: On finding the core of a tree with a specified length. *J. Algorithms*. **4**, 345–352 (1983)
20. Morgan, C. A., Slater, P. J.: A linear algorithm for a core of a tree. *J. Algorithms*. **1**, 247–258 (1980)
21. Nguyen, K., Vui, P.: The inverse p -maxian problem on trees with variable edge lengths. *Taiwanese J. Math.* **20**, 1437–1449 (2016)
22. Peng, S., Lo, W.: Efficient algorithms for finding a core of a tree with specified length. *J. Algorithms*. **20**, 445–458 (1996)
23. Puerto, J., Rodrigues-Chia, A. M., Tamir, A., Perez-Brito, D.: The bi-criteria doubly weighted center-median path problem on a tree. *Networks*. **47**, 237–247 (2006)
24. Raith, A., Ehrgott, M.: A comparison of solution strategies for biobjective shortest path problems. *Comput. Oper. Res.* **36**, 1299–1331 (2009)

25. Raith, A., Ehrgott, M.: A two-phase algorithm for the biobjective integer minimum cost flow problem. *Comput. Oper. Res.* **36**, 1945–1954 (2009)
26. Ramos, R. M., Ramos, M. T., Colebrook, M., Sicilia, J.: Locating a facility on a network with multiple median-type objectives. *Ann. Oper. Res.* **86**, 221–235 (1999)
27. Rana, R., Garg, D.: Algorithm for obnoxious facility location problem. *Int. J. Adv. Technol.* **5**, 96–106 (2014)
28. Slater, P. J.: Locating central paths in a graph. *Transport. Sci.* **16**, 1–18 (1982)
29. Steiner, S., Radzik, T.: Computing all efficient solutions of the biobjective minimum spanning tree problem. *Comput. Oper. Res.* **35**, 198–211 (2008)
30. Tamir, A., Puerto, J., Mesa, J. A., Rodriguez-Chia, A. M.: Conditional location of path and tree shaped facilities on trees. *J. Algorithms.* **56**, 50–75 (2005)
31. Tamir, A., Puerto, J., Perez-Brito, D.: The centroid subtree on tree network. *Discrete Appl. Math.* **118**, 263–278 (2002)
32. Ulungu, E. L., Teghem, J.: The two-phases method: an efficient procedure to solve bi-objective combinatorial optimization problems. *Found. Comput. Decision Sci.* **20**, 149–165 (1995)
33. Wang, H. L.: An optimal algorithm for the weighted backup 2-center problem on a tree. *Algorithmica.* **77**, 426–439 (2017)
34. Zaferanieh, M., Fathali, J.: Finding a core of a tree with pos/neg weight. *Math. Meth. Oper. Res.* **76**, 147–160 (2012)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
