Weighted admissible absolute 1-center problem

S. I. Fainshtein, A. S. Fainshtein, V. E. Torchinsky

G.I. Nosov Magnitogorsk State Technical University, Magnitogorsk, Russia

Abstract. This paper presents a polynomial algorithm for new generalization of the absolute 1-center problem (A1CP) in general undirected graph with each edge having a positive weight vector (length for the first coordinate and costs for all the other coordinates) and with each vertex having non-negative weight vector. We assume that the cost is a linear function of the length on edge. Non-negative cost boundaries are also given. AA1CP (admissible absolute 1-center problem) minimizes the weighted length of path between a point on edge and the farthest vertex provided that any weighted cost of path from the point to any vertex does not exceed the corresponding cost boundary.

Keywords: vertex-weighted absolute 1-center problem, admissible absolute 1-center problem.

DOI 10.14357/20718632200201

Introduction

Absolute 1-center problem

The absolute 1-center problem is defined as follows. Undirected connected graph $G = <V, E, l, w>$ without loops or multiple edges is given, $|V| = n$, $|E| = m$ with non-negative vertex weights $w(v)$, $v \in V$ and positive edge lengths $l(e)$, $e \in E$. We refer to the case of equal weights as the vertex-unweighted case. An edge (x, y) is identified with a line segment of length $l(x, y)$. By point τ in G we mean any point along any edge including its vertices. Point $\tau = (x, y; t)$ is characterized by its location at a distance of t and at $l(x, y) - t$ from x and y, respectively. The distance $d(t_1, t_2)$ between the points t_1 and t_2 is defined as the length of the shortest path in G between t_1 and t_2. Let us define the weighted distance between the point τ and the farthest vertex as $F(\tau)$: $F(\tau) = \max\{w(v)d(\tau, v): v \in V\}$ where $w(v)d(\tau, v)$ is the weighted distance from point τ to vertex v. Let us assume that τ^* is a point in G such that the distance between this point and the farthest vertex is minimum: $F(\tau^*) = \min\{F(\tau): \tau \in G\}$. The point τ^* is referred to as the absolute 1-center of G and $r_1(G) = F(\tau^*)$ is referred to as the absolute 1-radius of G.

A1CP for the vertex-unweighted case was originally stated and solved by Hakimi (1964) [1]. To find the absolute 1-center of the graph Hakimi used a graphical procedure for computing local absolute centers on each edge by determining the minimum value of the function $F(\tau)$ along such edge. The local absolute center of the edge corresponds to the point $\tau^*(e)$ along the edge e such that $F(\tau^*(e)) = \min\{F(\tau(e)): \tau \in e\}$ and $F(\tau^*) = r(e)$ is the local absolute radius. The absolute 1-center of the graph is chosen among the local absolute centers of edges so that $r_1 = \min\{r(e): e \in E\}$. Kariv and Hakimi (1979) later presented an algorithm for the general vertex-weighted graph with computational complexity $O(|E|n\log n)$ when the distance matrix was known and $O(|E|n \log n + n^3)$ when the distance matrix was unknown [2].

Goldman (1972) first studied A1CP for vertex-unweighted trees [3]. Kariv and Hakimi (1979) presented an algorithm with computational complexity $O(|E|n\log n)$ for a weighted tree [2] and

Megiddo (1983) presented a linear $O(n)$ algorithm [4]. Let us consider the case with weighted trees in more detail.

Vertex-weighted tree

Let us assume that τ^* is the 1-center of the vertex-weighted graph G. It can be easily verified that there are at least two vertices i and j such that the weighted distance from τ^* to i is equal to that from τ^* to j and the absolute radius: $w(i)d(i, \tau^*) = w(j)d(j, \tau^*) = r_1(G)$. The absolute center is the middle of the weighted path between i and j. Vertices i and j are referred to as peripheral vertices. When we add $w(j)d(i, \tau^*)$ to both sides of the previous equality we obtain the following expression for the absolute radius of the graph: $r_1(G) = \frac{w(i)\times w(j)}{w(i)+w(j)} \times l(i,j)$ where $l(i, j)$ is the length of the path between the peripheral vertices provided that it passes through the absolute center. When the graph G is a tree $T(V, E)$, the path between the vertices i and j is unique and $l(i, j) = d(i, j)$ [2]:

$$r_1(T) = \frac{w(i)\times w(j)}{w(i)+w(j)} \times d(i,j) \stackrel{\text{def}}{=} \frac{1}{2}\Delta(i,j), \quad (1)$$

and the peripheral vertices represent any pair of i and j such that $\Delta(i, j) \geq \Delta(u, v)$ for all $u, v \in V$ [2].

Extension of the vertex-weighted A1CP

In the above-cited papers the distance from the point to the farthest vertex is minimized without any constraints. Therefore, the absolute center can be located at any point of the edge. This represents a grave disadvantage from the perspective of practical applications.

Ding and Qui (2017) presented an extension of the classic vertex-weighted absolute 1-center problem (GA1CP) and FPTAS for solving it [5; p.1]: “Given a vertex-weighted undirected connected graph $G = (V, E, l, p)$ where each edge $e \in E$ has length $t(e) > 0$ and each vertex $v \in V$ has weight $p(v) > 0$, a subset $T \subseteq V$ of vertices and a set S consisting of all the points along the edges in a subset $E' \subseteq E$ of edges, the generalized absolute 1-center problem (GA1CP), an extension of the classic vertex-weighted absolute 1-center problem (A1CP), requires to find a point from S such that the weighted distance to the farthest vertex from T is minimized”.

In real problems, the decision maker must choose the optimal location using more than just a single criterion. The first approach is to treat the multi-objective problems as really multi-objective and to derive the Pareto-set. For example, R.M. Ramos, J. Sicilia and M.T. Ramos (1997) studied a bi-criteria problem of determining the absolute center of a graph with two objective functions using independent lengths on each edge [6]. The authors proposed a polynomial time algorithm to obtain the non-dominated location points on the graph.

Another approach to dealing with multicriteria discrete location problems is to fix a bound on the obnoxious effects as a set of constraints. For example, Ding and Qui (2015) studied the restricted absolute center problem (RA1SP) in general undirected graph with each edge having two weights, cost and delay, where the delay is a separable function of the cost on edge [7]. The distance involved in RA1SP is referred to as the length of the restricted shortest path (RSP) between two distinct nodes. The RSP problem is known to be *NP*-complete and has a fully polynomial approximation scheme (FPAS). RA1SP requires to find a restricted absolute 1-center in P (consisting of all the points along the edges in a subset $E' \subseteq E$) such that the cost of the most costly RSP from it to a given node subset is minimized. The problem is solved by $(1 + \varepsilon)$-approximation when the RSP distance matrix has a saddle point.

Our results

In this paper we present a restricted vector version of A1CP for the weighted case. Every edge $e \in E$ of the undirected connected graph $G = \langle V, E \rangle$ is assigned a positive weight vector $l_j(e)$ and every vertex $v \in V$ is assigned a non-negative weight vector $w_j(v)$, $j = 1 \ldots k$. We minimize the weighted length of path between a point $\tau_1 \in e \in E' \subseteq E$ and the farthest vertex for $j = 1$ and require to apply the following cost constraint for all the other coordinates $j \geq 2$: weighted cost of path between the point τ_j and any vertex does not exceed B_j. We assume that $\tau_1 = \lambda_j \tau_j$ where $\lambda_j = l_1(e)/l_j(e)$. The adaptation of the presented algorithm to the solution of the applied problem (“The toll road problem”) is reviewed.

Computational complexity of the algorithm is $O(k|E'|n^2)$ when the distance matrix is unknown. We do not use the technique of “unpromising” edge isolation elaborated in the papers of the above-cited authors. When $k = 1$, the algorithm finds all the equivalent absolute centers over the time $O(|E|n^2)$. When a finite system of admissible disjoint segments S is given in the graph, all the equivalent absolute centers in S can be found over the time of $O(|E'|n^2 + p\log n)$ where p is the number of admissible segments.

1. Median points and median segments

We can define the local absolute center of a segment similar to the definition of the local absolute center of an edge.

Definition 1. The local absolute center of the segment $s \in e$ is a point $\tau^*(s)$ on the segment s such that $F(\tau^*(s)) = \min\{F(\tau(s)): \tau \in s\}$. Let us define $F(\tau^*(s))$ as the local radius of s on the edge e and represent it as $r_e(s)$.

Let us assume that V' is a set of non-zero weight vertices. We select the edge $(x, y) \in E'$ where $l(x, y)$ is the edge length and $d(x, v)$ and $d(y, v)$ are the distances from the end-points to the vertex v.

We shall find a point on the edge (x, y), which is the farthest from the non-zero weight vertex. It is evident that the length of path between the point and the vertex v passing through the vertex x is equal to the length of path passing through the vertex y: $d(v, x) + t = l(x, y) - t + d(y, v)$ and

$$t = (d(v, x) + l(x, y) + d(y, v))/2. \qquad (2)$$

The point $t(v)$ has an important property: for any point on the edge to the left of the point $t(v)$ the shortest path to the vertex v passes through the vertex x and for any point on the edge to the right of that point the shortest path to the vertex v passes through the vertex y. For any zero weight vertex v_0, $v_0 \neq x$, $v_0 \neq y$ we assume by convention that $t(v_0) = l(x, y)$; for the end-points of the edge we assume that $t(x) = l(x, y)$ and $t(y) = 0$, regardless of the weight.

Definition 2. The median point of the vertex v on the edge (x, y) is called a point $t(v)$ such that:

$t(v) = (d(v, x) + l(x, y) + d(y, v))/2$, $v \in V'$, $v_0 \neq x$, $v_0 \neq y$;

$t(v) = l(x, y)$, $v \in V \backslash V'$, $v \neq x$, $v \neq y$;

$t(x) = l(x, y)$, $t(y) = 0$.

Based on (2), we shall find all the median points, arrange them in non-decreasing order and numerate them. We shall note that the median points may be the same. Equal median points are arranged in the non-increasing order of weights of their generating vertices. We shall define the list of median points of the edge (x, y) as $T(x, y) = \{t_1, .., t_n\}$, $t_1 = 0$, $t_n = l(x, y)$ where each point t_i is associated with its generating vertex $v(t_i)$.

Median points generate median segments. Median segment of non-zero length is generated by adjacent unique points. In case with equal median points the first point generated by the largest weight vertex is selected as the beginning and/or the end of the segment.

Definition 3. We shall refer to the segment $[t_{i-c}, t_i]$, $t_{i-c}, t_i \in T(x, y)$, $c \geq 1$, $i \geq 2$ such that the conditions (1)-(3) are satisfied as a median segment $ms(i)$ of non-zero length on the edge (x, y):

1) $t_i - t_{i-1} > 0$; 2) $t_{i-1} - t_{i-c} = 0$, $i - c \geq 1$; 3) $t_{i-c} - t_{i-c-1} > 0$ when $i - c \geq 2$.

The number of median segments will not be greater than $n - 1$. Since $t_1 = 0$ and $t_n = l(x, y)$, median segments $ms(i)$, $i = 2 \ldots n$ completely cover the edge (x, y). We shall assume that the local absolute radius of the zero length median segment is equal to $+\infty$. The local absolute radius of the edge will be selected as the minimum of local absolute radii of median segments: $r(e) = \min\{r(ms(i)): 2 \leq i \leq n\}$.

The median segment $ms(i)$ has the following important property: for any point on this segment the path to the vertices $v(t_1), ..., v(t_{i-1})$ passes through the vertex x and the path to the vertices $v(t_i), ..., v(t_n)$ passes through the vertex y. All the vertices in the graph are divided into two sets with respect to the segment $ms(i)$: $X0$ and $Y0$. $X0$ and $Y0$ contain vertices $v(t_1), ..., v(t_{i-1})$ and $v(t_i), ..., v(t_n)$, respectively.

For segment $ms(i)$ let us assume that $T_i = T(ms(i))$ corresponds to the tree of the lengths of the shortest paths connecting $ms(i)$ with all the vertices in the graph. Let us assume that $[t_{i-1}, t_i]$ represents a non-zero length median segment. The tree is constructed as follows: first we connect the points t_{i-1} and t_i to the vertex set V and then connect the vertices t_{i-1} and t_i with all the vertices of the sets $X0$ and $Y0$, respectively (Fig. 1). We shall assume that $d(t_{i-1}, v) = t_{i-1} + d(x, v)$ for all $v \in X0$ and $d(t_i, v) = l(x, y) - t_i + d(y, v)$ for all $v \in Y0$. It can be easily verified that the length of the path $d(\tau, v)_{Ti}$ in the distance tree T_i is equal to the length $d(\tau, v)_G$ in the original graph G for any point $\tau \in ms(i)$ and any vertex $v \in V'$.

Absolute center of the median segment

Based on (1), a tree has a unique absolute center, which is the middle of the weighted path between two peripheral vertices. We assume that at least one vertex of the graph has non-zero weight. Otherwise, any point in the graph would corre-

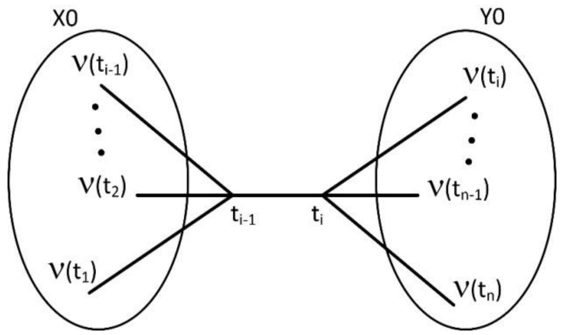

Fig. 1. Distance tree of the median segment $[t_{i-1}, t_i]$

spond to its absolute center. We shall consider the possible locations of the peripheral vertices and the absolute center of the median segment.

1. Let us assume that each of the sets $X0$ and $Y0$ contains at least one non-zero weight vertex.

a) Let us assume that the peripheral vertices belong to different sets. When the mid-point of the weighted path between them belongs to the segment $[t_{i-1}, t_i]$, it represents the local absolute center of the i-th median segment and the radius is calculated based on (1).

Let us assume that the mid-point of the path does not belong to the segment $[t_{i-1}, t_i]$ and, to be definite, is located to the left of the vertex t_{i-1}. The function $F(\tau) = \max\{w(v)d(\tau, v): v \in V\}$ is convex and piecewise linear on every simple path of the tree [4]. All the points of the segment $[t_{i-1}, t_i]$ are located along the path between the two peripheral vertices. The convex function $F(\tau)$ increases as we move away from its absolute minimum point. Thus, the local absolute center of the median segment $[t_{i-1}, t_i]$ is located at the point nearest to the absolute minimum point, i.e. at the vertex t_{i-1} and the farthest vertex from t_{i-1} belongs to the set $X0$. Similarly, when the absolute center of the tree is located to the right of t_i, the local absolute center $[t_{i-1}, t_i]$ is located at the vertex t_i and the farthest vertex belongs to the set $Y0$.

b) Let us assume that the peripheral vertices of the tree T_i belong to one set: $X0$, to be definite. Then the local absolute center of the segment $[t_{i-1}, t_i]$ will be located at the vertex t_{i-1}.

Therefore, to find the peripheral vertices we take one non-zero weight vertex from the set $X0$ and another one from the set $Y0$. We calculate the length $dd(u, v)$ of the unweighted path between vertices $u \in X0$ and $v \in Y0$ in the tree T_i. This path passes through the edge (x, y): $dd(u, v) = d(u, x) + l(x, t_{i-1}) + l(t_{i-1}, t_i) + l(t_i, y) + d(y, v) = d(u, x) + l(x, y) + d(y, v)$ and does not depend on the boundaries of the segment ms_i.

2. Let us assume that one of the sets – $X0$, to be definite – consists of only zero weight vertices. Then the local absolute center of the segment is located at the vertex t_i. Similarly, when $Y0$ consists of only zero weight vertices, the local absolute center of the segment is located at the vertex t_{i-1}.

2. Local absolute centers of the edge (A1CP)

In the following, by "*center*" and "*radius*" of the segment (edge) we shall mean the "local absolute center" and the "local absolute radius" of the segment (edge).

We shall consider a classic A1CP. We are given the undirected connected graph $G = <V, E>$, $|V| = n$, $|E| = m$ with non-negative weights of vertices $w(v)$, $v \in V$ and positive edge lengths $l(e)$, $e \in E$; V' is a set of non-zero weight vertices. It is required to find a point in G such that the distance between the point and the farthest vertex from V' is minimum.

We set the edge (x, y) and precalculate the matrix of weight coefficients $ww(i,j) = \frac{w(i)\times w(j)}{w(i)+w(j)}$, $i, j \in V'$ and the distance vectors $d(x, v)$ and $d(y, v)$, $v \in V$.

Algorithm 1

1. We form a set of median points $T(x, y)$ based on (2) and arrange it in non-decreasing order. We arrange equal points in the non-decreasing order of weights of their generating vertices.

2. We divide all the vertices into two sets $X0$ and $Y0$. $Y0$ initially contains the following: zero weight vertices (except x when $w(x) = 0$); vertex y; vertices whose shortest path to x passes through y.

For every vertex $i \in X0$, $i \in V'$ we find and memorize the farthest vertex $j \in Y0$, $j \in V'$ and half of the weighted distance between them: $maxd(i) = ww(i, j)\times (d(x, i) + l(x, y) + d(y, j))$; $maxv(i) = j$.

3. We pass through the median points of the edge (x, y) from point x to point y. After visiting the point t we transfer its associated vertex $v(t)$ from $X0$ to $Y0$ and recalculate $maxd(i)$ at this vertex for all $i \in X0$, $i \in V'$: $maxd(i) = \max\{maxd(i), ww(i, v(t))\times (d(x, i) + l(x, y) + d(y, v(t))\}$.

Note. The distance is also recalculated for all equal median points since the maximum distance does not necessarily have to be achieved at the largest weight vertex.

4. We find and memorize the radius and the center for every passed non-zero length segment $ms(i)$. For this purpose, we find the vertex i_0 with maximum value of $maxd(i_0)$ among the vertices $i \in X0$, $i \in V'$. When the mid-point of the weighted path between i_0 and $maxv(i_0)$ does not belong to $ms(i)$, we shift the center to the nearest end of the segment and find the weighted distance to the farthest vertex.

5. $r(x, y) = \min\{r(ms(i)): 2 \le i \le n\}$. To print all the equivalent centers of the edge we pass through the median segments and compare the memorized radius of the segment with the edge radius.

Computational complexity for a single edge $O(n^2)$. We shall mention that the idea of dividing the vertices into the two sets X and Y as well as the vertex transfer from X to Y is adopted in the Minieka's algorithm (1981) [8].

Algorithm 1 layout represented in the model graph

We shall find the local absolute radius and all the local absolute centers of the edge (4, 5) for the model graph (Fig. 2). The weights of all the vertices are equal to 1.

Graph radius = 9, the center is located at the vertex 1.

Set of median points with generating vertices:

$T(4, 5) = \{0.0\ (5), 0.0\ (7), 2.5\ (2), 3.5\ (1), 4.5\ (3), 7.0\ (4), 7.0\ (6)\}$.

Set $X0 = \{1, 2, 3, 4, 6\}$; set $Y0 = \{5, 7\}$.

Non-zero length segments, their radii, centers and peripheral vertices *px* and *py*:

[0.0, 2.5], r = 8.0, *center* = 2.0, *px* = 2, *py* = 7, transferred to $Y0$ vertex 2;

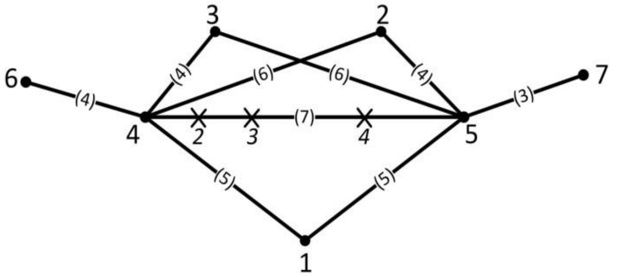

Fig. 2. Model graph, edge lengths

[2.5, 3.5], r = 8.0, *center* = 3.0, *px* = 1, *py* = 2, transferred to $Y0$ vertex 1;

[3.5, 4.5], r = 8.0, *center* = 4.0, *px* = 3, *py* = 1, transferred to $Y0$ vertex 3;

[4.5, 7.0], r = 8.5, *center* = 4.5, *px* = 6, *py* = 3.

Local absolute radius of the edge (4, 5) = 8.0, centers are located at points {2.0; 3.0; 4.0}.

3. Admissible absolute 1-center problem (AA1CP)

Instance. Undirected connected graph $G = <V, E>$, $|V| = n$, $|E| = m$ is given. Every edge $e \in E$ is assigned a positive weight vector $l_j(e)$ and every vertex $v \in V$ is assigned a non-negative weight vector $w_j(v)$, $j = 1 \dots k$. Set S, is given, which contains all the edge points from the specified subset $E' \subseteq E$. The non-negative cost boundaries B_j are given for all $j \ge 2$.

Question. It is required to find the point $\tau_1 \in e \in S$ that minimizes the function $\max\{w_1(v)d_1(\tau_1, v): v \in V\}$ provided that for all $j \ge 2$ and all $v \in V$ $w_j(v)d_j(\tau_j, v) \le B_j$, $d_j(\tau_j, v)$ is the length of shortest path from τ_j to v taken for the j-th coordinate and $\tau_1 = \lambda_j\tau_j$ where $\lambda_j = l_1(e)/l_j(e)$.

We shall set the edge $(x, y) \in E'$ and the coordinate $j \ge 2$. We shall precalculate the matrices of coefficients ww_j and the distance vectors $d_j(x, v)$ and $d_j(y, v)$, $v \in V$. We shall calculate the radii and the centers of the median segments $(ms_j(i))$, $i = 2 \dots n$ using the Algorithm 1.

Finding admissible segments based on one coordinate

Algorithm 2

1. We use the Algorithm 1 to calculate the local absolute radii and the centers of the median segments $(ms_j(i))$, $i = 2 \dots n$.

2. p_j:= 0; //current number of the admissible segment.

3. We pass through the median points of the edge (x, y) from point x to point y. When the radius of the i-th segment $r(ms_j(i)) \le B_j$, go to step 4.

4. We find the admissible segment of $ms_j(i)$, which satisfies the j-th constraint.

All the vertices associated with the median points $ii = 1 \dots i - 1$ belong to the set $Y0$. We find the left boundary of the admissible segment, i.e. the maximum distance

$$lb = \max\left\{\frac{B_j}{w_j(v(t_{ii}))} - d_j(y, v(t_{ii})): ii = 1 \cdots i - 1\right\}.$$

Similarly, all the vertices associated with the median points $ii = i \dots n$ belong to the set $X0$. We find the right boundary of the admissible segment, i.e. the minimum distance

$$rb = \min\left\{\frac{B_j}{w_j(v(t_{ii}))} - d_j(x, v(t_{ii})) : ii = i \cdots n\right\}.$$

When $lb \le rb$ and $[lb, rb] \cap ms_j(i) \ne \varnothing$, $p_j := p_j + 1$ and $as_j(p_j) := [lb, rb] \cap ms_j(i)$.

Finding admissible absolute centers that satisfy all the constraints

Algorithm 3

1. We use the Algorithm 2 for all $j \ge 2$ to find the admissible segments $as_j(i)$, $i = 1 \dots p_j$ that satisfy the j-th constraint.

2. We convert the boundaries of all the admissible segments $as_j(i).lb$ and $as_j(i).rb$, $i = 1 \dots p_j$, $j \ge 2$ into the edge measurement units for $j = 1$:

$$\lambda(j) = \frac{l_1(x, y)}{l_j(x, y)};$$
$$as_j(i).lb := \lambda(j) \times as_j(i).lb;$$
$$as_j(i).rb := \lambda(j) \times as_j(i).rb.$$

3. We apply the binary search to find the intersections of the admissible segments for all the coordinates $j \ge 2$. The number of the resulting admissible segments $as(i)$, $i = 1 \dots p$ will not be greater than $n - 1$ and all the cost constraints will be satisfied for every point on any segment.

4. We use the Algorithm 1 to find the median segments $ms_1(ii)$, $ii = 2 \dots n$, their local absolute radii and centers for $j = 1$.

5. We apply the binary search to find the intersections between the median segments $ms_1(ii)$, $ii = 2 \dots n$ and the admissible segments $as(i)$, $i = 1 \dots p$. We define the local absolute radius and the center for every non-empty intersection. The minimum radius will correspond to the local absolute radius of the edge and the respective centers will satisfy all the cost constraints.

Computational complexity of the algorithm is $O(k|E'|n^2)$ when the distance matrix is unknown.

Toll road problem

Let us assume that the edges of the model graph represent toll roads and the vertices represent clients. The road lengths ($j = 1$) and the fares ($j = 2$) are shown in Fig. 2 and Fig. 3, respectively. The weights of all the vertices are equal to 1. It is required to locate the service center on the edge (4, 5) in such a way that the fare for travelling there does not

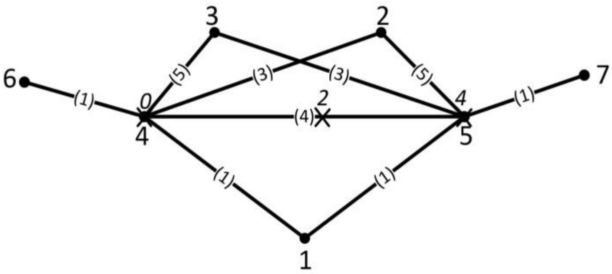

Fig. 3. Model graph, edge costs

exceed 5.5 for any client and the distance to the farthest client is minimum.

Set $X0 = \{1, 2, 3, 4, 6\}$; set $Y0 = \{5, 7\}$.

Set of median points with generating vertices:

$T(4, 5) = \{0.0\ (5),\ 1.00\ (3),\ 1.00\ (7),\ 2.00\ (1),\ 3.00\ (2),\ 3.00\ (6),\ 7.00\ (4)\}$.

Radii, centers and peripheral vertices px and py of non-zero length segments:

[0.00, 1.00], $r = 5.00$, $center = 0.00$, $px = 3$, $py = 5$, transferred to $Y0$ vertices 3 and 7;

[1.00, 2.00], $r = 5.00$, $center = 2.00$, $px = 2$, $py = 3$, transferred to $Y0$ vertex 1;

[2.00, 3.00], $r = 5.00$, $center = 2.00$, $px = 2$, $py = 3$, transferred to $Y0$ vertices 2 and 6;

[3.00, 4.00], $r = 5.00$, $center = 4.00$, $px = 4$, $py = 2$.

Local absolute radius of the edge (4, 5) = 5.00, the centers = {0.00; 2.00; 4.00}.

Admissible segments (in two measurement units), their centers and radii

[0.00, 0.50] → [0.00, 0.88], $r = 9.13$, $center = 0.88$;

[1.50, 2.00] → [2.63, 3.50], $r = 8.00$, $center = 3.00$;

[2.00, 2.50] → [3.50, 4.38], $r = 8.00$, $center = 4.00$;

[3.50, 4.00] → [6.13, 7.00], $r = 10.13$, $center = 6.13$.

Local absolute radius of the edge (4, 5) = 8.00, the admissible centers = {3.00; 4.00}.

Conclusion

1. An edge of the undirected connected graph over the time $O(|V'|\log|V'|)$ can be divided into not more than $n - 1$ median segments of non-zero length and each segment has a unique local absolute center where $|V'|$ is the number of non-zero weight vertices.

2. The admissible vector version of the absolute center problem (AA1CP) allows to find all the equivalent admissible absolute centers of the graph, which satisfy all the $k - 1$ constraints of the problem over the time $O(k|E'|n^2)$ where $|E'|$ is the

number of admissible edges. The bound of cost B_j, $j \geq 2$ considered in this paper is uniform for all nodes in V'. In fact, Algorithm 2 also can be applied to scenarios where the cost bounds of nodes in V' are nonuniform.

3. When a finite system of admissible disjoint segments S is given for a classic A1CP, all the equivalent admissible absolute centers of the graph can be found from S over the time $O(p\log|V'| + |E'|n^2)$ where p is the number of admissible segments.

4. The review of our study materials and translated textbooks [9]-[10] shows that either the graphic Hakimi algorithm is used for the classical A1CP or such problem is completely eliminated from consideration, for example [11]. The proposed Algorithm 1 features simple program implementation without using special data structures and can be used for educational purposes when studying the problems of optimum location for service centers.

Acknowledgements. The authors wish to thank Victor S. Shulman for his helpful discussions.

References

1. Hakimi, S.L. 1964. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. Operations Research 12:450–459. doi:10.1287/opre.12.3.450.
2. Kariv, O. & Hakimi, S.L. 1979. Algorithmic approach to network location problems, I: The p-Centers, SIAM J. Appl. Math. 37(3):513–537. doi:10.1137/0137040.
3. Goldman, A.J. Minimax location of a facility in a network. Transp. Sci. 6(4):407–418. doi:10.1287/trsc.6.4.407.
4. Megiddo, N. 1983. Linear-time algorithms for linear programming in R3 and related problems. SIAMJ. Comput. 12(4): 759–776. doi:10.1137/0212052.
5. Ding, W. & Qiu., K. 2017. FPTAS for generalized absolute 1-center problem in vertex-weighted graphs. Journal of Combinatorial Optimization. 34(4):1084–1095. doi: 10.1007/s10878-017-0130-4.
6. Ramos, R.M., Sicilia, J. & Ramos, M.T. 1997. The biobjective absolute center problem. TOP, 5(2): 187-199. doi: 10.1007/BF02568548.
7. Ding, W. & Qiu., K. 2015. Approximating the Restricted 1-Center in Graphs. In: Lu Z., Kim D., Wu W., Li W., Du DZ. (eds) Combinatorial Optimization and Applications. Lecture Notes in Computer Science, vol 9486: 647-659. doi: 10.1007/978-3-319-26626-8_47.
8. Minieka, E. 1981. Polynomial Time Algorithm for Finding the Absolute Center of a Network. Networks, 11: 351-355. doi:10.1002/net.3230110404
9. Minieka, E. 1978. Optimization Algorithms for Networks and Graphs. New York: Marcel Dekker. 356 p.
10. Christofides, N. 1975. Graph Theory: Algorithmic Approach. New York: Academic. 400 p.
11. Torchinsky, V.E. & Fainshtein, S.I. 2007. Struktury i algoritmy obrabotki dannykh na EVM [Structures and algorithms of computer data processing]. Magnitogorsk: Magnitogorsk State Univ. 139 p.

Fainshtein S. I. Assistant professor of G.I. Nosov Magnitogorsk State Technical University, Magnitogorsk, 38 Lenina Ave. Associate professor. Graduated from Lomonosov Moscow State University in 1982. 10 published articles. Topics of interest: discrete optimization, information technologies. Corresponding author. E-mail: sfainshtein@yandex.ru

Fainshtein A. S. Assistant professor of G.I. Nosov Magnitogorsk State Technical University, Magnitogorsk, 38 Lenina Ave. PhD, associate professor. Graduated from Azerbaijan State University in 1976. 40 published articles. Topics of interest: operators on Banach spaces, Lie algebras of operators, discrete mathematics. E-mail: swetlana@mgn.ru

Torchinsky V. E. Assistant professor of G.I. Nosov Magnitogorsk State Technical University, Magnitogorsk, 38 Lenina Ave. Associate professor, leading software engineer. Graduated from G.I. Nosov Magnitogorsk Mining and Metallurgical Institute in 1984. 8 published articles. Topics of interest: optimization and evolution algorithms, information technologies. E-mail: vet@magtu.ru