ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2013, Vol. 7, No. 2, pp. 142–152. © Pleiades Publishing, Ltd., 2013.
Original Russian Text © A.I. Erzin, P.V. Plotnikov, Yu.V. Shamardin, 2013, published in Diskretnyi Analiz i Issledovanie Operatsii, 2013, Vol. 20, No. 1, pp. 12–27.

On Some Polynomially Solvable Cases and Approximate Algorithms in the Optimal Communication Tree Construction Problem

A. I. Erzin[1,2*], R. V. Plotnikov[2], and Yu. V. Shamardin[1***]**

[1] *Sobolev Institute of Mathematics, pr. Akad. Koptyuga 4, Novosibirsk, 630090 Russia*

[2] *Novosibirsk State University, ul. Pirogova 2, Novosibirsk, 630090 Russia*

Received January 17, 2012; in final form, March 28, 2012

Abstract—Considering an arbitrary undirected n-vertex graph with nonnegative edge weights, we seek to construct a spanning tree minimizing the sum over all vertices of the maximal weights of the incident edges. We find some particular cases of polynomial solvability and show that the minimal span whose edge weights lie in the closed interval $[a, b]$ is a $\left(2 - \frac{2a}{a+b+2b/(n-2)}\right)$-approximate solution, and the problem of constructing a 1.00048-approximate solution is NP-hard. We propose a heuristic polynomial algorithm and perform its *a posteriori* analysis.

DOI: 10.1134/S1990478913020038

Keywords: *wireless communication network, spanning tree, approximate algorithm*

INTRODUCTION

Many communication networks use wireless communication for information exchange. The energy losses at their elements are proportional to d^s with $s \geq 2$, where d is the transmission distance [3]. In some networks, for instance in wireless sensor networks, the elements (sensors) have restricted energy reserves, and efficient energy use by them makes it possible to extend the network's lifetime [1, 10, 13, 14]. To save on energy use, the modern sensors are capable of adjusting the radio transmission distance. This leads to the problem of determining the transmission distance for every element of the network so that to minimize the total energy required to keep the graph connected. If we assume that the radio signal propagates in the same way in all directions then all elements in the transmission zone (not beyond the transmission distance) receive the message. In this case, we may assume that the communication network, a spanning subgraph whose edges carry the transmission, is a complete graph [3, 6, 9, 10]. However, it is not always true that the signal propagates in the same way in all directions and at all distances. Therefore, in general, we should assume that the communication graph $G = (V, E)$ can be an arbitrary spanning subgraph and the energy losses for the transmission along the edges may vary. Thus, if $c_{ij} \geq 0$ stands for the energy loss incurred in data transmission from $i \in V$ to $j \in V$ then, in the connected subgraph $T = (V, E')$ with $E' \subseteq E$, the energy loss at the vertex $i \in V$ equals

$$E_i(T) = \max_{j|(i,j)\in E'} c_{ij}.$$

The goal of this article is to study the problem of constructing a spanning subgraph T for which the sum $\sum_{i\in V} E_i(T)$ is minimal. Without loss of generality, we may assume that T is a spanning tree.

As we noted above, the problems of this kind arise, for instance, in wireless sensor networks, when the location of sensors is known and we are required to determine an energy-efficient graph connecting all sensors [13]. It is customary in the literature to consider as the communication graph of a sensor

* E-mail: adilerzin@math.nsc.ru
** E-mail: nomad87@ngs.ru
*** E-mail: orlab@math.nsc.ru

network a spanning tree of minimal weight P when the weight of the edge joining a pair of vertices equals the squared distance between these vertices [13]. However, the minimal span is not always the optimal communication graph of a sensor network.

The problem of determining the transmission distance of every vertex lying in a Eucludean space in order to find a strongly connected graph in which the total energy expense on communication is minimal was studied in [9]. In particular cases when the vertices lie on a straight line, some polynomial algorithms are proposed for solving this problem. The NP-hardness of the problem in the three-dimensional Euclidean space is established.

The following algorithms are proposed in [3]: an algorithm with asymptotic accuracy $5/3$; a polynomial algorithm which constructs an $11/6$-approximate solution; an exact algorithm, the branches-and-boundaries method, using a new formulation of the problem as an integer linear programming problem.

The problem of determining the power of radio transmitters for data transmission at two distances, "short" and "long", was considered in [6] and shown to be NP-hard. A polynomial algorithm is proposed which constructs a solution with the number of long-distance transmitters at most $11/6$ of their number in the optimal solution. An exponential $9/5$-approximate algorithm is also proposed. These results are obtained in the cases that the elements lie in an Euclidean space, but they are easy to generalize to an arbitrary metric.

In this article, we find some particular cases of polynomial solvability of the problem. We show that the problem of constructing a 1.00048-approximate solution is NP-hard. We propose an effective heuristic algorithm and run a simulation demonstrating its high efficiency.

1. STATEMENT OF THE PROBLEM AND COMPLEXITY ANALYSIS

Consider a simple undirected weighted graph $G = (V, E)$ with vertex set V, $|V| = n$, and edge set E. Let $c_{ij} \geq 0$ denote the weight of the edge $(i, j) \in E$. We are to find a spanning tree T^* for G which is the solution to the problem

$$W(T) = \sum_{i \in V} \max_{j \in N_i(T)} c_{ij} \ \rightarrow \ \min_{T}, \tag{1}$$

where $N_i(T)$ stands for the set of vertices adjacent to vertex i in T. In the English literature, this problem is usually called the *Min-Power Symmetric Connectivity Problem* [3]. Henceforth, we call every admissible solution to (1), which is a spanning tree, a *communication tree* (or a *subgraph*).

The proofs that this problem is NP-hard in the cases that the network elements lie in the three-dimensional Euclidean space appeared in [9], and in the two-dimensional Euclidean space, in [5]. In both cases, the edge weight corresponds to the squared Euclidean distance between the corresponding vertices. Naturally, this implies the NP-hardness of the problem in general, but we need the polynomial reduction presented below, which also proves the NP-hardness of the problem under consideration in the strong sense, to ascertain the approximability bounds of the problem.

Lemma 1. *The minimal vertex covering problem reduces polynomially to Problem* (1).

Proof. Let us polynomially reduce t he NP-hard, in the strong sense, minimal vertex covering problem to a particular case of Problem (1). In the minimal vertex covering problem, given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$, we seek a subset $V' \subseteq V$ of vertices of minimal cardinality such that, for every edge $(i, j) \in E$, at least one of the vertices i and j belongs to V'.

Consider the auxiliary minimal covering problem in which we are given some sets of objects J and elements I, as well as the parameters

$$a_{ij} = \begin{cases} 1, & \text{if } i \in I \text{ covers } j \in J, \\ +\infty, & \text{otherwise.} \end{cases}$$

We seek a subset $I' \subseteq I$ of minimal cardinality covering all objects.

The minimal vertex covering problem polynomially reduces to the minimal covering problem. In order to verify this, it suffices to put $J = E$ and $I = V$ and assume that the element $i \in I$ covers the object $e = (p, q) \in J$ whenever $i = p$ or $i = q$.

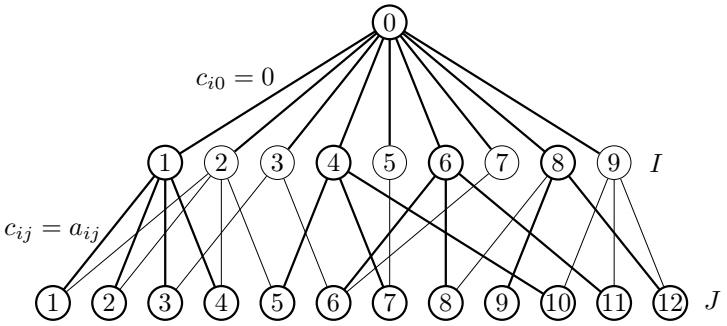


Fig. 1. Example of Problem (1) obtained from an individual minimal covering problem

Given an arbitrary minimal covering problem, construct the graph $G = (V, E)$ with the vertex set $V = \{0\} \cup I \cup J$ and edge set

$$E = \big\{(0, i) \mid i \in I\big\} \cup \big\{(i, j) \mid a_{ij} = 1,\ i \in I,\ j \in J\big\}.$$

Define the edge weights $c_{0i} = 0$ and $c_{ij} = a_{ij}$ for $i \in I$ and $j \in J$. Then, in every spanning tree T of G, for every $j \in J$, we have $\max_{i \in N_j(T)} c_{ij} = 1$, while, for $i \in I$, either $\max_{j \in N_i(T)} c_{ij} = 1$ if i is joined to some vertex $j \in N_i(T)$ or $\max_{j \in N_i(T)} c_{ij} = 0$ if i is joined only to vertex 0. A solution T^* to Problem (1), to which we refer as *Problem* $\mathcal{P}$, determines a minimal covering $I' \subseteq I$ containing the vertices in I adjacent to the vertices in J in the tree T^*.

Thus, we have polynomially reduced the minimal covering problem, which is NP-hard in the strong sense, to a particular case of Problem (1). The proof of Lemma 1 is complete. □

Fig. 1 depicts an example of Problem (1) constructed from an individual minimal covering problem using the polynomial reduction just described. The bold lines highlight an admissible solution to the problem, which determines a covering of all elements of J by the elements of I.

2. PARTICULAR CASES

Consider some particular cases of the problem arising in various applications [1, 13].

Lemma 2. *If the graph G is complete and the edge weights take two values, $c_{ij} \in \{a, b\}$ with $a < b$, then Problem* (1) *is polynomially solvable.*

Proof. Construct a minimal span with respect to the total weight of the included edges as follows: Start with the trivial graph $(V, \varnothing)$. Add the maximal number of edges of weight a to obtain acyclic connected components. Let m be the number of these connected components. To construct a spanning tree, we have to join these by $m - 1$ edges. Choose one vertex in every connected component and connect them by weight b edges to obtain a spanning tree.

The resulting minimal span in this case is the optimum of Problem (1) since in it the number of vertices adjacent to weight b edges is minimal. The proof is over. □

The two other simple cases of optimal tree construction are worth mentioning:

Case 1. Take two minimal values $a_1 < a_2$ of edge weights in a graph G. Add to the trivial tree $(V, \varnothing)$ an edge of length a_1 to obtain the minimal number of acyclic connected components. If we can choose one vertex in every connected component and join these vertices by edges of length a_2 then the resulting tree is the optimal solution to Problem (1).

Case 2. Take the minimal weight a of edges in a graph G and suppose that the weights of all edges in a minimal span P equal a. Then P is the optimal solution to Problem (1).

Remark 1. Regular coverings of a plane by disks of two radii are considered in [1]. The minimal spans connecting the sensors contain only edges of minimal length. Hence, a minimal span yields an optimal communication tree. On the other hand, [13] proposed a covering with disks of three radius (model III). The minimal span connecting the sensors in model III is not the optimal communication tree, and the conclusions of the authors of [13] concerning the energy efficiency of the link are false.

3. SOME ACCURACY ESTIMATES FOR POLYNOMIAL ALGORITHMS

Let us estimate the proximity of a minimal span to the optimal solution to Problem (1).

Lemma 3. *Given a spanning tree T, we have*

$$\sum_{(i,j)\in T} c_{ij} \le \sum_{i\in V} \max_{j\in N_i(T)} c_{ij} \le 2 \sum_{(i,j)\in T} c_{ij}. \tag{2}$$

Proof. The left inequality is obvious: once we have chosen an arbitrary vertex as the root of T, every vertex i has one predecessor vertex $p(i)$ on the way from the root (the root is its own predecessor), and we can express the weight of the tree as

$$\sum_{(i,j)\in T} c_{ij} = \sum_{i\in V} c_{i,p(i)} \le \sum_{i\in V} \max_{j\in N_i(T)} c_{ij}.$$

The right inequality in (2) follows from the fact that every edge is incident to two vertices, and consequently, its weight can occur in the objective function at most twice. The proof is over. □

Let T^* denote the solution to Problem (1); thus, $W(T^*) = W^* = \min_T W(T)$, while

$$C = \sum_{(i,j)\in P} c_{ij} = \min_T \sum_{(i,j)\in T} c_{ij}$$

is the weight of the minimal span P. Order the edges included into P according to their weights:

$$a = c_1 \le c_2 \le \cdots \le c_{n-1} = b.$$

Denote the numbers of edges of the tree P whose weights are counted in the objective function $W(P)$ two, one, and zero times by k, l, and m respectively. Then $k + l + m = n - 1$, while $2k + l = n$. This implies $l = n - 2k$ and $m = k - 1$. Hence,

$$W(P) \le 2\sum_{i=n-k}^{n-1} c_i + \sum_{i=n-k-l}^{n-k-1} c_i = 2\sum_{i=n-k}^{n-1} c_i + \sum_{i=k}^{n-k-1} c_i$$
$$= 2C - \left(\sum_{i=1}^{k-1} c_i + \sum_{i=1}^{n-k-1} c_i\right) \le 2C - a(n-2).$$

Therefore,

$$\varepsilon(P) = \frac{W(P)}{W(T^*)} \le 2 - \frac{a(n-2)}{C} = \varepsilon_1(C).$$

On the other hand, $\varepsilon(P) \le bn/C = \varepsilon_2(C)$. The weight C of the minimal span P is unknown beforehand. However, $\varepsilon_1(C)$ increases, while $\varepsilon_2(C)$ decreases, and consequently, there is $\overline{C} \in [a(n-1), b(n-1)]$ such that $\varepsilon_1(\overline{C}) = \varepsilon_2(\overline{C})$.

We have $\overline{C} = ((a+b)(n-2) + 2b)/2$ and

$$\varepsilon(P) \le \min_C\{\varepsilon_1(C), \varepsilon_2(C)\} \le \varepsilon_1(\overline{C}) = \varepsilon_2(\overline{C}) = 2 - \frac{2a}{a + b + 2b/(n-2)}.$$

This establishes

Theorem 1. *If the weights of the edges in the minimal span P belong to the closed interval $[a, b]$ then we have the attainable estimate*

$$\varepsilon(P) = \frac{W(P)}{W(T^*)} \le 2 - \frac{2a}{a + b + 2b/(n-2)}, \tag{3}$$

and

$$\varepsilon(P) \le \frac{2b}{a+b} \tag{4}$$

as $n \to +\infty$.

The resulting estimate is attainable and equals 2, for instance, in the case that the weights of the edges included into the minimal span P less than two times are zero. If the network elements are points in a Euclidean space then the distance between distinct points cannot be zero. However, estimates (3) and (4) are attained in the limit. In order to verify this, it suffices to consider the example in Fig. 5 of [3].

The covering model III of [13] uses as a communication network a minimal span connecting the nodes of an infinite lattice which includes edges of weights $c_1 \approx 0.071$, $c_2 \approx 0.095$, and $c_3 \approx 1.071$. Hence, according to (4), the relative error of the minimal spanning tree P_{III} in the limit equals

$$\frac{W(P_{III}) - W(T^*)}{W(T^*)} \le \frac{2 \cdot 1.071}{1.142} - 1 \approx 0.87.$$

Recall that, in the minimal vertex covering problem, we are given a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. Lemma 1 implies that every minimal vertex covering problem reduces to a particular case of Problem (1), namely, Problem $\mathcal{P}$.

Suppose that a polynomial algorithm A constructs a Q-approximate solution to Problem $\mathcal{P}$. Thus,

$$1 \le \frac{W(T_A)}{W^*} \le Q, \tag{5}$$

where $W^* = W(T^*) = n^* + m$ is the optimal value of the objective function of Problem $\mathcal{P}$, in which n^* elements of the set I cover all objects, while $W(T_A) = n' + m$ is the value of the functional of Problem $\mathcal{P}$ on the tree constructed by Algorithm A.

Suppose that the problem of constructing an R-approximate solution, with $R > 1$, to the minimal vertex covering problem is NP-hard. Then

$$\frac{n'}{n^*} = \frac{W(T_A) - m}{W^* - m} > R.$$

We infer from (5) that $W(T_A) \le QW^*$. Hence,

$$QW^* - m > R(W^* - m), \qquad Q > R - (R-1)m/W^*.$$

Suppose that the degree of the graph G, which is the maximal degree of its vertices, equals $\Delta \le k$. It is known [4] that the minimal vertex covering problem is NP-hard for every class of graphs with $k \ge 3$. Since, in this case, every element covers at most k objects, it follows that $n^* \ge m/k$. Consequently,

$$W^* \ge m/k + m, \qquad Q > 1 + (R-1)/(k+1).$$

In general R depends on k. For instance, if P≠NP then, for $k = 5$, there exists no 1.0029-approximate polynomial algorithm for the minimal vertex covering problem [4] (that is, $R = 1.0029$). Hence, $Q > 1.00048$, and therefore we have proved

Theorem 2. *If* P≠NP *then there exists no* 1.00048*-approximate polynomial algorithm for Problem* (1).

4. A HEURISTIC ALGORITHM

Here we propose a method for solving Problem (1) approximately. In short, it amounts to the following: To begin with, we construct a lower bound for the optimum of Problem (1) by solving the minimal span problem in a graph with special edge weights. Then we apply a local improvement procedure to the minimal span found in calculating the lower bound. The resulting tree is the required approximate solution to Problem (1).

Instead of a graph $G = (V, E)$ consider the complete n-vertex directed graph K_n, in which every pair $i, j \in V$ of vertices are joined by both an arc (i, j) of weight $c_{ij} \geq 0$ and an arc (j, i) of weight $c_{ji} \geq 0$. Put

$$N_i(T) = \{j \in V \mid (i, j) \in T\}.$$

Put

$$c_i = \min_{j \in V,\, j \neq i} c_{ij}, \quad \text{for } i \in V, \qquad c = \sum_{i \in V} c_i,$$

as well as $a_{ij} = c_{ij} - c_i$ for $i, j \in V$ with $i \neq j$. Then the functional of (1) becomes

$$W(T) = c + \sum_{i \in V} \max_{j \in N_i(T)} a_{ij}. \tag{6}$$

Consider a vertex i of K_n. Order the remaining vertices so that the weights a_{ij} are nondecreasing: $a_{ij_1} = 0 \leq a_{ij_2} \leq \cdots \leq a_{ij_{n-1}}$. Refer to the nondecreasing function $g_i(k) = a_{ij_k}$, $k = 1, \ldots, n-1$, as the *weight function* of vertex i. Take some nonnegative nondecreasing minorant $h_i(k)$ of $g_i(k)$:

$$h_i(1) = 0 \leq h_i(2) \leq \cdots \leq h_i(n-1), \qquad h_i(k) \leq g_i(k), \qquad k = 1, \ldots, n-1.$$

Let z_{ij} denote the increments of the minorants h_i relative to the arcs (i, j) and calculated as $z_{ij_1} = 0$ and $z_{ij_k} = h_i(k) - h_i(k-1)$ for $k = 2, \ldots, n-1$. Recalling (6) and the notation introduced, we obtain

$$W(T) - c = \sum_{i \in V} \max_{j_k \in N_i(T)} g_i(k) \geq \sum_{i \in V} \max_{j_k \in N_i(T)} h_i(k) \geq \sum_{i \in V} \sum_{k | j_k \in N_i(T)} z_{ij_k} = \sum_{(i,j) \in T} (z_{ij} + z_{ji}). \tag{7}$$

Let (b_{ij}) denote the symmetric matrix with entries $b_{ij} = b_{ji} = z_{ij} + z_{ji}$. Using (7), we obtain the following lower bound L for the optimal values of the objective function of Problem (1):

$$W^* = \min_T W(T) \geq c + \min_T \sum_{(i,j) \in T} b_{ij} = L. \tag{8}$$

Calculating L amounts to solving the minimal spanning tree problem for the graph K_n with the matrix (b_{ij}) of edge weights.

In calculating the lower bound (8) we propose to take as the minorants h_i the convex hulls of the weight functions g_i for $i \in V$. According to (2), the minimal weight C of the span in the initial graph with edge weights c_{ij} is another lower bound for the functional of (1). Hence, we can take $\max\{C, L\}$ as a lower bound for the objective function of Problem (1).

Observe that both C and L can be the best (largest) lower bound. Indeed, consider the following example: Place the network elements at the points on the plane with coordinates

$$(9, 0), \qquad (0, 0), \qquad (9, 1), \qquad (18, 0), \qquad (9, 11).$$

Assign to each edge (i, j) the weight c_{ij} equal to the squared distance between the points i and j. As the minorants for the functions g_i take their convex hulls. Then the tree

$$T = \{\{1, 3\},\ \{2, 3\},\ \{3, 4\},\ \{3, 5\}\},$$

which is the span of minimal weight for the complete graph with weights b_{ij}, is the optimal solution to the problem. Moreover, the weight of T (the lower bound L) coincides with the value of the objective function: $W(T) = L = 365$. Consider the minimal weight span

$$P = \{\{1, 2\},\ \{1, 3\},\ \{1, 4\},\ \{3, 5\}\}$$

for the graph with weights c_{ij}. Then $W(P) = 443 > W(T)$, while $C = 263 < L$. This example demonstrates that the passage to weights b_{ij} can prove useful both for improving the lower bound and for constructing a more accurate solution to Problem (1). In this example the lower bound L is 1.68 times the lower bound equal to the weight of the minimal span in the initial graph.

Now we propose a local improvement procedure applicable to an arbitrary spanning tree.

Consider an arbitrary spanning tree T and some numerical parameter d. In order to describe the local improvement procedure for T, we introduce some notation: Denote the current record tree by R and the tree obtained as the output of the procedure, by $D(T, d)$.

Initial step. Refer to an edge $(i, j) \in T$ as *black* whenever $\min\{a_{ij}, a_{ji}\} \geq d$, and as *white* otherwise. If all edges of the tree T turn out of one color then the algorithm stops, and the result is $D(T, d)$.

Suppose that the edges of T are of different colors. Denote its connected components formed by the white edges by $S_1, \ldots, S_M$. Put $R = T$.

In a loop for $m = 1, \ldots, M$ do

General step. Take the tree R and the component S_m. Denote by B the set of black edges incident to the vertices in S_m, and by N, the set of vertices incident to the edges in B and lying outside S_m.

For $s \in S_m$, put $E_s = \{(v, s),\ v \in N\}$ and $T_s = (R \setminus B) \cup E_s$. Choose from the trees R and T_s for $s \in S_m$ the best one and assign it the new R; that is, put

$$R = \arg\min\left\{W(R), \arg\min_{s \in S_m} W(T_s)\right\}.$$

Final step. Once the loop over m is complete, put $D(T, d) = R$.

Consider the minimal tree T obtained in calculating the lower bound (8). Put

$$d_i = \max_{j \in N_i(T)} a_{ij}, \qquad i \in V.$$

The tree

$$P = \arg\min\{W(D(T, d_i)),\ i \in V\}$$

is the required approximate solution to Problem (1).

5. SIMULATION

To analyze the effectiveness of the heuristic algorithm of Section 4, we ran a simulation placing the network elements randomly and uniformly inside the square of side length 100.

Denote by T_1 the minimal spanning tree on the graph with weights c_{ij}, and by T_2, the minimal spanning tree on the graph with weights b_{ij}; we determined them using the convex hulls of the functions g_i as the minorants. Denote by T_3 the minimal spanning tree on the graph with weights b_{ij}, which we determined using the functions g_i themselves as the minorants. Denote the weights of the trees T_1, T_2, and T_3 by L_1 L_2, and L_3.

We ran the experiment for $n = 5$, 10, 15, 20, 30, 40, 50, 60, 70, 80, and 90, and 100. For the same dimension we generated 50 distinct examples. For each example generated we started the algorithm with each of the three trees T_1, T_2, and T_3 as the initial tree. As the lower bounds we took $\max\{L_1, L_2, L_3\}$.

In order to construct the optimal solution, use the representation by a Steiner tree, proposed in [7], and write Problem (1) as an integer linear programming problem. To formulate the problem, assign one vertex of the graph (vertex 1) to be the root of the required tree T and assume that all arcs in T are directed away from the root. Put $V_j = \{i \in V \mid (i, j) \in E\}$. Denote by u_i the number of arcs from the root to the vertex i in T (and furthermore, $u_1 = 0$). Put $x_{ij} = 1$ if the arc (i, j) belongs to T and $x_{ij} = 0$ otherwise. Then we can express the problem as follows:

$$\sum_{i \in V} C_i \to \min_{x,u,C}, \tag{9}$$

$$c_{ij}x_{ij} \leq C_i, \qquad c_{ij}x_{ij} \leq C_j, \qquad i \in V, \quad j \in V \setminus \{1\}, \tag{10}$$

Table 1. Average values of relative accuracy

n	$\varepsilon(A(P))$	$\varepsilon(A(Q))$	$\varepsilon(A(T_1))$	$\varepsilon(A(T_2))$	$\varepsilon(A(T_3))$
5	1.02323	1.02199	1.02587	1.02323	1.02413
10	1.02046	1.02026	1.02545	1.02252	1.22374
15	1.03056	1.02703	1.02619	1.04297	1.45522
20	1.03592	1.03444	1.03303	1.08635	1.7503

$$\sum_{i\in V_j} x_{ij} = 1, \qquad j \in V \setminus \{1\}, \tag{11}$$

$$1 - n(1 - x_{ij}) \le u_j - u_i \le 1 + n(1 - x_{ij}), \qquad (i, j) \in E, \tag{12}$$

$$x_{ij} \in \{0, 1\}; \quad u_i \in \{1, \ldots, n-1\}, \quad i \ne 1; \qquad u_1 = 0. \tag{13}$$

To solve problem (9)–(13) for small dimensions $n \le 20$ we used the CPLEX package made available to the authors in the framework of the IBM Academic Initiative.

The branches-and-boundaries method of [3] is based on a different statement of Problem (1) as an integer linear programming problem. This method constructs the optimal solution on 30–40 vertices in acceptable time. The CPLEX package for (9)–(13) enabled us to solve Problem (1) on at most 20 vertices. For $n > 20$, we estimated the relative accuracy of the algorithm by

$$R = \frac{W(A)}{\max\{L_1, L_2, L_3\}}.$$

In order to demonstrate the results of our simulation, put

$$P = \arg\max\{L_1, L_2, L_3\}, \qquad Q = \arg \min_{T\in\{T_1,T_2,T_3\}} \{W(A(T))\}.$$

Denote the improvement of the solution by the algorithm by

$$I(T) = 100\% \cdot \frac{W(T) - W(A(T))}{W(T)}.$$

Tables 1–4 indicate that on average the algorithm constructs a 1.025-approximate solution for small dimensions $n \le 20$ and a 1.21-approximate solution for $n > 20$. For small dimensions ($n \le 10$) the local improvement algorithm with T_2 as the initial tree on average constructs a better solution than the same algorithm with the initial tree T_1, while, for $n = 5$, the quantity L_2 exceeds L_1 on average by more than 10%. For $n \ge 30$, the quantity L_1 is always the greatest of lower bounds for Problem (1), and we obtain the best solution if the initial tree is T_1; furthermore, for large n, in the case of an unfortunate choice of the initial tree, the value of the objective function on the resulting solution can exceed the optimal value by a factor of more than 5. We should also note that the estimate of the algorithm's quality changes slightly as n grows in the case that we choose T_1 as the initial tree, while T_2 and T_3 are reasonable choices only for small values of n.

In Table 4 we use the following notation: P_1 is the share of cases with $L_1 = \max\{L_1, L_2, L_3\}$, while P_2 is the share of cases with $L_2 = \max\{L_1, L_2, L_3\}$, in percent, $O(T)$ is the share of cases when algorithm A constructs the optimal solution with the initial tree T, in percent.

The graphs in Fig. 2 reflect the change in the running time of the proposed algorithm and CPLEX with the growth of dimension. Already for $n = 20$ the average running time of the CPLEX package is quite long: about 250 seconds. Fig. 3 depicts the changing share of the improvement of the solution by the algorithm with the growth of n for the choice of P as the initial tree. Observe that the data in Table 4 and Fig. 3 imply that, for $n \ge 20$, the algorithm improves the solution T_1 by 2–2.5% on average. Fig. 4 depicts the values of relative accuracy for all generated examples for the choice of T_1 as the initial tree.

Table 2. Average values of upper bounds for relative accuracy

n	$R(A(P))$	$R(A(Q))$	$R(A(T_1))$	$R(A(T_2))$	$R(A(T_3))$
30	1.23717	1.23072	1.23717	1.27233	2.63508
40	1.2309	1.2309	1.2309	1.45989	3.10877
50	1.21145	1.21145	1.21145	1.44256	3.40106
60	1.20451	1.20451	1.20451	1.53906	3.77661
70	1.20162	1.20162	1.20162	1.66828	4.22862
80	1.2069	1.2069	1.2069	1.74425	4.57752
90	1.20338	1.20338	1.20338	1.84641	4.94562
100	1.20327	1.20327	1.20327	1.90425	5.16939

Table 3. Average values of lower bounds and solutions constructed by the algorithm and the CPLEX package (W^* is the value of the objective function on the optimal solution)

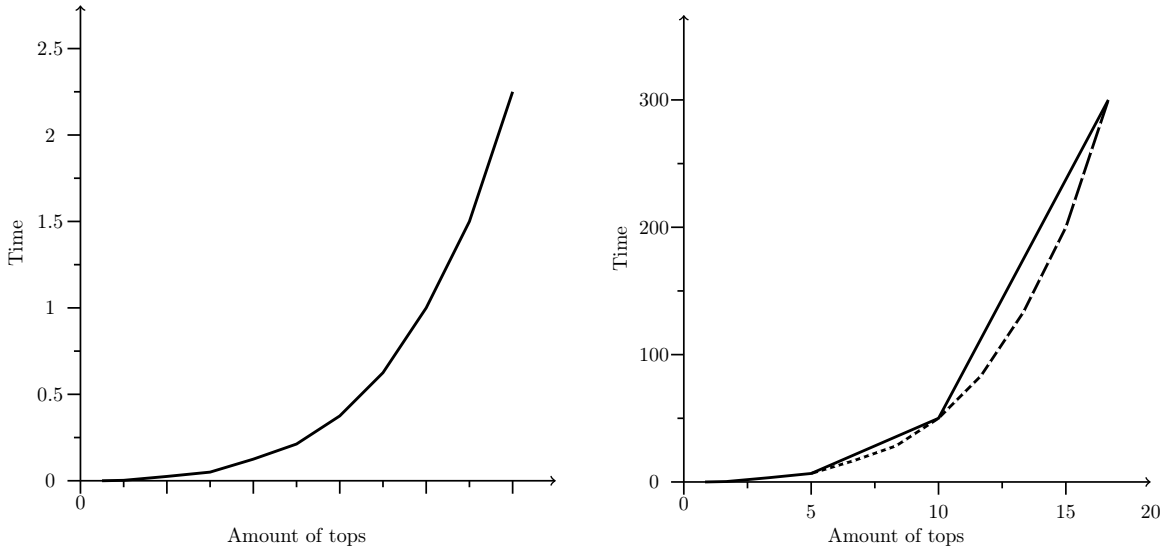
n	L_1	L_2	L_3	W^*	$W(A(T_1))$	$W(A(T_2))$	$W(A(T_3))$
5	137.831	151.929	141.164	196.182	201.213	200.661	201.205
10	205.192	200.587	183.04	260.276	267.26	266.347	318.325
15	255.102	244.644	217.732	314.196	322.573	327.784	455.07
20	306.074	287.666	253.887	370.694	383.06	402.655	645.289
30	380.788	353.942	313.516	—	470.951	484.616	999.965
40	431.685	387.242	340.200	—	531.051	629.013	1339.52
50	486.356	435.735	386.743	—	588.903	701.197	1652
60	528.040	466.710	414.695	—	635.942	812.682	1994.48
70	570.238	506.142	451.653	—	685.088	950.265	2409.22
80	606.016	528.798	472.553	—	731.176	1055.71	2769.35
90	640.283	555.639	497.559	—	770.364	1181.28	3160.63
100	671.691	577.833	519.057	—	808.079	1278.5	3469.25

6. CONCLUSIONS

In this article we gave a mathematical formulation of the NP-hard, in the strong sense, problem of combinatorial optimization which adequately reflects the problem of constructing communication networks in some technical systems (for instance, in wireless sensor networks). We found particular cases of polynomial solvability of the problem. We showed that the minimal span whose edge weights lie on the closed interval $[a, b]$ is a $\left(2 - \frac{2a}{a+b+2b/(n-2)}\right)$-approximate solution and that the problem of constructing a 1.00048-approximate solution is NP-hard. We proposed a heuristic polynomial algorithm. We ran a simulation demonstrating the high effectiveness of the algorithm proposed. On average it constructed a 1.025-approximate solution for small dimensions $n \leq 20$ and a 1.21-approximate solution for $n > 20$.

Table 4.

n	P_1	P_2	$O(P)$	$O(Q)$	$O(T_1)$	$O(T_2)$	$O(T_3)$
5	22	64	62	64	60	62	60
10	52	48	28	32	30	28	2
15	76	24	12	18	18	6	0
20	88	12	2	0	2	2	0
30	100	0	—	—	—	—	—
40	100	0	—	—	—	—	—
50	100	0	—	—	—	—	—
60	100	0	—	—	—	—	—
70	100	0	—	—	—	—	—
80	100	0	—	—	—	—	—
90	100	0	—	—	—	—	—
100	100	0	—	—	—	—	—



(a) (b)

Fig. 2. Average running time of algorithm A (a) and the CPLEX package (b)

The reason for this discontinuity is that to estimate the quality of the algorithm for large dimensions we used a lower bound instead of the optimal values of the objective function.

ACKNOWLEDGMENTS

The authors were supported by the Russian Foundation for Basic Research (projects nos. 12–01–33028_mol_a_ved and 13–07–00139–a) and the Federal Target Program "Scientific and Scientific-Pedagogical Personnel of Innovative Russia" for 2009–2013 (project no. 8227).

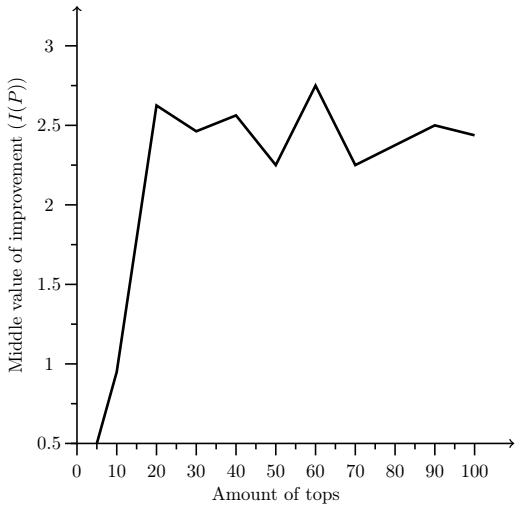


Fig. 3. The improvement of the tree P by the algorithm ($I(P)$)

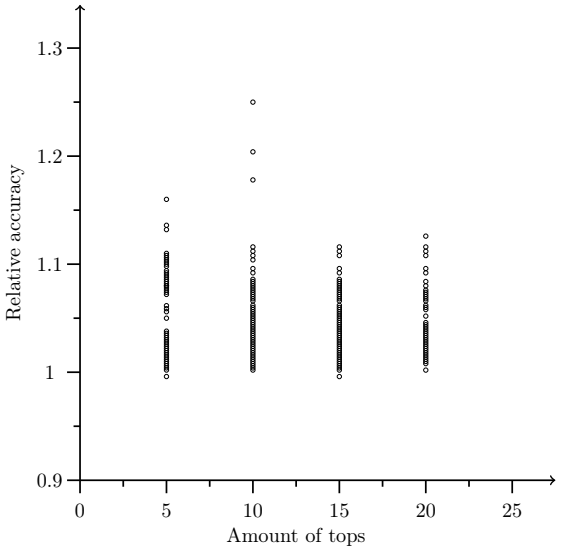


Fig. 4. All values of the ratio($A(T_1)$)

REFERENCES

1. S. N. Astrakov, A. I. Erzin, and V. V. Zalyubovskii, "Sensor Maps and Plane Covering with Rings," Diskret. Anal. Issled. Oper. **16** (3), 3–19 (2009).
2. T. F. Tot, *Layouts on a Plane, on a Sphere, and in Space* (Fizmatgiz, Moscow, 1958) [in Russian].
3. E. Althaus, et al. "Power Efficient Range Assignment for Symmetric Connectivity in Static Ad Hoc Wireless Networks," Wireless Networks **12** (3), 287–299 (2006).
4. P. Berman and M. Karpinski, "On Some Tighter Inapproximability Results," Electron. Colloquium Computational Complexity (ECCC), TR98–065 (1998).
5. A. E. F. Clementi, P. Penna, and R. Silvestri, "On the Power Assignment Problem in Radio Networks," Electron. Colloquium Computational Complexity (ECCC), TR00–054 (2000).
6. P. Carmi and M. L. Katz, "Power Assignment in Radio Networks with Two Power Levels," Algorithmica, No. 47, 183–201 (2007).
7. M. Diane and J. Plesnik, "An Integer Programming Formulation of the Steiner Problem in Graphs," Math. Methods Oper. Res. **37**, 107–111 (1993).
8. R. Kershner, "The Number of Circles Covering a Set," Amer. J. Math. **61** (3), 665–671 (1939).
9. L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power Consumption in Packet Radio Networks," Theor. Comput. Sci. No. 243, 289–305 (2000).
10. G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," Commun. ACM **43** (5), 51–58 (2000).
11. F. G. Tóth, "Covering the Plane with Two Kinds of Circles," Discrete Comput. Geometry **13** (3), 445–457 (1995).
12. J. Wu and F. Dai, "Virtual Backbone Construction in MANETs Using Adjustable Transmission Ranges," IEEE Trans. Mobile Comput. **5** (9), 1188–1200 (2006).
13. J. Wu and S. Yang, "Energy-Efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges," Int. J. Found. Comput. Sci. **16** (1), 3–17 (2005).
14. H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," Ad Hoc & Sensor Wireless Networks **1** (1–2), 89–124 (2005).