Information Technology and Management Science
ISSN 2255-9094 (online)
2023, vol. 26, pp. 17–26
https://doi.org/10.7250/itms-2023-0003
https://itms-journals.rtu.lv

1862
RIGA TECHNICAL UNIVERSITY

Building Minimum Spanning Trees under Maximum Edge Length Constraint

Vadim Romanuke[*]
Polish Naval Academy, Gdynia, Poland

***Abstract* – Given an initial set of planar nodes, the problem is to build a minimum spanning tree connecting the maximum possible number of nodes by not exceeding the maximum edge length. To obtain a set of edges, a Delaunay triangulation is performed over the initial set of nodes. Distances between every pair of the nodes in respective edges are calculated used as graph weights. The edges whose length exceeds the maximum edge length are removed. A minimum spanning tree is built over every disconnected graph. The minimum spanning trees covering a maximum of nodes are selected, among which the tree whose length is minimal is the solution. It is 1.17 % shorter on average for 10 to 80 nodes compared to a nonselected tree.**

***Keywords* – Delaunay triangulation, maximum edge length constraint, minimum spanning tree, root node.**

I. Minimum Spanning Trees under Constraints

The task of building minimum spanning trees arises when complex networks are designed in order to connect a set of nodes [1], [2]. Usually these are supply networks which should cover its nodes by paying the least possible cost of the coverage [1], [3], [4]. To the contrary, the number of covered nodes is naturally desired to be maximum possible.

The coverage cost is mainly determined by distances between every pair of nodes directly linked in the network [5], [6]. The sum of these distances is to be minimized. Built over a set of nodes connected with edges, a minimum spanning tree is a subset of the edges of an undirected graph that connects all the nodes without any cycles and with the minimum possible total edge length [5], [7], [8]. In other words, the minimum spanning tree algorithm connects all the nodes by minimizing the cost of the connection [1], [5], [9], [10].

In many practical problems, building minimum spanning trees is constrained to using a limited number of connections (edges). Besides, the maximum edge length cannot be unlimited. For instance, while building a network of electrical power converters supplying the desired voltage to industrial and individual customers from electrical power stations, the length of any direct connection between two nodes is limited in order to minimize electrical loss and leak [1], [11], [12]. A similar constraint is imposed on wired tree circuits [4], [6], computer [13], [14], measurement [15], broadcasting [16], telecommunication [17], and transportation networks [18].

II. Related Works and Motivation

Borůvka was the first who solved the minimum spanning tree problem dealing with a task of efficient electrical coverage [19]. The Borůvka's algorithm repeats finding the shortest-length edge incident to each node of the graph, and adding all of those edges to the forest. Each repetition reduces the number of trees, within each connected component of the graph, to at most half of this former value. Another two commonly used algorithms for finding a minimum spanning tree are the Prim's algorithm and Kruskal's algorithm. Step by step, the Prim's algorithm builds a tree by adding the cheapest possible connection from the currently built tree to another node [20], [21]. The starting node is arbitrary. The Kruskal's algorithm builds a tree by adding at each step the next shortest-length edge that will not form a cycle to the minimum spanning forest [6], [22], [23]. At the termination of the algorithm, the forest forms a minimum spanning forest of the connected graph, for which the forest has a single component and forms a minimum spanning tree. The algorithms have nearly the same asymptotic time complexity varying from linear to polynomial [24], but the Prim's algorithm is claimed to perform better on dense graphs [25], whereas the Kruskal's algorithm is believed to perform acceptably on sparser graphs [1], [23], [26]. However, building minimum spanning trees under maximum edge length constraint has not been studied yet. Moreover, the algorithms perform on a given set of edges (connections), which in real-world scenarios are yet to be determined for an initial set of nodes (locations) [27].

III. The Goal and Objectives

Given a set of N planar nodes (points whose locations are two-component vectors), the problem is to build a minimum spanning tree connecting the maximum possible number of nodes by not exceeding the maximum edge length. The root node, which is to be unconditionally included into the tree, is assigned at the start. This is, for instance, the starting node in the Prim's algorithm. The root node can be used as an identifier of a minimum spanning tree, if there are a few such trees. Although such an identifier will be not unique for a tree, disconnected trees will be identified uniquely. To achieve this goal, the following eight objectives are to be fulfilled:

[*] Corresponding author. E-mail: romanukevadimv@gmail.com

Article received 01.08.2023; accepted 27.10.2023

1. To define an initial set of planar nodes and describe how its topology is specified.
2. To suggest a way of obtaining a set of edges based on the given initial set of planar nodes.
3. To define the metric, by which the edge length is calculated. The lengths are to be used as weights of a graph, for which a minimum spanning tree will be built.
4. To impose a maximum edge length constraint on the problem solution. To suggest a way, by which the constraint is applied to the set of edges obtained from the given initial set of planar nodes.
5. To suggest a method of solving the problem under the maximum edge length constraint.
6. To compare the suggested method performance to solving the problem straightforwardly, without considering the root node (tree identifier) change.
7. To discuss the scientific and practical importance of the suggested method. Its drawbacks are to be described as well.
8. To conclude the study and outline what can be advanced in further research on building minimum spanning trees under constraints.

IV. Initial Set of Planar Nodes

Initially, as it is common in practice, no edges are given, but a set of N planar nodes

$$\left\{\mathbf{P}_i\right\}_{i=1}^{N} = \left\{\left[x_i \quad y_i\right]\right\}_{i=1}^{N} \subset \mathbb{R}^2 \tag{1}$$

is given, on which a minimum spanning tree is to be built by not exceeding the maximum edge length denoted by $l_{\max}$. Components x_i and y_i are horizontal and vertical coordinates of node $\mathbf{P}_i \in \mathbb{R}^2$. Although the topology of set (1) is not generally specified, the distance between the pair of any two nodes is calculated using the common Euclidean metric in $\mathbb{R}^2$ [28], [29].

V. Maximum Edge Length Constraint

To obtain a set of edges, over which a minimum spanning tree can be built, set (1) is triangulated. The triangulation is performed by the Delaunay approach [30], [31], mostly excluding sliver triangles [32]. Besides, the Delaunay triangulation does not maximize the edge-length of the triangles [33] that fits the constraint of not exceeding the maximum edge length. Instead, the Delaunay triangulation maximizes the minimum of all the angles of the triangles [34], [35]. This property usually shows up (Fig. 1), but sometimes the slivering is not avoidable (Fig. 2).

Upon the nodes of set (1) are triangulated, a set of Q edges

$$E = \left\{\mathbf{E}_q\right\}_{q=1}^{Q} = \left\{\left[j_q \quad k_q\right]\right\}_{q=1}^{Q} \tag{2}$$

emerges, where edge $\mathbf{E}_q$ is determined by nodes $\mathbf{P}_{j_q}$ and $\mathbf{P}_{k_q}$ connected by this edge for

$$j_q \in \left\{\overline{1, N}\right\} \text{ and } k_q \in \left\{\overline{1, N}\right\} \text{ by } j_q \neq k_q \quad \forall q = \overline{1, Q}. \tag{3}$$

The number of edges connecting planar nodes after they are triangulated is not necessarily the same for a given N. For

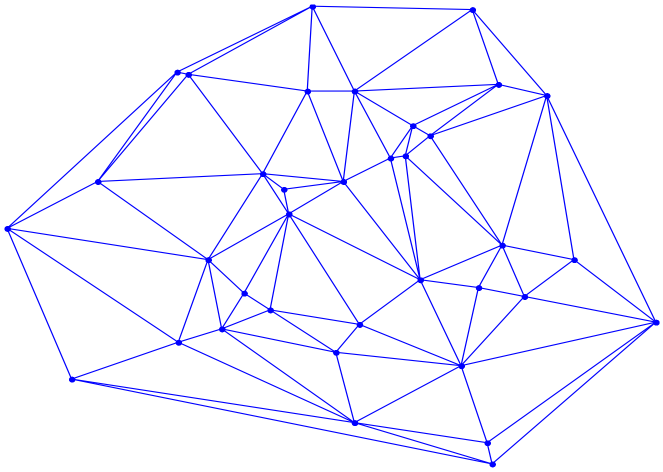

Fig. 1. A triangulated set of 36 nodes. The Delaunay triangulation “prefers” non-sliver triangles, although here are a few triangles with pretty sharp angles.

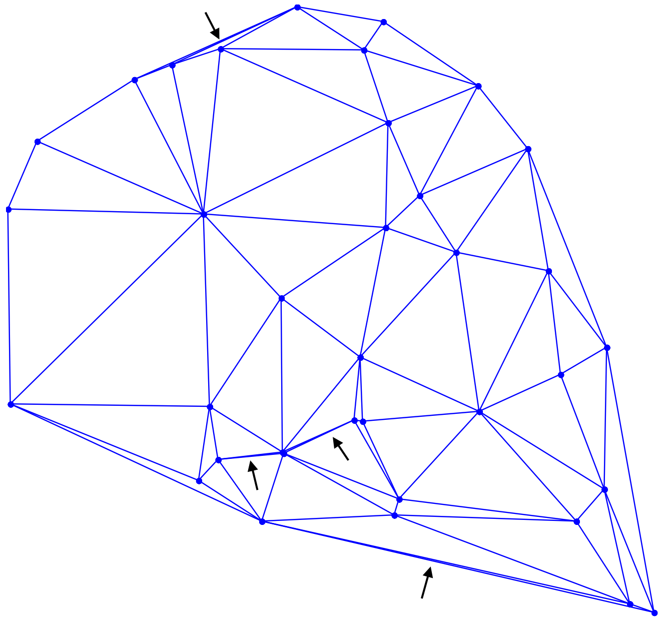

Fig. 2. An example of triangulating another set of 36 nodes, where the slivering is not avoidable. Here are (at least) four extremely sliver triangles (the arrows point at them), which visually almost stretched into their edges.

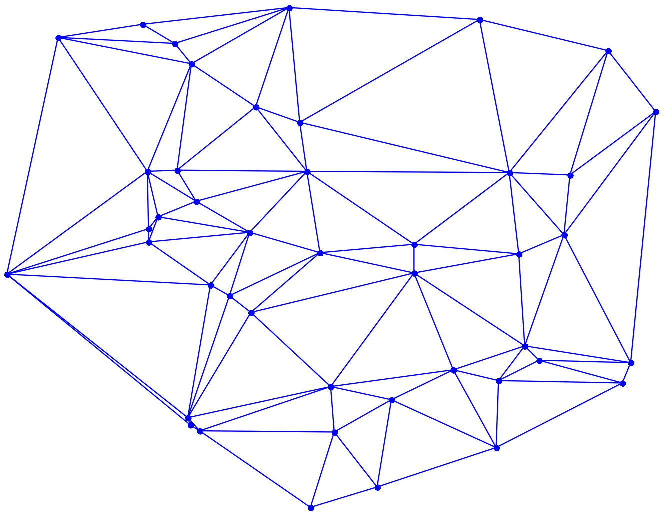

Fig. 3. A set of 115 edges (2) is a result of triangulating 44 nodes (Q = 115).

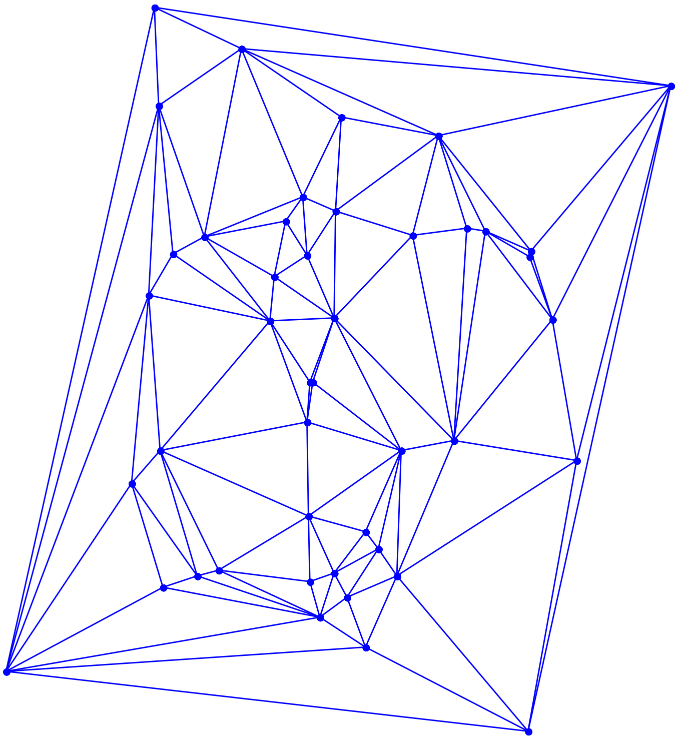

Fig. 4. A set of 125 edges (2) is a result of triangulating another set of 44 nodes ($Q = 125$). Unlike that in Fig. 3, a few sliver triangles are noticeable here. It is also noticeable that the set of these 44 nodes is kind of inscribed into a quadrangle being close to a rectangle. The quadrangle is the convex hull of the node set. In general, as the convex hull of the node set becomes more rectangle-shaped, the number of edges in set (2) tends to increase.

instance, there are 115 edges upon 44 nodes are triangulated (Fig. 3), but triangulating other 44 nodes results in 125 edges (Fig. 4). It would be generally correct to claim that set (2) depends on the shape of planar data.

By the Euclidean metric in $\mathbb{R}^2$, the length of edge $\mathbf{E}_q$ is

$$l_{\mathbb{R}^2}\left(\mathbf{E}_q\right) = \rho_{\mathbb{R}^2}\left(\mathbf{P}_{j_q}, \mathbf{P}_{k_q}\right) = \\ = \sqrt{\left(x_{j_q} - x_{k_q}\right)^2 + \left(y_{j_q} - y_{k_q}\right)^2} \text{ for } q = \overline{1, Q}. \tag{4}$$

If there was no maximum edge length constraint, a minimum spanning tree would be built for a graph with edges (2) and their respective weights

$$\left\{l_{\mathbb{R}^2}\left(\mathbf{E}_q\right)\right\}_{q=1}^{Q}. \tag{5}$$

In this case, the constraint still can exist, but only $l_{\max}$ is set either to a sufficiently great value (intentionally or not) or, more formally (intentionally), at infinity (Fig. 5). In a general case, those nodes whose distances between exceed $l_{\max}$ cannot be connected. Thus, the respective edges should be removed. Therefore, a subset of edges

$$\bar{E} = \left\{\bar{\mathbf{E}}_u\right\}_{u=1}^{U} = \left\{\left[j_u \quad k_u\right]\right\}_{u=1}^{U} = \\ = \left\{\mathbf{E}_q : l_{\mathbb{R}^2}\left(\mathbf{E}_q\right) \leqslant l_{\max}, q = \overline{1, Q}\right\} \subset E \tag{6}$$

is formed. Subset (6) contains only edges not exceeding $l_{\max}$, and then a minimum spanning tree is built over this subset.

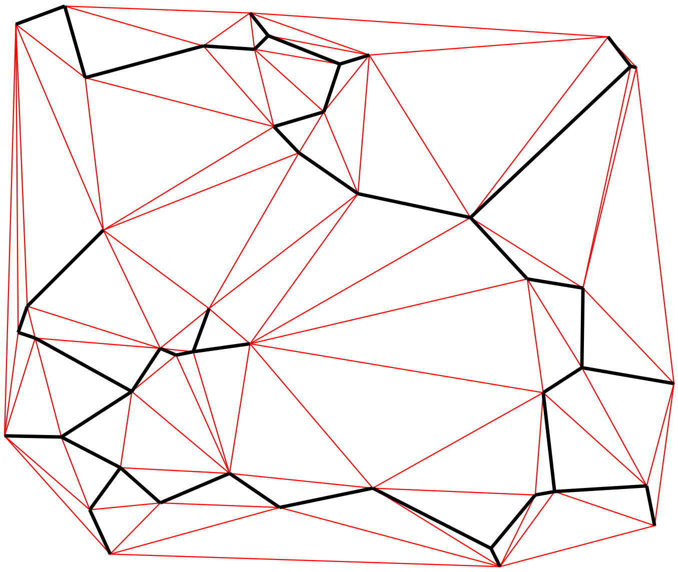

Fig. 5. A minimum spanning tree (highlighted bold) built over 47 nodes primarily triangulated. None of the initial $Q = 128$ edges is removed due to $l_{\max}$ is sufficiently great. It also formally means that subset (6) coincides with set (2), i. e. $U = 128$.

Under too short length $l_{\max}$, however, subset (6) may be torn into two or more disconnected sets (graphs). For instance, if L_* is the total edge length and for the set of 47 nodes in Fig. 5

$$l_{\max} = \frac{2L_*}{N-1} = \frac{L_*}{23}, \tag{7}$$

then 48 out of 128 edges are removed and subset (6) is of 80 edges, but this subset is torn into three disconnected graphs (see Fig. 6, where the removed edges are dash-lined). So, building a minimum spanning tree over subset (6) may cause uncertainty when such cases of the disconnection occur.

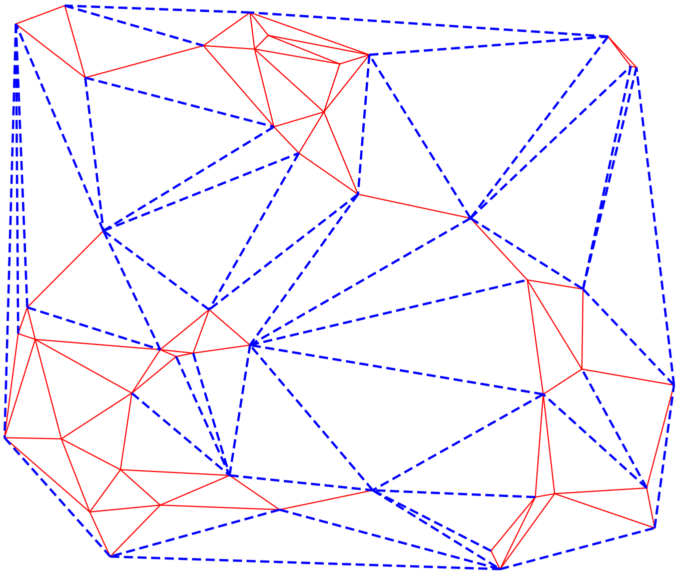

Fig. 6. Subset (6) for 47 nodes in Fig. 5 under maximum edge length (7). The subset is a union of the three disconnected graphs. One of these graphs is of just three edges (the close-to-be-sliver triangle in the top right).

VI. Running through Root Nodes

Let subset (6) be a set of T separate (disconnected) graphs, where graph t is of $W(t)$ edges of set

$$\overline{E}^{(t)} = \left\{\overline{\mathbf{E}}_w^{(t)}\right\}_{w=1}^{W(t)} \subset \overline{E} \tag{8}$$

and

$$\overline{E} = \bigcup_{t=1}^{T} \overline{E}^{(t)} \text{ by } \overline{E}^{(t_1)} \bigcap \overline{E}^{(t_2)} = \varnothing$$

$$\forall t_1 = \overline{1, T} \text{ and } \forall t_2 = \overline{1, T} \text{ for } t_1 \neq t_2 . \tag{9}$$

In a particular case, when subset (6) forms a single connected graph, $T = 1$ and $W(1) = U$, $\overline{E}^{(1)} = \overline{E}$. Let the edges in set (8) connect $M(t)$ nodes

$$\left\{\mathbf{P}^{(t)}(m)\right\}_{m=1}^{M(t)} \subset \left\{\mathbf{P}_i\right\}_{i=1}^{N}, \tag{10}$$

i. e., tree t is built over $M(t)$ nodes (10), where the nodes for each tree are indexed separately. Let us denote by $L_*(t)$ the total edge length of tree t built for graph t with a root node

$$\mathbf{P}_0^{(t)}(m_0) \in \left\{\mathbf{P}^{(t)}(m)\right\}_{m=1}^{M(t)} \text{ by } m_0 \in \left\{\overline{1, M(t)}\right\}. \tag{11}$$

Note that $L_*(t)$ does not depend on the root node. Inasmuch as the best tree should connect the maximum possible number of nodes, only trees that cover the maximum number of nodes are considered further. Thus, the maximal integer is selected from the set of all possible numbers $\left\{M(t)\right\}_{t=1}^{T}$. Subset

$$T_{\max} = \arg\max_{t=\overline{1, T}} M(t) \tag{12}$$

contains all indices of trees that cover the maximum number of nodes. Therefore, the maximum number of nodes is

$$M_{\max} = M(t_{\max}) \text{ for } t_{\max} \in T_{\max} . \tag{13}$$

Then the minimum of total edge length $L_*(t)$ is searched over trees with index $t = t_{\max} \in T_{\max}$ by running through any $|T_{\max}|$ root nodes

$$\left\{\mathbf{P}_0^{(t_{\max})}(m_0)\right\}_{t_{\max} \in T_{\max}}$$

in these trees, i. e., by taking a single node $m_0 \in \left\{\overline{1, M_{\max}}\right\}$ in every tree and calculating its $L_*(t_{\max})$. Thus, a subset

$$T_{\max}^* = \arg\min_{t_{\max} \in T_{\max}} L_*(t_{\max}) \subset T_{\max} \tag{14}$$

of indices of trees covering $M_{\max}$ nodes with the minimum of total edge length is found. Every tree $t_{\max}^* \in T_{\max}^*$ having the same length $L_{**} = L_*\left(t_{\max}^*\right)$ in subset (14) is a solution to the problem under the maximum edge length constraint.

Obviously, if $T = 1$ then $\left|T_{\max}^*\right| = \left|T_{\max}\right| = 1$ and the problem is solved by just calculating the single minimum spanning tree length $L_*(1)$. In this case, no maximum number of covered nodes by (12) is searched because it is already known as $M_{\max} = M(1)$. In other cases, when $T > 1$, the maximum number of covered nodes is found by (12) and (13), whereupon the set of all tree lengths $\left\{L_*(t_{\max})\right\}_{t_{\max} \in T_{\max}}$ is determined after (14) by running through $|T_{\max}|$ root nodes.

To compare how the suggested method performs versus solving the problem by straightforwardly (randomly or taking the first tree) picking a tree out of $|T_{\max}|$ trees (when $|T_{\max}| > 1$), the number of initial nodes is varied as

$$N \in \left\{10, 20, 30, 40, 50, 60, 70, 80\right\}. \tag{15}$$

The maximum edge length is set to

$$l_{\max} = \frac{1}{\beta Q} \cdot \sum_{q=1}^{Q} l_{\mathbb{R}^2}\left(\mathbf{E}_q\right) \tag{16}$$

by a constraint factor

$$\beta \in \left\{1, 1.1, 1.2, 1.3, 1.4, 1.5\right\}. \tag{17}$$

For $\beta = 1$ the maximum edge length by (16) is the average of the initial set of edge lengths (5) following the Delaunay triangulation. For each pair $\{N, \beta\}$ the initial set of nodes (1) is generated as

$$\begin{bmatrix} x_i & y_i \end{bmatrix} = \begin{bmatrix} 80 \cdot \theta_i + \vartheta_i + 50 & 80 \cdot \xi_i + \zeta_i + 50 \end{bmatrix}$$

$$\text{for } i = \overline{1, N}, \tag{18}$$

where θ_i, ξ_i are values of two independent random variables distributed uniformly on the open interval $(0; 1)$ and ϑ_i, ζ_i are values of two independent random variables distributed normally with unit variance and zero mean [36], [37]. For every triple

$$\left\{N, \left\{\mathbf{P}_i\right\}_{i=1}^{N}, \beta\right\} \tag{19}$$

the problem is regenerated for 100 times. An example of a problem with 50 initial nodes for $\beta = 1$ has two separate trees, and both cover 25 nodes. However, one tree has the total edge length of 209.8151 (Fig. 7), whereas the other one has the total edge length of 196.6618 (Fig. 8) being 6.269 % shorter. In this example, the maximal number of disconnected trees covering a maximum number of nodes is 2: $|T_{\max}| = 2$.

In the 4800 results of the computational set-up by (15)–(18), the maximal number of disconnected trees covering a maximum number of nodes varies between 1 and 5:

$$|T_{\max}| \in \left\{1, 2, 3, 4, 5\right\}.$$

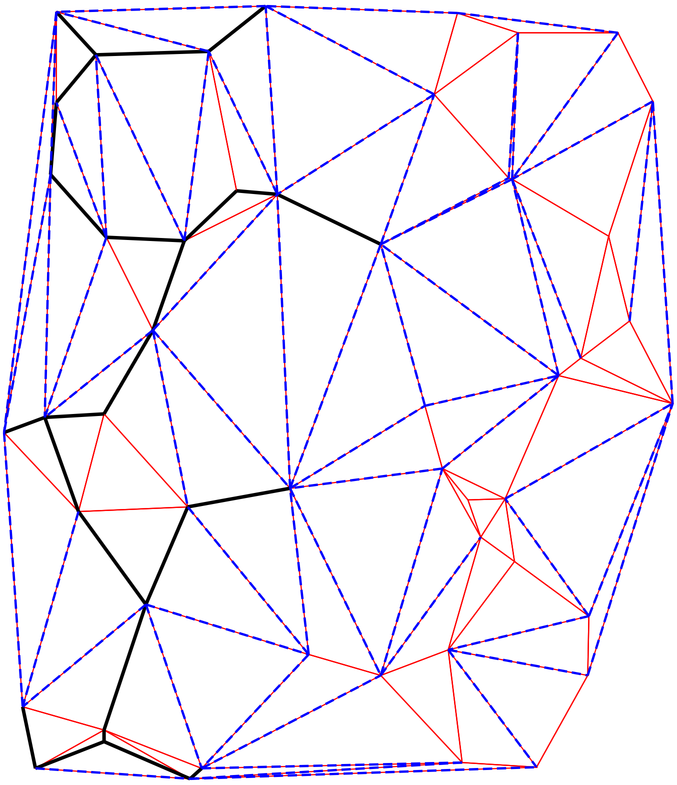

Fig. 7. The minimum spanning tree (highlighted bold; the removed edges are dash-lined) of one of two disconnected graphs for a problem with 50 nodes ($N = 50$) and β = 1. The tree covers exactly a half of the nodes ($M_{\max} = 25$) and its total edge length is 209.8151.

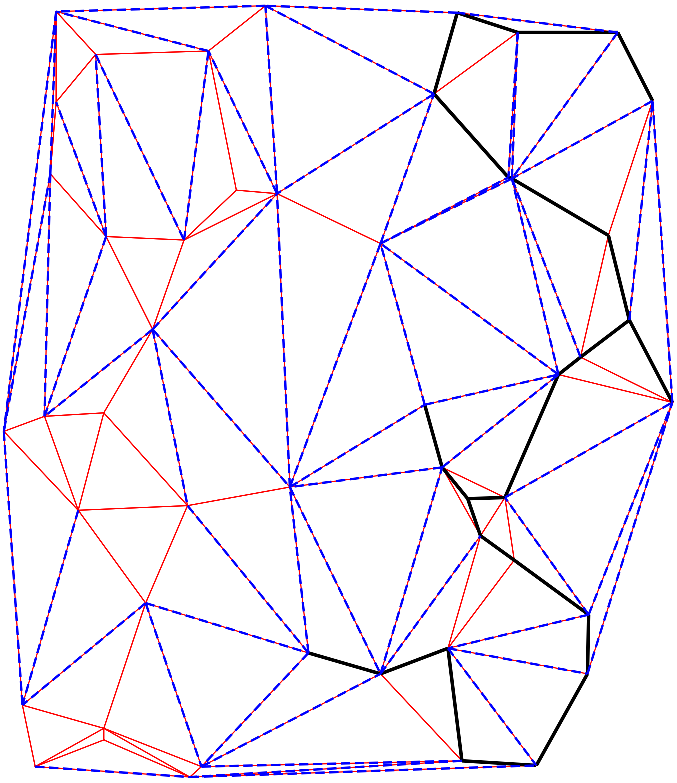

Fig. 8. The minimum spanning tree of the other disconnected graph for the problem with 50 nodes and β = 1 in Fig. 7. The tree covers the other half of the nodes ($M_{\max} = 25$), but its total edge length is 196.6618. This tree is 6.269 % shorter than that in Fig. 7.

There are 4523 problem instances with $|T_{\max}| = 1$ and 233 instances with $|T_{\max}| = 2$. The tree triple covering a maximum number of nodes ($|T_{\max}| = 3$) has occurred rarely – it is just 34 times, and the tree quadruple ($|T_{\max}| = 4$) has been registered 8 times. The rarest instance of five trees covering a maximum number of nodes ($|T_{\max}| = 5$) occurred only 2 times, both by the severest maximum edge length constraint (β = 1.5): one with 10 nodes and the other one with 30 nodes (Fig. 9). The latter is solved to a tree whose length is 21.6192, while the lengths of the other four trees are 24.4593, 24.6687, 28.8194, 32.6425 (listed in ascending order of their least root nodes). Without considering the root node change, the average loss in the total edge length would be

$$\frac{21.6192 + 24.4593 + 24.6687 + 28.8194 + 32.6425}{5 \cdot 21.6192} \approx 1.2231 .$$

The maximal loss is far more impressive:

$$\frac{\max\{21.6192,\, 24.4593,\, 24.6687,\, 28.8194,\, 32.6425\}}{21.6192} \approx 1.5099 .$$

In general, the average and maximal losses being relevant for those 277 instances of $|T_{\max}| > 1$ are respectively calculated as

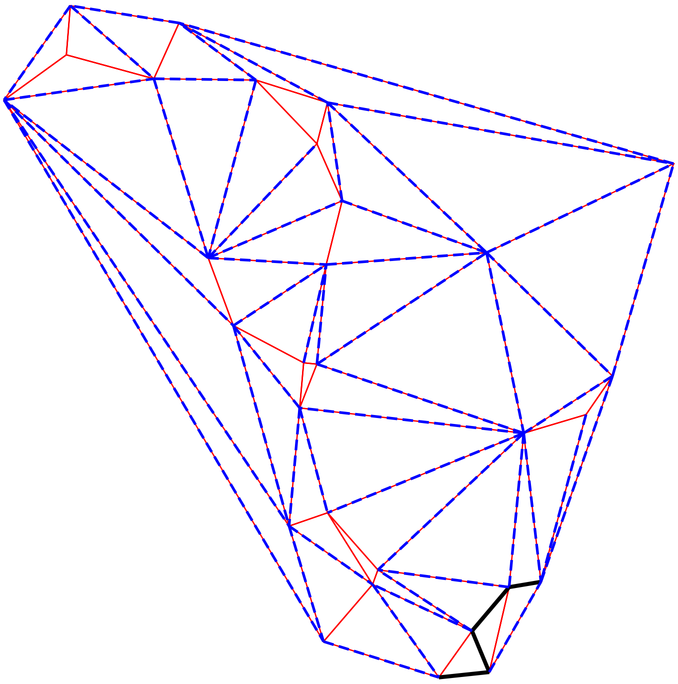

Fig. 9. The instance of 30 nodes and β = 1.5 (the severest maximum edge length constraint), for which the minimum spanning tree (highlighted bold) covers 5 nodes ($M_{\max} = 5$). The tree has the total edge length of 21.6192, while the other four trees covering 5 nodes have lengths 24.4593, 24.6687, 28.8194, 32.6425. These trees are easily spotted due to each of them comprises 4 edges. The remaining, sixth, tree comprises just 2 edges, and therefore $|T_{\max}| = 5$ here. Compared to the solution tree, the four trees are longer by 13.137, 14.1055, 33.3045, and 50.9887 %, respectively. These are indeed very significant differences, so determining the shortest tree in this instance would be quite important and efficient in practice.

$$\frac{\sum\limits_{t_{\max}\in T_{\max}} L_*\left(t_{\max}\right)}{\left|T_{\max}\right|\cdot L_{**}} \tag{20}$$

and

$$\frac{\max\limits_{t_{\max}\in T_{\max}} L_*\left(t_{\max}\right)}{L_{**}} \tag{21}$$

for every triple (19). Average losses (20) averaged over 100 instances (19) are presented in Table I. Here and below maximal values over the number of nodes are highlighted bold; maximal values over the constraint factor are highlighted dim. These values show that the average loss tends to increase as the maximum edge length constraint is made severer. The rightmost column, however, shows that the average loss decays as the number of nodes increases. Nevertheless, the overall average (here and below highlighted with larger font) implies that, without solving problem (14), the solution tree is more than 1 % longer on average. Table II presenting maximal losses (21) averaged over 100 instances (19) confirms these trends. The bold-and-dim patterns in Tables I and II coincide. The averaged maximal loss is also at 10 nodes by the severest maximum edge length constraint. The overall average is 1.0233 that is the averaged maximal loss approaches 2.33 %. It is noteworthy that there are 300 instances (50, 60, 80 nodes for β = 1.1, β = 1, β = 1, respectively) solved by $\left|T_{\max}\right| = 1$.

TABLE I
Average Losses (20) Averaged over 100 Instances (19)

β N	1	1.1	1.2	1.3	1.4	1.5	Average
10	1.0052	1.0157	**1.0544**	**1.0551**	**1.0514**	**1.0722**	**1.0423**
20	**1.0066**	**1.0276**	1.0072	1.0142	1.0244	1.0418	1.0203
30	1.0065	1.003	1.0094	1.0092	1.0256	1.0195	1.0122
40	1.0006	1.0047	1.0004	1.0044	1.0048	1.0112	1.0043
50	1.0003	1	1.0016	1.0064	1.0042	1.003	1.0026
60	1	1.0004	1.0006	1.0075	1.0077	1.0144	1.0051
70	1.0013	1.0009	1.0005	1.0024	1.0098	1.0049	1.0033
80	1	1.0018	1.0003	1.0022	1.0073	1.0065	1.003
Average	1.0026	1.0067	1.0093	1.0127	1.0169	1.0217	1.0117

TABLE II
Maximal Losses (21) Averaged over 100 Instances (19)

β N	1	1.1	1.2	1.3	1.4	1.5	Average
10	1.0104	1.0323	**1.1097**	**1.116**	**1.1081**	**1.138**	**1.0858**
20	**1.0133**	**1.0488**	1.0145	1.0285	1.0491	1.0891	1.0405
30	1.013	1.0061	1.0189	1.0161	1.0474	1.0411	1.0238
40	1.0012	1.0093	1.0008	1.0079	1.0099	1.0227	1.0086
50	1.0007	1	1.0032	1.0128	1.0084	1.006	1.0052
60	1	1.0007	1.0012	1.0149	1.015	1.0258	1.0096
70	1.0026	1.0018	1.0011	1.005	1.0206	1.0099	1.0068
80	1	1.0034	1.0007	1.0044	1.0146	1.0131	1.006
Average	1.0051	1.0128	1.0188	1.0257	1.0341	1.0432	1.0233

The worst-case scenario cannot be excluded, too. Thus, Table III presents maxima of average losses (20) over 100 instances (19). The bold-and-dim pattern slightly differing from that in Tables I and II fits the abovementioned loss trends. The hugest loss has occurred for an instance of 20 nodes and β = 1.1; the minimal maximum of average losses (20) has occurred for an instance of 60 nodes and the same constraint factor. The overall average implies that, without solving problem (14), the solution tree is more than 42 % longer in the worst case of the average loss. Table IV presenting maxima of maximal losses (21) over 100 instances (19) has the same worst-case scenario pairs of N and β. The worst loss is 5.217, but there are three trees ($\left|T_{\max}\right| = 3$) covering just 4 nodes ($M_{\max} = 4$). More practically valuable are the instances with 40 to 80 nodes, where more nodes are eventually covered by the best tree (just like the instance illustrated in Figs. 7 and 8), although the potential loss is not that huge. Such an instance is illustrated in Fig. 10 for 70 nodes and β = 1.3, where the best tree covers 15 nodes having the length of 70.6549 (the lengths of the other two trees covering the same number of nodes are 78.7802 and 91.191, listed in ascending order of their least root nodes; the trees are shown in Fig. 11 in a simplified presentation style). Herein, the maximal loss is 1.2907 and the average loss is 1.1352 (both are seen in Tables IV and III), which are very significant if solving problem (14) is ignored. In fact, there are many such examples among those 27 instances.

TABLE III
Maxima of Average Losses (20) over 100 Instances (19)

β N	1	1.1	1.2	1.3	1.4	1.5	Average
10	1.2546	1.7384	**2.5371**	**2.5371**	**2.4575**	**2.9739**	**2.2498**
20	1.327	**3.426**	1.7246	1.9052	1.4753	1.6537	1.9186
30	**1.5025**	1.1019	1.3496	1.6758	1.6758	1.3873	1.4488
40	1.0589	1.2047	1.0283	1.1956	1.1913	1.2808	1.1599
50	1.0334	1	1.1035	1.212	1.1856	1.1272	1.1103
60	1	1.0204	1.0612	1.1673	1.2028	1.6735	1.1875
70	1.1288	1.0876	1.0401	1.1352	1.4674	1.2397	1.1831
80	1	1.0946	1.0332	1.1483	1.1651	1.1691	1.1017
Average	1.1632	1.4592	1.3597	1.4971	1.4776	1.5631	1.42

TABLE IV
Maxima of Maximal Losses (21) over 100 Instances (19)

β N	1	1.1	1.2	1.3	1.4	1.5	Average
10	1.5093	2.5735	**4.0742**	**4.0742**	**3.9216**	**4.5607**	**3.4522**
20	1.6541	**5.217**	2.4492	2.8104	1.9507	2.3074	2.7315
30	**2.005**	1.2038	1.6991	2.1227	2.1227	1.8851	1.8397
40	1.1178	1.4094	1.0565	1.3912	1.4057	1.5615	1.3237
50	1.0669	1	1.207	1.424	1.3713	1.2543	1.2206
60	1	1.0407	1.1224	1.3346	1.4056	2.0235	1.3211
70	1.2575	1.1752	1.0803	1.2907	1.9348	1.4794	1.3696
80	1	1.1892	1.0665	1.2966	1.3301	1.3382	1.2034
Average	1.3263	1.8511	1.7194	1.968	1.9303	2.0513	1.8077

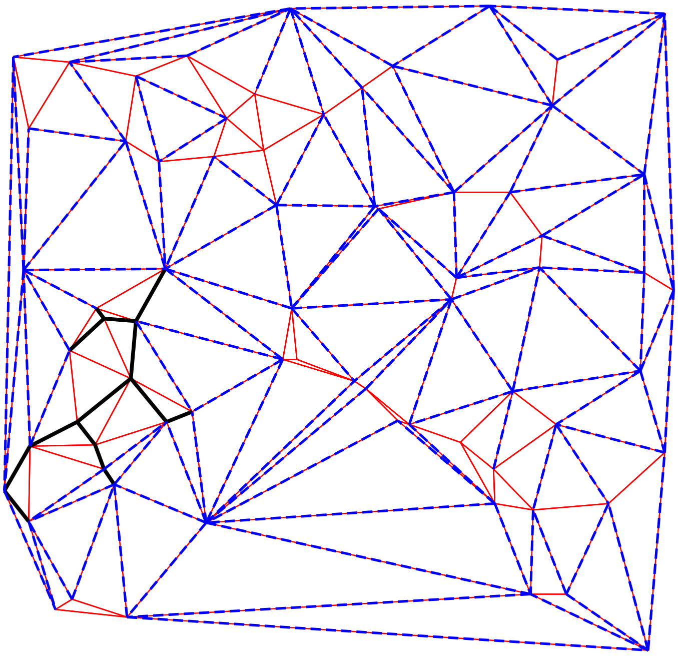

Fig. 10. The instance of 70 nodes and β = 1.3, for which the minimum spanning tree (highlighted bold) covers 15 nodes ($M_{\max}$ = 15). The tree has the total edge length of 70.6549, while the other two trees (shown in Fig. 11) covering 15 nodes have lengths 78.7802 and 91.191. Compared to the solution tree, the two trees are longer by 11.5 and 29.07 %, respectively.

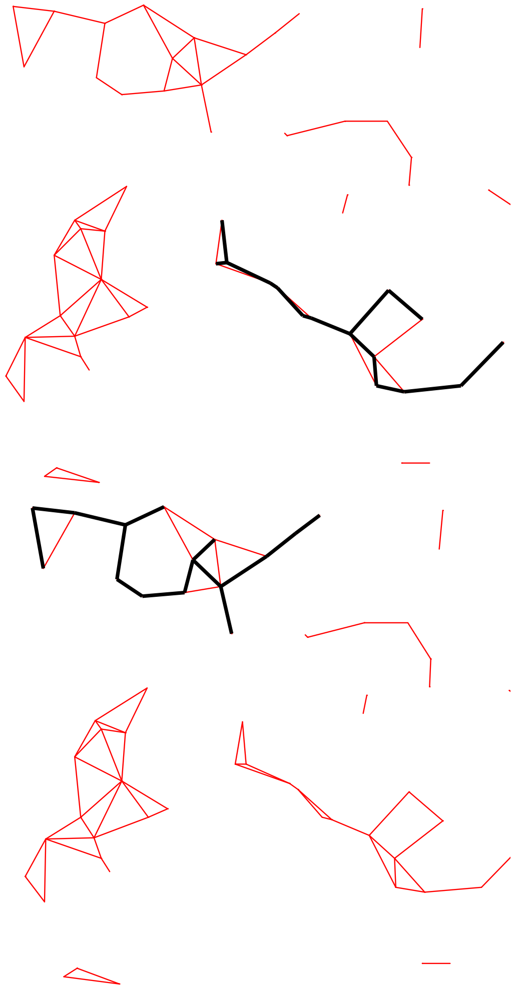

Fig. 11. The two trees in Fig. 10, where each covers 15 nodes, but is longer by 11.5 % (top) and 29.07 % (bottom). The removed nodes are not shown.

The bold-and-dim patterns in Tables III and IV also coincide. The overall average is 1.8077, i. e., is the averaged maximum of maximal loss exceeds 80 %, which is quite significant. All those results in Tables I–IV remain roughly the same in their trends if the part of normal distribution in generating node locations is increased. If the part is 10 times increased, that is

$$\begin{bmatrix} x_i & y_i \end{bmatrix} = \begin{bmatrix} 80 \cdot \theta_i + 10 \cdot \vartheta_i + 50 & 80 \cdot \xi_i + 10 \cdot \zeta_i + 50 \end{bmatrix} \quad \text{for } i = \overline{1, N}, \tag{22}$$

there are 4592 problem instances with $|T_{\max}| = 1$, 186 instances with $|T_{\max}| = 2$, 17 instances with $|T_{\max}| = 3$, and 3 instances with $|T_{\max}| = 4$. The rarest instance of five trees covering a maximum number of nodes ($|T_{\max}| = 5$) occurred only 2 times, both for 10 nodes, but by β = 1.4 and β = 1.5 – unlike the instances generated by (18). The overall averages are less than those in Tables I–IV (now they are 1.0092, 1.0175, 1.3204, 1.5908, respectively), but the general pattern described by Tables I–IV remains.

For larger sets of initial nodes this pattern remains also, but the losses are weaker and rarer. Thus, when the number of initial nodes is varied as

$$N \in \{100, 200, 300, 400\} \tag{23}$$

by constraint factor (17) and (22), and the problem is regenerated for 100 times for every triple (19), the overall averages of average losses (20) and of maximal losses (21) are 1.0009 and 1.0018 being far less than those in Tables I and II. The overall averages of maxima of average losses (20) and of maxima of maximal losses (21) are 1.045 and 1.0899, though. Thus, in the worst-case scenario of a larger initial set of planar nodes, the worst average loss is about 4.5 %, whereas the worst maximal loss (on average) is about 8.99 %. The relationship between these two overall averages is almost the same as it is for problems with (15) generated by (18) and problems with (15) generated by (22): the worst maximal loss percentage is nearly twice as greater compared to the worst average loss. For problems with (23) generated by (22), however, the worst average and worst maximal losses are about 10 times as lesser compared to the overall averages in Tables III and IV. Nevertheless, there is an instance of 400 nodes by β = 1.5 (Fig. 12), for which the maximal loss by (21) is 1.324 (or, in percentage terms, 32.4 %). This is the instance with $|T_{\max}| = 2$, where the solution tree length is 65.0188, and the other tree has the length of 86.083. Each of the trees covers 33 nodes ($M_{\max}$ = 33), and eliminating the 32.4 % loss is an efficient operation. The average loss by (20) is, obviously, 1.162 (its respective percentage 16.2 % is due to there are only two trees). Another instance with an impressive potential loss is a problem of 100 nodes by β = 1.3, where three trees cover 15 nodes each having various lengths (Fig. 13). Despite the maximal loss elimination here is 12.2 %, the solution tree is longer than that in Fig. 12 and it covers a relatively greater number of nodes compared to the trees in Fig. 12.

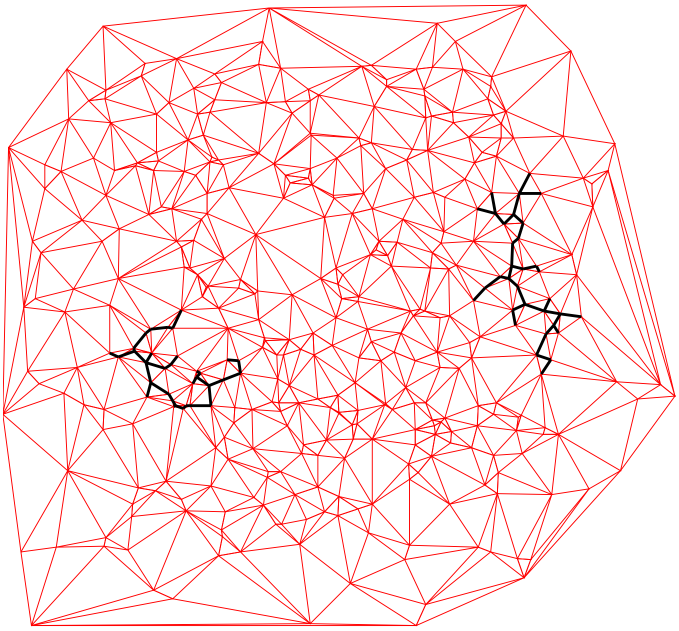

Fig. 12. The problem of 400 nodes and β = 1.5, where $|T_{\max}| = 2$ (the solution tree is on the left; these two trees are shown along with initial edges, but the removed edges are not highlighted to simplify the visualization). Each of the two trees covers 33 nodes ($M_{\max} = 33$), which is 8.25 % of the number of initial nodes. The solution tree total edge length is 65.0188. The other tree total edge length is 86.083, i. e., this tree is 32.4 % longer. If the tree selection is random, the potential loss is interpreted as the average one by (20), and it is $0.5 \cdot (86.083 + 65.0188)/65.0188 \approx 1.162$ (or, in percentage terms, 16.2 %).

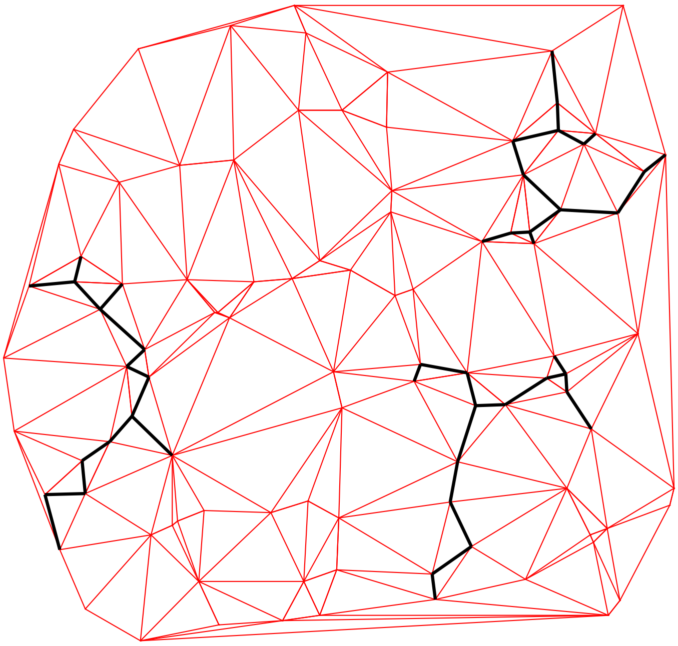

Fig. 13. The problem of 100 nodes and β = 1.3, where $|T_{\max}| = 3$ (these trees are shown along with initial edges, but the removed edges are not highlighted). Each of the three trees covers 15 nodes ($M_{\max} = 15$), which is 15 % of the number of initial nodes. Thus, the trees cover a relatively greater number of nodes compared to the trees in Fig. 12. The total edge lengths of the trees, from the left to the right, are 78.7003, 72.0263, and 70.1428 (the upper tree is the solution). Compared to the solution tree, the two trees are longer by 12.2 and 2.69 %, respectively. The longest tree is on the left, but it is hardly recognizable visually. The maximal loss is 1.122 (or 12.2 %). The average loss is $(1/3) \cdot (78.7003 + 72.0263 + 70.1428)/70.1428 \approx 1.0496$ (or 4.96 %).

VII. Discussion

Determining the number of disconnected graphs, upon removing the edges whose length exceeds $l_{\max}$, begins from finding a first graph ($t = 1$) and checking whether every next node (apart from the nodes in this graph) belongs to this graph. If there are no disconnections, i. e., subset (6) forms a single connected graph, the problem is solved by building the (single possible) minimum spanning tree, where any of the known efficient methods can be used. This case appears to be very likely for loose maximum edge length constraints. As the constraint is made severer, the disconnections are inevitable, and the likelihood of two or more disconnected graphs grows. The probability of two or more trees covering the same maximum of nodes (when $|T_{\max}| > 1$) is greater for fewer initial nodes. This can also be indirectly inferred from Tables I–IV.

The suggested method firstly removes too lengthy edges, and then it builds either minimum spanning trees over disconnected graphs or the single minimum spanning tree if there are no disconnections. However, might building the trees be the first, followed by removing too lengthy edges? No, it would be unreasonable because in this case the removal might tear some trees already built, whereupon re-building trees had to be run.

It may seem that (12) and (14) constitute a two-criterion optimization problem. In this problem, firstly, the number of nodes covered by a minimum spanning tree is maximized, and only then the tree length is minimized. Might these operations be accomplished inversely? The answer is negative because these operations are not of the same priority or order. Obviously, a minimum spanning tree which would connect the maximum possible number of nodes is to be built. This is the prime purpose. The operation of the tree length minimization is run only if there are two and more trees each of which connects the same (maximum) number of nodes.

The suggested method builds the best minimum spanning tree under maximum edge length constraint with respect to planar nodes by using Euclidean metric (4) to calculate graph weights (5). In fact, this is the Euclidean minimum spanning tree on the plane. Nevertheless, it does not mean that the suggested method is limited to the Euclidean metric. Any other metric can be used to calculate weights of Q edges (2), and still the method will be valid. Moreover, the edge weight is not necessarily to be tied to its length. Then, generally speaking, the suggested method will build the best minimum spanning tree under maximum edge weight constraint with respect to planar nodes, whichever metric is used. The matter is that using the Euclidean metric is very common in practice to evaluate the edge weight, which is a linear function of the edge length. In the considered model, the linear function is formally the non-scaled direct proportion without bias, by which distances (5) are taken as weights. However, even if the function that maps the edge length into its weight is nonlinear, it nonetheless relies on the Euclidean metric. This is why the constraint of the maximum edge length is pinned on the model. In other words, the suggested method will build still the best minimum spanning tree, whichever approach is used to obtain the weights of Q edges (2).

The study is an important scientific contribution to the domain of graph theory in combinatorial optimization. It further extends and supplements the theory of minimum spanning trees whose practical importance is very high. The main drawbacks are additional identification of disconnected trees (if any) and ignorance of potentially efficient routes based on trees covering fewer than $M_{\max}$ nodes. For instance, if the best minimum spanning tree is 10 % longer than a minimum spanning tree covering $M_{\max} - 1$ nodes, this may induce vagueness in making a decision on the solution tree for $M_{\max} > 10$. Such an issue is to be addressed in further research.

VIII. Conclusion

In order to build a minimum spanning tree connecting the maximum possible number of planar nodes by not exceeding the maximum edge length, an approach has been suggested that successively involves triangulating the initial nodes, removing too lengthy edges, determining the disconnected graphs, selecting the minimum spanning trees covering a maximum of nodes, and selecting the tree whose length is minimal. The application of this operation sequence is more relevant for not large sets of initial nodes with severe maximum edge length constraints. If the last operation in the suggested method is not accomplished, but the tree selection is random, the solution tree becomes 1.17 % longer on average (see Table I) with the worst average case of 42 % (see Table III) for 10 to 80 nodes. If the worst tree is selected (the tree covering the maximum of nodes, but being the longest among the trees covering the same number of nodes), the solution tree becomes 2.33 % longer on average (see Table II). In the worst case, the loss exceeds 80 % (see Table IV).

In addition to the optimal coverage, the suggested method has also a strong relation and practical application to the metric facility location problem [3], [4], [38], [39], where planar objects are considered. The study can be advanced in further research on building minimum spanning trees for three-dimensional nodes. Minimum spanning trees of spatial graphs have practical application as well [40], [41]. The case when the node is of four or more coordinates is a matter of a generalized research in this direction.

Acknowledgment

The research has technically been supported by the Faculty of Mechanical and Electrical Engineering at the Polish Naval Academy, Gdynia, Poland.

References

[1] R. L. Graham and P. Hell, “On the history of the minimum spanning tree problem,” *Annals of the History of Computing*, vol. 7, no. 1, pp. 43–57, Jan.–Mar. 1985. https://doi.org/10.1109/MAHC.1985.10011

[2] H. Ahmadi and J. R. Martí, “Minimum-loss network reconfiguration: A minimum spanning tree problem,” *Sustainable Energy, Grids and Networks*, vol. 1, pp. 1 – 9, Mar. 2015. https://doi.org/10.1016/j.segan.2014.10.001

[3] B. Stojanović, T. Rajić, and D. Šošić, “Distribution network reconfiguration and reactive power compensation using a hybrid Simulated Annealing – Minimum spanning tree algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 147, May 2023, Art. no. 108829. https://doi.org/10.1016/j.ijepes.2022.108829

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Chapter 23: Minimum Spanning Trees,” in *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001, pp. 561–579.

[5] R. E. Tarjan, “Chapter 6. Minimum spanning trees. 6.2. Three classical algorithms,” *Data Structures and Network Algorithms*, in *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 44, Society for Industrial and Applied Mathematics, 1983, pp. 72 – 77.

[6] H. Loberman and A. Weinberger, “Formal procedures for connecting terminals with a minimum total wire length,” *Journal of the ACM*, vol. 4, no. 4, pp. 428 – 437, Oct. 1957. https://doi.org/10.1145/320893.320896

[7] J. C. Gower and G. J. S. Ross, “Minimum spanning trees and single linkage cluster analysis,” *Applied Statistics*, vol. 18, no. 1, pp. 54 – 61, 1969. https://doi.org/10.2307/2346439

[8] W. B. March, P. Ram, and A. G. Gray, “Fast Euclidean minimum spanning tree: algorithm, analysis, and applications,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, July 25 – 28, 2010, pp. 603–612. https://doi.org/10.1145/1835804.1835882

[9] D. Eppstein, “Spanning trees and spanners,” in J.-R. Sack and J. Urrutia (eds.), *Handbook of Computational Geometry*. Elsevier, 1999, pp. 425–461. https://doi.org/10.1016/B978-044482537-7/50010-3

[10] G. Robins and J. S. Salowe, “Low-degree minimum spanning trees,” *Discrete & Computational Geometry*, vol. 14, no. 2, pp. 151–165, 1995. https://doi.org/10.1007/BF02570700

[11] H. Li, W. Mao, A. Zhang, and C. Li, “An improved distribution network reconfiguration method based on minimum spanning tree algorithm and heuristic rules,” *International Journal of Electrical Power & Energy, Systems*, vol. 82, pp. 466–473, Nov. 2016. https://doi.org/10.1016/j.ijepes.2016.04.017

[12] X. Q. Liao, T. Su, and L. Ma, “Application of neutrosophic minimum spanning tree in electrical power distribution network,” *CAAI Transactions on Intelligence Technology*, vol. 5, no. 2, pp. 99–105, Apr. 2020. https://doi.org/10.1049/trit.2019.0100

[13] A. Capone, D. Corti, L. Gianoli, and B. Sansó, “An optimization framework for the energy management of carrier ethernet networks with Multiple Spanning Trees,” *Computer Networks*, vol. 56, no. 17, pp. 3666–3681, Nov. 2012. https://doi.org/10.1016/j.comnet.2012.08.002

[14] B. Fortz, L. Gouveia, and M. Joyce-Moniz, “Optimal design of switched Ethernet networks implementing the Multiple Spanning Tree Protocol,” *Discrete Applied Mathematics*, vol. 234, pp. 114–130, Jan. 2018. https://doi.org/10.1016/j.dam.2016.07.015

[15] M. M. Devi and M. Geethanjali, “Hybrid of genetic algorithm and minimum spanning tree method for optimal PMU placements,” *Measurement*, vol. 154, Mar. 2020, Art. no. 107476. https://doi.org/10.1016/j.measurement.2020.107476

[16] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder, “Minimum-energy broadcasting in static ad hoc wireless networks,” *Wireless Networks*, vol. 8, no. 6, pp. 607–617, Nov. 2002. https://doi.org/10.1023/a:1020381720601

[17] A. E. F. Clementi, G. Huiban, G. Rossi, Y. C. Verhoeven, and P. Penna, “On the approximation ratio of the MST-based heuristic for the energy-efficient broadcast problem in static ad-hoc radio networks,” in *Proceedings of 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, Nice, France, April 22–26, 2003, p. 222. https://doi.org/10.1109/IPDPS.2003.1213407

[18] M. Hajiaghaei-Keshteli, S. Molla-Alizadeh-Zavardehi, and R. Tavakkoli-Moghaddam, “Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic algorithm,” *Computers & Industrial Engineering*, vol. 59, no. 2, pp. 259–271, Sep. 2010. https://doi.org/10.1016/j.cie.2010.04.007

[19] J. Nešetřil, E. Milková, and H. Nešetřilová, “Otakar Borůvka on minimum spanning tree problem: translation of both the 1926 papers, comments, history,” *Discrete Mathematics*, vol. 233, no. 1–3, pp. 3–36, Apr. 2001. https://doi.org/10.1016/S0012-365X(00)00224-7

[20] R. C. Prim, “Shortest connection networks and some generalizations,” *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957. https://doi.org/10.1002/j.1538-7305.1957.tb01515.x

[21] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959. https://doi.org/10.1007/BF01386390

[22] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956. https://doi.org/10.1090/S0002-9939-1956-0078686-7

[23] J. Kleinberg and É. Tardos, *Algorithm Design*. Boston, Pearson, Addison-Wesley, 2006.

[24] R. Jothi and B. Raghavachari, "Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design," *ACM Transactions on Algorithms*, vol. 1, no. 2, pp. 265–282, Oct. 2005. https://doi.org/10.1145/1103963.1103967

[25] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the Association for Computing Machinery*, vol. 49, no. 1, pp. 16–34, Jan. 2002. https://doi.org/10.1145/505241.505243

[26] D. A. Bader and G. Cong, "Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs," *Journal of Parallel and Distributed Computing*, vol. 66, no. 11, pp. 1366–1378, Nov. 2006. https://doi.org/10.1016/j.jpdc.2006.06.001

[27] D. Lin, Z. Lin, L. Kong, and Y. L. Guan, CMSTR: "A constrained minimum spanning tree based routing protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 146, Jul. 2023, Art. no. 103160. https://doi.org/10.1016/j.adhoc.2023.103160

[28] R. E. Edwards, *Functional Analysis: Theory and Applications*. Holt, Rinehart and Winston, 1965.

[29] V. V. Romanuke, "Optimization of a dataset for a machine learning task by clustering and selecting closest-to-the-centroid objects," *Herald of Khmelnytskyi National University. Technical Sciences*, no. 6, vol. 1, pp. 263–265, 2018.

[30] H. Edelsbrunner, T. S. Tan, and R. Waupotitsch, "An $O\left(n^2 \log n\right)$ time algorithm for the minmax angle triangulation," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 4, pp. 994–1008, 1992. https://doi.org/10.1137/0913058

[31] J. A. De Loera, J. Rambau, and F. Santos, "Triangulations, Structures for Algorithms and Applications," in *Algorithms and Computation in Mathematics*, vol. 25. Springer, 2010. https://doi.org/10.1007/978-3-642-12971-1

[32] G. Xia, "The stretch factor of the Delaunay triangulation is less than 1.998," *SIAM Journal on Computing*, vol. 42, no. 4, pp. 1620–1659, 2013. https://doi.org/10.1137/110832458

[33] F. Hurtado, M. Noy, and J. Urrutia, "Flipping edges in triangulations," *Discrete & Computational Geometry*, vol. 22, no. 3, pp. 333–346, Oct. 1999. https://doi.org/10.1007/PL00009464

[34] C. Du, "An algorithm for automatic Delaunay triangulation of arbitrary planar domains," *Advances in Engineering Software*, vol. 27, no. 1–2, pp. 21–26, Oct.–Nov. 1996. https://doi.org/10.1016/0965-9978(96)00004-X

[35] L. J. Guibas, D. E. Knuth, and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, vol. 7, no. 1–6, pp. 381–413, Jun. 1992. https://doi.org/10.1007/BF01758770

[36] V. V. Romanuke, "Speedup of the *k*-means algorithm for partitioning large datasets of flat points by a preliminary partition and selecting initial centroids," *Applied Computer Systems*, vol. 28, no. 1, pp. 1–12, Aug. 2023. https://doi.org/10.2478/acss-2023-0001

[37] V. V. Romanuke, "Accurate detection of multiple targets by uniform rectangular array radar with threshold soft update and area rescanning," *Information and Telecommunication Sciences*, vol. 13, no. 2, pp. 62–71, Dec. 2022. https://doi.org/10.20535/2411-2976.22022.62-71

[38] S. Li, "A 1.488 approximation algorithm for the uncapacitated facility location problem," in *Automata, Languages and Programming. Lecture Notes in Computer Science*, L. Aceto, M. Henzinger, and J. Sgall, Eds. vol. 6756. Springer, 2011, pp. 77–88. https://doi.org/10.1007/978-3-642-22012-8_5

[39] N. Megiddo and A. Tamir, "On the complexity of locating linear facilities in the plane," *Operations Research Letters*, vol. 1, no. 5, pp. 194–197, Nov. 1982. https://doi.org/10.1016/0167-6377(82)90039-6

[40] R. Zhou, L. Shu, and Y. Su, "An adaptive minimum spanning tree test for detecting irregularly-shaped spatial clusters," *Computational Statistics & Data Analysis*, vol. 89, pp. 134–146, Sep. 2015. https://doi.org/10.1016/j.csda.2015.03.008

[41] Y. Li, T. Chu, Y. Liu, H. Zhang, F. Dong, Q. Gai, Y. Shi, H. Ma, F. Zhao, K. Che, N. Mao, and H. Xie, "Classification of major depression disorder via using minimum spanning tree of individual high-order morphological brain network," *Journal of Affective Disorders*, vol. 323, pp. 10–20, Feb. 2023. https://doi.org/10.1016/j.jad.2022.11.029

Vadim V. Romanuke graduated from the Technological University of Podillya (Ukraine) in 2001. In 2006, he received the Degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. The Candidate Dissertation suggested a way of increasing interference noise immunity of data transferred over radio systems. The degree of Doctor of Technical Sciences in Mathematical Modelling and Computational Methods was obtained in 2014. The Doctor-of-Science Dissertation solved a problem of increasing efficiency of identification of models for multistage technical control and run-in under multivariate uncertainties of their parameters and relationships. In 2016, he received the status of Full Professor.

He is a Professor of the Faculty of Mechanical and Electrical Engineering at the Polish Naval Academy. His current research interests concern decision making, game theory, statistical approximation, combinatorial optimization, and control engineering based on statistical correspondence. He has 410 published scientific articles, one monograph, one tutorial, eight methodical guidelines in Functional Analysis, Mathematical and Computer Modeling Master Thesis preparation, Conflict-Controlled Systems, Database Design and Management, Software Design and Analysis, Algorithms and Data Structures. Before January 2018, Vadim Romanuke was the scientific supervisor of a Ukrainian budget grant work concerning minimization of water heat transfer and consumption. Also during 2015–2019 he led a branch of fitting statistical approximators at the Centre of Parallel Computations in Khmelnitskiy, Ukraine. Since 2019 he has been participating as a scientific collaborator in two budget grant works concerning automation in navigation.

Address for correspondence: 69 Śmidowicza Street, Gdynia, Poland, 81-127.
E-mail: romanukevadimv@gmail.com
ORCID iD: https://orcid.org/0000-0003-3543-3087