UDC 519.172.1+004.722.25

BUILDING MINIMUM SPANNING TREES BY LIMITED NUMBER OF NODES OVER TRIANGULATED SET OF INITIAL NODES

Vadim V. Romanuke

Vinnytsia Institute of Trade and Economics of State University of Trade and Economics, Ukraine

Background. The common purpose of modeling and using minimum spanning trees is to ensure efficient coverage. In many tasks of designing efficient telecommunication networks, the number of network nodes is usually limited. In terms of rational allocation, there are more possible locations than factually active tools to be settled to those locations.

Objective. Given an initial set of planar nodes, the problem is to build a minimum spanning tree connecting a given number of the nodes, which can be less than the cardinality of the initial set. The root node is primarily assigned, but it can be changed if needed.

Methods. 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. These distances being the lengths of the respective edges are used as graph weights, and a minimum spanning tree is built over this graph.

Results. The problem always has a solution if the desired number of nodes (the number of available recipient nodes) is equal to the number of initially given nodes. If the desired number is lesser, the maximal edge length is found and the edges of the maximal length are excluded while the number of minimum spanning tree nodes is greater than the desired number of nodes.

Conclusions. To build a minimum spanning tree by a limited number of nodes, it is suggested using the Delaunay triangulation and an iterative procedure in order to meet the desired number of nodes. Planar nodes of an initial set are triangulated, whereupon the edge lengths are used as weights of a graph. The iterations to reduce nodes are done only if there are redundant nodes. When failed, the root node must be changed before the desired number of nodes is changed.

Keywords: *minimum spanning tree; triangulation; edge lengths; redundant nodes; root node.*

1. Introduction

One of the primal practical uses of minimum spanning trees was an efficient electrical coverage [1]. Spread out quickly since, minimum spanning trees have been successfully applied to design computer, broadcasting, telecommunication, and transportation networks [2], [3]. The latter include as natural resources supply networks, as well as electrical power supply grids [4], [5].

Given a set of nodes connected with edges, the purpose of the minimum spanning tree is to connect all the nodes by minimizing the cost of the connection. Basically, 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 weight [6], [7].

Along with the very first approach [1], another two commonly used algorithms for finding a minimum spanning tree are the Prim's algorithm and Kruskal's algorithm. The Prim's algorithm performs by building such a tree, at each step adding the cheapest possible connection from the currently built tree to another node [8], [9]. The Kruskal's algorithm performs by adding at each step the next lowest-weight edge that will not form a cycle to the minimum spanning forest [10], [11]. At the termination of the algorithm, the forest forms a minimum spanning forest of the graph. For the connected graph, the forest has a single component and forms a minimum spanning tree.

Whereas the Prim's algorithm is commonly said to perform better on dense graphs [12], the Kruskal's algorithm is believed to perform acceptably on sparser graphs [2], [6], [7]. However, the efficiency of the Prim's algorithm on sparse graphs still has not been denied [13]. Both the algorithms have nearly the same asymptotic time complexity varying from linear to polynomial [14].

2. Problem statement

Given a set of N planar points (nodes), the problem is to build a minimum spanning tree connecting a given number M of the points, where $M \leqslant N$. The root node is assigned as well. This problem relates to tasks of designing efficient telecommunication networks, where the number of network nodes is usually limited. For example, the task is to build a network of mobile base stations, connected in a mesh on a given set of possible locations, with the minimum possible total distance to ensure efficient maintenance and signal level. Another example is to build a network of electrical power converters in order to supply the desired voltage to industrial and individual customers from electrical power stations. In both examples, just like in others similar, the number of minimum spanning tree nodes is limited and usually less than the number of all possible locations.

3. Delaunay triangulation

Let

$$\{\mathbf{P}_i\}_{i=1}^{N} = \{[x_i \quad y_i]\}_{i=1}^{N} \qquad (1)$$

be a set of planar nodes, on which a minimum spanning tree is to be built having maximum M nodes, $M \leqslant N$. First of all, in order to obtain a set of edges, a Delaunay triangulation is performed over set (1). For further consideration, denote the set of edges after the triangulation of set (1) by

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

where edge $\mathbf{E}_q$ is determined by nodes $\mathbf{P}_{j_q}$ and $\mathbf{P}_{k_q}$ connected by this edge for $j_q \in \left\{\overline{1, N}\right\}$, $k_q \in \left\{\overline{1, N}\right\}$, $j_q \neq k_q$, and Q is the total number of edges. Although the Delaunay triangulation does not maximize the edge-length of the triangles, it maximizes the minimum of all the angles of the triangles in the triangulation [15], [16]. This allows to mostly exclude creating sliver triangles [17]. For instance, a set of 17 nodes issuing a set of 289 edges (Fig. 1) is triangulated and thus a set of only 38 edges is formed (Fig. 2), over which a minimum spanning tree can be potentially built.

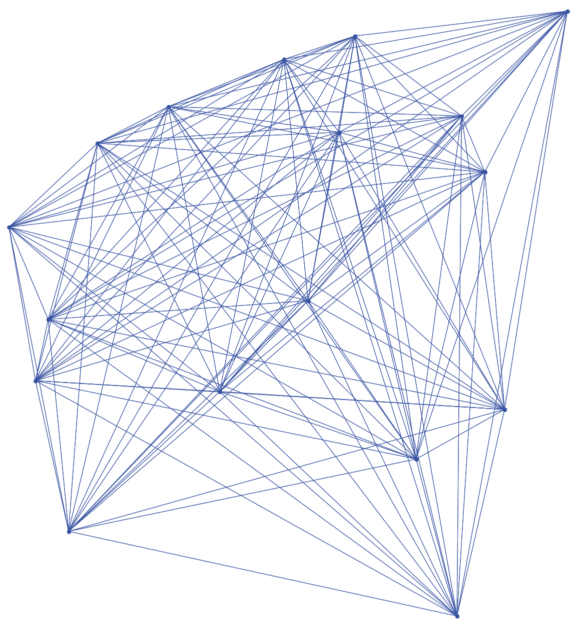

Fig. 1. A set of 17 planar nodes connected pairwise with a set of 289 edges

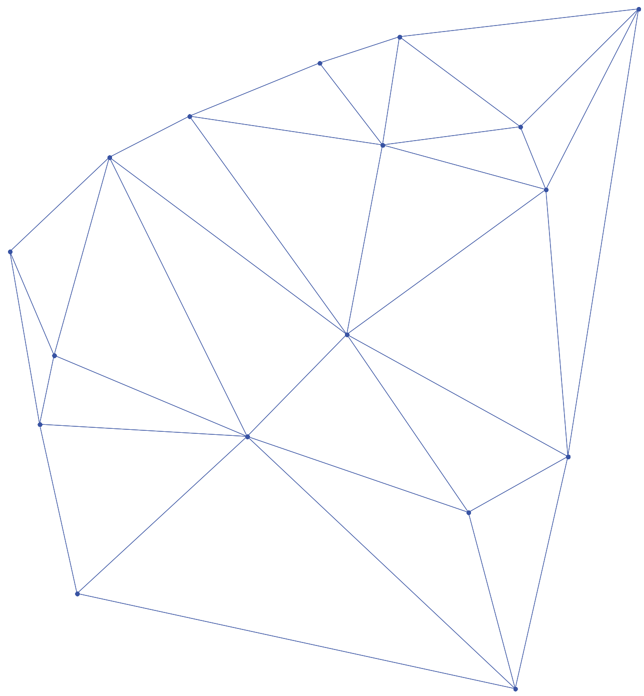

Fig. 2. The triangulated planar nodes from Fig. 1 and a set of the respective 38 edges

It is worth noting that the number of edges connecting planar nodes after they are triangulated is not necessarily the same for a given N. In a case with 17 planar nodes, it can vary between 36 and 44. As the number of nodes increases, the variation becomes wider. For instance, it is 93 to 104 edges for 37 nodes (Fig. 3, 4), 204 to 217 edges for 75 nodes (Fig. 5, 6), 280 to 292 edges for 100 nodes (Fig. 7, 8), although the result depends on the shape of planar data.

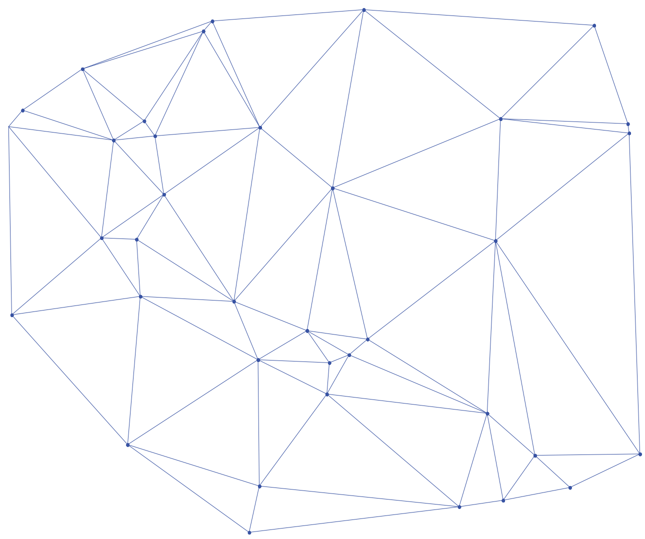

Fig. 3. A minimum of 93 edges for a set of 37 planar nodes after triangulation

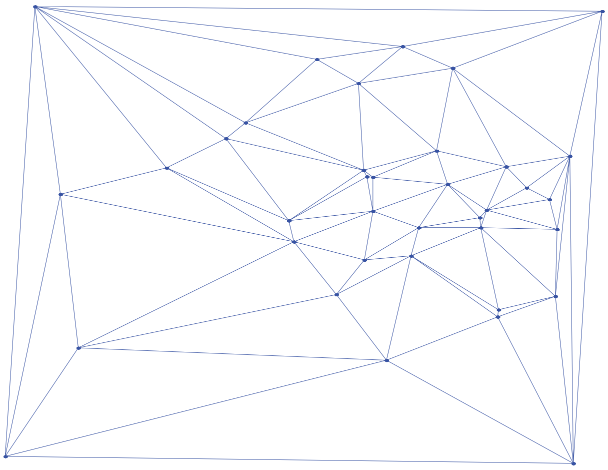

Fig. 4. A maximum of 104 edges for another set of 37 planar nodes after triangulation

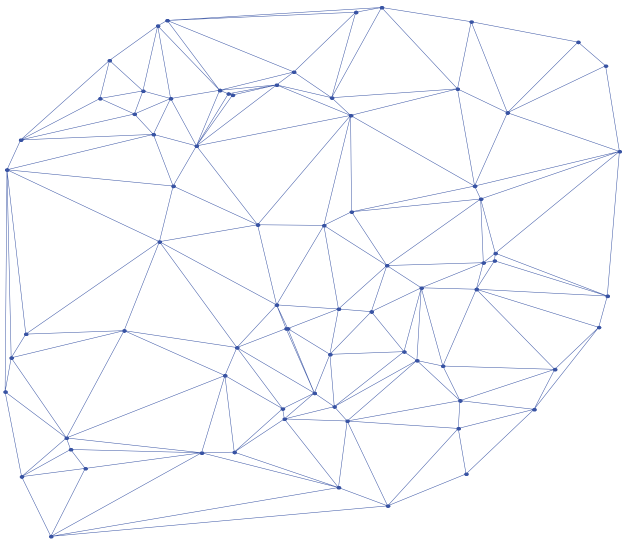

Fig. 5. A minimum of 204 edges for a set of 75 planar nodes after triangulation

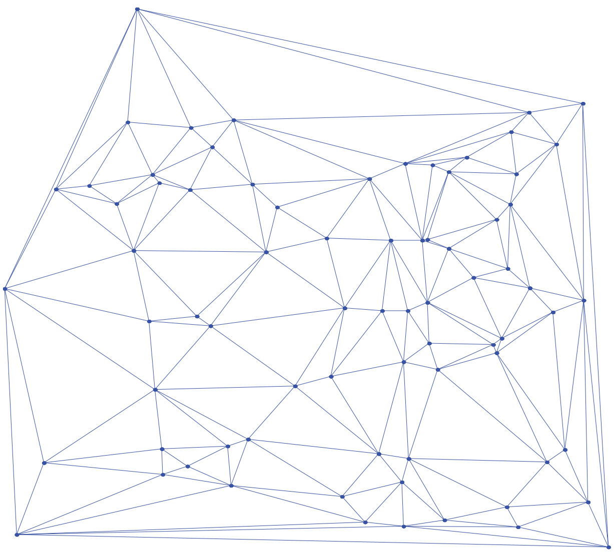

Fig. 6. A maximum of 217 edges for another set of 75 planar nodes after triangulation

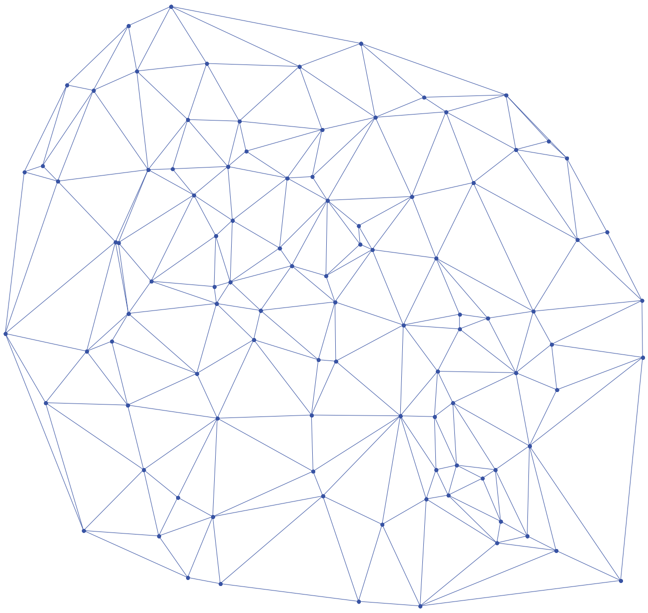

Fig. 7. A minimum of 280 edges for a set of 100 planar nodes after triangulation

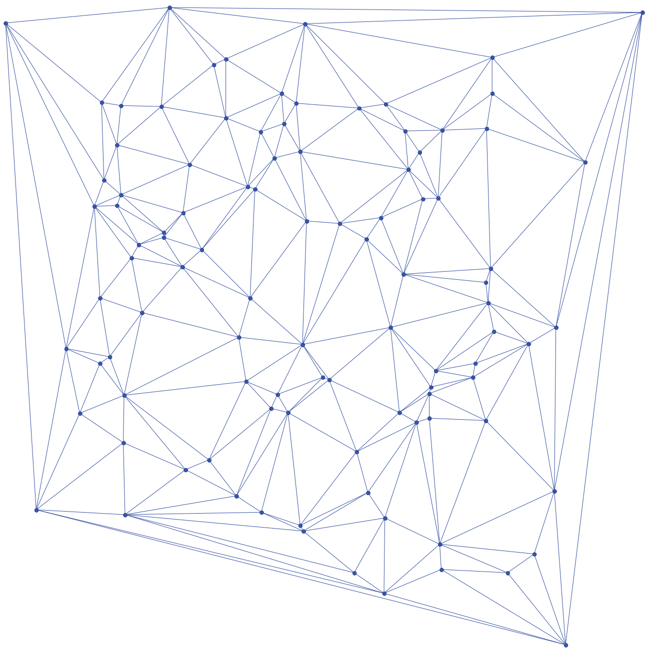

Fig. 8. A maximum of 292 edges for another set of 100 planar nodes after triangulation

It is seen from Fig. 3 — 8 that the minimum of edges is typical for a circular-shaped node sets. As the convex hull of the node set becomes more square-shaped (or, more generally, polygon-shaped), the number of edges increases.

4. Minimum spanning tree iterations

To obtain weights for the graph edges, the distances between every pair of the nodes in edges (2), being the lengths of these edges, are calculated as

$$\rho_{\mathbb{R}^2}\left(\mathbf{P}_{j_q}, \mathbf{P}_{k_q}\right) = \sqrt{\left(x_{j_q} - x_{k_q}\right)^2 + \left(y_{j_q} - y_{k_q}\right)^2} = \\ = l_{\mathbb{R}^2}\left(\mathbf{E}_q\right) \text{ for } q = \overline{1, Q}. \quad (3)$$

Then a minimum spanning tree is found for the graph with edges (2) and their respective weights (3). The problem is solved if $M = N$. Otherwise, when $M < N$, then

$$N^* = N, \; Q^* = Q, \; E^* = \left\{\mathbf{E}_q^*\right\}_{q=1}^{Q^*} = E,$$

$$l_{\mathbb{R}^2}^*\left(\mathbf{E}_q^*\right) = l_{\mathbb{R}^2}\left(\mathbf{E}_q\right) \text{ for } q = \overline{1, Q^*},$$

and the following routine is executed while the number of nodes connected by edges in the minimum spanning tree is greater than M (i. e., while $N^* > M$).

First, the edges whose length is maximal are excluded from set E^*:

$$E^{*(\text{obs})} = E^*, \; E^* = E^{*(\text{obs})} \setminus \left\{\mathbf{E}_h^*\right\}_{h \in H}$$

$$\text{by } H = \arg \max_{q=\overline{1, Q^*}} l_{\mathbb{R}^2}^*\left(\mathbf{E}_q^*\right). \quad (4)$$

Second, the respective distances are excluded from the set of distances

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

by

$$Q^{*(\text{obs})} = Q^*, \; Q^* = \left|\left\{\overline{1, Q^{*(\text{obs})}}\right\} \setminus H\right|,$$

whence a new set $\left\{l_{\mathbb{R}^2}^*\left(\mathbf{E}_q^*\right)\right\}_{q=1}^{Q^*}$ of distances (edge lengths or weights, in other words) is formed. Then a minimum spanning tree is found for the graph with new edges (4) and their respective weights (5).

Generally speaking, this routine cannot ensure that the eventual number of nodes in the minimum spanning tree be equal to M. In other words, equality $N^* = M$ does not always hold as the while condition is broken and the algorithm stops returning a minimum spanning tree connecting N^* nodes. An example is presented in Fig. 9, where the task is to build a minimum spanning tree connecting 98 nodes out of 100 nodes. In this particular example, the minimum spanning tree connects 97 nodes — a node less than desired. A noticeable spot is the separated triangle of three nodes. The triangle became separated after cutting off too lengthy edges. The vertices of the triangle are those three nodes, one of which is desired to be in a minimum spanning tree. However, another problem with 100 nodes, where only one node is redundant (in terms of building a tree with a limited number of nodes), is solved successfully (Fig. 10).

Fig. 11 shows the result for a set of 150 nodes, among which five nodes are redundant, wherein a node in the minimum spanning tree is missing like that one in Fig. 9. A separated set of four nodes is seen here (two triangles — one within another). Another two nodes are “turned off” in the upper left side of the planar data shape.

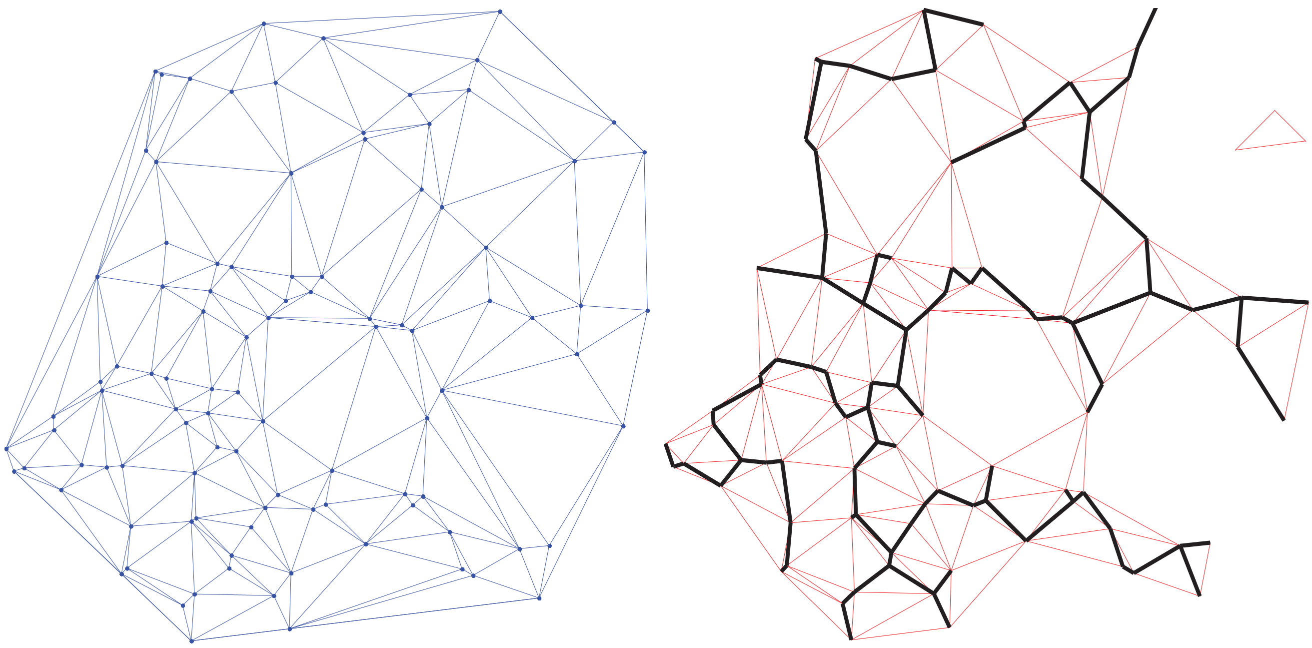

Fig. 9. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 96 edges connecting 97 nodes over an initial set of 100 nodes ($M = 98$)

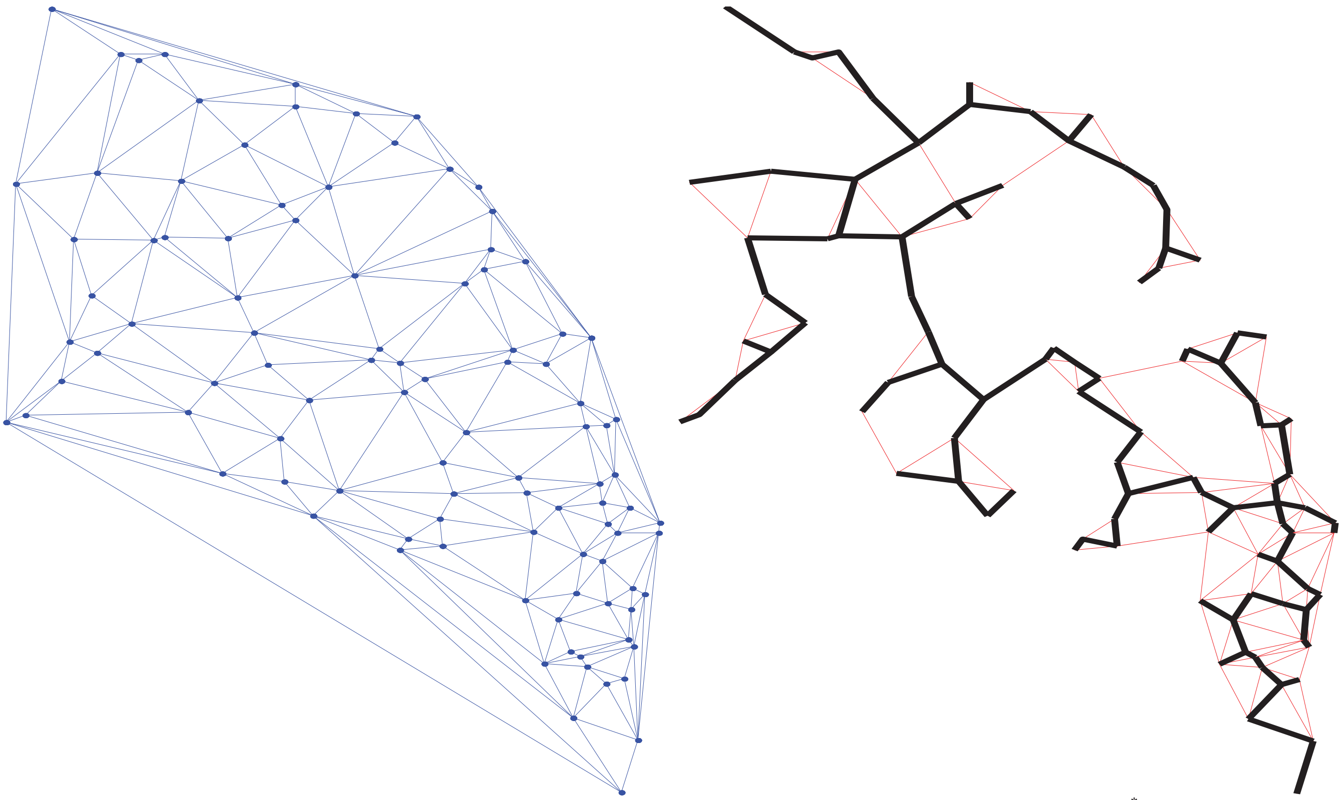

Fig. 10. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 98 edges connecting 99 nodes over an initial set of 100 nodes ($M = 99$)

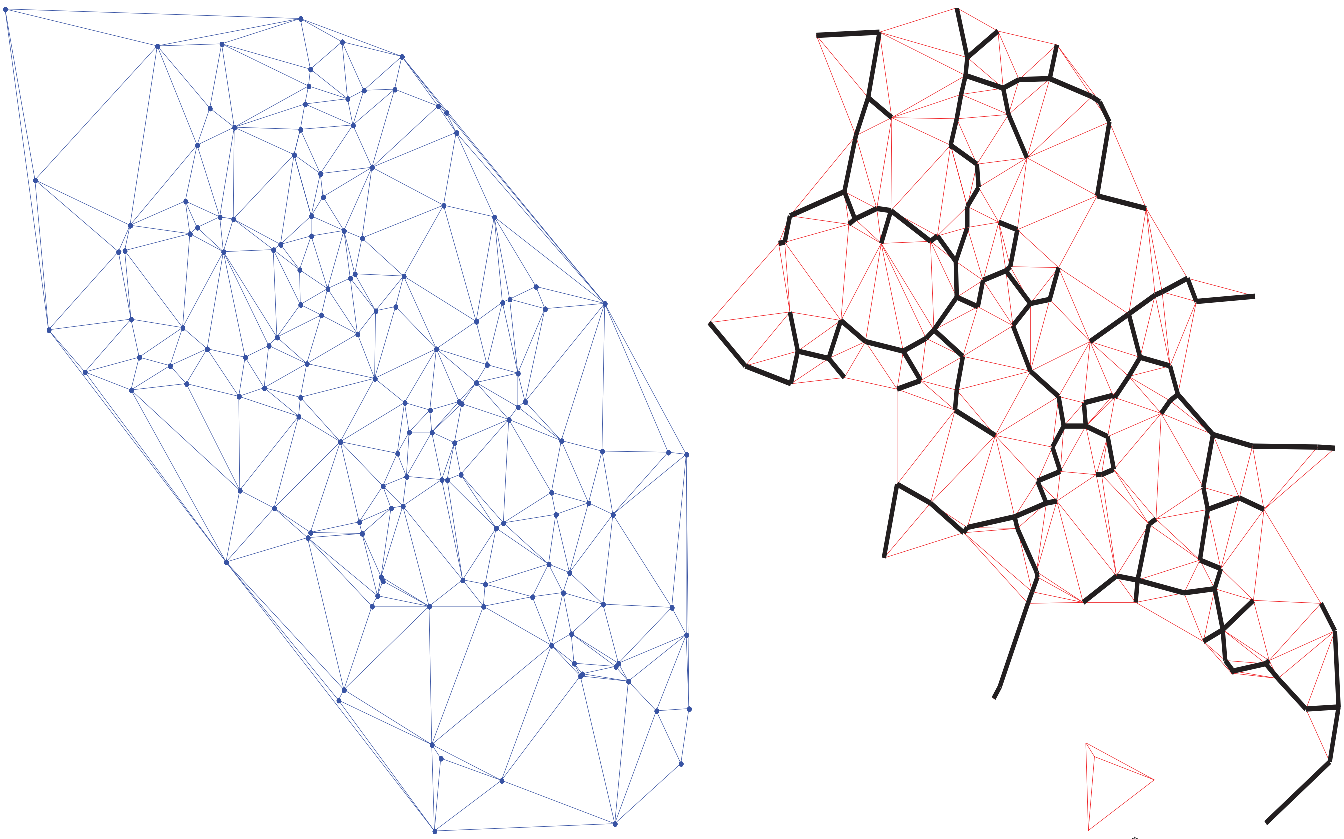

Fig. 11. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 143 edges connecting 144 nodes over an initial set of 150 nodes ($M = 145$)

A similar example is presented in Fig. 12 for a specific shape of the planar data. The shape resembles a circular arc. Aiming at building a minimum spanning tree connecting 147 nodes out of 150 ones, one node becomes "turned off". This is the node at the bottom, whose missing makes an edge separated (this is the very edge at the bottom on the right subplot). Nevertheless, for another specific shape, resembling a denser circular arc, the tree is successfully built connecting 247 nodes out of 250 ones (Fig. 13).

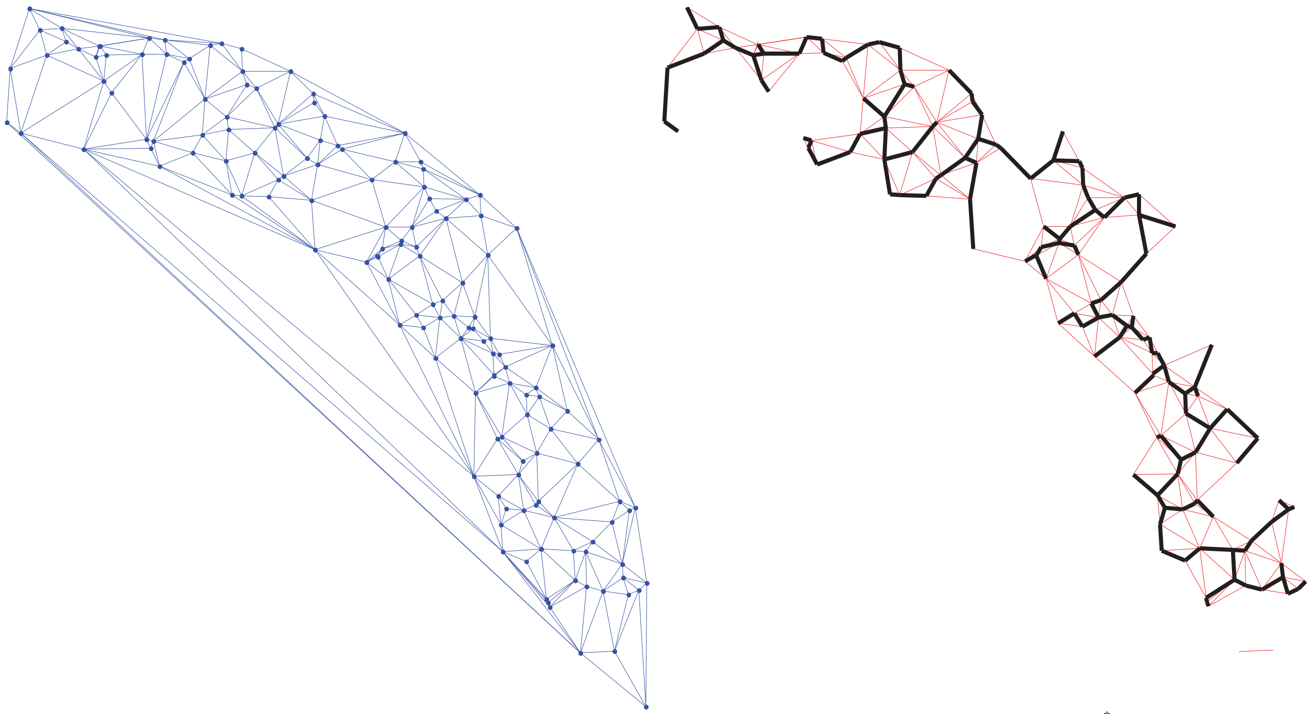

Fig. 12. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 145 edges connecting 146 nodes over an initial set of 150 nodes ($M = 147$)

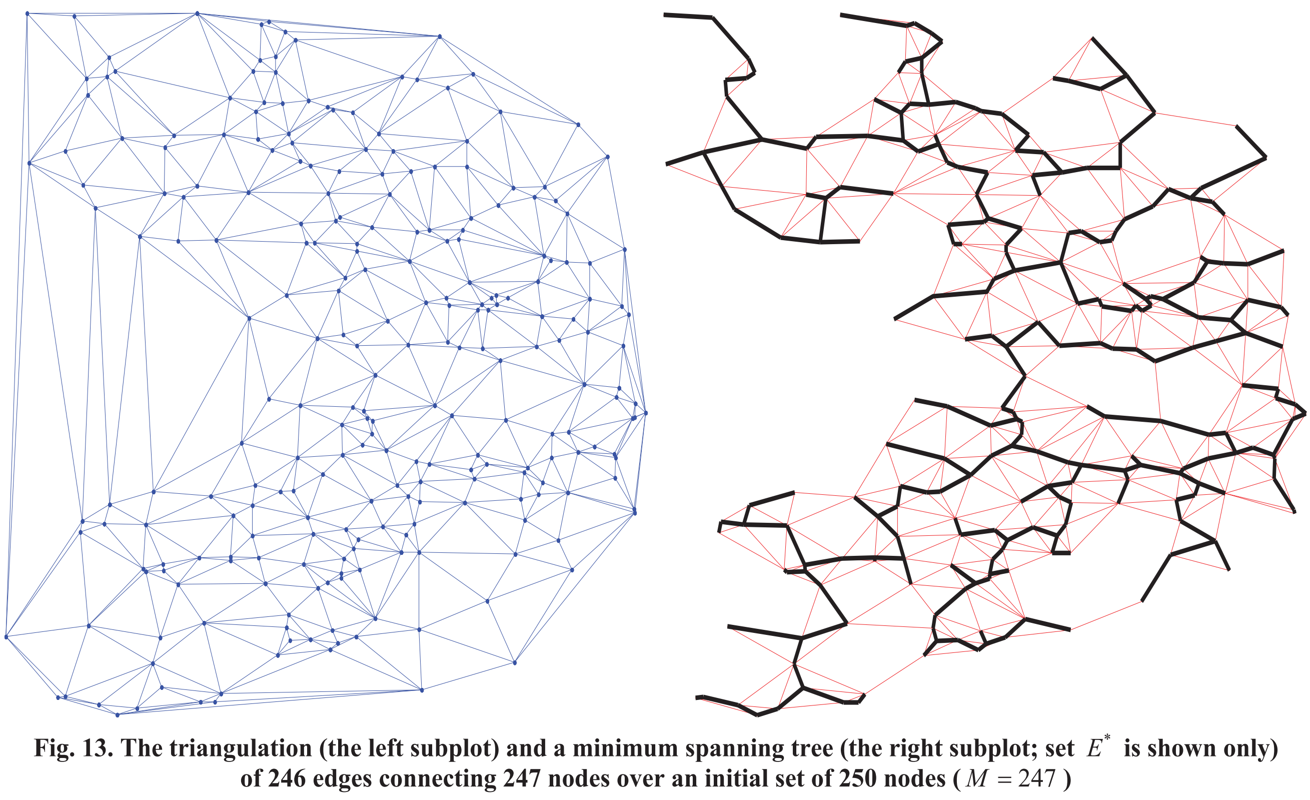

Fig. 13. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 246 edges connecting 247 nodes over an initial set of 250 nodes ($M = 247$)

An example of a total fail is presented in Fig. 14 for a semicircular shape of 250 nodes, where trying to exclude one to three nodes results in a tree with just 33 nodes. The problem without reducing nodes is solved normally (Fig. 15), and all those too long edges are not included into the tree.

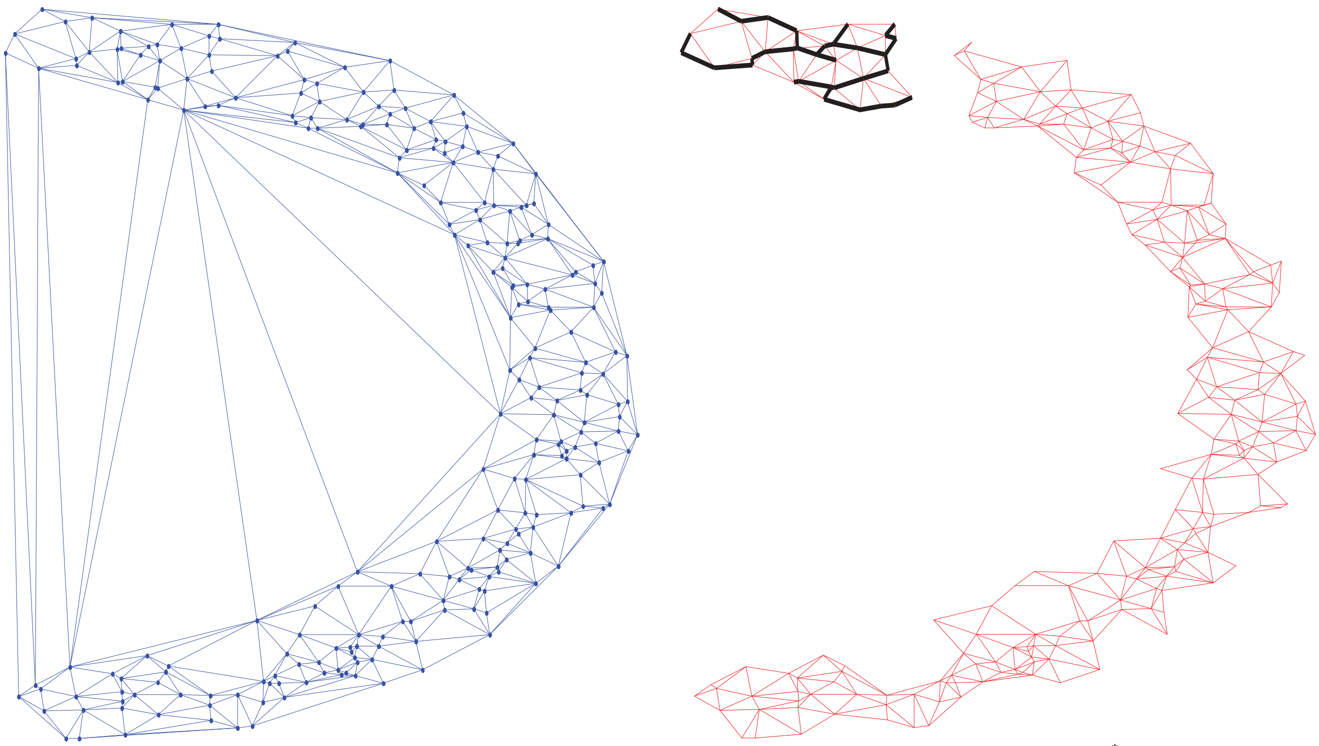

Fig. 14. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^{*} is shown only) of just 32 edges connecting 33 nodes over an initial set of 250 nodes ($M \in \{247, 248, 249\}$)

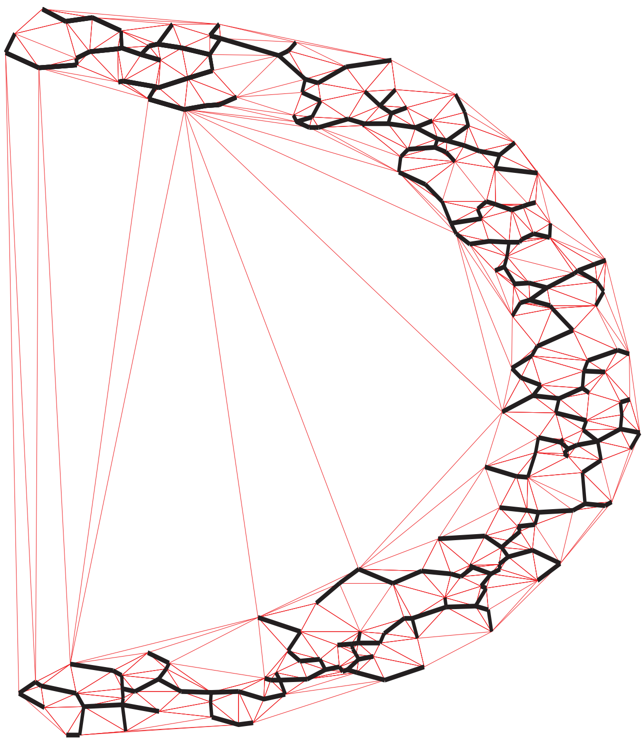

Fig. 15. A minimum spanning tree of 249 edges connecting all 250 nodes from Fig. 14 ($M = 250$)

A far more complicated case is shown in Fig. 16, where a minimum spanning tree is to be built over 248 or 249 nodes out of 250 ones. The result is the same whether $M = 248$ or $M = 249$: two nodes are "turned off", which can be easily spotted in the lower left side of the planar data shape. If the number of desired nodes is decreased to 247, the problem does not have an exact solution. There exists only one minimum spanning tree of 211 edges connecting 212 nodes. If the number of desired nodes is decreased further, the result does not change until $M = 212$ (Fig. 17). Thus, in this particular example, whichever the number of desired nodes is, being varied between 213 and 247, the result is the same, where 38 nodes are "turned off" from the initial set (Fig. 16), and the problem does not have an exact solution. Obviously, if $M = 212$ then the problem does have an exact solution which is a minimum spanning tree of 211 edges shown in Fig. 17.

The considered examples and their results visualized in Fig. 9 — 17 convince that thinner-shaped sets of nodes are more prone to have no solution if $M < N$. So, it is better to select number M warily. If the resulting tree has much fewer nodes, the root node must be changed selected from the missing nodes.

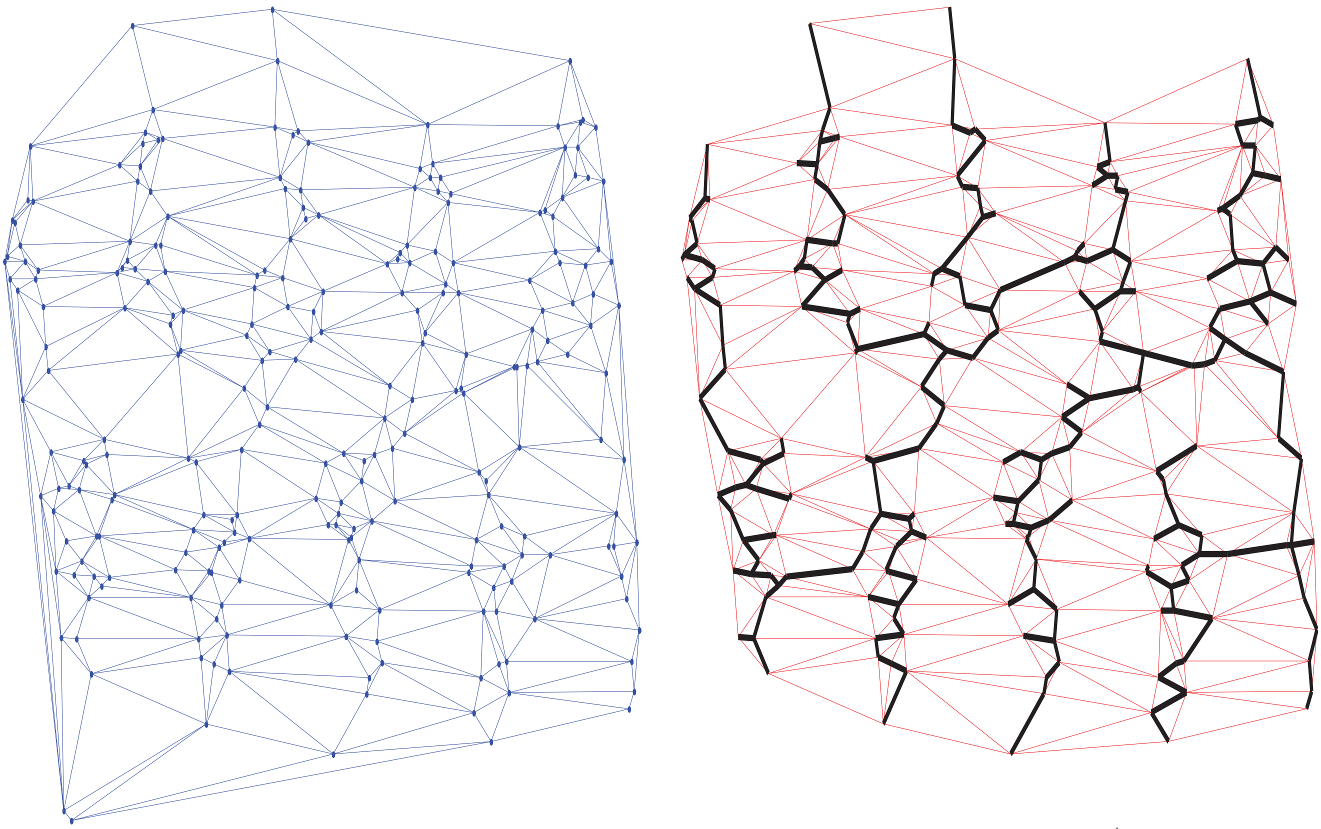

Fig. 16. The triangulation (the left subplot) and a minimum spanning tree (the right subplot; set E^* is shown only) of 247 edges connecting 248 nodes over an initial set of 250 nodes ($M \in \{248, 249\}$)

Fig. 17. A minimum spanning tree of 211 edges connecting 212 nodes from Fig. 16 by $M = 247$ down to $M = 212$ (set E^* is shown only)

5. Discussion

The time complexity of the suggested approach comprises the time complexity of the Delaunay triangulation and the algorithm of building a minimum spanning tree, whether it is the Prim's or Kruskal's algorithm (although the Prim's algorithm is preferable). In the case of a few redundant nodes (i. e., when $M < N$), the iterations during which the distances (lengths of edges) are compared with the currently maximal edge length may slow down the solution process. However, as the number of desired nodes is decreased, and the number of redundant nodes correspondingly decreases, it does not necessarily lead to a significant slowdown. The reason is the number of the minimum spanning tree edges may drop abruptly as the number of desired nodes is decreased by 1 (just like in the example in Fig. 16, 17). In such cases, it is reasonable to build a few trees for a few distant values of number M and see how the tree coverage changes. Then, if possible, number M is corrected (adjusted) so that the number of the minimum spanning tree edges be the closest to M. If it is impossible to correct number M, or the resulting tree covers too fewer nodes, changing the root node selected from the missing nodes may help. Otherwise, the problem has no solution.

6. Conclusion

To build a minimum spanning tree by a limited number of nodes, it is suggested using the Delaunay triangulation and an iterative procedure in order to meet the desired number of nodes (recipients). Given an initial set of planar nodes, they are triangulated, whereupon the distances between every pair of the nodes in respective edges are calculated. These distances being the lengths of the respective edges are used as graph weights. The problem always has a solution if the desired number of nodes (being commonly equal to the number of available recipient nodes) is equal to the number of initially given nodes. If the desired number is lesser, the maximal edge length is found and the edges of the maximal length are excluded while the number of minimum spanning tree nodes is greater than the desired number of nodes. When this problem is not solved to an exact desired number of nodes, the eventual number of tree nodes is less than desired. While it is so, the root node must be changed by selecting it from the missing nodes, but it does not ensure the exact solution. After all the missing nodes are tried and still the problem is not solved, the desired number of nodes must be changed by the least possible value [18]. A further research must be focused on such an adjustment.

References

1. 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, iss. 1 — 3, pp. 3 — 36, 2001. https://doi.org/10.1016/S0012-365X(00)00224-7
2. R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, iss. 1, pp. 43 — 57, 1985. https://doi.org/10.1109/MAHC.1985.10011
3. Y. K. Dalal and R. M. Metcalfe, "Reverse path forwarding of broadcast packets," *Communications of the ACM*, vol. 21, iss. 12, pp. 1040 — 1048, 1978. https://doi.org/10.1145/359657.359665
4. 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, Article ID 108829, 2023. https://doi.org/10.1016/j.ijepes.2022.108829
5. 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, 2015. https://doi.org/10.1016/j.segan.2014.10.001
6. S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the Association for Computing Machinery*, vol. 49, iss. 1, pp. 16 — 34, 2002. https://doi.org/10.1145/505241.505243
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Chapter 23: Minimum Spanning Trees," in *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001, pp. 561 — 579.
8. R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, iss. 6, pp. 1389 — 1401, 1957. https://doi.org/10.1002/j.1538-7305.1957.tb01515.x
9. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, iss. 1, pp. 269 — 271, 1959. https://doi.org/10.1007/BF01386390
10. T. Cormen, C. E. Leiserson, and R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Third Edition. MIT Press, 2009.
11. H. Loberman and A. Weinberger, "Formal procedures for connecting terminals with a minimum total wire length," *Journal of the ACM*, vol. 4, iss. 4, pp. 428 — 437, 1957. https://doi.org/10.1145/320893.320896
12. 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.
13. S. Chakraborty, A. Mukherjee, V. Raman, and S. Rao Satti, "Frameworks for designing in-place graph algorithms," *Journal of Computer and System Sciences*, vol. 123, pp. 1 — 19, 2022. https://doi.org/10.1016/j.jcss.2021.07.004
14. R. Jothi, S. K. Mohanty, and A. Ojha, "Fast approximate minimum spanning tree based clustering algorithm," *Neurocomputing*, vol. 272, pp. 542 — 557, 2018. https://doi.org/10.1016/j.neucom.2017.07.038
15. 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, iss. 4, pp. 994 — 1008, 1992. https://doi.org/10.1137/0913058
16. 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.
17. G. Xia, "The stretch factor of the Delaunay triangulation is less than 1.998," *SIAM Journal on Computing*, vol. 42, iss. 4, pp. 1620 — 1659, 2013. https://doi.org/10.1137/110832458
18. V. V. Romanuke, "Fast-and-smoother uplink power control algorithm based on distance ratios for wireless data transfer systems," *Studies in Informatics and Control*, vol. 28, iss. 2, pp. 147 — 156, 2019. https://doi.org/10.24846/v28i2y201903

Романюк В.В.

Побудова мінімальних сполучних дерев за обмеженої кількості вузлів на триангульованій множині початкових вузлів

Проблематика. Загальна мета моделювання та використання мінімальних сполучних дерев полягає у забезпеченні ефективного покриття. У багатьох завданнях проєктування ефективних телекомунікаційних мереж кількість вузлів мережі зазвичай є обмеженою. У термінах раціонального розміщення це означає, що фактично існує більше потенційних місць розташування, ніж наявних засобів для їх розміщення у цих місцях.

Мета дослідження. Для даної початкової множини вузлів на площині задача полягає у побудові мінімального сполучного дерева, що поєднує задану кількість вузлів, котра може бути меншою за кількість елементів початкової множини. Кореневий вузол першопочатково задається, однак за необхідності він може бути змінений.

Методика реалізації. Для отримання множини ребер виконується триангуляція Делоне на початковій множині вузлів. Обчислюються відстані між кожною парою вузлів у відповідних ребрах. Ці відстані, котрі є довжинами відповідних ребер, використовуються як ваги для графа, і на цьому графі будується мінімальне сполучне дерево.

Результати дослідження. Дана задача завжди має розв’язок за умови, якщо бажана кількість вузлів (кількість доступних вузлів-приймачів) рівна кількості початково даних вузлів. Якщо бажана кількість є меншою, знаходиться максимальна довжина ребра, й усі ребра максимальної довжини виключаються доки кількість вузлів мінімального сполучного дерева є більшою за бажану кількість вузлів.

Висновки. Для побудови мінімального сполучного дерева за обмеженої кількості вузлів запропоновано використовувати триангуляцію Делоне та ітеративну процедуру задля досягнення бажаної кількості вузлів. Виконується триангуляція вузлів з початкової множини на площині, після чого довжини ребер використовуються як ваги графа. Ітерації задля скорочення вузлів виконуються лише за наявності зайвих вузлів. У випадку відсутності розв’язку мусить бути змінений кореневий вузол перед тим, як змінювати бажану кількість вузлів.

Ключові слова: *мінімальне сполучне дерево; триангуляція; довжини ребер; зайві вузли; кореневий вузол.*