

UDC 519.172.1+004.722.25

PRUNING MINIMUM SPANNING TREES AND CUTTING LONGEST EDGES TO CONNECT A GIVEN NUMBER OF NODES BY MINIMIZING TOTAL EDGE LENGTH

Vadim V. Romanuke

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

Background. Whereas in many tasks of designing efficient telecommunication networks, the number of network nodes is limited, the initial choice of nodes is wider. There are more possible locations than factually active tools to be settled to those locations to further satisfy consumers. This induces an available node constraint problem.

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 is less than the cardinality of the initial set. Therefore, the available node constraint problem aims at building an optimally minimum spanning tree to connect a given number of planar nodes being less than an initial number of nodes by minimizing the tree length.

Methods. The initial set of nodes is triangulated. This gives a set of edges, whose lengths are calculated and used as graph weights. A minimum spanning tree is built over this graph. The desired number of nodes is reached by pruning the minimum spanning tree connecting the initial number of nodes, where free edges whose weights are the largest are iteratively removed from the tree. The other approach, the cutting method, removes longest edges off the initial minimum spanning tree, regardless of whether they are free or not.

Results. Unlike the pruning method, the method of cutting longest edges may result in a minimum spanning tree connecting fewer nodes than the desired number. However, the cutting method often outputs a shorter tree, especially when the edge length varies much. Besides, the cutting method is slower due to it iteratively rebuilds a minimum spanning tree.

Conclusions. The problem is initially solved by the pruning method. Then the cutting method is used and its solution is compared to the solution by the pruning method. The best tree is shorter. A tradeoff for the nodes and tree length is possible.

Keywords: *minimum spanning tree; triangulation; edge lengths; pruning the tree; cutting longest edges.*

1. Available node constraint

In many tasks of designing efficient transportation and telecommunication networks, the number of available network nodes is less than the total number of nodes [1], [2]. For a rational allocation problem, in other words, there are more possible (recipient) locations than factually active tools to be settled to those locations to further satisfy consumers. Such problems are solved by using minimum spanning trees to ensure efficient coverage with the minimum of cost [3], [4]. In particular, article [5] considered a problem of building a minimum spanning tree connecting a given number of the nodes, which can be less than the cardinality of the initial set of planar nodes. This problem was suggested to solve by using the Delaunay triangulation [6], [7] and an iterative procedure in order to connect the desired number of nodes. In more detail, the set of edges is obtained via a Delaunay triangulation performed over the initial set of nodes. Then 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. It is obvious that this problem is always solved if the desired number of nodes (the number of available recipient

nodes) is equal to the number of initially given nodes. If there is an available node constraint, i. e. the number of available nodes 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. However, this problem may be not solved to an exact number of available nodes, and the eventual number of tree nodes will be less than desired. Even when the root node is changed by selecting it from the missing nodes, it does not ensure the exact solution. In this case, the number of available nodes may be either decreased or increased just by 1, but this adjustment is not always possible.

2. Goals and tasks to achieve it

Given a set of N planar points (nodes), primarily not connected with edges, the goal is to build a minimum spanning tree connecting a given number M of the points, where $M < N$. To achieve the goal, obtaining a set of edges is to be formalized first. Then the minimum spanning tree built over N nodes is to be further processed to obtain a tree over M nodes. Third, the method from article [5] is to be applied to the minimum spanning tree built over N nodes with a purpose to try further shortening the total edge length.

While it is fulfilled, the root node can be changed, but number M is not changeable.

3. Obtaining a set of edges

A set of Q edges

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

is obtained from a set of N planar nodes

$$\{\mathbf{P}_i\}_{i=1}^N = \{[x_i \quad y_i]\}_{i=1}^N, \quad (2)$$

where edge \mathbf{E}_q connects nodes \mathbf{P}_{j_q} and \mathbf{P}_{k_q} for

$$j_q \in \{1, N\}, \quad k_q \in \{1, N\}, \quad j_q \neq k_q,$$

via the Delaunay triangulation [5]. The triangulation performed over set (2) does not depend on number M . However, the number of edges Q connecting planar nodes after they are triangulated is not necessarily the same for a given number N [5], [8], [9]. The result depends on the shape of planar data [10], [11]. In general, edge set (1) and its cardinality Q depend on the topology [12] of the initial set of planar nodes (2).

4. Pruning minimum spanning trees

In edge set (1), the length of edge \mathbf{E}_q is calculated using the common Euclidean metric in \mathbb{R}^2 :

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

It is quite clear that lengths

$$\{l_{\mathbb{R}^2}(\mathbf{E}_q)\}_{q=1}^Q \quad (4)$$

can be used as weights for any graph containing respective edges from set (1). So, length (3) is the weight of edge \mathbf{E}_q . A minimum spanning tree for the graph with edges (1) and their respective weights (3) contains $N - 1$ edges [1], [3], [13], [14]

$$E^{(\text{MST})} = \{\mathbf{E}_n^{(\text{MST})}\}_{n=1}^{N-1} \subset E \quad (5)$$

whose respective weights are

$$\{l_{\mathbb{R}^2}(\mathbf{E}_n^{(\text{MST})})\}_{n=1}^{N-1} \subset \{l_{\mathbb{R}^2}(\mathbf{E}_q)\}_{q=1}^Q, \quad (6)$$

where edge $\mathbf{E}_n^{(\text{MST})}$ connects nodes \mathbf{P}_{j_n} and \mathbf{P}_{k_n} for

$$j_n \in \{1, N\}, \quad k_n \in \{1, N\}, \quad j_n \neq k_n.$$

As the task is to build a minimum spanning tree over M nodes (i. e., the tree should be of $M - 1$ edges), the tree over N nodes is pruned to a tree containing $M - 1$ edges [15], [16]. At doing this, only free edges are removed from the tree. A free edge of the minimum spanning tree is such that one of its two nodes belongs only to this edge (and does not belong to any other). Thus, such a node can be called free as well. It is easy to see that a minimum spanning tree contains at least two free edges. In this case, the tree is just a polyline having no branching. Commonly a minimum spanning tree has more than two free edges (Fig. 1).

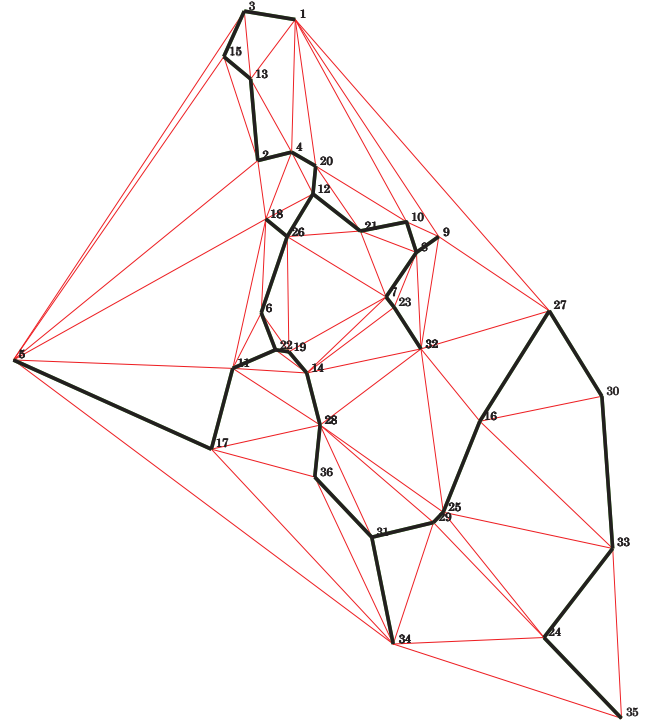


Fig. 1. A minimum spanning tree of 35 edges connecting 36 nodes by 7 free edges (whose free nodes have numbers 1, 5, 9, 18, 32, 34, 35); altogether there are $Q = 97$ edges after the triangulation (they are shown with thinner line)

To solve the problem with the available node constraint, $N - M$ free edges should be removed from the tree. Denote the set of free edges by

$$F = \{\mathbf{F}_u\}_{u=1}^U = \{[j_u \quad k_u]\}_{u=1}^U \subset E^{(\text{MST})}, \quad (7)$$

where either node \mathbf{P}_{j_u} or \mathbf{P}_{k_u} belongs only to edge \mathbf{F}_u . Obviously, no pruning is required if $M = N$; otherwise, temporary denotations

$$\begin{aligned} N^* &= N, \\ E^{(\text{MST})^*} &= \{\mathbf{E}_n^{(\text{MST})^*}\}_{n=1}^{N^*-1} = E^{(\text{MST})} \subset E, \\ F^* &= \{\mathbf{F}_u^*\}_{u=1}^U = F \end{aligned} \quad (8)$$

are done, and

$$C^{(-)} = \min\{N^* - M, U\} \quad (9)$$

free edges whose weights are the largest are removed from the tree:

$$\begin{aligned} E^{(MST)^*(obs)} &= E^{(MST)^*}, \\ E^{(MST)^*} &= E^{(MST)^*(obs)} \setminus \{\mathbf{F}_x^*\}_{x \in W} \text{ by} \\ W &= \arg \max_{u=1, U} l_{\mathbb{R}^2}(\mathbf{F}_u^*) \text{ and} \\ U^{(obs)} &= U, U = U^{(obs)} - 1, \\ F^{*(obs)} &= F^*, F^* = F^{*(obs)} \setminus \{\mathbf{F}_x^*\}, \end{aligned} \quad (10)$$

where operations (10) are repeated $C^{(-)}$ times, and only one free edge $u \in W$ is removed at once (even if there are multiple free edges whose weights are maximal). If the pruned tree still has more edges than $M - 1$ (i. e. $N^* > M$), it is pruned further. Set (7) as F^* is found again, and $C^{(-)}$ free edges by (9) are removed by subroutine (10). This routine is executed until $N^* = M$.

An example of optimally connecting 32 nodes out of 36 nodes from Fig. 1 is presented in Fig. 2, where 4 free edges are removed in one round of pruning by (7) — (10). Here and below the removed edges are shown with dashed thinner line. If the available node

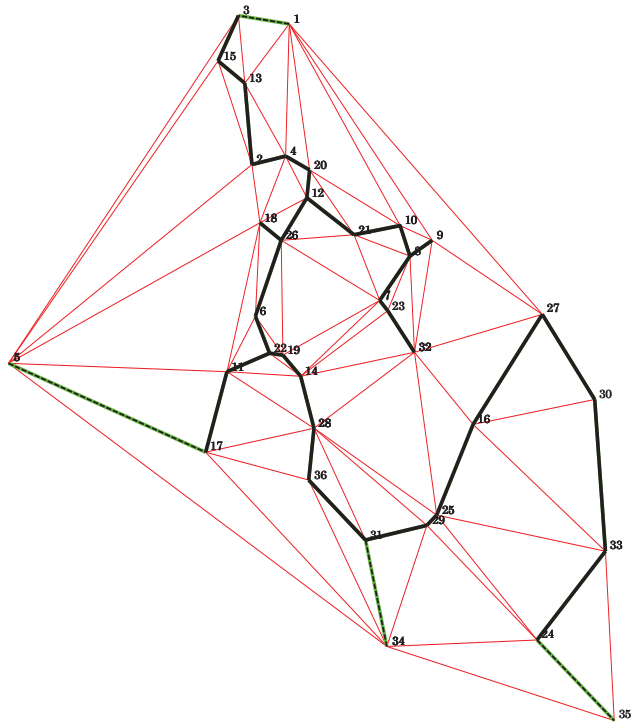


Fig. 2. The pruned minimum spanning tree from Fig. 1 for $M = 32$, where 4 free edges (whose free nodes have numbers 1, 5, 34, 35) are removed in one round of pruning by (7) — (10)

constraint is made severer, another round of pruning is required. Fig. 3 presents the solution for $M = 27$, for which 9 free edges are removed in two rounds. This is so because primarily there are only 7 free edges (see back Fig. 1), and pruning in one round cannot solve the problem. The pruned minimum spanning tree after the first round contains 4 free edges whose free nodes have numbers 3, 17, 23, 24. The edges with free nodes 3 (connected with node 15) and 23 (connected with node 7) are shorter (the axes here are scaled almost equally) than the edges with free nodes 17 (connected with node 11) and 24 (connected with node 33), and therefore the latter two edges are removed. Surely, when the number of desired nodes is decreased down to 25, i. e. 11 edges are to be removed from the tree in Fig. 1, the edges with free nodes 3 and 23 are removed at the second round (along with the edges with free nodes 17 and 24), and the problem is solved in two pruning rounds as well (Fig. 4).

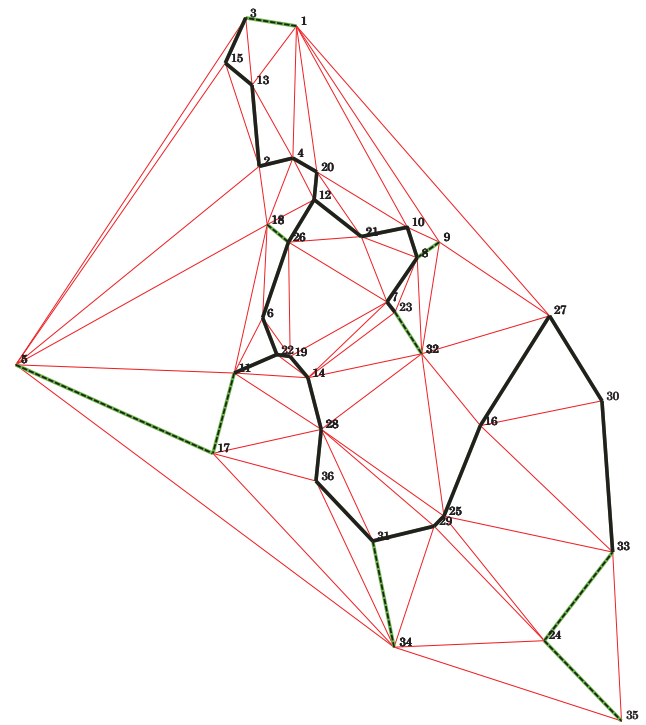


Fig. 3. The pruned minimum spanning tree from Fig. 1 for $M = 27$, where 9 free edges are removed in two rounds of pruning (first 7 free edges whose free nodes have numbers 1, 5, 9, 18, 32, 34, 35 are removed, and then 2 free edges whose free nodes have numbers 17, 24 are removed)

The total edge length is 332.2455 of the pruned tree in Fig. 4. It is noticeable that, at the second round of pruning, the free edge connecting nodes 23 and 7 is much shorter than, say, the non-free edge connecting nodes 33 and 30. So why do we remove the shorter

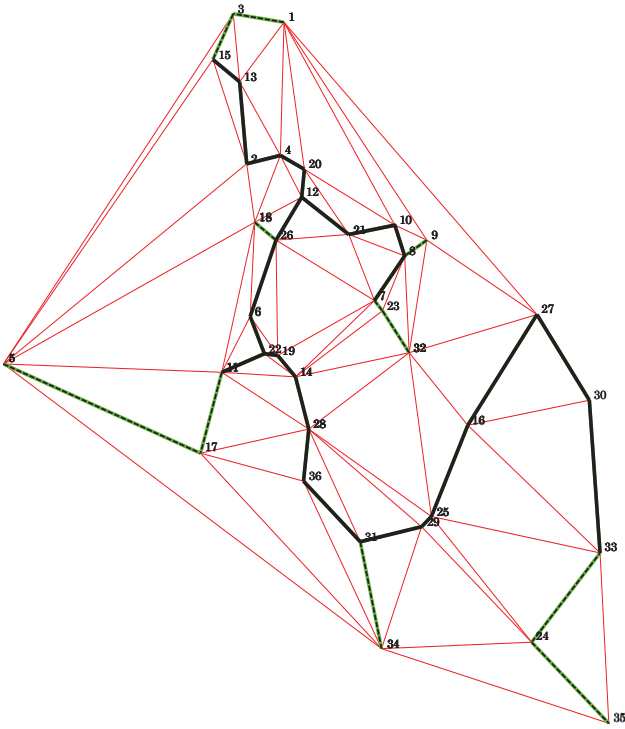


Fig. 4. The pruned minimum spanning tree from Fig. 1 for $M = 25$, where 11 free edges are removed in two rounds of pruning (first 7 free edges whose free nodes have numbers 1, 5, 9, 18, 32, 34, 35 are removed, and then 4 free edges whose free nodes have numbers 3, 17, 23, 24 are removed)

edge instead of removing the much longer one? This is so because, along with minimizing the total edge length, the pruned tree must cover the nodes fairly enough. For example, as nodes 34 and 35 are removed, removing the non-free edge with node 33 would make it far harder to get to farther nodes 34 and 35 (by ending at node 30). To the contrary, the shorter free edge with node 23 is removed, and it is far easier to get to node 32 (by ending at node 7). The latter version of the pruned tree has a greater total edge length, though. However, pruning by (7) — (10) contains a tradeoff for the sake of not losing the fair coverage.

5. Cutting longest edges

The pruning method is juxtaposed with article [5] which suggested to remove (cutting) longest edges off the set of all edges (1) right after the triangulation while the minimum spanning tree has more than $M - 1$ edges. According with article [5], temporary denotations

$$N^* = N, Q^* = Q, E^* = \{\mathbf{E}_q^*\}_{q=1}^{Q^*} = E, \quad l_{\mathbb{R}^2}^*(\mathbf{E}_q^*) = l_{\mathbb{R}^2}(\mathbf{E}_q) \text{ for } q=1, \overline{Q^*} \quad (11)$$

are done for the case of inequality $M < N$, and the

following two-step routine is executed while the number of nodes connected by edges in the minimum spanning tree is greater than M (i. e., while $N^* > M$). At the first step, the edges whose length is maximal are excluded (cut off) from set E^* :

$$E^{*(\text{obs})} = E^*, E^* = E^{*(\text{obs})} \setminus \{\mathbf{E}_h^*\}_{h \in H} \quad \text{by } H = \arg \max_{q=1, \overline{Q^*}} l_{\mathbb{R}^2}^*(\mathbf{E}_q^*). \quad (12)$$

At the second step, the respective edge weights (distances) are excluded from the set of distances

$$\{l_{\mathbb{R}^2}^*(\mathbf{E}_q^*)\}_{q=1}^{\overline{Q^*}} \quad (13)$$

by

$$Q^{*(\text{obs})} = Q^*, Q^* = \left\{ \overline{\{1, Q^{*(\text{obs})}\}} \setminus H \right\}, \quad (14)$$

whence a new set of edge weights (13) is formed. Then a minimum spanning tree is found for the graph with new edges (12) and their respective weights (13). In general, it is unlikely to have multiple edges of maximal length, and thus set H is a singleton. To make the routine stricter, only one edge of maximal length is excluded from set E^* . Therefore, update (14) is to be re-written as

$$Q^{*(\text{obs})} = Q^*, Q^* = Q^{*(\text{obs})} - 1, \quad (15)$$

whereupon the while condition $N^* > M$ is checked again.

Unlike the pruning method, cutting longest edges does not always solve the available node constraint problem. Nevertheless, it may outperform: Fig. 5 shows

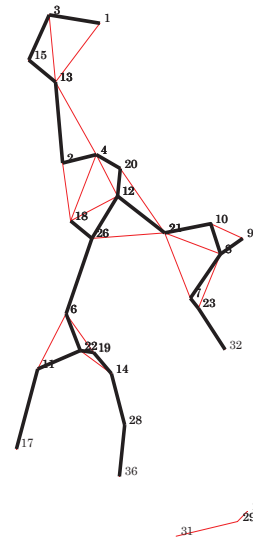


Fig. 5. Cutting longest edges from Fig. 1 for $M = 25$ produces a shorter spanning tree than that in Fig. 4

the result of applying the method of cutting longest edges to the problem with 25 available nodes out of 36 (see Fig. 1 and Fig. 4, where the scaling in Fig. 5 is maintained the same). Compared to the solution in Fig. 4, cutting longest edges produces a shorter spanning tree, whose total edge length is 254.0223, which is 23.5438 % less than the total edge length of the pruned tree in Fig. 4. This is indeed a huge difference, but the coverage is not fair here. Although the connection with node 5 is lost by both methods, a big subset of neighbouring nodes in southeast is lost while longest edges are cut off. In fact, these 10 nodes are 34, 31, 29, 25, 16, 27, 30, 33, 24, 35. In spite of the huge gain, the solution by cutting longest edges may be unacceptable in some real-world tasks of coverage with constraints due to the solution is kind of biased. Otherwise, when any biases matter not that much, the method of cutting longest edges may significantly outperform the pruning method.

Another merit of cutting longest edges is applicable to solving the available node constraint problem approximately (if approximation is admissible at all). Consider a problem with an initial set of 800 nodes, where only a half is available. The pruning method solves this problem with the total edge length of 1354.1323 (Fig. 6). The method of cutting longest edges does not solve this problem, whichever root node is selected. A minimum spanning tree nonetheless is built over 398 nodes (Fig. 7). Its total edge length is 1115.6912, which is 17.6084 % less than the total edge length of the pruned tree in Fig. 6. Thus, it appears that “turning off” just two nodes leads to a gigantic shortening of the length (that is surely connected with the cost of coverage). Why does it happen, anyway? The matter is, in this particular example, many short free edges are removed by pruning, whereas cutting longest edges avoids removing short edges. In this way, this method can be more efficient than pruning.

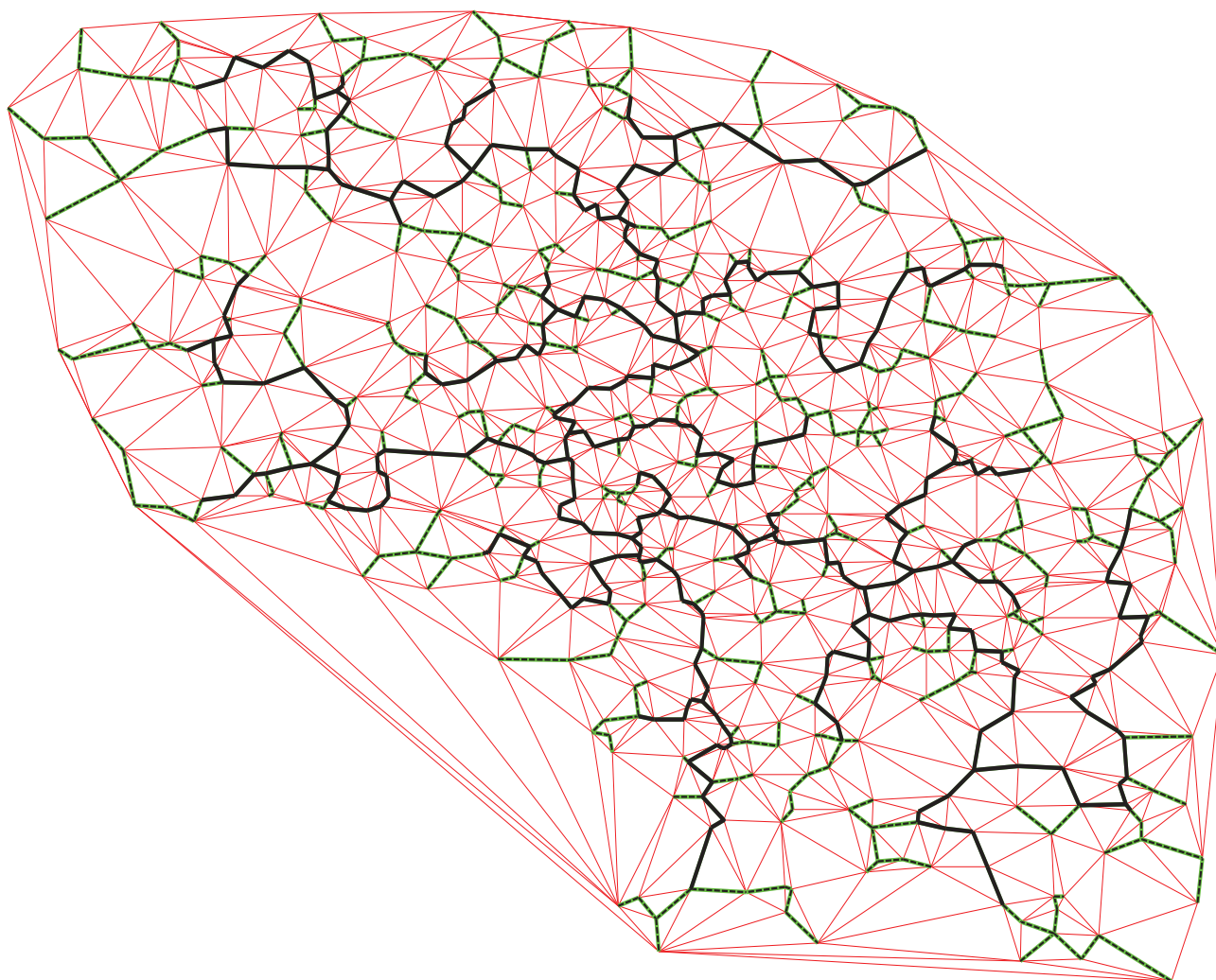


Fig. 6. The pruned minimum spanning tree over an initial set of 800 nodes, where the total edge length is 1354.1323 for $M = 400$ (exactly a half of the initial nodes is removed)

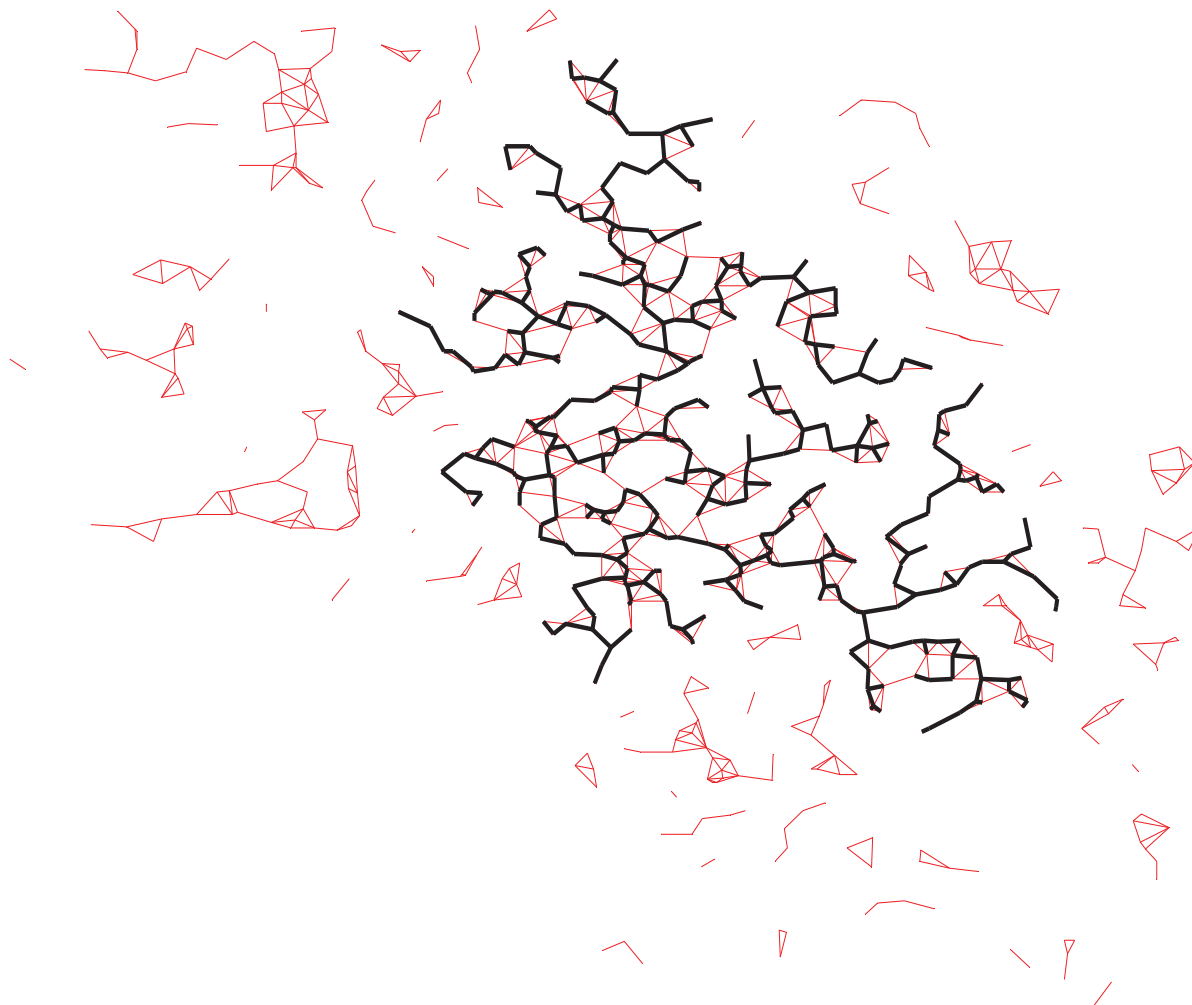


Fig. 7. Cutting longest edges from the initial edge set in Fig. 6 for $M = 398$ results in a 17.6084 % shorter spanning tree than that in Fig. 6 (“turning off” just two nodes leads to a gigantic shortening of the total edge length)

The best way is to use both methods. The initial tree is pruned first and the total edge length of the pruned tree is found. Then the set of all edges (1) is tried by cutting longest edges off it right after the triangulation while the minimum spanning tree has more than $M - 1$ edges. If the cutting results in a tree covering exactly M nodes and its total edge length is less than that of the pruned tree, then this tree is a solution. Otherwise, when the cutting results in a tree covering less than M nodes or its total edge length is greater than that of the pruned tree, the pruned tree is a solution. The case of a tie is still possible, when both trees have the same total edge length. Then an additional criterion must be introduced to select the better tree. The fair coverage can be such a criterion [17], [18].

6. Discussion

Although the pruning method is significantly faster, it may fail to produce the shortest possible tree. This property vanishes as number M is decreased with

respect to the number of initial nodes (i.e., as difference $N - M$ becomes bigger). Moreover, the volume of the initial node set does not seem to affect much the possible successfulness of the method of cutting longest edges. Thus, an example is shown in Fig. 8, where the total edge length of the pruned tree is 814.5889, but it is 778.3002 upon having cut longest edges. So, the cutting solving this particular problem exactly results in a 4.4549 % shorter tree. A huger example with 1172 initial nodes, wherein 1128 nodes are available (44 nodes are redundant), is solved by pruning to a tree whose total edge length is 1984.293, but the cutting results in an exact solution with a tree whose total edge length is 1957.8496 being 1.3326 % shorter. In another example of the same volume, where the number of redundant nodes is doubled, the pruning results in a tree whose total edge length is 1865.0095, but the cutting results in an exact solution with a tree whose total edge length is 1837.7613 being 1.461 % shorter. This means that the gain by applying the cutting can be very

significant even for voluminous node sets and big differences $N - M$. In a third example for $N = 1172$, $M = 1084$, the pruned tree has the length of 1881.6969

(Fig. 9), whereas the cutting results in an exact solution with a tree whose total edge length is 1842.5508 being 2.0804 % shorter (Fig. 10).

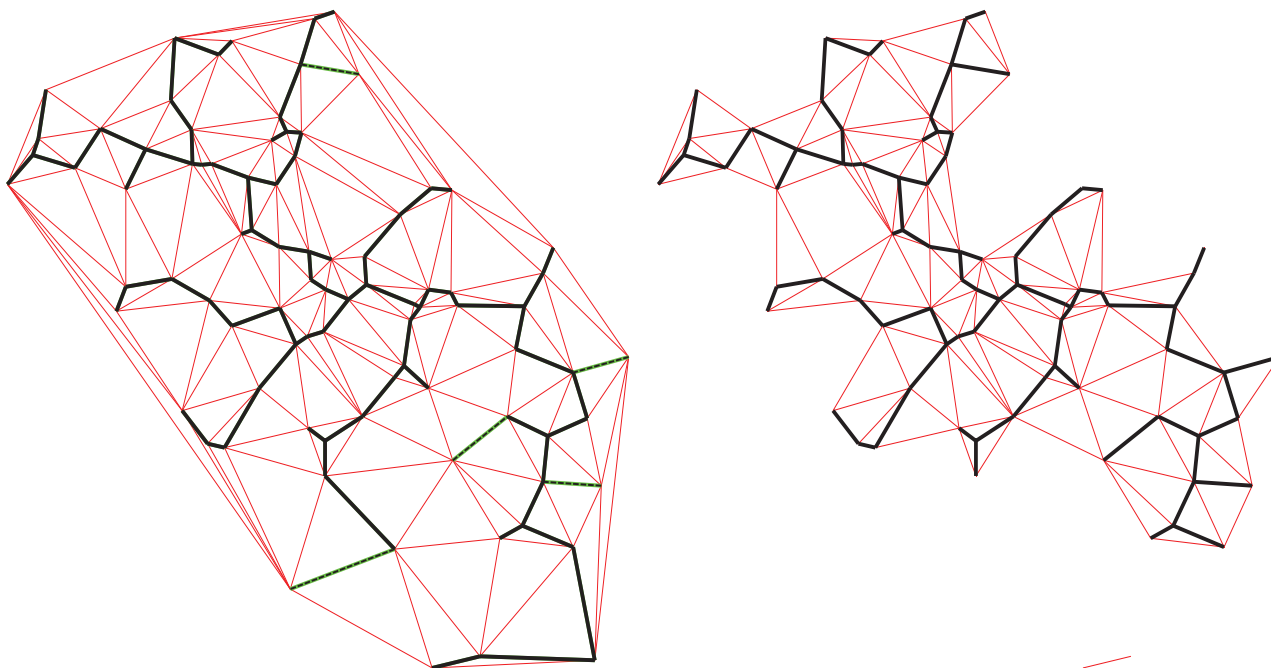


Fig. 8. The pruned tree (left) and a 4.4549 % shorter spanning tree (right) upon cutting longest edges ($N = 86$, $M = 81$)

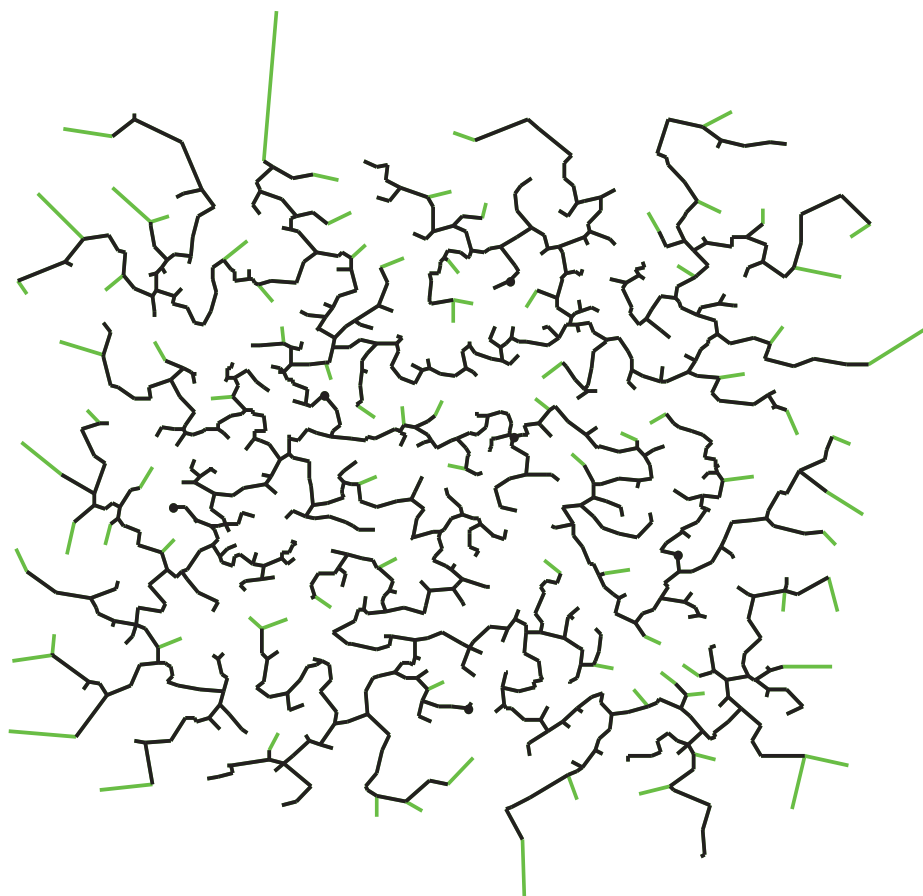


Fig. 9. The pruned tree for another example of $N = 1172$ and $M = 1084$ whose total edge length is 1881.6969

In these examples, the initial node set is generated by drawing random values from uniform and normal distributions, where the values distributed uniformly are added to the values distributed normally, and the latter are multiplied by a factor so that the part of the normal distribution is less than the part of the uniform distribution. The bigger the part of the normal distribution, the greater the difference is between the shortest and the longest edges. This might imply that the cutting method would fail more as the normal distribution part is increased (or, in other words, the uniform distribution part is decreased). However, it is

not always so. Fig. 11 presents the pruned tree for a problem with 1600 nodes, among which 100 nodes are unavailable. The tree whose total edge length is 1977.0938 resembles a square shape (just as the initial node set) because the normal distribution part here is decreased compared to the set in Fig. 9. Nevertheless, the cutting method solving this problem exactly results in a spanning tree whose total edge length is 2013.8545 being 1.8593 % longer (Fig. 12). Therefore, the pruning method surely can significantly outperform even when the edge length does not vary much (i. e., the normal distribution part is lesser).

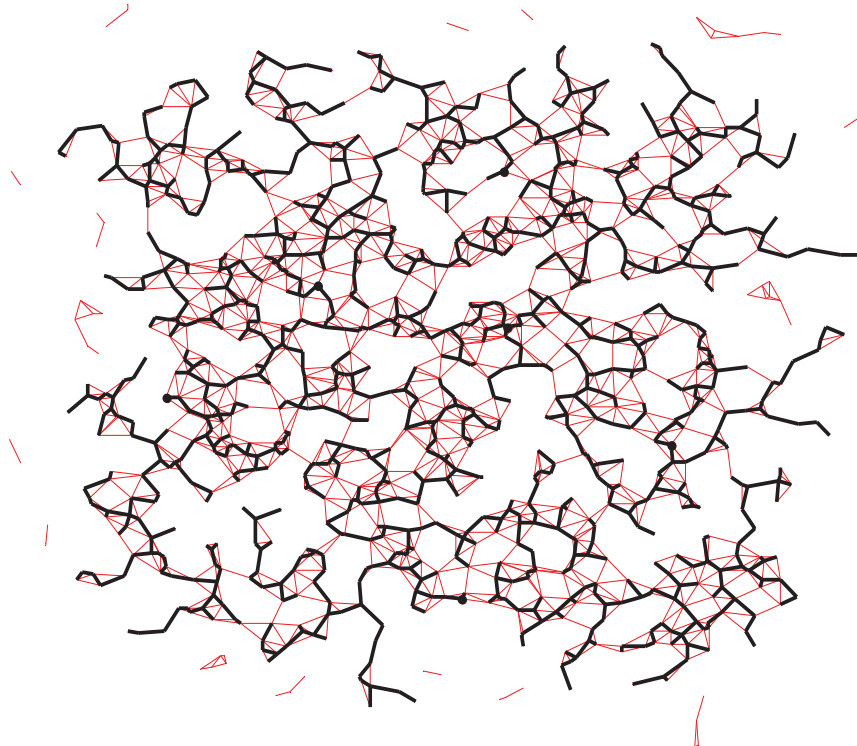


Fig. 10. Cutting longest edges from the initial edge set of 1172 nodes for $M = 1084$ results in a tree covering exactly 1084 nodes; the total edge length of the tree is 1842.5508 which is 2.0804 % shorter than that in Fig. 9 (it is noticeable that too long edges on the boundaries of the initial node set shape have been cut off)

The cutting method not necessarily producing an exact solution, often outputs a very good approximation. In another example with 1600 nodes, among which 100 nodes are unavailable, the pruned tree covering 1500 nodes has the length of 2911.3346, but the cutting method results in a spanning tree over 1499 nodes. This tree total edge length is 2801.7781, which is 3.7631 % shorter. In a one more example of the same volume, the pruned tree covering 1500 nodes has the length of 2867.4715, and the cutting method once again results in a spanning tree over 1499 nodes with the total edge length of 2712.4066. This time the gain is 5.4077 %, which seems really admissible to concede just one node.

The gain can be huger, though. A problem with an initial set of 1600 nodes, among which 1400 nodes are available, is solved by the pruning method to a tree whose total edge length is 2624.9907 (Fig. 13). Meanwhile, the cutting method produces a tree whose total edge length is 2318.3744 being 11.6807 % shorter, although the tree covers just a node less (Fig. 14). This is a gigantic gain if “turning off” a one node additionally is admitted. Besides, it is worth noting that the normal distribution part in this example is far bigger compared to the previous examples. The initial node set has a circular shape, and the edge length varies more severely (this is distinctly seen in Fig. 13). The cutting method thus shortens the edge length variation.

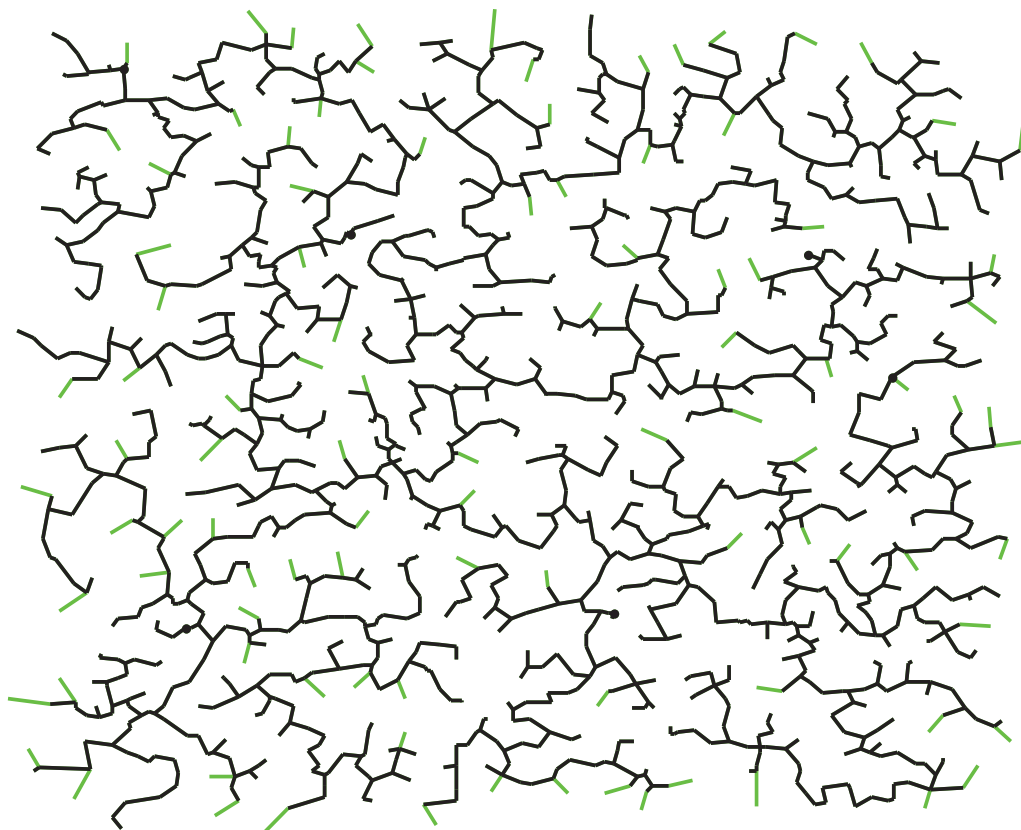


Fig. 11. The pruned tree of a 1977.0938 length over a square-shaped set of 1600 nodes with 100 unavailable nodes

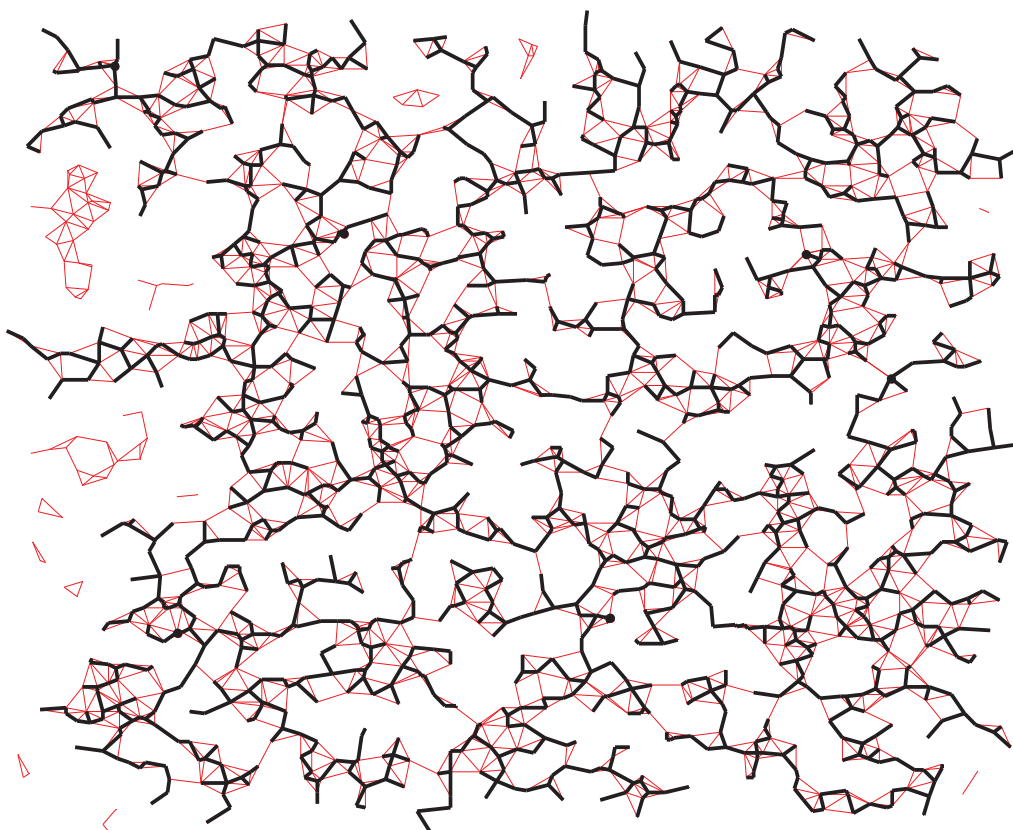


Fig. 12. A spanning tree of a 2013.8545 length upon cutting longest edges over the square-shaped set of 1600 nodes in Fig. 11 with 100 unavailable nodes (it is an exact solution, but the tree is 1.8593 % longer than that in Fig. 11)

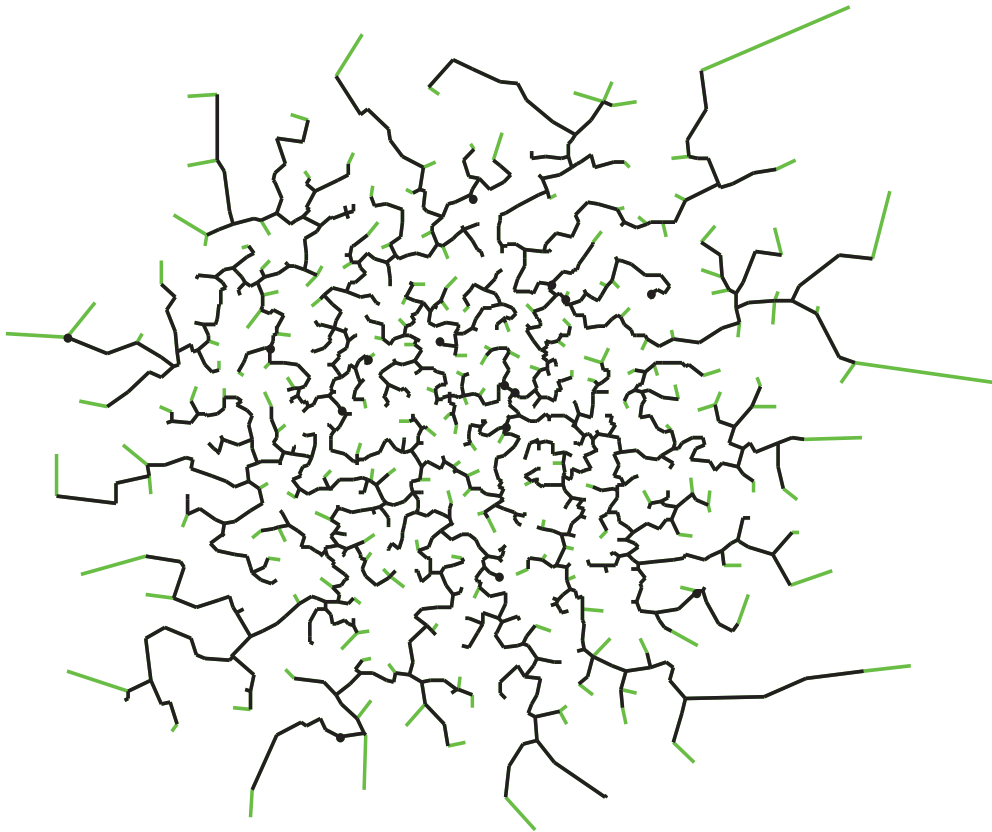


Fig. 13. The pruned tree of a 2624.9907 length for a problem with 1400 available nodes out of 1600

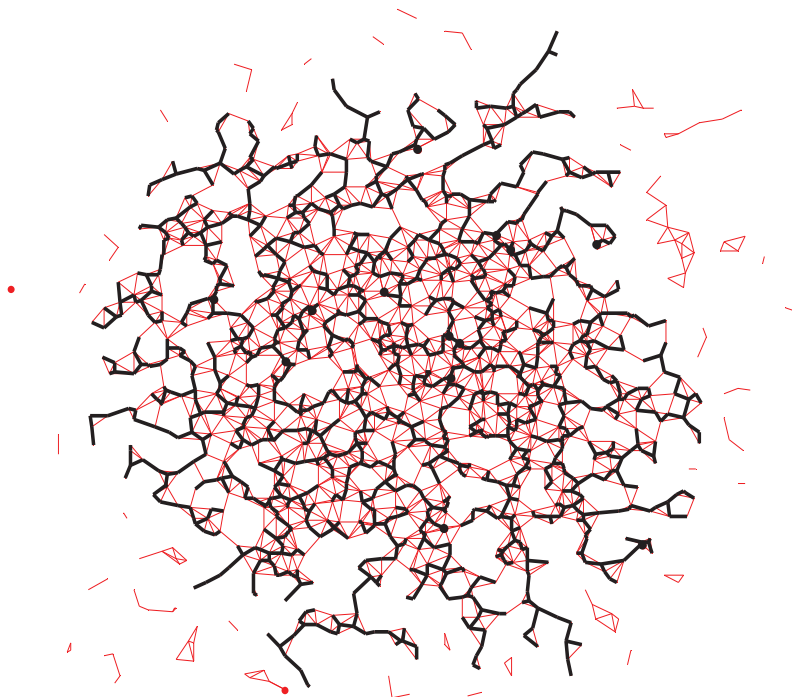


Fig. 14. A tree covering 1399 nodes (a node less) by the cutting method; it is 11.6807 % shorter than the tree in Fig. 13

The cutting method seems tending to solve the available node constraint problem exactly if there are just a few unavailable nodes. Thus, for another problem with 1600 initial nodes, among which only two nodes

are unavailable, both methods produce the exact solution — a spanning tree of a 2136.3556 length. Here, however, the cutting method is about as twice as slower than the pruning method. The slowdown can be even

more significant. Thus, a pruned tree of a 1948.1791 length (Fig. 15) for a problem with 1500 available nodes out of 1600 is found about 4.77 times faster than a spanning tree by the cutting method (Fig. 16).

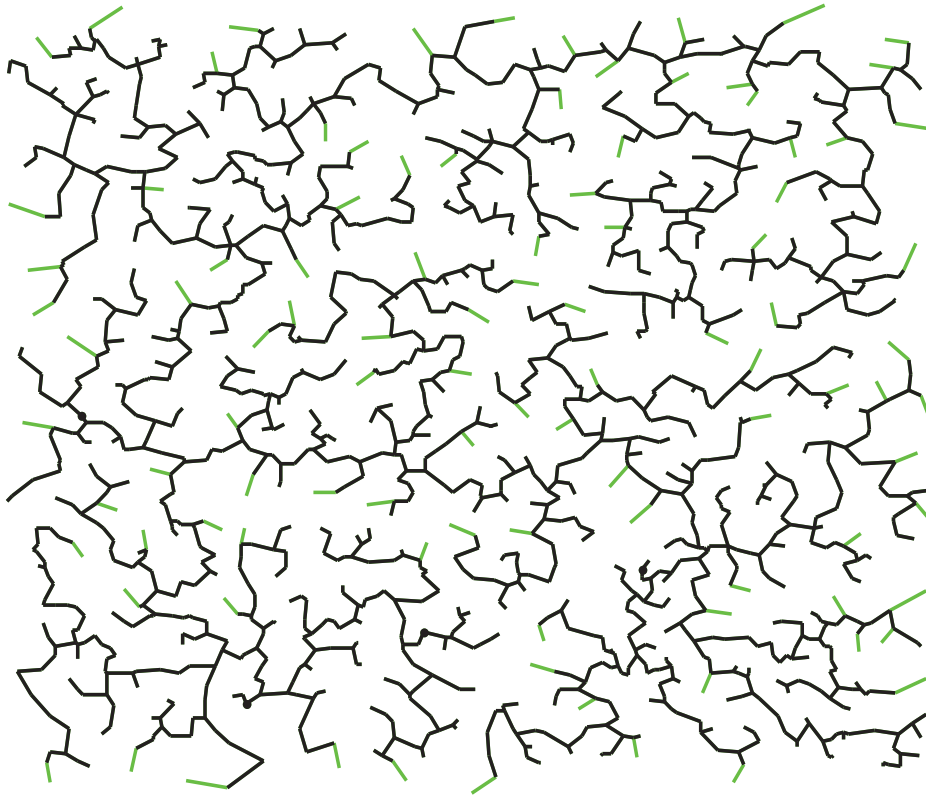


Fig. 15. The pruned tree of a 1948.1791 length for a problem with 1500 available nodes out of 1600

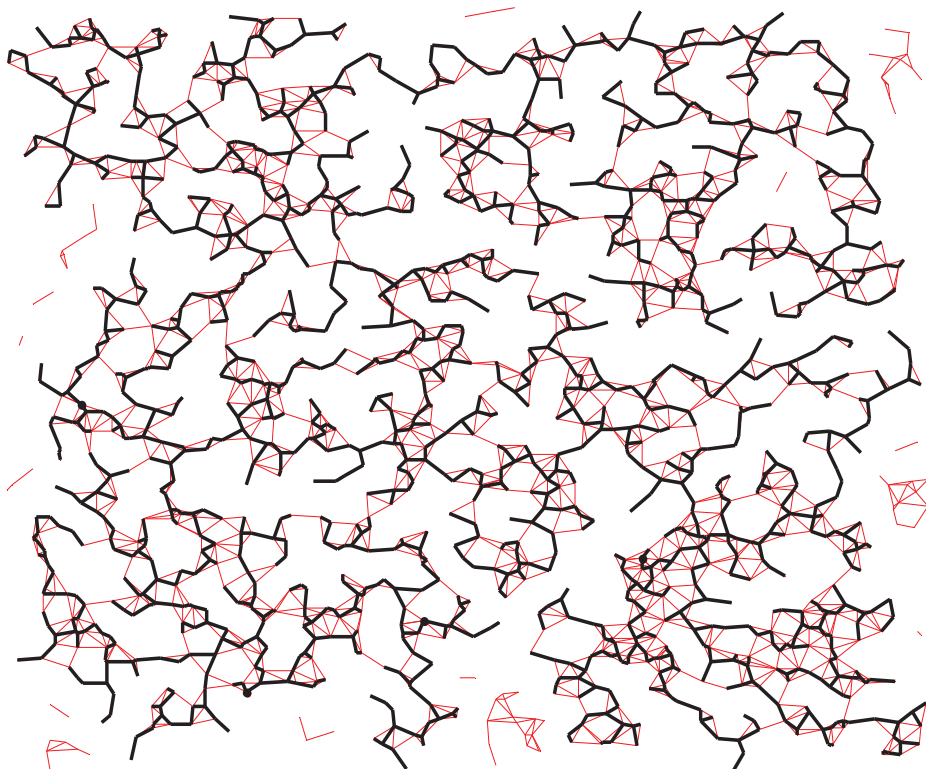


Fig. 16. A tree covering 1500 nodes by the cutting method; found by 4.77 times slower, it is 1.3433 % longer (Fig. 15)

Overall, the cutting method slowdown is caused by the cutting method requires rebuilding a minimum spanning tree while $N^* > M$, whereas the pruning method operates on the same tree built at the start of the solution process. Nevertheless, as the part of the normal distribution is bigger and difference $N - M$ is smaller, the slowdown becomes smaller. Besides, the cutting method in such cases usually provides shorter trees, although the coverage may lack for a few nodes (but difference $M - N^*$ is still not that big). As difference $N - M$ is increased, difference $M - N^*$ grows on average.

7. Conclusion

The available node constraint problem aims at building an optimally minimum spanning tree to connect a given number of planar nodes being less than an initial number of nodes by minimizing the tree length. The suggested pruning method allows reaching the desired number of nodes by pruning the minimum spanning tree connecting the initial number of nodes, where free edges whose weights are the largest are iteratively removed from the tree. The other approach, the cutting method, relies on removing longest edges off the initial minimum spanning tree, regardless of whether they are free or not [15], [19]. This process iteratively lasts while the rebuilt minimum spanning tree connects more than the desired number of nodes. Eventually, unlike the pruning method, the method of cutting longest edges may result in a minimum spanning tree connecting fewer nodes than the desired number.

However, the cutting method often outputs a shorter tree, especially when the edge length varies much. Therefore, the available node constraint problem is initially solved by the pruning method. Then the cutting method is used and its solution is compared to the solution by the pruning method. If the cutting method solution has the same number of nodes and its tree length is less than that of the pruned tree, then the cutting method solution tree is a solution to the available node constraint problem. If the cutting method solution tree connects fewer nodes than the desired number, the further matter is the relationship between the lengths of the trees produced by the methods. When the pruning method tree is shorter (or not longer, generally speaking), it is the final solution. Otherwise, the cutting method tree is shorter, although connecting not exactly the given number of nodes. So then a tradeoff for the nodes and tree length must be made, if “turning off” the respective number of nodes is admitted. In such a case, for instance, dropping

connection to just a few nodes can be more favourable by shortening the tree, if the length shortening percentage is greater than the percentage of the disconnected nodes [17], [20], [21].

Obviously, the research must be furthered by profoundly studying statistics of how both the methods perform on datasets whose edge length variation range is changeable. In this way, the slowdown of the cutting method along with its performance is to be estimated on average versus the edge length variation range and the ratio of the desired nodes to the number of initially given nodes. In addition, the tradeoff must be formulated stricter in order to automatically select the best tree, when the cutting method tree is shorter but it connects fewer nodes than the desired number.

References

1. 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>
2. 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>
3. 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>
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. V. V. Romanuke, “Building minimum spanning trees by limited number of nodes over triangulated set of initial nodes,” *Information and Telecommunication Sciences*, vol. 14, no. 1, pp. 41 — 50, 2023.
6. 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.
7. H. Edelsbrunner, T. S. Tan, and R. Waupotitsch, “An $O(n^2 \log n)$ 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>
8. 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>
9. T. Biedl, G. Kant, and M. Kaufmann, “On triangulating planar graphs under the four-connectivity constraint,” *Algorithmica*, vol. 19, pp. 427 — 446, 1997.

<https://doi.org/10.1007/PL00009182>

10. T. Biedl, “Triangulating Planar Graphs While Keeping the Pathwidth Small,” in E. Mayr (Ed.), *Graph-Theoretic Concepts in Computer Science. WG 2015*. Lecture Notes in Computer Science, vol. 9224, Springer, Berlin, Heidelberg, 2016. https://doi.org/10.1007/978-3-662-53174-7_30

11. G. Kant and H. L. Bodlaender, “Triangulating planar graphs while minimizing the maximum degree,” in O. Nurmi and E. Ukkonen (Eds.), *Algorithm Theory — SWAT’92. SWAT 1992*. Lecture Notes in Computer Science, vol. 621, Springer, Berlin, Heidelberg, 1992.

https://doi.org/10.1007/3-540-55706-7_22

12. 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.

13. 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>

14. P. Angelini, T. Bruckdorfer, M. Chiesa, F. Frati, M. Kaufmann, and C. Squarcella, “On the area requirements of Euclidean minimum spanning trees,” *Computational Geometry: Theory and Applications*, vol. 47, iss. 2, part B, pp. 200 — 213, 2014.

<https://doi.org/10.1016/j.comgeo.2012.10.011>

15. R. Ravi, R. Sundaram, M. Marathe, D. Rosenkrantz, and S. Ravi, “Spanning trees short or small,” *SIAM Journal on Discrete Mathematics*, vol. 9, iss. 2, pp. 178 — 200, 1996.

<https://doi.org/10.1137/S0895480194266331>

16. M. Chlebík and J. Chlebíková, “The Steiner tree problem on graphs: Inapproximability results,” *Theoretical Computer Science*, vol. 406, iss. 3, pp. 207 — 214, 2008.

<https://doi.org/10.1016/j.tcs.2008.06.046>

17. 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>

18. V. V. Romanuke, “Multiple state problem reduction and decision making criteria hybridization,” *Research Bulletin of NTUU “Kyiv Polytechnic Institute”*, no. 2, pp. 51 — 59, 2016.

<https://doi.org/10.20535/1810-0546.2016.2.61603>

19. J. M. Boyer and W. J. Myrvold, “On the cutting edge: Simplified $O(n)$ planarity by edge addition,” *Journal of Graph Algorithms and Applications*, vol. 8, iss. 3, pp. 241 — 273, 2005. <https://doi.org/10.7155/jgaa.00091>

20. V. V. Romanuke, “A couple of collective utility and minimum payoff parity loss rules for refining Nash equilibria in bimatrix games with asymmetric payoffs,” *Visnyk of Kremenchuk National University of Mykhaylo Ostrogradskyi*, iss. 1 (108), pp. 38 — 43, 2018.

<https://doi.org/10.30929/1995-0519.2018.1.38-43>

21. A. K. Sahu, C. Mandal, and S. Agarwal, “Approximate Minimum Spanning Tree for Points Moving in a Euclidean Two-Dimensions Plane,” in N. Chaki, N. Meghanathan, and D. Nagamalai (Eds.), *Computer Networks & Communications (NetCom)*. Lecture Notes in Electrical Engineering, vol. 131, Springer, New York, NY, 2013.

https://doi.org/10.1007/978-1-4614-6154-8_67

Романюк В.В.

Підрізання мінімальних сполучних дерев та відрізання найдовших ребер для сполучення даної кількості вузлів при мінімізації загальної довжини ребер

Проблематика. У багатьох завданнях проєктування ефективних телекомунікаційних мереж кількість вузлів мережі є обмеженою, але вибір вузлів є ширшим. Для подальшого задоволення споживачів фактично існує більше потенційних місць розташування, ніж наявних засобів для їх розміщення у цих місцях. Це призводить до виникнення задачі з обмеженням доступних вузлів.

Мета дослідження. Для даної початкової множини вузлів на площині задача полягає у побудові мінімального сполучного дерева, що поєднує задану кількість вузлів, котра є меншою за кількість елементів початкової множини. Тому задача з обмеженням доступних вузлів має на меті побудову оптимально мінімального сполучного дерева для сполучення заданого числа вузлів на площині, що є меншим за початкове число вузлів, за мінімізації довжини цього дерева.

Методика реалізації. Виконується триангуляція початкової множини вузлів. Це дає набір ребер, чії довжини обчислюються і використовуються як ваги графа. На цьому графі будується мінімальне сполучне дерево. Бажана кількість вузлів досягається підрізанням цього мінімального сполучного дерева, з якого ітеративно видаляються вільні ребра, чії ваги є найбільшими. Інший підхід, метод відрізання, видаляє найдовші ребра з початкового мінімального сполучного дерева незалежно від того, чи вони є вільними.

Результати дослідження. На відміну від методу підрізання, метод відрізання найдовших ребер може призвести до мінімального сполучного дерева, що поєднуватиме менше вузлів, ніж потрібно. Однак метод відрізання часто видає коротше дерево, особливо коли довжина ребра сильно варіюється. Крім того, метод відрізання є повільнішим через те, що він ітеративно перебудовує мінімальне сполучне дерево.

Висновки. Спочатку задача розв’язується методом підрізання. Тоді застосовується метод відрізання і його розв’язок порівнюється з розв’язком за методом підрізання. Коротше дерево є найкращим. Можливий компроміс числа вузлів і довжини дерева.

Ключові слова: мінімальне сполучне дерево; триангуляція; довжини ребер; підрізання дерева; відрізання найдовших ребер.