

Evolutionary Approach to Constrained Minimum Spanning Tree Problem – Commercial Software Based Application

Anna Pagacz¹, Günther Raidl², Stanisław Zawiślak³

¹ Vienna University of Technology, Socrates student 2004/2005; regular since 2005-onwards, Vienna, Austria, e-mail: annapagacz@poczta.onet.pl

² Vienna University of Technology, Institute of Computer Graphics and Algorithms, Vienna, Austria, e-mail: raidl@ads.tuwien.ac.at

³ University of Bielsko-Biała, Faculty of Mechanical Engineering and Computer Science, Bielsko-Biała, Poland, e-mail : szawislak@ath.bielsko.pl

Abstract. The constrained minimum spanning tree problem is considered in the paper. We assume that the degree of any vertex should not exceed a particular constraint d . In this formulation the problem turns into NP-hard one, therefore the evolutionary approach is applicable. The edge set representation of a chromosome was utilized for a tree in the algorithm. The evolutionary algorithm was worked out and the related computer program has been written. Interfaces between the core program and MS Visio as well as the data base system were prepared. The results obtained by means of the system are shown.

1 Introduction

Graph theory problems are solved by means of versatile algorithms which have been developed since the origins of the theory in the eighteen century. Some NP-hard problems are solved by means of evolutionary algorithms.

The goal of this paper is to give a short review of the papers dealing with algorithms for generalized spanning tree problems i.e. in non-classical formulated versions as well as to describe a computer program system (using commercial software) developed to solve this problem for one particular formulation.

It is a well known fact that an exact solution can be found for some classical problems in plain formulations e.g. minimum spanning tree problem (MST-P) or the shortest path problem. However, even those are turned into NP-hard problems when constraints are introduced or some additional assumptions are made. For example, it has been proved that quadratic minimum spanning tree (q-MST) is NP-hard [18]. However an EA approach is reasonable because of their multi-objective formulation. This area is especially suitable for EA because in the multi-objective approach we consider a Pareto set of solutions which can be analyzed iteration by iteration considering the whole population of solutions instead of one solution [4]. Furthermore, it has been proven that an evolutionary approach to some graph theory problems is comparably more effective than some other approaches e.g. ant colonies, neuronal networks and other approaches belonging to AI domain. Moreover EA-based methodology is especially flexible, robust and handy [3]. At present, telecommunication is one of a very fast growing field of science. At present, telecommunication presents a challenging range of difficult design

and optimization problems. An increasing interest in the application of evolutionary computation has been observed in recent years [17]. An application of AI based methodologies to telecommunication problems is connected with a new approach to optimisation task in general i.e. we are not only interested in finding the best solution, but also we are interested in finding the whole set of approximate solutions (a neighbourhood of an optimal one or a neighbourhood of a Pareto front). Such strategy allows us to choose proper solutions, and take into consideration different criteria and constraints [2,9]. The area of application of EA to tree-based problems is not restricted only to networks. The vast related field consists also in investigations of trees as a data structure as well as decision trees e.g. [6].

In this paper we consider that our solution should represent a network without loops with minimal total connection cost between all nodes (i.e. spanning tree) under a given degree constraint. It means that the number of edges adjacent to each vertex must be less than, or equal to, a given upper bound d . The natural way of representing a network is a graph so we formulate the problem using graph theory notation. There were several attempts to solve this problem by means of EA in many different formulations [5, 12-15, 19].

The present paper focuses its attention on building an effective, user-friendly system to solve the MST-P in the formulation given in chapter 2 using the data input interface (via MS-Visio).

2 Problem Formulation

The problem of identifying a minimum spanning tree (MST) of a connected, undirected graph belongs to the group of classical combinatorial optimization problems which can be solved by greedy heuristics: i.e. Kruskal's or Prim's in polynomial time. Unfortunately there exist several practically relevant variants of MST problem that were proven to be NP-hard. One of these problems is the degree-constrained MST problem. The essence of the problem consists in identifying the shortest spanning tree whose node degrees do not exceed an upper bound $d \geq 2$. In [1], it was shown that the DCMST is NP-hard by reducing it to an equivalent symmetric TSP. Thus it is likely that any exact solution approach will be inefficient as the size of problem instance increases.

Let's consider an undirected complete graph $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ is the set of n nodes (or vertices) and $E = \{e_1, \dots, e_m\}$ is the set of m arcs (or edges) with given costs c_{ij} for each arc $e \in E$, connecting vertices v_i and v_j . The *degree constrained minimum spanning tree* (DCMST) problem on G is to find a spanning tree of minimum total cost, such that the degree of each node is at most a given value d_i . Degree is defined as the number of arcs incident to v_i . For simplicity v_i is denoted by i in figures and some descriptions.

The algorithm to find a minimum spanning tree has been implemented. The problem of finding the minimum spanning tree includes a constraint as follows: degree-constrained d , which expresses that a node degree do not exceed an upper bound d . With reference to a specific device it expresses a number of lines that can leave a node. Second aim is to minimize the objective function (cost of building the network's connections).

Let C represents a whole current cost of network's connections. Because network is composed of network devices (servers, personal computers, routers, etc.) let each device to be represented by one node. In order to connect two nodes we have to use a link of a length (i, j) , where i represents the begin of the path, and j represents the end of path between these two devices. For simplicity reasons, the path's length that simultaneously expresses the cost of connection between two devices, is calculated in Euclidean space (Figure 1), and it is computed by the following formula:

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

Because a network can be represented by a graph, in which each vertex represents a single node, therefore it is necessary to minimize the objective function:

$$C = \sum_{(i,j)} c_{ij} \rightarrow \min \quad (2)$$

It means, we have to find a graph with the minimal total cost, in other words we have to find a minimum spanning tree that contains all network's devices, which are laid out by user in MS Visio.

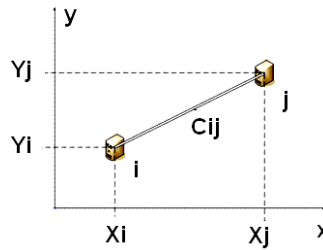


Figure 1. Calculating the weight c_{ij} of an edge (i, j) ; part of MS Visio screen editor

We wanted to find a minimal total cost of connection between all nodes, for the given layout of devices, and taking into account degree-constraint condition. Given constraints are as follows:

$$C = \sum_{(i,j)} c_{ij} \rightarrow \min$$

$$\text{deg}(i) \leq \mathbf{d}, \quad i \in V. \quad (3)$$

For this formulation of the problem, the computer program was created and tested for up to 60-vertex-graphs.

3 Evolutionary Approach

Several consecutive activities have to be done to develop an evolutionary approach to a particular problem. According to the well-known textbook [8], it demands among other - to chose encoding structures and to create adequate (tailored) evolutionary operations. These problems were considered for spanning trees in some papers [7, 11, 14, 16].

However, based on the reference review, in order to fulfil most of foregoing assumptions, the edge-set representation has been chosen and implemented. In this representation a single edge $e \in E$ is represented by just two nodes $v_i, v_j \in V$. It is a very simple and effective method of coding of spanning trees, when the mutation and crossover operators are specially built. The edge-set representation has been proposed by Raidl [12, 14-15]. The exemplary spanning trees G and H as well as their representations (i.e. Ch_G and Ch_H) are presented in Figure 2. The set representations forces the user to define adequate evolutionary operations.

A single genotype represents a single feasible solution, in other words a single possible connection between all nodes of graph, with given constrains. To initialise a population of solutions, modified Prim's algorithm has been used. Aiming for preservation of a diversity of an initial population, Prim's algorithm has been modified to a random manner [14-15]. To implement this algorithm, linked lists have been utilized. After each iteration a proper node has been discarded from a proper linked list which represents a set of connected or unconnected nodes.

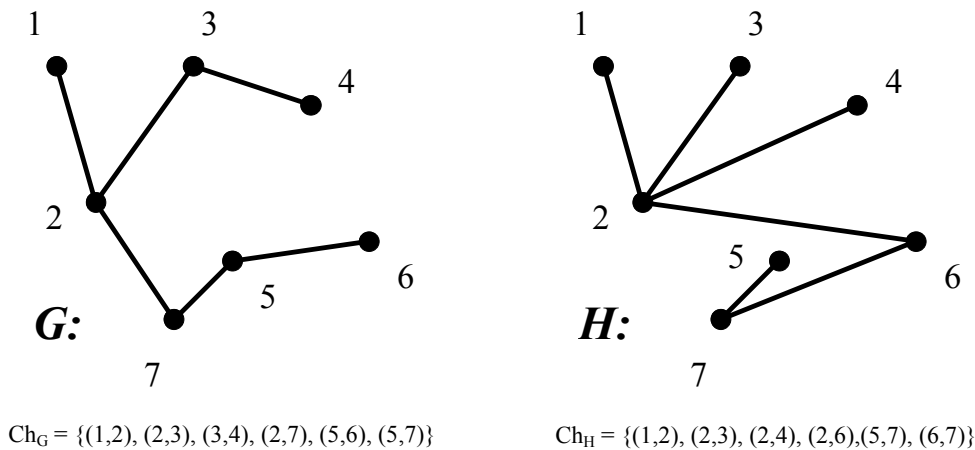


Figure 2. Two spanning trees and their representations as sets of edges.

In each iteration a number of elements held in each linked list were decreasing, resulting faster performance of the algorithm. In order to preserve a diversity of an initial population, the above procedure has been modified. The modification includes shuffling of elements in the set of not connected nodes. To generate an initial population - both procedures are used with a ratio 50% - 50 %. First part of individuals is generated by the procedure without shuffling and second part by the procedure with shuffling.

To preserve a good heritability, a recombination operator should build an offspring spanning tree (ST) that consists mostly or entirely of edges found in the offspring's parents. (Here the name recombination is more frequently used than crossover.) This can be done for any of the random tree algorithms. Let T_1 and T_2 to be sets of parental edges. If a problem does not include constraints it is possible to generate a new offspring using only parental edges. To specify a new offspring always the set $T_1 \cup T_2$ is generated. When we consider the constrained problem, generation of an offspring using only parental edges may violate constraints and produce an infeasible trees. When recombination runs out of feasible edges, it must include feasible edges from the set $E - \{T_1 \cup T_2\}$. Usually only a few non-parental edges are necessary to build a valid offspring. This strategy avoids complex repair and penalty mechanisms. The recombination applies random Prim's algorithm. Two nodes are chosen at random: one in set C and its degree must be less than d, and the second one, in the set of unconnected nodes. An edge that connects this two nodes is added to the tree T.

To build an offspring which includes as much as possible parental genetic information, edges in the set of edges contained in both parents ($T_1 \cap T_2$) own the highest priority. In next step all edges contained in the set $E' = (T_1 \cup T_2) - (T_1 \cap T_2)$ are checked for inclusion. If a new

edge can not be chosen from parental edges, a new edge is chosen at random in set $E'' = E - E'$ (Figure 3).

To find a global optimum in reasonably good time, a local heuristic has been implemented. To favour low-cost edges, the operator chooses each time the next edge from E' in a 2-tournament with replacement. By applying this simple and efficient technique low-cost edges are always favoured but more expensive ones are not excluded. It does not increase the time complexity of the operator.

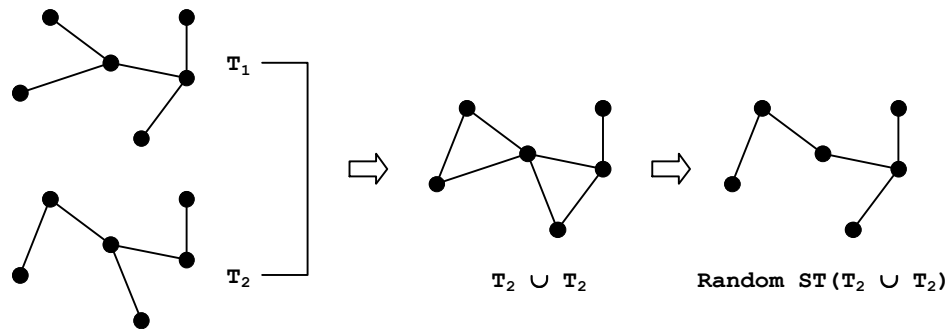


Figure 3. Recombination of spanning trees

Mutation should introduce genetic information into the population. To provide a high locality, a mutation operator should usually make a small change in a parent solution. The mutation replaces one spanning tree edge with another feasible edge.

An arbitrary edge is chosen at random from $E - T$ and then it is added to T (bold line). This operation creates a cycle, that has to be detected (stripped lines) and at the next step, the operator chooses an edge from the cycle at random and removes it from T . This process is shown in Figure 4.

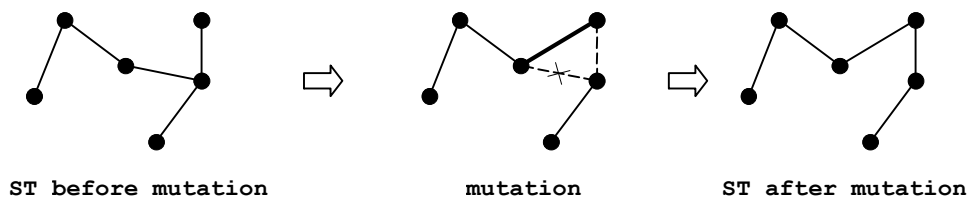


Figure 4. Mutation which introduces a new random edge.

The new edge which is added to the tree is marked with bold line, a cycle is distinguished by dashed lines, and an edge that is removed is marked by a cross.

In the implemented algorithm, a condition concerning degree of each vertex is checked. It does not violate the evolutionary approach but such a checking is impossible to perform in classical algorithms which have been proven on the basis of certain strict assumptions. Degrees of vertices which create an edge can not violate a degree constraint. If degree of both nodes equals d then a new edge is a random one. The edge - that is deleted - is dependent on its degree and priority, it means if it is or is not a root node. An operator has been constructed in

such a way to always create a feasible spanning tree without violating a degree-constraint. Before introducing a new edge a mutation operator uses a local heuristic which favours the low-cost edge via 6-tournament selection. This process leads to minimization of the objective function that express a total connection cost of all nodes. In [14] Raidl has proposed the strategy that consists in sorting all edges by their cost, then each edge has a rank with ties broken randomly. In the implemented algorithm as a local heuristic a 6-tournament selection has been involved. It provides an effective searching of the space of possible solutions and leads to minimization of the objective function. Selection of individuals to recombination process is made by a tournament selection, with the coefficient $k = 2$.

Two evolutionary algorithms have been implemented. First implemented Algorithm 1 is based on a standard genetic algorithm. In each iteration of generated offspring, the representatives are selected via a tournament selection. In Algorithm 2 after evaluation of new generated population, the best and the worst representatives are found. If the best representative in current population is worse than the best found in one of the previous iterations, then the worst individual in a current population is replaced with the best found in one of the previous iterations.

4 Computer System

User's interface has been designed in C++ language. MS Visio application has been adopted to place the network's devices (according to a planned layout), and ASA 6.0 database server has been used to hold positions of the placed devices.

A practical version of an idea is described underneath. On the following picture (Figure 5) every single stage of a user's work can be seen, with constraints/conditions from 1 to 10.

They have following meaning:

- (1) Designing of the distribution/layout of network devices by a user;
- (2) Setting-up the correct parameters before exporting data to the database; PinX, PinY we set to millimetres, and a type of data estimation to integer;
- (3) Selection of a database and a table, where data are suppose to be sent;
- (4) Selection of the database, that we want to connect to (to where the object positions have been exported);
- (5) Data importing from the database to application (C++). Values saved as a millimetre unit are changed to kilometre unit. To make the data importing possible a class named 'problem' has been created. After connecting to the database, and executing a query, a matrix of distances between every two nodes is generated. Object of this class are the best individuals found in consecutive generations, and also it is used to initialise objects of classes: 'population' and 'chromosome';
- (6) Possible changing of parameters by a user and setting in motion one of the two adopted evolutionary algorithms, or Prim's algorithm as a standard solution;
- (7) Adequately designed operators of mutation and crossover and initialisation procedure of the initial population, find the global minimum, or the another value which is very close to the global optimum (especially in the case of a big number of nodes and a big value of constraints);
- (8) The gained results are saved to the text file;
- (9) Giving the best result – it is shown in kilometre unit;
- (10) Because Visio by default draws objects in inch units, therefore position of every device are being changed into inch units.

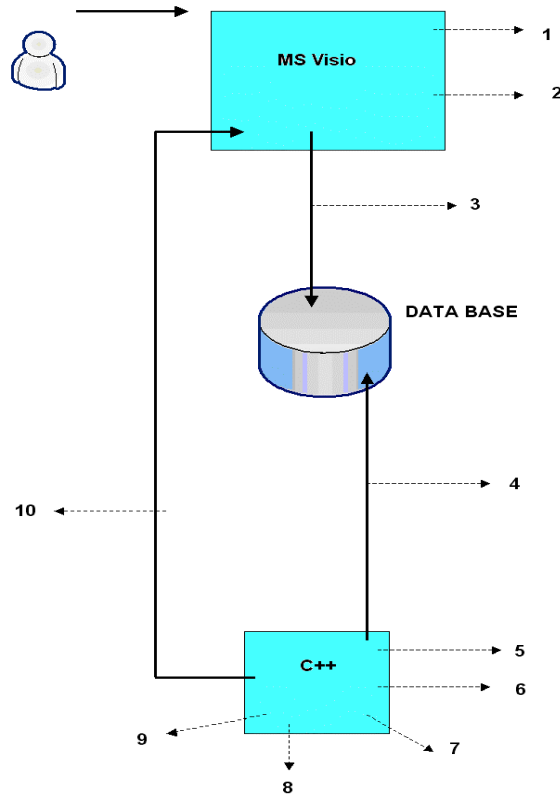


Figure 5. Scheme of realization of the assumed aim.

5 Exemplary Results

The algorithms and all cases were tested on population holding 100 chromosomes. For each case 20 trials were performed, and for each case an average value and the best founded representative of the solution were calculated. The number of nodes was changeable, as follows: 20 nodes, 30 nodes, 40 nodes and 60 nodes. The algorithm was tested with different values of probability of crossover and mutation in order to find optimal values of coefficients for the given problem. Some results are depicted in Table 1 and 2.

Tests have been made for up to 60 nodes. As we can see in Table 1 – it is more difficult to find optimal connection reaching the global optimum when the number of nodes increases, even if the number of iteration also increases. When the number of data and constraints grow, the program slows down, therefore the total time of solving the problem grows. The growth number of initial population and the growth of number of generations, increases the storage capacity. The higher probability of mutation and crossover increases the computational time of the algorithm, because along with growth of probability the number of recombined and mutated individuals grows.

Table 1. Genetic algorithm (ALGORITHM1) – exemplary results

Number of nodes	Constrain Deg(i)< d	Mutation probability	Crossover probability	Number of generation	The best Found solution	An average value after 20 trials
20	4	0.6	0.6	1000	0,1203	0,1203
	5	0.6	0.6	1000	0,1203	0,1203
	6	0.6	0.6	1000	0,1203	0,1203
	7	0.6	0.6	1000	0,1203	0,1203
30	4	0.8	0.8	1000	0,1362	0,13621
	5	0.8	0.8	1000	0,1362	0,136204
	6	0.8	0.8	1000	0,1362	0,136204
	7	0.8	0.8	1000	0,1362	0,13623
40	4	0.8	0.8	1000	0,1749	0,1749
	5	0.8	0.8	1000	0,1749	0,1750
	6	0.8	0.8	1000	0,1749	0,17507
	7	0.8	0.8	1000	0,1749	0,17497
40	4	0.6	0.6	1000	0,1749	0,1749
	5	0.6	0.6	1000	0,1749	0,1750
	6	0.6	0.6	1000	0,1749	0,1749
60	4	0.8	0.8	2000	0.2068	0.2073
	6	0.8	0.8	2000	0.2077	0.2084

Table 2. Evolutionary algorithm (ALGORITHM 2) – exemplary results

Number of nodes	Constrain Deg(i)< d	Mutation probability	Crossover probability	Number of generation	The best Found solution	An average value after 20 trials
40	4	0.6	0.6	1000	0,1749	0,1749
	5	0.6	0.6	1000	0,1749	0,17497
	6	0.6	0.6	1000	0,1749	0,1749
	7	0.6	0.6	1000	0,1749	0,1749

Although the increment of mutation probability slows down the effects of program, however it leads to better results. Moreover, the increment of the size of initial population causes growth (widening) of a search space. It is effective when the crossover operator is involved in searching.

Along with the increment of node's degree, the algorithm needs more time to find global optimum, or it finds the solution which is close to global optimum. When the value of degree is small the global optimum is found. The results of algorithms: ALGORITHM1 and ALGORITHM2 (Table 2) differ insignificantly, sometimes the results of ALGORITHM2 are better than results of operation of ALGORITHM1, but computational time of ALGORITHM2 is longer because of additional operations of finding the best and the worst representative and substitution's operations. Due to random nature of the algorithm, for each case 20 trials were performed, and average value of results was calculated. The exemplary results are presented in tables above and in Figure 6.

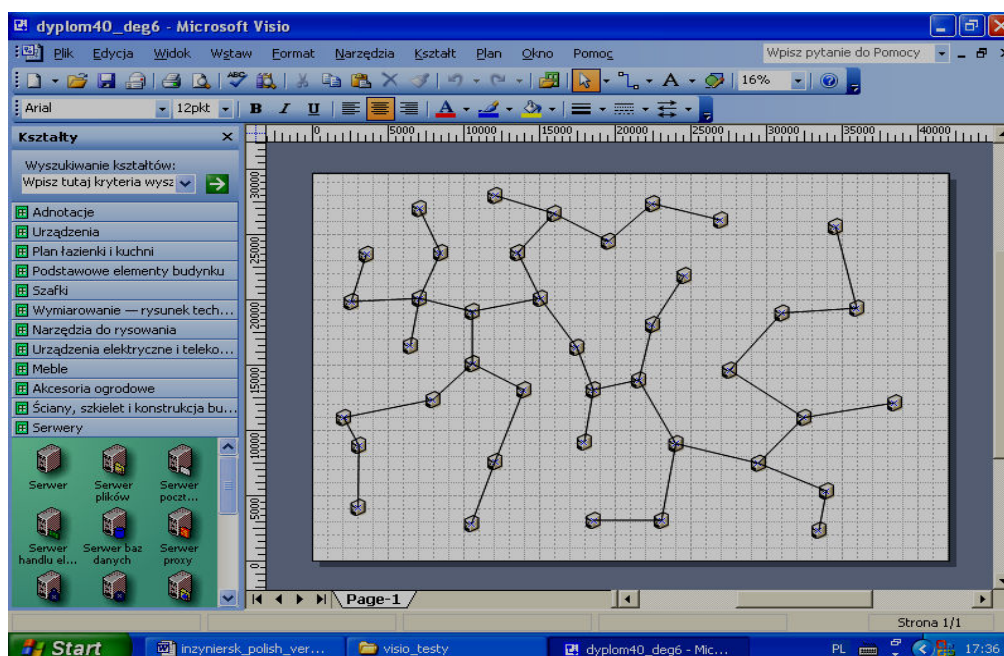


Figure 6. Graphical presentation of the connection found via the evolutionary algorithm; nodes number = 40, constraint degree = 6, probability of mutation and crossover = 0.8.

The best results were returned when using the probability with value 0.8 for both operators: mutation and crossover, although when the value of degree was small and the value of probability was 0.6 the results were identical or insignificantly worse. With small number of nodes (network's devices) and small value of degree the algorithm was able to solve the problem and find the minimum spanning tree even in fifty first iterations.

Other details and wider number of computer program outcomes are enclosed in the B.Eng.-thesis [10].

6 Final Remarks and Conclusions

The evolutionary algorithm which uses a simple but an effective representation of chromosome, where the solution is a set of all edges in a spanning tree, has been utilized. The proper operators for a such representation of a chromosome have been implemented.

The genetic operators always construct the feasible solutions, similar to initialization procedure, which allows us to avoid introducing any repair procedures or penalty function for individuals, which violate constraints. The mutation operator leads to minimization of the objective function faster than the crossover operator. The crossover/recombination operator usually takes advantage of the parental edges and this limits the search space and simultaneously slows down the searching process.

In each iteration the new edge is chosen in a tournament selection. Similarly, the crossover operator and the initially procedure use local heuristics, which favour low-cost edges. These greedy heuristics speeds up the action of algorithm, and usually provides better results, but extends the computational time of the algorithms.

The linked list used for operations of mutation, crossover and initial procedure speed up the operation of the genetic operators, because in each iteration the number of nodes in the random set changes. The time of a program performance was reduced by tournament selection, which is a simple way to select a better individual for the recombination process. When using tournament selection and above mentioned operators and initialisation; the new solution can be generated and evaluated in time $O(|V|)$.

The increase of problem's parameters: number of nodes and degree constraint causes an increase of computational time and makes it difficult to find a minimum spanning tree.

The user's interface of program has been designed which allows for designing a network using MS Visio, and then in next step connections between all devices having the minimal total cost are found. To communicate between MS Visio and application written in C++ the database based on ASA 6.0 server has been used. Having performed tests and based on comparison of the results, the standard parameters were adjusted for C++ application. However the user has rights to change the parameters and tool setting.

Bibliography

- [1] Garey M.R., Johnson D.S. Computers and Intractability, A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco, 1979.
- [2] Gen M., Kumar A., Kim J.R., Recent network design techniques using evolutionary algorithms, *Int. J. of Production Economics*, 98, 2006, 251-261.
- [3] Kirshnamoorthy M., Ernst A. T., Comparison of Algorithms for the Degree Constrained Minimum Spanning Tree, *Journal of Heuristic*, 7, 2001, 587-611.
- [4] Knowles J.D., Corne D.W., A Comparison of Encodings and Algorithms for Multi-objective Minimum Spanning Tree Problems, University of Reading, UK, <http://www.rdg.ac.uk/~ssr97jdak>.
- [5] Lin G-H., Xue G., Steiner problem with minimum number of Steiner points and bounded edge-length, *Information Processing Letters*, 69, 1999, 53-57.
- [6] Mak B., Blanning R., Ho S., Genetic algorithm in logic tree decision modeling, *European Journal of Operation Research* 170, 2006, 597-612.
- [7] Mehlhorn K., Data Structures and algorithms, Vol. II, Graph Algorithms and NP-Completeness, Springer Verlag 1984.

- [8] Michalewicz Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, PWN, Warszawa 2002.
- [9] Ovalle-Martinez F.J, Stojmenovic I, Garcia-Nocetti F., Solano-Gonzalez J., Finding minimum transmission radii for preserving connectivity and constructing minimal spanning tree in ad hoc and sensor networks, *J. Parallel Distrib. Comput.*, 65, 2005, 132-141.
- [10] Pagacz A. The system of optimal network designing based on evolutionary algorithms, B.Eng.-thesis, University of Bielsko-Biała, Poland, Supervisor (Host University) S. Zawislak, Co-supervisor (SOCRATES placement) G. Raidl (TU Vienna), Bielsko-Biała, Poland, 2005.
- [11] Palmer C.C., Kershenbaum A., "Representing trees in genetic algorithms," in Proceedings of the First IEEE Conference on Evolutionary Computation, David Scharrer, Hans-Paul Schwefel, and David B. Fogel, Eds., IEEE Press, 1994, 379–384.
- [12] Raidl G.R., An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In C. Fonseca et al., editors, *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, IEEE Press, 2000, 104-111.
- [13] Raidl G. R., Drexel C., A Predecessor Coding in an Evolutionary Algorithm for the Capacitated Minimum Spanning Tree Problem, in *Late-Breaking-Papers Proc. of the 2000 Genetic and Evolutionary Computation Conference*, Las Vegas, NV, July 2000, 309-316.
- [14] Raidl G. R., Julstrom B. A., Edge-sets: An effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, 7(3): 2003, 225-239.
- [15] Raidl G.R., Julstrom B. A., Greedy Heuristics and an Evolutionary Algorithm for the Bounded-Diameter Minimum Spanning Tree Problem, in *Proc. of the 2003 ACM Symposium on Applied Computing*, March 2000, 747-752.
- [16] Rothlauf F, *Representations for Genetic and Evolutionary Algorithms*, Studies in Fuzziness and Soft Computing. Physica, Heidelberg, 2002.
- [17] Sinclair, M.C., *Evolutionary Telecommunications: A Summary*, Proc. GECCO'99 Workshop on Evolutionary Telecommunications: Past, Present and Future, Orlando, Florida, USA, July 1999, 209-212.
- [18] Xu W.: On the quadratic minimum spanning tree problem. *Proceedings of 1995 Japan-China International Workshop on Information Systems*, eds. M. Gen and W. Xu, Ashikaga, 1995, 141-148.
- [19] Zhou G., Gen M., An effective genetic algorithm approach to the quadratic minimum spanning tree problem, *F. J. Computers Ops Res.*, Vol. 25 (3), 1998, 229-237.