Traffic & Transportation

Cost Optimisation Tool for Multicommodity Network Flow Problem in Telecommunications

Snežana MLADENOVIĆ[1], Ivana STEFANOVIĆ[2], Slađana JANKOVIĆ[3], Ana UZELAC[4], Goran MARKOVIĆ[5], Stefan ZDRAVKOVIĆ[6]

Review
Submitted: 16 Dec 2023
Accepted: 25 Mar. 2024

[1] snezanam@sf.bg.ac.rs, University of Belgrade, Faculty of Transport and Traffic Engineering

[2] Corresponding author, ivanas@viser.edu.rs, Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering

[3] s.jankovic@sf.bg.ac.rs, University of Belgrade, Faculty of Transport and Traffic Engineering

[4] ana.uzelac@sf.bg.ac.rs, University of Belgrade, Faculty of Transport and Traffic Engineering

[5] g.markovic@sf.bg.ac.rs, University of Belgrade, Faculty of Transport and Traffic Engineering

[6] s.zdravkovic@sf.bg.ac.rs, University of Belgrade, Faculty of Transport and Traffic Engineering

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

In this paper, we consider the problem of minimising the cost of data transmission as a function of the capacity of telecommunication links. To solve this problem, we first formulated a mathematical model, and then we designed and developed a software that enables the optimisation of the given or randomly generated telecommunications network. Declarative programming is a good choice for optimisation problems because it is enough to specify only the relations that must be satisfied, without giving any effective procedure for finding the values for the decision variables. To test the application, we developed a software that randomly generates a telecommunications network that meets the given requirements. This enables us to test the application on an arbitrary number of different telecommunication networks with different numbers of nodes and links, and analyse the impact of changing network parameters on the flow and results of the optimisation. As telecommunications networks operate in conditions of uncertainty, the subject of special analysis was the potential failure of some of the network links. The paper presents and thoroughly analyses the optimisation results for several selected networks, as well as summary results for a number of telecommunications networks.

KEYWORDS

data transmission cost optimisation; capacity of telecommunication links; network flow; link failure; declarative programming.

1. INTRODUCTION

The amount of data exchanged between users follows the growth in the number of users. Users demand the service to be available anytime and anywhere, with fast, high-quality and reliable data transmission at the lowest possible price. On the other hand, data transfer often occurs in conditions of uncertainty, such as unavailability or limited availability of some resources of the telecommunications network. Therefore, it is necessary to optimise telecommunication networks to make the best possible use of available resources. Optimisation of telecommunications networks most often means minimising the cost of transmission, delay and losses, that is, maximising network bandwidth, spectral and energy efficiency etc. This problem is challenging for several reasons. One of these is the mobility of users as well as the dynamic, changeable, uncertain environment in which communication occurs. Another, significant reason that complicates the practical implementation of optimisation is the large number of parameters that need to be considered.

In this paper, we discuss the problem of minimising the cost of data transmission as a function of the capacity of telecommunication links, where data exchange between nodes in the network is one-way. The optimisation problem is reduced to determining the optimal values of the loading of telecommunications links for which the total cost of data transmission is the lowest. This optimisation problem has been widely discussed in the literature. However, we did not find an adequate educational tool that would offer useful "what if" analyses in the processes of planning, designing and recovering telecommunication networks with a given optimisation criterion. For this reason, we have implemented the original *TESTGEN* software module for generating test networks. One of the important contributions of the research is link failure modelling and optimisation under such conditions.

The rest of the paper is organised in the following way: in the second section, a diagram flow of the research is presented. The third section presents a brief literature review of the considered optimisation problem. In the fourth section, we modelled the described optimisation problem as a linear programming task. The fifth section presents an optimisation application which was implemented in the declarative paradigm. In the sixth section, the most significant research results are presented and analysed. At the end of the paper, conclusions are provided and the direction of improvement of the optimisation application is pointed out.

2. RESEARCH DESIGN

The flow diagram in *Figure 1* provides an illustration of the methodology and steps in this research. A literature review is provided following the definition of the research problem. Our hypothesis is that we can create software that is able to find the optimal solution if it exists, that is, identify conflicts and offer relaxations otherwise. After the mathematical modelling of the optimisation problem, the new optimisation tool, *TESTGEN* software, was created and tested with a wide range of various telecommunications networks. The results of the research were then gathered and analysed.

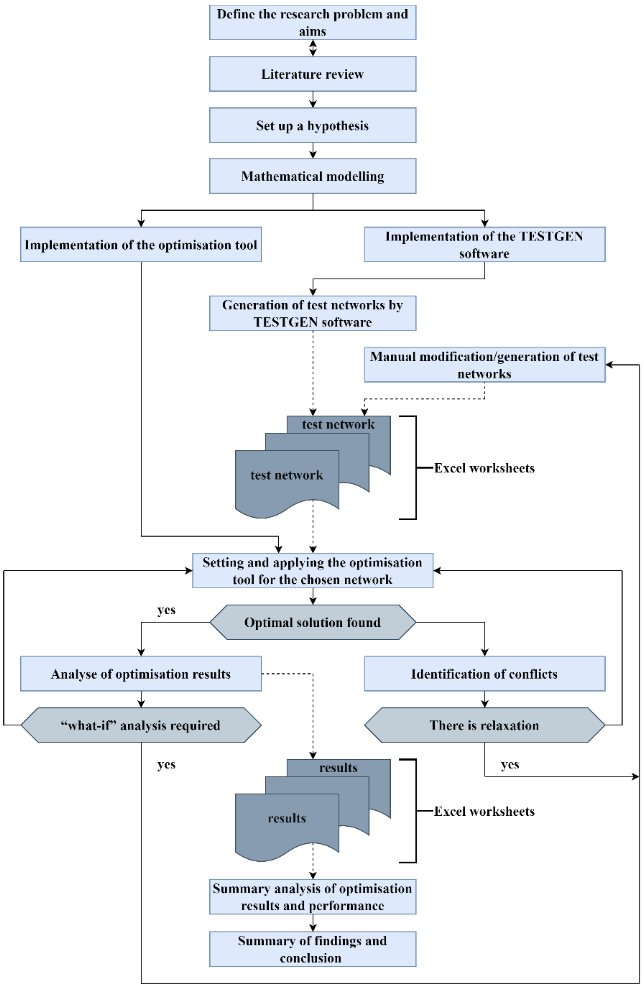
Define the research problem and aims
Literature review
Set up a hypothesis
Mathematical modelling
Implementation of the optimisation tool
Implementation of the TESTGEN software
Generation of test networks by TESTGEN software
Manual modification/generation of test networks
test network
test network
Excel worksheets
Setting and applying the optimisation tool for the chosen network
yes
Optimal solution found
Analyse of optimisation results
Identification of conflicts
"what-if" analysis required
There is relaxation
yes
yes
results
results
Excel worksheets
Summary analysis of optimisation results and performance
Summary of findings and conclusion

Figure 1 – Diagram flow of the research

3. LITERATURE REVIEW

A large number of research papers dealing with optimisation in telecommunications are available. By applying declarative programming as a convenient approach, numerous software solutions for the mentioned problems have been developed. The most commonly used tools for optimising telecommunications networks are *IBM ILOG CPLEX Optimization Studio*, *LP* (*Linear Programming*) *solver* and *MATLAB*.

Many researchers deal with the problem of minimising the cost of data transmission, which is also the subject of this work. The problem of minimising the cost of data transmission is an integral part of various textbooks and manuals on optimisation in telecommunications [1 – 4]. The abbreviation MCNF (*Multi-Commodity Network Flows*) is used to describe the considered problem. According to Salimifard and Bigharaz [5], the MCNF approach is used to solve complicated problems in complex systems, most often in telecommunications and transport. In their paper, an analysis and a classification of 263 papers based on the MCNF approach and published from 2000 to 2019 were given. The MCNF approach is frequently used in telecommunications for traffic routing, i.e. in determining the optimal route for minimal data transmission cost, as well as in network planning. Eriskin [6] used this approach to determine optimal link capacity values to satisfy the demands of all users. The MCNF does not lose its importance even today, as recently published papers also use this approach [7, 8].

The *NETGEN* software is used in a large number of analysed papers. It was created specifically for the MCNF problem and is used to generate telecommunications networks. The *NETGEN* software can generate a network with several thousand nodes and links and there is always a solution to the MCNF problem for that network. It is important to highlight that *NETGEN* is used exclusively for the generation of telecommunications networks, while *IBM ILOG CPLEX Optimization Studio* and *LP* solver are mostly used for their optimisation. Kovács [9] also used the *NETGEN* software to generate telecommunication networks. In the same paper, fifteen different solvers for minimum-cost flow problems were evaluated, most of them written in *C* and *C++*, while one is a component of the *IBM ILOG CPLEX Optimization Studio* package.

In a recently published paper, Chaturvedi et al. [10] aimed to maximise throughput in the case where users' demands were larger than the edge capacities. To solve the problem, an approach similar to the one applied in our research was used: modelling the telecommunications network using a directed graph and linear programming. Dilpriya et al. [11] presented a strategy for improving the public switched telephone network of Sri Lanka using graph theory and linear programming. The goal was to maximise the average data transfer rate, taking into account the cost, delay, throughput and the number of base stations. The *PHP Simplex* software was used for optimisation.

Antić [12] proposed a new routing strategy to maximise the throughput or minimise the cost of the network while using shortest path routing and traffic balancing. To solve the optimisation problem, both linear programming and the *LP solver* optimisation tool were used. The problem was analysed on regular and real network topologies. In the case of real network topology, networks of different dimensions were analysed, where the smallest network was 79x294 and the largest one was 315x1944. Copula theory and fuzzy programming have been used to demonstrate effectiveness in solving large-scale network problems by Khezri and Khodayifar [13]. The paper considered on MCNF problem with several types of costs, like the transition cost, the transition time, the delay time associated to each arc are taken into account.

4. MATHEMATICAL MODEL

Graph theory is very often used for modelling and optimisation of telecommunications networks. The graph $G = (N, E)$ is an ordered pair consisting of a set of nodes (also called vertices or points) N and a set of edges (also called branches, arcs, links or lines) E. Nodes in a telecommunications network model can represent base stations, computers, servers, routers, switching centres, sensors, various smart devices such as phones, tablets, watches, IoT devices etc. Depending on the optimisation problem itself, edges can represent the distance between transmitter and receiver, link attenuation, transmitter power, data rate, data transmission cost etc.

Xu and Vanier [14] addressed the problem of energy efficiency in wireless sensor networks in a recently published paper. In this paper, an approach similar to the one applied in this research was used, where in directed graph modelling a wireless sensor network is randomly generated using the *CPLEX* optimisation package. Fowler et al. [15] considered the problem of minimising the delay and maximising the total amount of power for 5G device-to-device communication with fifth-generation mobile communications, using linear and nonlinear programming and graph theory. Ali Raayatpanah et al. [16] also discussed the optimisation of wireless telecommunications networks where the network is modelled by a graph whose nodes are mobile switching centres and whose edges are links between them. The goal was to minimise the cost of the bandwidth assignment.

In this paper, the modelling of telecommunications networks is done using directed graphs due to one-way data transmission between network nodes. Let us consider a network with m nodes, $N=\{N_1, N_2, \ldots, N_m\}$. Each node N_i has its node flow b_i, $i=1, ..., m$. Similarly, if the network has n edges, a set of edges is $E=\{e_1, e_2, ..., e_n\}$, where e_j, $j=1, ..., n$, is an ordered quintuple of the following form $(l_j, u_j, c_j, s_j, d_j)$, where:

l_j – link capacity lower bound e_j,
u_j– link capacity upper bound e_j,
c_j– data transfer cost along the edge e_j,
s_j– source node of the edge e_j,
d_j– destination node of the edge e_j.

The graph is uniquely joined by an incidence matrix, let us call it A. The matrix is of dimensions mxn, and it defines the network topology. The elements of A are defined by *Equation 1*:

$$a_{ij} = \begin{cases} 1, \text{if edge } e_j \text{ comes out of node } n_i \\ \text{-}1, \text{if edge } e_j \text{ enters node } n_i \\ 0, \text{if edge } e_j \text{ and } n_i \text{ are not incidental} \end{cases} \tag{1}$$

where $i=1,2,...,m$, $j=1,2,\ldots,n$.

It is clear that the rows corresponding to the nodes that exclusively send data do not contain any -1, i.e. rows corresponding to nodes that exclusively receive data do not contain any 1 element. In each column of the matrix, there are exactly one elements 1 and -1, which define the source and destination nodes of the edge corresponding to that column.

Let S be the set of all nodes from which at least one edge starts, and let D be the set of all nodes in which at least one edge ends. The mathematical formulation of the problem of minimising the data transfer cost in the function of the capacity of telecommunications links as a task of integer linear programming can be presented as follows:

$$\min F(x) = \sum_{j=1}^{n} c_j x_j \tag{2}$$

Subject to:

$$l_j \leq x_j \leq u_j, \text{for } j = 1,\ldots,n \tag{3}$$

$$\sum_{i \mid i \in S \wedge j \in D} x_j - \sum_{j \mid j \in S \wedge i \in D} x_j = b_i, \text{for } i = 1,\ldots,m \tag{4}$$

$$\sum_{i=1}^{m} b_i = 0 \tag{5}$$

$$s_j \neq d_j, \text{for } j = 1,\ldots,n \tag{6}$$

where $F(x)$ is the objective function, while x_j represents decision variables (load of edges) that should be determined.

The constraint given by *Equation 3* refers to link capacity constraints. The constraint in *Equation 4* is related to flow conservation equations. The flow of a node is therefore equal to the difference between the received and generated traffic in that node and can be positive or negative. The next constraint, *Equation 5*, is related to network traffic balancing. According to this limitation, the sum of received and generated traffic in all network nodes is equal to 0. The constraint given by *Equation 6* prohibits loops in the network.

5. DEVELOPMENT OF OPTIMISATION APPLICATION

For the development of the optimisation application, both the optimisation package *IBM ILOG CPLEX Optimization Studio* and the declarative programming language *OPL* (*Optimization programming language*) were used [17]. Special attention was paid to testing, i.e. to automatic generation of test networks, for which the original *TESTGEN* software was developed.

5.1 Implementation of optimisation model

The formulated problem of minimising the cost of data transmission as a function of the capacity of telecommunications links belongs to the shortest path problem class, the one for which there are several algorithms. Sifaleras [18] described some of these algorithms.

In this research, declarative programming was used to solve the formulated optimisation problem. In Section 3, it has already been shown that declarative programming is a good choice for this class of problems since it specifies relations that must be satisfied, without specifying the effective procedure itself. Also, *IBM ILOG CPLEX Optimization Studio* is the most frequently used tool for solving this class of problems in telecommunications, as well as in logistics, railway and road traffic [19–22]. The tool has two solvers: *CPLEX* for mathematical programming and *CP* for constraint programming. Its integrated development environment (IDE) enables the creation and modification of models using the *OPL* declarative language, the execution of models with the help of existing solvers, as well as combination and management of models using the *OPL Script* procedural language. By setting certain options in the environment, the researcher significantly determines the flow and outcome of the execution of the optimisation application. The researcher can choose one of the offered algorithms, limit the number of iterations, set lower and upper objective value limits, conflict resolution method etc. A suitable setting can significantly reduce the search space and thus speed up the process of reaching the optimal solution. Moreover, if the number of iterations is limited, and the optimal solution is not found, we can stop at the current best feasible solution. This is a very important feature in cases when the optimisation is actually reoptimisation, i.e. finding a "good enough" solution as a response to the resulting disturbances. The environment, furthermore, enables a display of obtained solutions in different formats, supporting external data sources/destinations (*Excel*, *Access*), variables monitoring, as well as tracking time and memory used for solving problems. If it is not possible to find an optimal solution, the software finds conflicts and offers solutions to modify the constraints that lead to the conflict. A free version of the software is available on the IBM ILOG website. This version has limitations and can only be used with up to 1000 variables and 1000 constraints.

The coding of the mathematical model, i.e. declaring decision variables, defining the objective function and setting constraints is short and concise in *OPL*. For example, the constraint given by *Equation 4*, which refers to the capacity limits of telecommunications links in *OPL*, has the following form:

```
forall(e in Edges) e.l <= x[e] <= e.u;
```

Input-output communication was realised through *Excel* files. The telecommunications network is defined by loading network parameters from *Excel* worksheets, which includes quintuple, (l_j, u_j, c_j, s_j, d_j), for the set of edges and node flow, b_i, for each node. The optimisation results are also stored in an *Excel* workbook for tabular and graphical presentation. As the set problem belongs to linear programming, *CPLEX* solver was used to execute the model.

5.2 Automatic generation of test networks

It has already been mentioned that in a large number of papers that deal with the same or similar optimisation problem, the *NETGEN* software was used for the generation of test telecommunications networks. In this research, the authors decided to develop their own software module for the purposes of automatic generation of test cases (telecommunications networks) with given properties, called *TESTGEN*. This scenario of test case creation is often used in the software validation and verification processes.

As the model at this stage of development was created primarily with the idea of being used as an educational tool, there are several reasons for this approach to testing. The first is that the presence of the source code gives us the ability to modify it independently. Furthermore, this scenario avoids the import of data from other software and the possible need for conversion into a suitable format supported by the optimisation tool. Moreover, the usage of ready-made software for the generation of telecommunications networks often implies the modification of the obtained data for the purposes of optimisation, and the effort to implement such modifications can be significant. In an automatically generated network, the analyst can easily make "small" changes that enable very useful "what if" analyses in the educational process. *NETGEN* is a classic generator whose first version was developed by Klingman et al. [9] back in 1974. It produces random instances of the MCNF as well as of other network optimisation problems. Different *NETGEN* families define different relations between the number of nodes and the number of edges. In contrast, our *TESTGEN* generates test networks only for MCNF problems, with a mutually independent number of nodes and the number of edges.

The *NETGEN* software generates a network that always has a solution to the MCNF problem. In contrast, *TESTGEN* can generate telecommunications networks without an optimal solution. Experiments have shown that about 35% of the generated telecommunications networks do not have an optimal solution because the constraints are in conflict. In that case, the software identifies the constraints that are in conflict and offers a relaxation of them in order to find an optimal solution. Since the primary purpose of *TESTGEN* software is educational, the generation of networks without an optimal solution, the identification of conflicts and their elimination are essential for a better understanding of the MCNF problem. An example of a network for which an optimal solution was not found, because it is not possible to fulfil the constraint given by *Equation 4,* is analysed in detail in subsection 6.2.

The *TESTGEN* software is also developed in *OPL*, as well as the optimisation software itself. The global algorithm of generation particular network is shown by *Algorithm 1*. The function ***RandomInt***(a,b) for integers a and b returns a random number on the interval [a,b]. The procedure ***FillSet***(P,t,Q), based on the input matrix P and the integer t, determines the set Q with the ordinal numbers of those rows of the matrix P that contain only the values 0 and t.

Algorithm 1 – Global algorithm of TESTGEN software

```
procedure TESTGEN
inputs    minNN, maxNN, minEN, maxEN, maxEC, maxTP, maxNF
output    Edges, NodeFlow
begin
  nodesNumber ⟵ RandomInt(minNN, maxNN)
  edgesNumber ⟵ RandomInt(minEN, maxEN)
  forall (j | 1≤ j ≤ edgesNumber)
    Edges[j].l ⟵ 0
    Edges[j].u ⟵ RandomInt(1, maxEC)
    Edges[j].c ⟵ RandomInt(1, maxTP)
  end forall
  forall ((i,j) | 1 ≤ i ≤ nodesNumber ∧ 1 ≤ j ≤ edgesNumber)
    A[i,j] ⟵ 0
  end forall
  forall (j | 1 ≤ j ≤  edgesNumber)
    sn ⟵ RandomInt(1, nodesNumber)
    A[sn,j] ⟵ 1; Edges[j].s ⟵ sn
    repeat
      dn ⟵ RandomInt(1, nodesNumber)
    until sn ≠ dn
    $A[dn,j] \longleftarrow -1$; $Edges[j].d \longleftarrow dn$
  end forall
  FillSet($A$, 1, $NodesExclusivelySend$)
  FillSet($A$, -1, $NodesExclusivelyReceive$)
  $Sum \longleftarrow 0$
  forall ($i \mid i \in NodesExclusivelySend$)
    $NodeFlow[i] \longleftarrow$ RandomInt(1, $maxNF$)
    $Sum \longleftarrow Sum + NodeFlow[i]$
  end forall
  forall ($i \mid i \in NodesExclusivelyReceve$)
    $NodeFlow[i] \longleftarrow$ -RandomInt(1, $maxNF$)
    $Sum \longleftarrow Sum + NodeFlow[i]$
  end forall
  forall ($i \mid (1 \leq i \leq nodesNumber) \wedge i \notin (NodesExclusivelyReceve \cup NodesExclusivelySend)$)
    $NodeFlow[i] \longleftarrow$ RandomInt(-$maxNF$, $maxNF$)
    $Sum \longleftarrow Sum + NodeFlow[i]$
  end forall
  $bn \longleftarrow$ RandomInt($1, nodesNumber$)
  $NodeFlow[bn] \longleftarrow NodeFlow[bn]-Sum$
  EnterInExcelWorksheet ($Edges$)
  EnterInExcelWorksheet ($NodeFlow$)
end procedure
```

As can be seen from *Algorithm 1*, the parameters *minNN*, *maxNN*, *minEN*, *maxEN*, *maxEC*, *maxTP* and *maxNF* are input, set by the user, and *TESTGEN* procedure writes the output parameters in the *Edges* and *NodeFlow* worksheets. The examples of the *Edges* and *NodeFlow* worksheets are shown in *Table 1* and *Table 2*, respectively.

Table 1 – An overview of the Edges worksheet

Edge	Link capacity lower bound (*l*)	Link capacity upper bound (*u*)	Data transfer cost (*c*)	Source node (*s*)	Destination node (*d*)
e_1	0	5	3	2	1
e_2	0	15	2	2	3
e_3	0	12	9	2	3
e_4	0	9	2	3	4
e_5	0	5	3	5	4
e_6	0	7	2	5	1
e_7	0	5	3	6	3
e_8	0	15	9	6	4
e_9	0	10	2	6	3
e_{10}	0	7	8	6	1

Table 2 – An overview of the NodeFlow worksheet

Node	Node Flow (*b*)
N_1	-2
N_2	12
N_3	-16
N_4	-9
N_5	3
N_6	12

From the above, it is clear that in the telecommunications network generated in this way, the number of nodes and edges, sources and destinations of edges, the upper limit of the capacity of edges, the flow of nodes and data transfer prices are random values in given intervals. This is why 35% of the generated telecommunications networks do not have any solutions because it is not always possible to satisfy the constraints given by *Equation 4* and/or *Equation 5* in networks generated in this way. The network can also be set manually by entering appropriate values in corresponding *Excel* worksheets or by modifying the values previously entered in them by the *TESTGEN* software.

6. RESULTS

In this section, we try to prove the hypothesis that the software is able to find an optimal solution if it exists, that is, identify conflicts and offer relaxations in order to reach a solution. We first analysed optimisation results for one selected network where an optimal solution was found, and then for one where the optimal solution was not found. In the third subsection of this section, the summarised results of the optimisation of several telecommunications networks are presented. Let all networks satisfy the following: *minNN*=5, *maxNN*=10, *minEN*=10, *maxEN*=25, *maxTP*=10 and *maxNF*=20. All examples are performed on a computer: CPU: Intel(R) Core (TM) i5-1035G7 1.2 GHz 4 cores, 8GB RAM on Windows 11 Pro 64-bit OS with an x64-based processor.

6.1 Analysis of a network where optimal solution was found

Example 1. Consider the 6x10 network shown in *Figure 2*. Nodes with the positive flow are marked with green colour, and for the negative flow red colour is used. The lower link capacity, upper link capacity, data transfer cost, the source and the destination node are noted in parentheses next to each edge. *Table 1* and *Table 2* show the input parameters for the optimisation of this network, which were previously entered in the *Edges* and *NodeFlow* worksheets.

From *Figure 2* it can be seen that there are several edges with the same source and destination nodes, such as edges e_2 and e_3. Edges e_2 and e_3 have different upper capacity limits and prices. In telecommunications networks, the data between nodes can be transmitted via different links. For example, a Wi-Fi link and LTE (*Long-Term Evolution*) link can be used for data transmission from node N_2 to node N_3. Which link will be used for transmission depends on link availability, link capacity, transmission price etc. Backup links are often used as a means of increasing the certainty of data transmission in telecommunications networks. This is why the constraint for the number of edges that can have the same source and destination nodes is not included in the model.

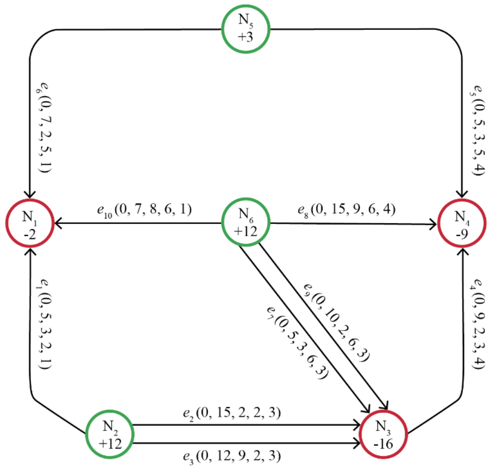

Figure 2 – The telecommunications network for which optimisation is done in Example 1

The optimisation software finds and displays the optimal solution [0 12 0 8 1 2 2 0 10 0]. The minimum price of data transfer is 73, whereby the edges used for data transfer are e_2, e_4, e_5, e_6, e_7 and e_9, whose specific optimal link loads are 12, 8, 1, 2, 2 and 10, respectively.

As telecommunications networks are complex, uncertain systems, for various reasons (defective equipment, cable breaks, weather etc.), some of the telecommunications links may become unavailable for data transmission during a certain period. According to Bugarčić et al. [23], the process of choosing the optimal route from source to destination node is a challenging task due to frequent link breaks and performance degradation. A number of papers take into account the complex system and uncertain environment, the recognition of failures and disruptions in the operation of telecommunications systems in such circumstances, and the response to them. According to Aktaş et al. [24], mobile operators invest significant funds in tools for network monitoring and optimisation, in order to perform network reconfiguration in the shortest possible time in case of unavailability of telecommunications links. Many models have been developed for the prediction of telecommunications link failures [24–25], as well as for network recovery after their detection [26–27]. The prediction of telecommunications link failures given by Aktaş et al. [24] was based on key network parameters and weather data. Patel and Pathak [25] presented a model for time estimation of connection interruption based on the signal strength of received packets. Moshiri et al. [26] presented a network recovery strategy that involves finding a set of spare E^R links to be used after a set of E^A network links becomes unavailable for data transmission. Bakhshi Kiadehi et al. [27] used declarative programming to find the minimum cost of data transmission in SDN (software-defined network) networks, taking into account the cost of data transmission, link utilisation and link delay between source and destination nodes. To model link failure cases, the input data was supplemented with the indicator related to the availability of each telecommunications link during the optimisation.

Further testing went in the direction of making individual edges unavailable. *Table 3* provides an overview of the minimum data transfer price, the percentage increase of minimum transfer price concerning the case when all links are available and the obtained values of decision variables. The total CPU time of code execution, the percentage of CPU time spent on data loading from the total execution time of the code and the percentage of memory used for the optimisation procedure from the total memory engaged for code execution, depending on the unavailability of telecommunications links, are also given in *Table 3.* Because of the network's dimensions, the amount of data that is loaded from *Excel* is on the order of KB.

Table 3 – An overview of the optimisation results and performance in function of unavailability of individual telecommunications links

Unavailable link	Min cost	Cost increase (%)	Optimal solution	Total time of code execution (s)	Time of data loading (%)	Amount of memory spent on optimisation (%)
e_1	73	0.00	[0 12 0 8 1 2 2 0 10 0]	1.832	88	35
e_2	157	115.07	[0 0 12 8 1 2 2 0 10 0]	2.470	89	36
e_3	73	0.00	[0 12 0 8 1 2 2 0 10 0]	1.791	86	36
e_4	111	52.05	[0 12 0 0 3 0 0 6 4 2]	1.557	88	30
e_5	79	8.22	[1 13 0 9 0 3 2 0 10 0]	1.607	88	35
e_6	81	10.96	[0 12 0 6 3 0 0 0 10 2]	2.536	92	40
e_7	81	10.96	[0 12 0 6 3 0 0 0 10 2]	2.444	92	41
e_8	73	0.00	[0 12 0 8 1 2 2 0 10 0]	1.923	87	36
e_9	111	52.05	[0 12 0 1 3 0 5 5 0 2]	2.457	90	40
e_{10}	73	0.00	[0 12 0 8 1 2 2 0 10 0]	2.267	88	37

From *Table 3*, we can see that the availability of edges e_1, e_3, e_8 and e_{10} does not affect the minimum cost, since the mentioned edges are not used for data transmission when all links are available. Moreover, the total code execution time when turning off edges e_1, e_3, e_8 and e_{10} is shorter compared to the code execution time when turning off other edges. It should be emphasised that when disconnecting individual edges of the network in this example, the optimal solution was always found. In doing so, each edge of the network is used for data transmission in at least one of the cases shown by the rows in *Table 3*, so this is an example of a well-designed network. An increase of the total cost of data transmission is visible: as much as 115.07%, in the case of edge e_2 not being available. The increase in the total price is a consequence of the high price of data transmission in edge e_3, which is at the same time maximally loaded. The disconnection of edges e_4 and e_9 was followed by a significant price increase by 52.05%, which was primarily caused by the high price of data transmission through edge e_8. Another factor that slightly affects the price increase is the price of data transmission through edge e_{10}, but the load on this edge is of a smaller volume.

The problem is more complex if several links are not simultaneously available for data transmission. In the following tests, from the set of edges e_7, e_8, e_9 and e_{10} we "excluded" 2 by 2 edges from the traffic. *Table 4* provides an overview of these experiments.

Table 4 – An overview of the optimisation results and performance in function of unavailability of 2 selected telecommunications links

Unavailable link	Min cost	Cost increase (%)	Optimal solution	Total time of code execution (s)	Time of data loading (%)	Amount of memory spent on optimisation (%)
e_7, e_8	81	10.96	[0 12 0 6 3 0 0 0 10 2]	1.802	89	36
e_7, e_9	162	121.92	[4 15 1 0 3 0 0 6 0 6]	1.809	90	37
e_7, e_{10}	81	10.96	[0 12 0 6 1 2 0 2 10 0]	2.786	90	39
e_8, e_9	155	112.33	[5 15 2 6 3 0 5 0 0 7]	1.825	87	34
e_8, e_{10}	73	0.00	[0 12 0 8 1 2 2 0 10 0]	1.720	89	35
e_9, e_{10}	111	52.05	[0 12 0 1 1 2 5 7 0 0]	1.792	89	37

In the case of unavailability of these pairs of edges, we can conclude the following:

- e_7 and e_8 – we get the same optimal solution as when edges e_6 or e_7 are individually turned off;
- e_7 and e_9 – a new optimal solution with the largest price increase, 121.92%;
- e_7 and e_{10} – the same value of the objective function as when e_6 or e_7 are switched off individually, or e_7 and e_8 are simultaneously switched off, but the values of the decision variables are different;
- e_8 and e_9 – a new optimal solution with a price increase of 112.33%;
- e_8 and e_{10} – do not affect the optimal solution, the minimum price remains 73;
- e_9 and e_{10} – a price increase of 52.05%, the same as when e_4 or e_9 are turned off individually, but the values of the decision variables are different.

It is interesting to note that several feasible solutions give the same value of the objective function, e.g. solutions [0 12 0 6 3 0 0 0 10 2] and [0 12 0 6 1 2 0 2 10 0] correspond to the value of the objective function 81; solutions [0 12 0 0 3 0 0 6 4 2], [0 12 0 1 3 0 5 5 0 2] and [0 12 0 1 1 2 5 7 0 0] correspond to the value of the objective function 111. Also, we can see from *Table 3* and *Table 4* that the individual unavailability of edges e_8 and e_{10}, as well as the unavailability of both of them simultaneously, does not lead to an increase in price. However, that does not mean that the mentioned edges are redundant. From *Table 3*, we can note that edge e_8 is used for data transfer when edges e_4 and e_9 are unavailable, while in the case of edges e_4, e_6, e_7 and e_9 being unavailable, edge e_{10} is used. For example, in the case of unavailable edge e_4, there is no feasible solution without using edge e_8. This kind of "what-if" analysis makes it possible to find the weaknesses and weak points of the network and represents the greatest convenience offered by the *TESTGEN* software.

6.2 Analysis of network where no optimal solution was found

Example 2. *Figure 3* shows a network without a solution due to the conflict between the flow limit of the node N_1 and the upper capacity limits of edges e_1 and e_7. Nodes and edges are represented using the same notation as in Example 1. For orange nodes, the sum of received and generated traffic is equal to 0. The outgoing edges e_1 and e_7, from node N_1, are marked red. The software finds conflicts and offers solutions to modify the constraints that lead to the conflict. In this case, the conflict can be resolved by increasing the upper capacity limit of edge e_1 and/or edge e_7.

By increasing the upper capacity limit of edge e_7 from 5 to 10, the optimisation tool determines the minimum price of data transfer, which is 74, while edges e_1, e_6, e_7 and e_{11} are used for data transfer, with optimal link load values being 5, 5, 10 and 12, respectively. If it is not possible to change the upper capacity limit of link e_7, while it is possible to increase the upper capacity limit of link e_1 to 10, the newly obtained minimum price of data transmission would be 104, which corresponds to a price increase of 40.54%. It has been observed that the unavailability of links e_7 and e_1 can significantly affect the cost of data transmission.

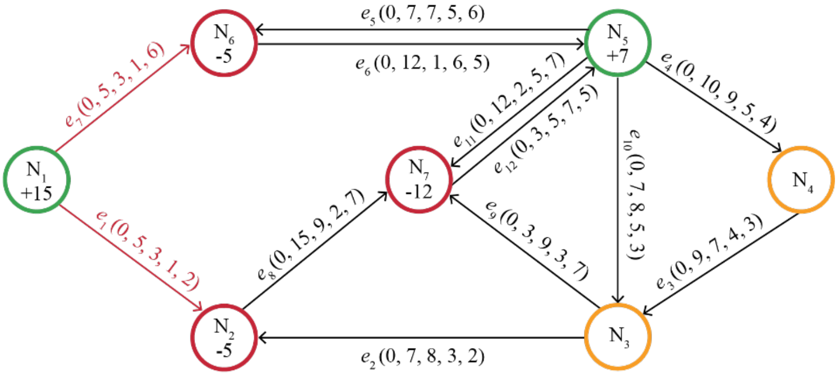

Figure 3 – The telecommunications network without feasible solutions

For this reason, the upper capacity limit of edges e_1 and e_7 was increased to 15 and the optimisation of the telecommunications network was carried out under those conditions. When link e_7 is unavailable for data transmission, a price increase of 135.13% occurs. The sudden increase in the price of data transmission is caused by the high cost of data transmission on link e_8. The cost of data transmission on link e_8 also contributed to the 40.54% price increase discussed earlier in this example. At the same time, e_8 is the only edge that enables data transmission from node N_2 to node N_7. This information is of great importance during network design. An additional link that would enable data transmission from node N_2 to node N_7, at a lower cost of transmission, would certainly result in a solution with a lower optimal cost of data transmission.

6.3 Summary analysis

During the experiments with the *TESTGEN* software, a large number of telecommunications networks were generated, whereby the optimisation software was able to find the optimal solution for approximately 65% of them. *Table 5* shows the summary results for 20 selected, randomly generated networks for which the optimal solution was found.

It is interesting to note that in case there are no conflicting constraints, the optimisation was executed very quickly. The average time spent to reach the optimal solution is 0.021s, i.e. 5.73% of the total CPU time was spent on code execution. On the other hand, the average memory used to reach the optimal solution is 2.497 MB, which is 84.65% of the total memory space engaged during the execution of the optimisation program.

As already shown, it is possible to generate a telecommunication network where a large number of edges have the same source and destination node. In 55% of randomly generated telecommunication networks, at least two edges with the same source and destination nodes appear two or three times. In 15% of the generated networks there are no edges with the same source and destination nodes. The maximum number of edges with the same source and destination node is 3, and this number of edges appears only in 5% of the generated telecommunication networks – only in one network from *Table 5*.

The average cost of data transfer is 111.15, where the absolute minimum cost of data transfer is 16, for the network 2, and the absolute maximum is 269, for the network 8. It is interesting to note that the networks with the minimum and the maximum transfer costs have similar dimensions, 7x16 and 7x15, respectively. The price of data transmission does not depend on network dimensions, but on the volume of network traffic, that is, the flow of nodes, the capacity of links, as well as the connections between nodes and links, and the price of data transmission by individual links.

In the following figures, network links are on the abscissa. *Figure 4* shows the obtained loading and the data transfer cost which correspond to the optimal solution for each link of network 2. For the network 8, these data are shown in *Figure 5*.

From *Figure 4*, we can see that only three edges, e_{10}, e_{14} and e_{16}, are used for data transmission in the network 2, and all of them are equally loaded. Unlike the network 2, the traffic volume increased in the network 8, *Figure 5*. Edges e_1, e_3, e_4, e_7, e_9, e_{11} and e_{14} are used for data transmission. The largest amount of traffic is transmitted by the link e_3, which also has the highest transmission cost.

Table 5 – Optimisation performance of selected randomly generated telecommunications networks for which an optimal solution was found

No network	Optimal cost	Average link load	Average link upper capacity limit	Time spent to find optimal solution		Memory amount spent on optimisation		Network dimensions (m×n)
				(s)	(%)	(MB)	(%)	
1	96	2.74	10.74	0.014	6.06	2.475	97	6×22
2	16	1.00	12.24	0.045	10.47	2.945	93	7×16
3	218	4.85	14.69	0.076	15.90	2.428	84	10×12
4	51	1.36	12.43	0.030	9.17	3.042	95	7×13
5	211	3.06	9.33	0.035	10.03	3.109	83	5×15
6	78	1.60	10.36	0.037	8.53	2.585	98	10×24
7	81	2.17	10.21	0.006	2.24	1.845	72	7×11
8	269	2.47	12.50	0.015	4.36	2.265	86	7×15
9	108	1.73	8.54	0.019	6.11	2.808	97	10×25
10	33	1.42	9.58	0.012	5.24	1.940	77	6×11
11	79	2.55	9.36	0.011	4.14	2.859	92	5×10
12	77	2.46	9.62	0.008	2.20	1.975	73	9×12
13	36	1.47	9.53	0.004	1.02	2.720	86	9×18
14	100	2.11	9.42	0.015	3.88	2.472	78	6×18
15	180	2.93	10.20	0.014	3.52	2.296	76	10×14
16	204	4.42	12.25	0.013	4.44	2.269	82	8×11
17	87	1.67	10.52	0.014	4.86	2.358	74	7×20
18	48	2.67	9.83	0.015	4.00	2.183	77	8×11
19	44	1.95	9.50	0.017	4.70	2.767	89	8×18
20	207	3.35	7.80	0.016	3.71	2.605	84	7×19

The total cost of data transmission per edge e_3 is 120, which represents as much as 44.61% of the total cost of data transmission. From *Figure 5*, it can be seen that a large amount of data is transmitted by the edges e_7 and e_{14}. The price of data transmission through the edges e_7 and e_{14} is 8·6+7·8=104, which represents 38.66% of the total cost. Using *Figure 4* and *Figure 5*, we can easily calculate the cost of data transmission for individual links as well as the overall cost of data transfer for the observed network.

After analysing the optimisation results, we can confirm the hypothesis and conclude that the *TESTGEN* software is capable of finding an optimal solution, as well as identifying conflicts and providing relaxations to find a solution.

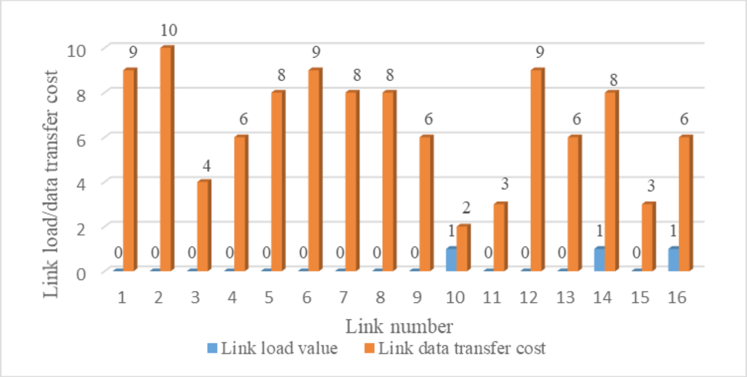

Figure 4 – Link load values in optimal solution and data transfer cost for links of network 2

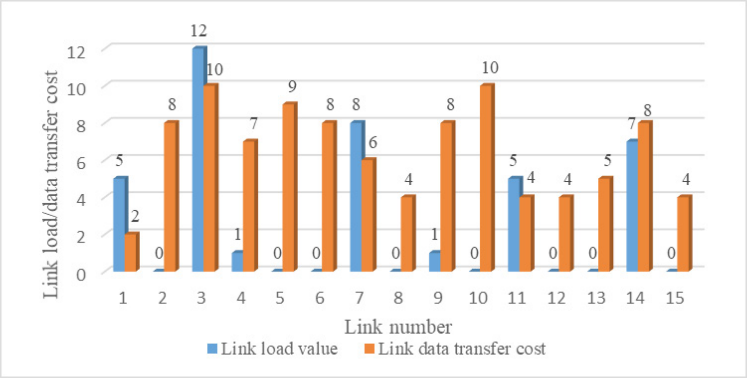

Figure 5 – Link load values in optimal solution and data transfer cost for links of network 8

7. CONCLUSION

In this paper, we first formulated a mathematical model, and then designed and implemented software that enables cost optimisation of a given or randomly generated telecommunications network. The original *TESTGEN* software was developed to generate test telecommunication networks with specific characteristics.

Although there are ready-made generators of telecommunication networks, the authors of this research decided to design and implement their own generator for several reasons. The first reason is that the presence of source code enables independent modification. Furthermore, this scenario avoids importing data from other software and the eventual need to convert it to a format supported by the optimisation tool. Over the automatically generated network, users can easily make “small” changes that enable very useful “what if” analyses. Also, the generation of networks without an optimal solution, identification of conflicts and their elimination are essential parts of the educational process, not only for initial network design and planning but also for network recovery after detected link failures. Validation of the implemented software has proven that it can find an optimal solution if it exists, that is, identify conflicts and offer relaxations to reach a solution, while using acceptable time and memory resources.

The *TESTGEN* software has no limitations on the dimensions of the network it creates. However, with the increase in the number of nodes and links, the number of constraints that must be satisfied also increases, so the optimisation process can be very time-consuming. However, according to Lagos et al. [28], linear programming techniques have been combined with well-known metaheuristics (tabu search, simulated annealing and genetic algorithms) for solving optimisation problems for years. The choice of suitable metaheuristics or heuristics and their synergy with mathematical programming techniques are the first direction to take for future research and improvement of the optimisation application so that it is not only educational but also realistically usable for complex telecommunications networks.

Another direction of improving the optimisation application is the possibility of choosing one of the offered criteria for minimisation (costs, delay and losses) or maximisation (bandwidth, spectral and energy efficiency) in the telecommunications network.

Improving one parameter of network performance sometimes worsens another, for example, increasing energy efficiency can negatively affect delay and cost. For this reason, the third direction of our future work will be upgrading the existing software to multi-criteria optimisation software.

ACKNOWLEDGEMENTS

This research was funded by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia.

REFERENCES

[1] Oki E. *Linear programming and algorithms for communication networks: A practical guide to network design, control, and management.* Broken Sound Parkway NW, USA: Taylor & Francis Group; 2013.

[2] Pioro M, Medhi D. *Routing, flow, and capacity design in communication and computer networks.* San Francisco, USA: Elsevier Inc; 2004.

[3] Medhi D, Ramasamy K. *Network routing algorithms, protocols, and architectures.* San Francisco, USA: Elsevier Inc; 2007.

[4] Resende M, Pardalos P. *Handbook of optimization in telecommunications.* Boston, MA, US: Springer; 2006.

[5] Salimifard K, Bigharaz S. The multicommodity network flow problem: State of the art classification, applications, and solution methods. *Operational Research.* 2022;22:1-47. DOI: 10.1007/s12351-020-00564-8.

[6] Eriskin L. A Lagrangean relaxation-based solution approach for multicommodity network design problem with capacity violations. *Journal of Naval Sciences and Engineering.* 2021;17(2):241-263. https://dergipark.org.tr/en/pub/jnse/issue/65720/932377.

[7] Tsai K, et al. Multi-commodity flow routing for large-scale leo satellite networks using deep reinforcement learning. *Proceedings of the IEEE Wireless Communications and Networking Conference 2022, 10–13 April 2022, Austin, TX, USA*. 2022. p. 626-631. DOI: 10.1109/WCNC51071.2022.9771734.

[8] Max-Onakpoya E, Baker CE. Assessing a synergistic use of alternate broadband delivery models in rural areas. *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) 2022, 21-25 March 2022, Pisa, Italy*. 2022. p. 5-8. DOI: 10.1109/PerComWorkshops53856.2022.9767252.

[9] Kovács P. Minimum-cost flow algorithms: An experimental evaluation. *Optimization Methods and Software*. 2015;30(1):94-127. DOI: 10.1080/10556788.2014.895828.

[10] Chaturvedi A, et al. Improved throughput for all-or-nothing multicommodity flows with arbitrary demands. *ACM SIGMETRICS Performance Evaluation Review*. 2021;49(3):22-27. DOI: 10.1145/3529113.3529121.

[11] Dilpriya TAH, Lanel GHJ, Vidanage BVNC. A strategy to strengthen and enhance the telecommunication network in Sri Lanka by using concepts of graph theory and linear programming models. *International Journal of Natural Sciences Research*. 2022;10(1):1-20. DOI: 10.18488/63.v10i1.2913.

[12] Antić M. *Optimization of non-blocking packet networks using the practical routing protocol with load balancing*. PhD Thesis. University of Belgrade, School of Electrical Engineering; 2014.

[13] Khezri S, Khodayifar S. Joint chance-constrained multi-objective multi-commodity minimum cost network flow problem with copula theory. *Computers & Operations Research*. 2023;156(106260). DOI: 10.1016/j.cor.2023.106260.

[14] Xu L, Haddad Vanier S. Branch-and-price for energy optimization in multi-hop wireless networks. *Networks*. 2022;80(1):123-148. DOI: 10.1002/net.22083.

[15] Fowler S, Li Y, Pollastro A, Napoli S. Simple network design and power allocation for 5g device-to-device communication. *Proceedings of the IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, *1-3 Dec. 2014, Athens, Greece.* 2014. p. 203-207. DOI: 10.1109/CAMAD.2014.7033235.

[16] Raayatpanah MA, et al. Design of survivable wireless backhaul networks with reliability considerations. *Computers & Operations Research*. 2023;151(106120). DOI: 10.1016/j.cor.2022.106120.

[17] IBM. *IBM ILOG CPLEX Optimization studio CPLEX user's manual*. 2017;12(8). https://perso.ensta-paris.fr/~diam/ro/online/cplex/cplex1271_pdfs/usrcplex.pdf [Accessed 28th November 2022].

[18] Sifaleras A. Minimum cost network flow: Problems, algorithms, and software. *Yugoslav Journal of Operation Research*. 2013;23(1):3-17. DOI: 10.2298/YJOR121120001S.

[19] Eshtehadi R, Demir E, Huang Y. Solving the vehicle routing problem with multi-compartment vehicles for city logistics. *Computers & Operations Research.* 2020;115(104859). DOI: 10.1016/j.cor.2019.104859.

[20] Mladenović S, et al. Heuristic based real-time train rescheduling system. *Networks*. 2016;67(1):32-48. DOI: 10.1002/net.21625.

[21] Tran VM, Vu THN. Leveraging CPLEX to solve the vehicle routing problem with time windows. *Proceedings of the 13th International Conference on Knowledge and Systems Engineering (KSE)*, *10-12 Nov. 2021, Bangkok, Thailand*. 2021. p. 1-6. DOI: 10.1109/KSE53942.2021.9648591.

[22] Guo Y, Shahraki AA. Selection of rail station locations on an intercity route regarding maximum users' economic profits. *Promet – Traffic&Transportation.* 2023;35(4):595-606. DOI: 10.7307/ptt.v35i4.241.

[23] Bugarčić P, Jevtić N, Malnar M. Reinforcement learning-based routing protocols in vehicular and flying ad hoc

networks–a literature survey. *Promet – Traffic&Transportation.* 2022;34(6):893-906. DOI: 10.7307/ptt.v34i6.4159.

[24] Aktaş S, Alemdar H, Ergüt S. Towards 5G and beyond radio link diagnosis: Radio link failure prediction by using historical weather, link parameters. *Computers and Electrical Engineering*. 2022;99(107742). DOI: 10.1016/j.compeleceng.2022.107742.

[25] Patel S, Pathak H. A mathematical framework for link failure time estimation in MANETs. *Engineering Science and Technology, an International Journal*. 2022;25(100984). DOI: 10.1016/j.jestch.2021.04.003.

[26] Moshiri M, Safaei F, Samei Z. A novel recovery strategy based on link prediction and hyperbolic geometry of complex networks. *Journal of Complex Networks*. 2021;9(4):cnab007. DOI: 10.1093/comnet/cnab007.

[27] Bakhshi Kiadehi K, Rahmani AM, Sabbagh Molahosseini A. A fault-tolerant architecture for internet-of-things based on software-defined networks. *Telecommunication Systems*. 2021;77:155-169. DOI: 10.1007/s11235-020-00750-1.

[28] Lagos C, at al. Combining tabu search and genetic algorithms to solve the capacitated multicommodity network flow problem. *Studies in Informatics and Control*. 2014;23(3):265-276. DOI: 10.24846/v23i3y201405.

Snežana Mladenović, Ivana Stefanović, Slađana Janković, Ana Uzelac, Goran Marković, Stefan Zdravković

Alat za optimizaciju troškova za problem višestrukog mrežnog protoka u telekomunikacijama

Rezime

U ovom radu razmatran je problem minimizacije cene prenosa podataka u funkciji kapaciteta telekomunikacionih linkova. U cilju rešavanja ovog problema, prvo je formulisan matematički model, a zatim je dizajniran i implementiran softver koji omogućava optimizaciju zadate ili slučajno generisane telekomunikacione mreže. Poznato je da je deklarativno programiranje dobar izbor za optimizacione probleme, jer je dovoljno specificirati relacije koje moraju biti zadovoljene, bez specifikacije efektivne procedure za pronalaženje vrednosti promenljivih odlučivanja. Za potrebe testiranja kreirane aplikacije, razvijen je originalni softver koji slučajno generiše telekomunikacionu mrežu sa zadatim osobinama. Ovo daje mogućnost da se aplikacija testira na proizvoljnom broju različitih telekomunikacionih mreža, sa različitim brojem čvorova i linkova, i analizira uticaj promene parametara mreže na tok i rezultat optimizacije. Kako telekomunikacione mreže funkcionišu u uslovima neizvesnosti, predmet posebne analize su bili potencijalni otkazi nekih od linkova mreže. U radu su predstavljeni i detaljno analizirani rezultati optimizacije za nekoliko odabranih mreža, kao i sumarni rezultati za veći broj telekomukacionih mreža.

Ključne reči

optimizacija cene prenosa podataka; kapacitet telekomunikacionih linkova; protok mreže; otkaz linka; deklarativno programiranje.