

Graph Theory Approach to Transportation Systems Design and Optimization

S. Guze

Gdynia Maritime University, Gdynia, Poland

ABSTRACT: The main aim of the paper is to present graph theory parameters and algorithms as tool to analyze and to optimise transportation systems. To realize these goals the 0-1 knapsack problem solution by SPEA algorithm, methods and procedures for finding the minimal spanning tree in graphs and digraphs, domination parameters problems accurate to analyse the transportation systems are introduced and described. Possibility of application of graph theory algorithms and parameters to analyze exemplary transportation system are shown.

1 INTRODUCTION

Nowadays, researchers have two main problems. On the one hand people are dependent on critical systems such as transportation, electricity, water supply, sewage, ICT. According to the official definition, the critical infrastructure is a term used to describe assets that are essential for the functioning of a society and economy. The following facilities are related to this subject [1]:

- electricity and heating generation, transmission and distribution;
- gas and oil production, transport and distribution;
- telecommunication;
- water supply;
- agriculture, food production and distribution;
- public health (hospitals, ambulances);
- transportation systems (fuel supply, railway network, airports, harbours, inland shipping);
- financial services (banking);
- security services (police, military).

There are several regional critical-infrastructure protection programmes, which main aims are:

- to identify important assets,
- to analyze a risk based on major threat scenarios and the vulnerability of each asset,
- to identify, select and make prioritisation of counter-measures and procedures.

These goals are common for all facilities presented above. It is very important to keep these systems in good conditions [8], [9]. Thus, the risk and reliability analyses is needed to understand the impact of threats and hazards [4], [5], [8]. Unfortunately, these problems are more and more complex, because of existing strong interdependencies both within and between infrastructure systems.

The second problem is finding optimal solutions. It is met in many areas of modern science, technology and economics. For example, the navigator's main aim is optimizing the route of the ship due to safety, time of passage, fuel and costs [16], [21]. The solving of these real-life problems is looking for a mathematical model or function that best approximates or fit to the data collected during the operation process or experiment [11]. If we can specify three elements: a model of the phenomenon of

distinguished decision variables, objective function - also known as a quality criterion - and limitations, each of these can be (generally) formulated strictly as an optimization problem [3]. This approach can be applied to reliability, safety and risk analysis [3], [4]. But, because of the current complexity of the technical systems [9] the criteria for their safe and reliable operations are made important more and more [11], [12]. This implies that more than one object is taken into account for solving the optimization problems ([3], [5], [10], [12], [14], [16], [20], [22] - [23]). Some tools for solving the problems of complex technical systems operation, reliability, availability, safety and cost optimization are presented in [10] - [12]. The methods of the reliability prediction and optimization of complex technical systems related to their operation processes are introduced in [10].

The methods of operational research can be classified as deterministic or non-deterministic [15], [18], [20], [22]. There are many publications concerning deterministic or non-deterministic optimization methods for engineering and management [20] - [23]. The methods for solving the maritime transportation optimization problems, i.e. weather routing or minimizing fuel consumption are introduced in [14], [21], respectively.

In this paper the review of known graph theory algorithms and parameters [2] - [7], [13], [15], [17] - [19], [23] which can be applied to design, analyze and optimize of transportation systems is done.

Whereas, the presented approach can be used to any of listed above facilities.

2 MULTI-CRITERIA OPTIMIZATION METHODS REVIEW

In the general words, the single-objected optimization problem is defined as following mathematical model (minimizing or maximizing problem):

$$F(x_i) \rightarrow \min \text{ or } F(x_i) \rightarrow \max, \quad (1)$$

$$l_j(x_i) \leq 0, \quad l_j(x_i) \leq 0, \quad x_i \geq 0, \quad i, j = 1, 2, \dots, n$$

where

x_i - decision variables, $i = 1, 2, \dots, n$;

$F(x_i)$ - goal function;

$l_j(x_i)$ - limits function (low or high) for decision variables, $i, j = 1, 2, \dots, n$.

Whereas, the multi-objective (multi-criteria) optimization model can be described as a vector function f that maps a tuple of m decision variables to a tuple of n objectives. The formal notation is as follows [12]:

$$y = f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \rightarrow \min/\max$$

subject to

$$x = (x_1, x_2, \dots, x_m) \in X, \quad (2)$$

$$y = (y_1, y_2, \dots, y_n) \in Y,$$

where

x - decision variable,

X - parameter (variable) space,

y - objective vector,

Y - objective space.

The set of multi-objective optimization problem solutions consists of all decision vectors for which the corresponding objective vectors cannot be improved in any dimension without degradation in another. These vectors are known as Pareto optimal, what is related to the concept of domination vector by vector. It is simple to explain after introduce following definitions.

Definition 1

Let us take into account a maximization problem and consider two decision vectors $a, b \in X$, then a is said to dominate b if and only if

$$\forall i \in \{1, 2, \dots, n\} : f_i(a) \geq f_i(b)$$

$$\wedge$$

$$\exists j \in \{1, 2, \dots, n\} : f_j(a) > f_j(b). \quad (3)$$

Definition 2

All decision vectors which are not dominated by another decision vector are called non-dominated.

When the decision vectors are non-dominated within the entire search space, they are denoted as Pareto optimal (Pareto-optimal front).

These general formulations for single and multi-objective optimization problem are common for different types of engineering problems, which can be related to different optimization problems presented on Figure 1.

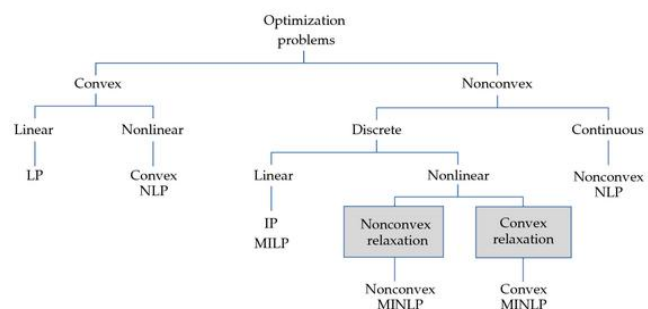


Figure 1. Different types of problem related to optimization problems [3], [15].

The basic classification of the optimization methods consists in their division due to the number of criteria (one or multi-criteria). As it is shown on Figure 2, there is possibility to distinguish seven most frequently used methods of multi-criteria optimization and three for one-criterion [3].

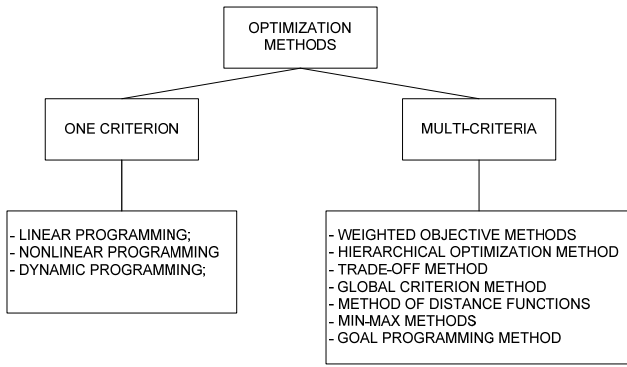


Figure 2. Chosen optimization methods [3].

These methods represent some general approaches for optimization, i.e.:

- deterministic,
- non-deterministic,
- heuristic,
- evolutionary/genetic.

The above approaches can provide general tools for solving optimization problems to obtain a global optimum. In second case the best way is using the evolutionary or genetic algorithms. The schema of basic genetic algorithm is presented on Figure 3.

General operation of genetic or evolutionary algorithms is based on the following steps (see Figure 3) [3]:

- 1 Initialization.
- 2 Calculate fitness.
- 3 Selection/Recombination/Mutations (parents and children).
- 4 Finished.

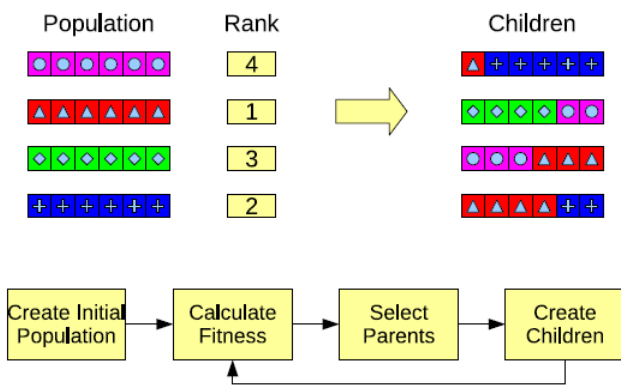


Figure 3. Basic genetic algorithm [3], [22].

The each chromosome of the genetic or evolutionary algorithm is represented as a string of bits.

In the paper the Strength Pareto Evolutionary Algorithm (SPEA) is considered [3], [23].

The basic notations for above algorithm are as follows:

- t - number of generation,
- P_t - population in generation t ,
- \bar{P} - external set in generation t ,
- \bar{P}' - temporary external set,
- P' - temporary population.

Additionally, it is necessary to give following input parameters:

- N - population size,
- \bar{N} - maximum size of external set,
- T - maximum number of generations,
- p_c - crossing probability,
- p_m - mutation probability,
- A - set of non-dominated solutions.

The Strength Pareto Evolutionary Algorithm is as follows [3], [23]:

Step 1. Initialization:

The initial population P_0 is generated according to procedure:

- 1 To get item i .
- 2 To add item i to set P_0 .

Next, the empty external set \bar{P}_0 is generated, where $t=0$.

Step 2. The complement of the external set is done.

Let $\bar{P}' = \bar{P}_t$

- 1 To copy non-dominated items from population P_t to population \bar{P}' .
- 2 To remove dominated items from set \bar{P}' .
- 3 To reduce the cardinality of the set \bar{P}' to value \bar{N} , using clustering and the solution give into P_{t+1} .

Step 3. Determination fit function.

The value of the fit function F for items from sets P_t i \bar{P}_t can be found according to following procedure:

The real value $S \in [0,1)$ is assigned for every item $i \in P_t$ (called power). This value is proportional to number of items $j \in P_t$, which represents the solutions dominated by item i .

The adaptation of item j is calculated as sum of all items from external set, represents solution dominated by item j , increased by 1.

The aim of addition 1 is to ensure that items $i \in \bar{P}_t$ will have better value of fit function than items from set P_t , i.e.

$$S(i) = \frac{n}{N+1}, \quad (4)$$

where:

$S(i)$ - power of item i ,
 n - number of items in population dominated by item i .

It is assumed that value of fit function for item i is equal to his power, i.e.

$$F(i) = S(i). \quad (5)$$

Step 4. Selection

Let $P' = \emptyset$.

For $i = 1, 2, \dots, k$ do

- 1 To choose randomly two items $i, j \in P_t \cup \bar{P}_t$.

- 2 If $F(i) < F(j)$ then $P' = P' \cup \{i\}$ else $P' = P' \cup \{j\}$, under assumption that value of fit is minimizing.

Step 4. Recombination.

Let $P'' = \emptyset$.

For $i = 1, 2, \dots, N/2$ do:

- 1 To choose two items $i, j \in P'$ and to remove it from P' .
- 2 To create items: k, l by crossing the items i, j .
- 3 To add items k, l to set P'' with probability p_c , else add items i, j to set P'' .

Step 5. Mutation

Let $P'' = \emptyset$.

For every item $i \in P''$ do:

- 1 To create item j by mutation the item i with probability p_m .
- 2 To add item j to set P'' .

Step 6. Finished

Let $P_{t+1} = P''$ and $t = t + 1$.

If $t \geq T$ then return A – non-dominated solution from population P_t and finish else back to Step 2.

3 REVIEW OF GRAPH THEORY TOPICS

Bellow chapter is showing general definitions, parameters and algorithms of the domination in graph theory and other related topics. Let $G = (V, E)$ be a connected simple graph (Figure 1) where V - set of n vertices, E - set of m edges. The set of neighbors of vertex v in G is denoted by $N_G(v)$.

3.1 Basics on domination in undirected graphs

In the following, the different type of domination sets and numbers' definitions will be introduced, i.e. general, connected and independent for undirected graphs [4], [6], [7].

Definition 1

A set $D \subseteq V(G)$ is the dominating set of G if for any $v \in V$ either $v \in D$ or $N_G(v) \cap D \neq \emptyset$. The domination number $\gamma(G)$ of a graph G is the minimum cardinality of a dominating set of G .

Definition 2

A set $D_c \subseteq V(G)$ is a connected dominating set of G if every vertex of $V \setminus D_c$ is adjacent to a vertex in D_c and the subgraph induced by D_c is connected. The minimum cardinality of a connected dominating set of G is the connected domination number $\gamma_c(G)$.

Definition 3

A set $D_I \subseteq V(G)$ is an independent dominating set of G if no two vertices of D_I are connected by any edge of G . The minimum cardinality of an independent dominating set of G is the independent domination number $\gamma_I(G)$.

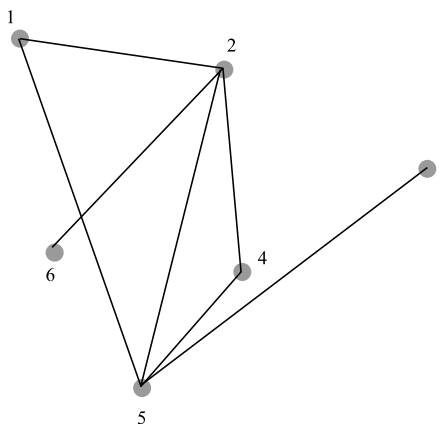


Figure 3. The exemplary undirected graph G .

For the exemplary graph G presented in Figure 3 the particular domination sets and numbers (Definitions 1 – 3) are as follows:

$$D = \{5, 6\}, \gamma(G) = 2;$$

$$D = \{1, 2, 5\}, \gamma_c(G) = 2;$$

$$D = \{2, 3\}, \gamma_I(G) = 2.$$

The above definitions refer to the general concept of undirected graphs. There are topics related to weight functions, what allows defined the vertex-weighted graph ([4], [6], [7]).

Definition 4

A vertex-weighted graph (G, w_v) is defined as a graph G together with a positive weight-function on its vertex set $w_v : V(G) \rightarrow R > 0$.

According to above definition we get next definitions.

Definition 5

The weighted domination number $\gamma_w(G)$ of (G, w_v) is the minimum weight $w(D) = \sum_{v \in D} w(v)$ of a set $D \subseteq V(G)$ such that every vertex $x \in V(D) - D$ has a neighbor in D .

Definition 6

The weighted connected domination number $\gamma_{Cw}(G)$ of (G, w_v) is the minimum weight $w(D_c) = \sum_{v \in D_c} w(v)$ of a set $D_c \subseteq V(G)$ such that every vertex of $V \setminus D_c$ is adjacent to a vertex in D_c and the subgraph induced by D_c is connected.

Definition 7

The weighted independent domination number $\gamma_{Iw}(G)$ of (G, w_v) is the minimum weight $w(D_I) = \sum_{v \in D_I} w(v)$ of a set $D_I \subseteq V(G)$ such that if

no two vertices of D_i are connected by any edge of (G, w) .

Similarly, it can be done for directed graphs.

Taking into account the complexity of the minimum dominating set, we should state, that in general is NP-hard problem. Efficient approximation algorithms do exist under assumption that any dominating set problem can be formulated as a set covering problem. Thus, the greedy algorithm for finding domination set is an analog of one that has been presented in [4], [18]. This algorithms is formulated as follows ([4], [18]):

Algorithm 1:

- 1 Let $V = \{1, \dots, n\}$, and define $D = \emptyset$.
- 2 Greedy add a new node to D in each iteration, until D forms a dominating set.
- 3 A node j , is said to be covered if $j \in D$ or if any neighbor of j is in D . A node that is not covered is said to be uncovered.
- 4 In each iteration, put into D the least indexed node that covers the maximum number of uncovered nodes.
- 5 Stop when all the nodes are covered.

For graph in Figure 1, Greedy will return $D = \{1, 2\}$.

In case of the minimum connected domination set, the Greedy algorithm is also used. However, to define them some preliminaries are necessary [4], [19].

We consider graph G and subset C of vertices in G . We can divided all vertices into three classes:

- belong to C are called black v_b ;
- not belong to C but adjacent to C are called gray v_g ;
- not in C and not adjacent to C are called white v_w .

Under assumption that C is connected dominating set if and only if there is no white vertex and the subgraph induced by black vertices is connected. The sum of the number of white vertices and the number of connected components of the subgraph induced by black vertices (black components) equals 1. The greedy algorithm with potential function equal to the number of white vertices plus number of black components is as follows [4], [19].

Algorithm 2:

Set $w := 1$;

while $w = 1$ **do**

If there exists a white or gray vertex such that coloring it in black and its adjacent white vertices in gray would reduce the value of potential function

then choose such a vertex to make the value of potential function reduce in a maximum amount

else set $w := 0$;

3.2 Spanning trees

The appropriate tool for the transportation system analysis in terms of its infrastructure connecting each

node the spanning tree is proposed. It can be done for both undirected and directed cases.

Definition 8

The spanning tree T of a connected, undirected graph G is a tree composed of all the vertices and some (or perhaps all) of the edges of G .

Informally, a spanning tree of G is a selection of edges of G that form a tree *spanning* every vertex. It means, that every vertex lies in the tree, but no cycles (or loops) are formed [4].

According to Definition 5, the edge-weighted graph is introduced.

Definition 9

An *edge-weighted graph* (G, w_e) is defined as a graph G together with a positive weight-function on its edge set $w_e : E(G) \rightarrow R > 0$.

Furthermore, for edge-weighted graphs it is possible to find minimum spanning tree, which is defined as follows.

Definition 10

A *minimum spanning tree (MST)* of an edge-weighted graph is a spanning tree whose weight (the sum of the weights of its edges) is no greater than the weight of any other spanning tree.

It can be done according to two well-known algorithms: Kruskal's and Prim's. They can be shown as follows [2], [4]:

Algorithm 3 (Kruskal's)

- 1 Find the cheapest edge in the graph (if there is more than one, pick one at random). Mark it with any given colour, say red.
- 2 Find the cheapest unmarked (uncoloured) edge in the graph that doesn't close a coloured or red circuit. Mark this edge red.
- 3 Repeat *Step 2* until you reach out to every vertex of the graph (or you have $n-1$ coloured edges).

The red edges form the desired minimum spanning tree.

Algorithm 4 (Prim's)

- 1 Pick any vertex as a starting vertex - v_{start} . Mark it with any given colour (red).
- 2 Find the nearest neighbor of v_{start} (call it P_1). Mark both P_1 and the edge $v_{start} P_1$ red. Cheapest unmarked (uncoloured) edge in the graph that doesn't close a coloured circuit. Mark this edge with same colour of Step 2.
- 3 Find the nearest uncoloured neighbor to the red subgraph (i.e., the closest vertex to any red vertex). Mark it and the edge connecting the vertex to the red subgraph in red.
- 4 Repeat *Step 3* until all vertices are marked red.

The red subgraph is a minimum spanning tree.

The above knowledge (Sections 3.1 and 3.2) can be applied to analyze transportation systems, particularly its infrastructure (even critical). For instance it is useful to choose the routes and nodes classified as critical infrastructures [4], [8]. To show possible applications of discussed methods, the academic example is shown below.

Example

Let us consider the hypothetical map of road connections, what is given by schema presented in Figure 4. We choose eleven nodes and describe them the consecutive number and the time of red light (in seconds). The edges between the nodes are described by the number of kilometers.

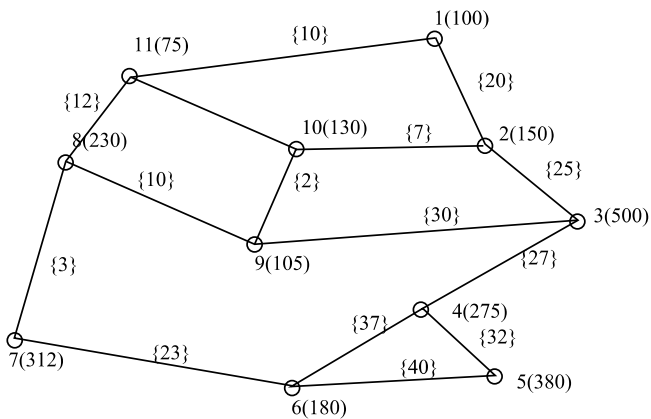


Figure 4. The exemplary scheme of road infrastructure.

Our main goal is to choose the minimal spanning tree and minimal independent domination set, i.e. finding an independent domination number.

According to the Kruskal's Algorithm, the minimal spanning tree is given in Figure 5 as the set of double edges.

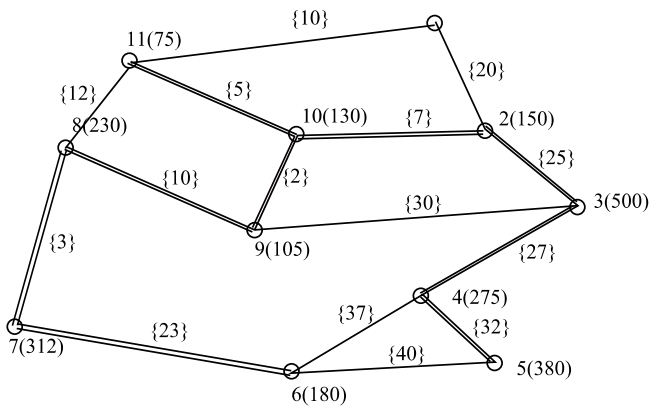


Figure 5. Minimal spanning tree (Kruskal's Algorithm)

In this way, the minimal number of kilometers is equal to 134 [km]. According to Algorithm 1, the time of red lights is equal to 430 [seconds]. It is minimal dominating set, what is marked in Figure 6 with black nodes (vertices).

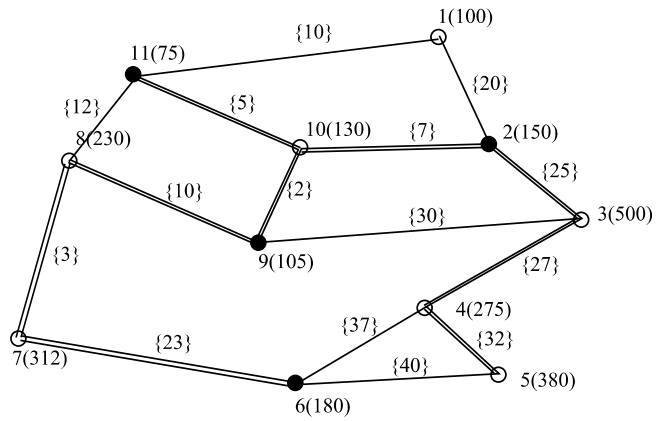


Figure 6. Minimal weighted independent domination set.

3.3 Introduction to the knapsack problem

One of the most known problem in graph theory is the knapsack problem. It has been studied since 1897 and is combinatorial optimization problem. General description is based on given a set of items, each with a mass and a value [3]. There is determined the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible (according to (1)). The knapsack problem has many modified forms, i.e. to form of the 0-1 knapsack problem. In this way is formulated as multi-objective optimization problem [3].

General assumption about a 0-1 knapsack problem is it consists of a set of items, weight and profit associated with each item, and an upper bound for the capacity of the knapsack. We want find a subset of items which maximizes the profits and all selected items fit into the knapsack, i.e., the total weight does not exceed the given capacity [23].

After assuming an arbitrary number of knapsacks, the single-objective problem is extended directly to the multi-objective case. Formally, the multi-objective 0-1 knapsack problem can be defined in the following way [23]:

Given a set of m items and a set of n knapsacks, with

$p_{i,j}$ = profit of item j according to knapsack i ,

$w_{i,j}$ = weight of item j according to knapsack i ,

c_i = capacity of knapsack i ,

find a vector $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \{0,1\}^m$, such that

$$\forall i \in \{1, 2, \dots, n\}: \sum_{j=1}^m w_{i,j} \cdot x_j \leq c_i \tag{6}$$

and for which $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$ is maximum, where

$$f_i(\mathbf{x}) = \sum_{j=1}^m p_{i,j} \cdot x_j \tag{7}$$

and $x_j = 1$ if and only if when item j is chosen.

Nowadays, the best solutions of knapsack problem are described in terms of a genetic methods. As it is

shown in [3] it can be very useful tool for multi-objective optimization of transportation systems.

3.4 Flows in transportation systems

The graph theory is the basis for analyzing a traffic flow in transportation systems and networks [13], [17]. In section 3 the definition of undirected graph was introduced. But, for more detailed analysis of traffic flows, the digraphs should be defined. Thus, the graph $G^d = (V, A)$ is directed graph or digraph with a set V , whose elements are called vertices or nodes and set A of ordered pairs of vertices, called arcs, directed edges, or arrows (Figure 7).

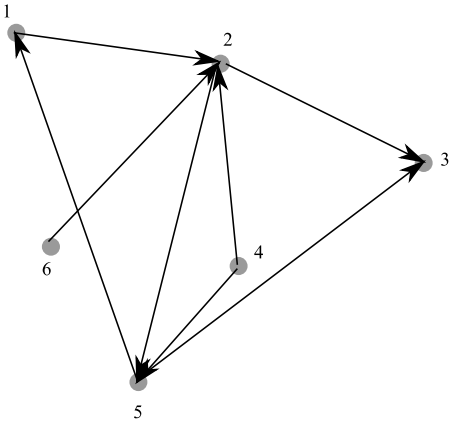


Figure 7. The exemplary directed graph G .

The main reason for the creation of this Subsection is fact that transportation engineers need know the traffic flow theory as a tool that helps understand and express the problems of evaluating the capacity of existing roadways or designing new ones. Thus, the following definitions are important [17].

Definition 11

Let $G_{st} = (V, A, s, t)$ be a network with $s, t \in A$ being the source and the sink of G_{st} respectively.

Definition 12

The capacity of an edge of network G_{st} is mapping $c: A \rightarrow R^+$, which represents the maximum amount of flow that can pass through an edge. It is denoted as c_{uv} , where $u, v \in A$.

Definition 13

A flow in network G_{st} is mapping $f_{uv}: A \rightarrow R^+$, where $u, v \in A$, which subjects to the following two constrains:

- 1 $\forall_{(u,v) \in A} f_{uv} \leq c_{uv}$, (capacity constraint: the flow of an edge cannot exceed its capacity)
- 2 $\forall_{v \in V \setminus \{s, t\}} \sum_{u: (u,v) \in A} f_{uv} = \sum_{u: (u,v) \in A} f_{vu}$, (conservation of flows: the sum of the flows entering a node must equal the sum of the flows exiting a node, except for the source and the sink nodes).

Definition 14

The value of flow $|f_{uv}| = \sum_{v: (s,v) \in A} f_{sv}$, where s is the source of G_{st} .

Generally, the main problem in traffic flows is maximize value of $|f_{uv}|$, which is called maximum flow problem. One of the solutions is using residual network, which is defined as follows [17].

Definition 15

For given G_{st} and flow f_{uv} , we define the residual network G_{st}^f as follows:

- 1 The node set of G_{st}^f is the same as that of G_{st} ;
- 2 Each edge $e = (u, v)$ of G_{st}^f is with capacity $c_e - f_e$;
- 3 Each edge $e' = (v, u)$ of G_{st}^f is with capacity f_e .

Furthermore $c_p^f = \min\{c_{uv}^f : (u, v) \in p\}$ is residual capacity of path p in network G_{st}^f .

According to above definitions, the algorithm of finding the maximal flow in network is given as follows [13], [17].

Algorithm 5 (Ford-Fulkerson)

For each edge $e = (u, v)$ **do**

Begin

$$f_{uv} = 0$$

$$f_{vu} = 0$$

End

While exists path p from s to t in G_{st}^f **do**

Begin

$$c_p^f = \min\{c_{uv}^f : (u, v) \in p\}$$

For each $e = (u, v) \in p$ **do**

begin

$$f_{uv} = f_{uv} + c_p^f$$

$$f_{vu} = f_{vu} - c_p^f$$

End

End

Return f .

Based on algorithm 5, the exemplary residual network is proposed in Figure 8.

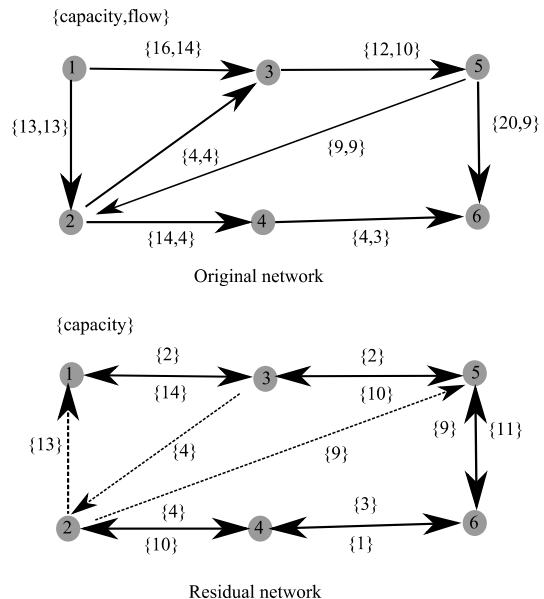


Figure 8. Exemplary original and residual network, where $s=1$, $t=6$.

4 CONCLUSION

In the paper, the concept of critical infrastructures has been presented. Some review and classifications of well-known information about optimization, graph theory and network flow theory have been done. The SPEA algorithm has been described step-by-step and the knapsack problem with its binary modification has been presented. It has also been used the SPEA algorithm to solve the 0-1 knapsack problem. Furthermore, the selected definitions, parameters and algorithms of graph theory have been introduced and applied to system transportation analysis. As the example of possible application, the road transportation system with the shortest time of red light in nodes and the shortest kilometers have been determined. The examples, from Section 3, are only showing potential applications of methods, algorithms and parameters, which are described in the article.

REFERENCES

- [1] COMMISSION OF THE EUROPEAN COMMUNITIES (2006), Communication from the Commission on a European Programme for Critical Infrastructure Protection, Brussels.
- [2] Cormen, T. H. & al. (2009). Introduction to Algorithms, Third Edition. MIT Press, ISBN 0-262-03384-4. Section 23.2: The algorithms of Kruskal and Prim, pp. 631–638.
- [3] Guze, S. (2014). Application of the knapsack problem to reliability multi-criteria optimization. *Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars*, Vol. 5, No 1, pp. 85 – 90.
- [4] Guze, S. (2014). The graph theory approach to analyze critical infrastructures of transportation systems. *Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars*, Vol. 5, No 2, pp. 57-62.
- [5] Guze, S. Smolarek, L. (2011). *Methods for risk minimizing in the process of decision-making under uncertainty*. Journal of Polish Safety and Reliability Association - JPSRA, Vol. 2, Number 1, 123 – 128, Gdańsk-Sopot.
- [6] Harary, F. (1969). Graph Theory. Addison-Wesley, Reading.
- [7] Haynes T. W., Hedetniemi, S., Slater, P. (1988). Fundamentals of Domination in Graphs. CRC Press.
- [8] Kołowrocki, K. (2013). Safety of critical infrastructures. *Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars*, Vol. 4, No. 1, pp. 51-74.
- [9] Kołowrocki, K. (2004). *Reliability of Large Systems*. Elsevier, Amsterdam - Boston - Heidelberg - London - New York - Oxford - Paris - San Diego - San Francisco - Singapore - Sydney - Tokyo.
- [10] Kołowrocki, K. & Soszyńska, J. (2010). *Optimization of complex technical systems operation processes*. Maintenance Problems, No 1, 31-40, Radom.
- [11] Kołowrocki, K. & Soszyńska-Budny, J. (2011). *Reliability and Safety of Complex Technical Systems and Processes, Modeling – Identification – Prediction – Optimization*, Springer-Verlag.
- [12] Kołowrocki, K. Soszyńska-Budny, J. (2013). *Reliability prediction and optimization of complex technical systems with application in port transport*. Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars – SSARS 2013, Volume 3, Number 1-2, 263 – 279, Gdańsk-Sopot.
- [13] Leeuwen, Van, J. (1986). Graph Algorithms. Book.
- [14] Marie, S. & Courteille, E. (2009) *Multi-Objective Optimization of Motor Vessel Route*. TransNav, The International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 3, No. 2, 133-141, Gdynia.
- [15] Martello, S. & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Chichester, U.K.: Wiley.
- [16] Ming-Hua, L., Jung-Fa, T. and Chian-Son Y. (2012), *A Review of Deterministic Optimization Methods in Engineering and Management*, Mathematical Problems in Engineering, Volume 2012.
- [17] Newell, G. F. (1980). Traffic flow on transportation networks. MIT Press Series in transportation studies, Monograph 5.
- [18] Parekh, A. K. (1991). Analysis of Greedy Heuristic for Finding Small Dominating Sets in Graphs. *Information Processing Letters*, Volume 39, Issue 5, pp. 237 – 240.
- [19] Ruan, L. & al., (2004). A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science*, Volume 329, Issues 1–3, pp 325–330.
- [20] Sun, W. and Yuan, Y.-X. (2006). *Optimization theory and methods: nonlinear programming*. Springer-Verlag.
- [21] Szłapczyńska, J. (2013). *Multicriteria Evolutionary Weather Routing Algorithm in Practice*. TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 7, No. 1, 61-65, Gdynia.
- [22] Venter, G. (2010). *Review of Optimization Techniques, Encyclopedia of Aerospace Engineering*, John Willey and Sons, Ltd.
- [23] Zitzler, E. & Thiele, L. (1999). *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. IEEE Transactions on Evolutionary Computation, VOL. 3, NO. 4., 257 – 271.