Zeszyty Naukowe Akademii Morskiej w Gdyni
Scientific Journal of Gdynia Maritime University

Nr 107/2018, 25–39
ISSN 2451-2486 (online)
ISSN 1644-1818 (printed)
DOI: 10.26408/107.02

Złożony/submitted: 01.11.2018
Zaakceptowany/accepted: 12.11.2018
Opublikowany/published: 28.12.2018

REVIEW OF METHODS AND ALGORITHMS FOR MODELLING TRANSPORTATION NETWORKS BASED ON GRAPH THEORY

PRZEGLĄD METOD I ALGORYTMÓW MODELOWANIA SIECI TRANSPORTOWYCH OPARTYCH NA TEORII GRAFÓW

Sambor Guze
Gdynia Maritime University, 81-87 Morska, 81-225 Gdynia, Faculty of Navigation, Department of Mathematics, e-mail: s.guze@wn.umg.edu.pl, ORCID 0000-0003-1635-1924

Streszczenie: Jednym z najlepszych sposobów modelowania sieci transportowej jest użycie grafu z wierzchołkami i krawędziami. Reprezentują one odpowiednio węzły i łuki takiej sieci. Teoria grafów daje możliwość użycia dziesiątek parametrów lub charakterystyk, w tym spójności, drzew spinających lub różnych typów liczb dominowania i związanych z tym problemów. Głównym celem artykułu jest przedstawienie metod i algorytmów teorii grafów pomocnych w modelowaniu i optymalizacji sieci transportowej. Po pierwsze, wprowadzono opisy podstawowych pojęć w teorii grafów. Następnie zaprezentowano koncepcje dominowania, liczby zniewolenia czy podziału krawędzi grafu oraz ich implementacji do opisu i modelowania sieci transportowej. Ponadto przedstawiono algorytmy do wyszukiwania drzewa opinającego i maksymalnego przepływu w sieciach. Wreszcie pokazano możliwe sposoby wykorzystania wyróżnionych koncepcji do przykładu sieci transportowej. Na zakończenia przedstawiono wnioski i przyszłe kierunki prac.

Słowa kluczowe: problem plecakowy, liczba dominowania, liczba zniewolenia spójnego, MST, maksymalny przepływ, sieć transportowa, wrażliwość sieci.

Abstract: One of the best ways of modelling a transport network is to use a graph with vertices and edges. They represent nodes and arcs of such network respectively. Graph theory gives dozens of parameters or characteristics, including a connectivity, spanning trees or the different types of domination number and problems related to it. The main aim of the paper is to show graph theory methods and algorithms helpful in modelling and optimization of a transportation network. Firstly, the descriptions of basic notations in graph theory are introduced. Next, the concepts of domination, bondage number, edge-subdivision and their implementations to the transportation network description and modeling are proposed. Moreover, the algorithms for finding spanning tree or maximal flow in networks are presented. Finally, the possible usage of distinguishing concepts to exemplary transportation network is shown. The conclusions and future directions of work are presented at the end of the paper.

Keywords: knapsack problem, domination number, bondage-connected number, MST, maximal flow, transportation network, vulnerability.

1. INTRODUCTION

The main problem in transportation network analysis is proper description and selection of the right tool. Natural way to describe a structure of transportation network is the graph theory. According to definition of graph, there is a set of vertices and edges. In case of description of transportation networks, the nodes and arcs (links) are used respectively. In this way of thinking, the graph is a simplified mathematical model of the physical network. Many solutions of the real problems in transport networks have been reduced to finding the right parameters in the corresponding graph. The well-known problems solving according to graph theory are Traveling Salesman Problem (TSP) [Harrary 1969; Leeuwen 1986; Cascetta 2001; Cormen et al. 2009] with later modification [Neumann 2016], Chineese Postman Problem [Harrary 1969; Leeuwen 1986], maximal flow in network [Newell 1980; Leeuwen 1986], knapsack problem [Martello and Toth 1990; Cormen et al. 2009]. Due to technological developments, it is necessary to continue to explore the possibility of writing complex technical problems in networks using the simplest but also the most faithful mathematical graph theory models.

The basic task of the transport network is to enable you to reach from point A to point B. Because of this, it is reasonable to search for a way to describe the transport network using the theory of reliability and safety [Kołowrocki and Soszyńska-Budny 2011; 2013] also with regard to its operation processes [Kołowrocki and Soszyńska-Budny 2011].

Sometimes, the solution is well-known, but it is not enough. Thus, the important part of research on transportation networks is optimization. Earlier, the one criterion of optimization gave full solution. There should be mentioned here algorithms for: TSP [Cascetta et al. 2001; Cormen et al. 2009; Neumann 2016], maximal flow in network [Newell 1980; Leeuwen 1986], minimum spanning tree [Leeuwen 1986; Cormen et al. 2009; Guze 2014b]. But now the multi-criteria is much more important. There are some algorithms in graph theory, which serve as multi-criteria optimization tools. There should be mentioned here algorithms for 0-1 knapsack problem [Zitzler and Thiele 1999; Guze 2014a; 2015] or minimal cost of maximal flow in network [Newell 1980; Leeuwen 1986; Cormen et al. 2009; Guze 2014b].

The domination theory in graphs [Harrary 1969; Haynes, Hedetniemi and Slater 1998] and related concepts as bondage number or sub-division number [Fink et al. 1990; Hartnell and Rall 1994] can also be used to analyze and optimize the transportation network. The proposition of the usage mentioned conceptsare presented in [Guze 2014b; 2015; 2017].

The main aim of this paper is to review and to show usefulness of the models based on graph theory in network analysis and optimization.

2. GRAPH THEORY BASIC NOTATIONS

We consider the connected, simple, undirected graph $G = (V,E)$, where V is the set of vertices (nodes) and E is the set of edges (arcs). Sometimes, we use the fuller notation for graph G, where *V(G)* and *E(G)* denote respectively its vertex-set and the edge-set. The assumption about connectivity is very important, because it is the fundamental to functioning of transport or logistics networks.

Some basic definitions of graph theory are necessary to understand ways of description of networks graphs. The set of all adjacent vertices to vertex $v \in V$ in G is called neighborhood and denoted by $N_G(v)$ or $N(v)$. The close neighborhood of this vertex is defined as $N_G(v) \cup \{v\}$ and denoted by $N_G[v]$. The other basic parameter for graphs is degree of vertex $v \in V$, what is defined as the number of vertices in $N_G(v)$ and denoted by $\deg(v)$. The minimum and maximum degree are defined as $\delta(G) = \min\{x \in V : \deg(x)\}$ and $\Delta(G) = \max\{x \in V : \deg(x)\}$, respectively. Moreover, the set of all edges, incident to the vertex $v \in V$ is denoted by $I_G(v)$. For any set $A \subseteq V$, neighborhood is given by $N(A) = \bigcup_{v \in A} N(v)$. The induced subgraph defined on A is denoted by $G[A]$.

Regarding the assumption about connectivity in graphs, it is said that two *vertices u* and *v* in graph *G* are called connected if *G* contains a path from *u* to *v*. Otherwise, they are called *disconnected*. A graphis said to be connected if every pair of vertices in the graph is connected. If graph is disconnected, there are a connected component, which are maximal connected subgraphs of *G*. Each vertex belongs to exactly one connected component, as does each edge.

3. REVIEW OF METHODS FOR ANALYSIS OF TRANSPORTATION NETWORKS

3.1. Domination and related topics in graphs

The domination theory has various application, where the analysis of communications network is the one of most discussed in literature. Thus, the idea is to transfer selected methods to the transportation network research field.

One of these methods, more precisely, the parameter is the dominating set and domination number. According to results given in [Haynes 1988] a set $D \subseteq V(G)$ is a *dominating set* of graph G if for any $v \in V$ either $v \in D$ or $N_G(v) \cap D = \emptyset$. While the minimum cardinality of a dominating set of graph G is called *domination number* of G and denoted as $\gamma(G)$.

The example of dominating set is presented in Fig. 2.

Fink et al. have examined a question concerning the vulnerability of the communications network under link failure. They supposed situation, where someone does not know which nodes in the network act as transmitters but does know that the set of such nodes can build a minimum dominating set in the related graph. Thus, they asked about the fewest number of communication links that he must sever so that at least one additional transmitter would be required in order that communication with all sites be possible. In this way, they introduced the new parameter called the bondage number of a graph. It is defined in following way. The bondage number $b(G)$ of nonempty graph G is the minimum cardinality among all sets of edges E for which $\gamma(G-E)>\gamma(G)$ [Fink et al. 1990; Hartnell and Rall 1994]. Thus, the bondage number of graph G describes the smallest number of edges whose removal from *G* results in a graph with domination number larger than that of *G* (see Fig. 3 or Fig. 4).

Because the definition of bondage network is not enough for transportation network analysis, in [Guze 2017] author defined the bondage-connected number $bc(G)$ of nonempty graph *G*as the minimum cardinality among all sets of edges *E* for which $\gamma(G-E)>\gamma(G)$ and graph $G-E$ is connected.

The above definitions refer to the general concept of undirected graphs. There are topics related to weight functions, what allows defined the vertex-weighted and edge-weighted graphs [Harrary 1969; Haynes, Hedetniemi and Slater 1988; Guze 2014b].

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

These definitions are needed to define different types of weighted dominiation numbers in an undirected graph. Two of them are as follows:

- 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;
- 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 [Parekh 1991; Guze 2014]. This algorithm is formulated as follows [Parekh 1991; Guze 2014]:

Algorithm 1:
1. Let $V = \{1, ..., n\}$, and define $D = \varphi$.
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.

In case of the minimum connected domination set, the greedy algorithm is also used. However, to define them some preliminaries are necessary [Guze 2014b].

3.2. Spanning tree

The main role of the transportation networks is to guarantee reliable transfer form point A to point B. Thus, the connectivity of the network is the most important thing. As the appropriate tool for the transportation network analysis in terms of its links between each node is finding the spanning tree of this network. It can be done for both undirected and directed cases or for simple or edge-weigthed graphs.

Generally, 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*. In other words, 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 according to tree definition [Guze 2014b].

4. REVIEW OF METHODS FOR OPTIMIZING TRANSPORTATION NETWORKS

4.1. Minimal spanning tree

The first optimization problem disscussed in this section is minimum spanning tree. It can be done for the edge-weighted graphs. For this type of graphs, a *minimum spanning tree* (MST) 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.

The solution of this problem can be done according to two well-known algorithms: Kruskal's and Prim's. They can be shown as follows [Cormen et al. 2009; Guze 2014]:

Algortihm 2 (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 does not 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.

Algortihm 3 (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 does not 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.

4.2. The knapsack problem

Second optimization problem mentioned in this paper is one of the most known problem in graph theory - the knapsack problem. First studies in this problem started in 1897. It is combinatorial optimization problem. General description is based on given a set of items, each with a mass and a value [Martello and Toth 1990; Guze 2014a]. 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)). One of the most applicable modification of the knapsack problem is the 0-1 knapsack problem. In this way is formulated as multi-objective optimization problem [Guze 2014a].

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 to 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 [Zitzler and Thiele 1999; Guze 2014a; 2015].

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 [Zitzler and Thiele 1999; Guze 2015]:

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 $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in \{0,1\}^m$, such that $\forall i \in \{1,2,...,n\}: \sum_{j=1}^{m} w_{i,j} \cdot x_j \le c_i$

and for which $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$ is maximum, where $f_i(\mathbf{x}) = \sum_{j=1}^{m} p_{i,j} \cdot x_j$

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 exemplary in [Zitzer and Thiele 1999; Guze 2015] it can be very useful tool for multi-objective optimization of transportation networks and complex systems, particulary in their safety and reliability aspects [Guze 2015].

4.3. Flows in networks

The graph theory is the basis for analyzing a traffic flow in transportation systems and networks [Newell 1980; Leeuwen 1986]. In Section 2 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.

The following definitions are important [Newell 1980]:

- 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;
- the capacity of an edge of network G_{st} is mapping $c: A \to R^+$, which represents the maximum amount of flow that can pass through an edge. It is denoted as c_{uv}, where $u, v \in A$;
- a flow in network G_{st} is mapping $f_{uv}: A \to R^+$, where $u, v \in A$, which subjects to the following two constraints:

1 $\underset{(u,v)\in A}{\forall} f_{uv} \le c_{uv}$, (capacity constraint: the flow of an edge cannot exceed its capacity),

2 $\underset{v\in V\setminus\{s,t\}}{\forall} \sum_{u:(u,v)\in A} f_{uv} = \sum_{u:(u,v)\in A} f_{vu}$, (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).

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 G_{st}. The definition of this type of network, according to [Newell 1980], for given G_{st} and flow f_{uv}, is 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 [Newell 1980; Leeuwen 1986; Cormen et al. 2009].

Algorithm 4 (Ford-Fulkerson)

1. for each edge $e = (u,v)$ do
2. begin
3. $\quad\quad f_{uv} = 0$ and $f_{vu} = 0$;
4. end
5. while exists path p from s do t in G_{st}^f do
6. begin
7. $\quad\quad c_p^f = \min\{c_{uv}^f : (u,v) \in p\}$;
8. $\quad\quad$ for each $e = (u,v) \in p$ do
9. $\quad\quad$ begin
10. $\quad\quad\quad\quad f_{uv} = f_{uv} + c_p^f$ and $f_{vu} = -f_{uv}$;
11. $\quad\quad$ end
12. end
13. return *f*.

The extension of this problem is minimal cost maximal flow problem, where addtionally each arc $e=(u,v)$ has a cost k_{uv}. The cost sending a flow f_{uv} by the edge (u,v) is equal to $k_{uv} \cdot f_{uv}$. The total cost of a flow over all edges is given by $\sum_{(u,v)\in E} k_{uv} \cdot f_{uv}$,with some constraints:

- $\forall_{(u,v)\in A} f_{uv} \le c_{uv}$, (capacity constraint: the flow of an edge cannot exceed its capacity);
- $\forall_{v\in V\setminus\{s,t\}} \sum_{u:(u,v)\in A} f_{uv} = \sum_{u:(u,v)\in A} f_{vu}$, (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);
- Skew symmetry: $f_{uv} = -f_{vu}$,
- Requiered flow: $\sum_{w\in V} f_{sw} = d$ and $\sum_{w\in V} f_{wt} = d$, where s is the sink and t is the source.

There is many algorithms for minimal cost maximum flow problem. The base is the mixture of the Bellamnn-Ford algorithm, which helps in detecting of negative cycles in cost network, and the Ford-Fulkerson for maximal flow in a network (see Algorithm 5, source [http:www./hackerearth.com/practice…. 2018]).

```
Algorithm 5 (Cycle Cancelling Algorithm)
function: CostNetwork(Graph G, Graph Gf):
    Gcis empty graph
    fori in edges E(G):
        if E(u,v) in G:
            cf(u,v) = c(u,v)
        else if E(u,v) in Gf:
            cf(u,v) = -c(u,v)
function: MinCost(Graph G):
    Find a feasible maximum flow of G using Ford Fulkerson and construct residual
    graph(Gf)
      Gc = CostNetwork(G, Gf)
      while(negativeCycle(Gc)):
          Increase the flow along each edge in cycle C by minimum capacity in the
          cycle C
          Update residual graph(Gf)
          Gc = CostNetwork(G,Gf)
    mincost = sum of Cij*Fij for each of the flow in residual graph
    return mincost
```

In this solution, the main problem is detecting and reflecting the negative cycle. In cost network it is a cycle where sum of costs of all the edges in the cycle is negative number. These cycles can be detected using Bellman Ford algorithm. They should be eliminated because flow through such cycles cannot be allowed.

5. EXAMPLE OF APPLICATION IN TRANSPORTATION NETWORKS

The main aim of this section is to present the possible application of selected graph theory topics presented in Sections 3 and 4. The urban area of city can be divided into communication regions, what helps in analysis and management of traffic flow. The number of these regions depends on detail of provided analyses.

In our example, let us consider, the exemplary small urban transportation network with eight communication regions, what is represented by graph G given in Fig. 1. This is the network's graph consisting of set of 8 nodes {A, B, C, D, E, F, G, H} with degrees {3, 3, 4, 4, 5, 4, 4, 3} respectively. The minimal degree of graph $\delta(G) = 3$ for vertices {A, B, H} and maximal degree $\Delta(G) = 5$ for only one vertex $\{E\}$.

We assume, that the minimal dominating set represents the most important nodes for traffic flow management. In this way, we can formulate the problem of how many broken links in the network will cause to add at least one node as the most important for traffic flow management. In other words, the question is how vulnerable for disruptive events this network is.

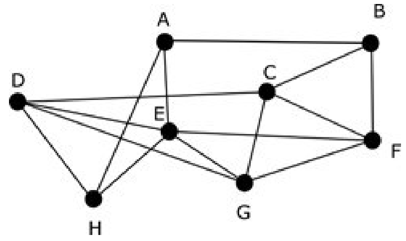

Fig. 1. *Part of urban transportation network with 8 nodes*

Rys. 1. Fragment miejskiej sieci transportowej z 8 węzłami

To solve this problem, we can use the domination number and the bondage-connected number mentioned in Subsection 3.1. This can be considered in two ways.

First way is the general approach. After deleting arcs in graphs, the new minimal dominating set has to be found. In the second approach, after operation of deleting the edges, the new minimal dominating set is extension the beginning one.

In the Fig. 2, the minimal dominating set of considered network is given as blue vertices {C, E} and it is found with Algorithm 1. Thus, the domination number $\gamma(G)$ of this network is equal to 2. This is the initial set of main nodes in aspect of traffic flow management.

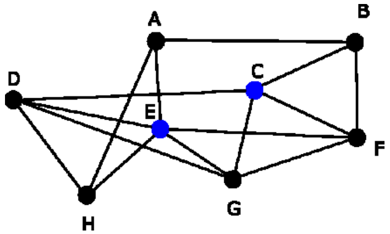

Fig. 2. *Minimal domination set (blue nodes), domination number* $\gamma(G) = 2$

Rys. 2. Minimalny zbiór dominujący (niebieskie wierzchołki), liczba dominowania $\gamma(G) = 2$

Now, we can consider the vulnerability of this network in general case. The resulting graph is presented in Fig. 3 according to the definition of bondage-connected number of graph G, $bc(G)$, mentioned in Section 3.1.

As we see, in Fig. 3, the cardinality of the minimum dominating set increases only after deleting five edges (C,D), (D, E), (E.H), (D,G), (E,F). Thus, the $bc(G) = 5$ and blue vertices are elements of new minimal dominating set $\{A,G,H\}$. These nodes of urban transportation network are now important in traffic flow management point of view.

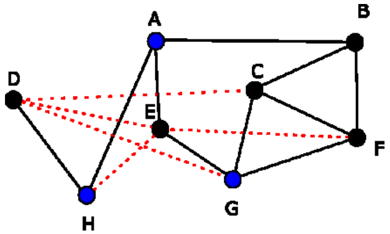

Fig. 3. *The bondage-connected number of transportation network* $b(G) = 5$

Rys. 3. Liczba zniewolenia spójnego w sieci transportowej $b(G) = 5$

When we take into account second approach, where the minimal dominating set presented in Fig. 2 is fixed and after deleting the edges, the initial minimal dominating set is extended by addition of one vertex. As it is presented in Fig. 4, the only one edge erasing is enough to increase the domination number of graph G.

This approach is the simplier than this presented firstly, because traffic flow management system needs less time to react for changes in transportation network. But on the other side, in this approach, the urban transportation network is more vulnerable for disruptive events. Only one edge makes problems in considered network.

Necessary deleting five red-dotted edges: (C,D), (D, E), (E.H), (D,G), (E,F), the blue vertices form the minimal domination set.

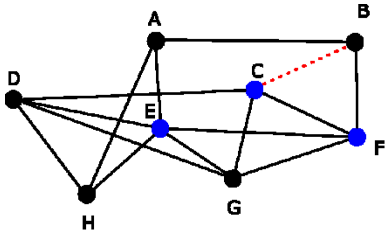

Fig. 4. The bondage number of transportation network $bc(G) = 1$ *for fixed minimal dominating set*

Rys. 4. Liczba zniewolenia w sieci transportowej $bc(G) = 1$ przy ustalonym najmniejszym zbiorze dominującym

The second aim for this section is to show the possible usage of the computer program "Algorytmy grafowe" programmed by M. Syslo [Syslo 2011]. The based algorithms for:

- minimal spanning tree;
- the shortest path problem;
- the depth first search (DFS);
- the breadth-first search (BFS)

are implemented.

We demonstrate only how to use this programme for finding minimal spanning tree for the considered transportation network with random weights. First, we introduce the graph represented by urban transportation network, what is shown in Fig. 5. In addition to drawing of the graph, the programme also presents the matrix and the list of neighborhood in graph *G*. The matrix of neighborhood is the best representation of graph for computers.

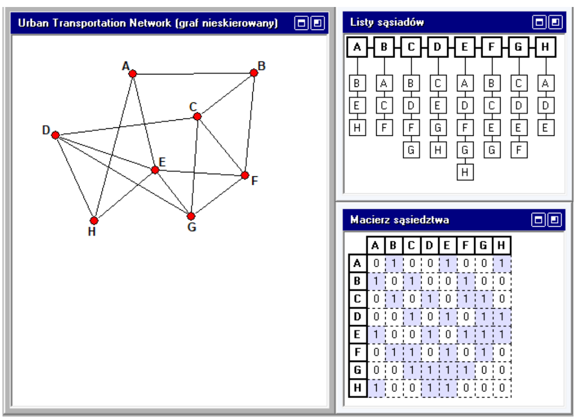

***Fig. 5.** The part of transportation network drawn in "Algorytmy grafowe"*

Rys. 5. Fragment sieci transportowej narysowany w programie „Algorytmy grafowe"

We consider the simple, undirected graph as the model of transportation network. To find the minimal spanning tree, we need to use the edge weighted graph (*G, we*). The weights are random, generated by computer programme in the number range presented by interval [11;75]. The edge-weighted graph is shown in Fig. 6.

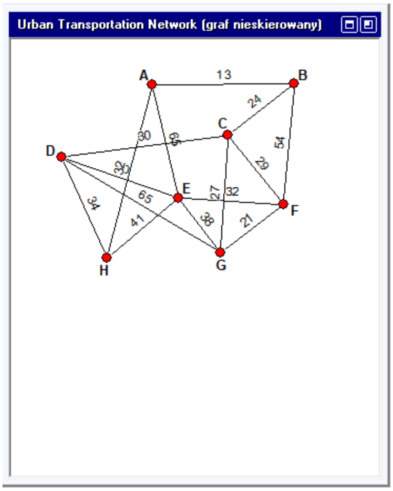

Fig. 6. *The edge-weighted graph generated by computer programme*

Rys. 6. Graf z wagami krawędziowymi wygenerowany przez program komputerowy

For this edge-weighted graph, the minimum spanning tree sought by the algorithms Kruskal's or Prim's is presented in Fig. 7.

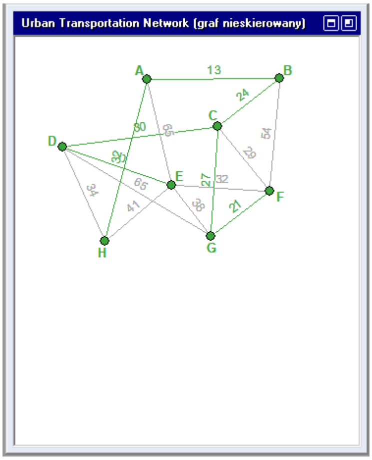

Fig. 7. *The minimum spanning tree for edge-weighted graph* (G, w_e)

Rys. 7. Najmniejsze drzewo spinające dla grafu (G, w_e)

The green edges form the minimum spanning tree (A,B), (F,G), (B,C), (C,G), (D,E), (C,D), (A,H) with respective set of weights {13,21,24,27,30,30,32}. Thus, the weight of this minimum spanning tree is equal to 177.

6. CONCLUSIONS

In the paper selected graph theory concepts have been presented, i.e. domination number, bondage and bondage-connected number, minimum spanning tree with application. First, the descriptions of basic notations in graph theory have been introduced. Next, the concepts of domination, bondage number, bondage-connected and their implementations to the transportation network description and modelling have been proposed. Moreover, the algorithms for finding spanning tree or maximal flow in networks have been presented.

Finally, the applications of presented methods have been shown for part of urban transporation network. These applications show that, the graph theory methods used as the tools for analysing and optimazing the transportation networks are helpful. They are simple but at the same time they model the real systems and networks accurately. In the future, the weighted-algorithms for vulnerability should be presented. To achieve this goal, first of all, the algorithm for finding the weighted-domination number has to be proposed. Next, the weighted bondage-connected number would be defined. This approach can give the opportunity to take into account the significance of the transport network nodes.

REFERENCES

Cascetta, E., 2001, *Transportation Systems in Transportation Systems Engineering: Theory and Methods*, Cascetta, E. (ed.), Springer US, Boston, MA, s. 1–22.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2009, *Introduction to Algorithms*, 3rd ed., MIT Press, s. 631–638.

Fink, J.F., Jacobson, M.S., Kinch, L.F., Roberts, J., 1990, *The Bondage Number of Graph*, Discrete Mathematics, vol. 86, no. 1–3, s. 47–57.

Guze, S., 2014a, *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, s. 85–90.

Guze, S., 2014b, *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, s. 57–62.

Guze, S., 2014c, *Graph Theory Approach to Transportation Systems Design and Optimization,* TransNav, International Journal on Marine Navigation and Safety of Sea Transportation, vol. 8, no. 4, s. 571–578.

Guze, S., 2015, *Numerical Application of the SPEA Algorithm to Reliability Multi-Objective Optimimization,* Journal of Polish Safety and Reliability Association, Summer Safety and Reliability Seminars, vol. 6, no. 1, s. 101–114.

Guze, S., 2017, *An Application of the Selected Graph Theory Domination Concepts to Transportation Networks Modelling*, Zeszyty Naukowe Akademii Morskiej w Szczecinie, nr 52(124), s. 97–102.

Harrary, F., 1969, *Graph Theory*, Addison-Wesley, Reading, MA, USA.

Hartnell, B.L., Rall, D.F., 1994, *Bounds on the Bondage Number of a Graph*, Discrete Mathematics, vol. 128, no. 1, s. 173–177.

Haynes, T.W., Hedetniemi, S., Slater, P.J, 1998, *Fundamentals of Domination in Graphs*, CRC Press, New York.

https://www.hackerearth.com/practice/algorithms/graphs/minimum-cost-maximum-flow/tutorial/ (access 30.10.2018).

Kołowrocki, K., Soszyńska-Budny, J. (2011), *Reliability and Safety of Complex Technical Systems and Processes. Modeling – Identification – Prediction – Optimization,* Springer-Verlag, London.

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, Gdańsk-Sopot, vol. 3, no. 1–2, s. 263–279,

Leeuwen van, J., 1990, *Graph Algorithms,* w: Leeuwen van, J. (ed.), *Handbook of Theoretical Computer Science,* vol. A, Elsevier Science Publisher B.V., Amsterdam, s. 525–631.

Martello, S., Toth, P., 1990, *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, Chichester, U.K.

Neumann, T., 2016, *The Shortest Path Problem with Uncertain Information in Transport Networks*, w: Mikulski J. (ed.), *Challenge of Transport Telematics*, Springer International Publishing, s. 475–486.

Newell, G.F., 1980, *Traffic Flow on Transportation Networks,* MIT Press Series in transportation studies*,* Monograph 5.

Parekh, A.K., 1991, *Analysis of Greedy Heuristic for Finding Small Dominating Sets in Graphs*, Information Processing Letters, vol. 39, no. 5, s. 237–240.

Rodrigue, J.P., Comtois, C., Slack, B., 2017, *The Geography of Transport Systems,* 4th ed., Routledge, Taylor & Francis Group, New York.

Syslo, M., 2011, *Źródło programu "Algorytmy grafowe",* http://mmsyslo.pl/Materialy/Oprogramowanie /Algorytmy-grafowe (dostęp 30.10.2018).

Velammal, S., 1997, *Studies in Graph Theory: Covering, Independence*, *Domination and Related Topics. Ph.D. Thesis,* Manonmaniam Sundaranar University, Tirunelveli, India.

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., s. 257–271.