Application of Graphs and Networks Similarity Measures for Analyzing Complex Networks

C. BARTOSIAK, R. KASPRZYK, Z. TARAPATA
cezary.bartosiak@gmail.com

Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

In the paper we focus on the research of graphs and networks similarity measures for analyzing complex networks. This kind of researches has a very wide range of applications in the military and civilian domains and tasks such as: law enforcement, criminal investigation, counter-terrorism as well as algorithms used in web search engines, analysis of bio-systems or chemical compounds and many others. Using a tool, which we have implemented, we show an experimental analysis of an airlines network. Afterwards we present opportunities of making use of our methods and tool for analyzing real systems which can be modelled using graphs and network models.

Keywords: graph similarity measures, complex networks, gephi.

1. Introduction

Link analysis is a technique revealing complex patterns by linking together values from datasets. This kind of analysis, also called network analysis, concentrates on the associations between entities such as people, places, organizations and so on, not just on entities themselves. Network analysis can be used to clarify and explain data revealing otherwise hidden patterns thanks to very powerful measures based on network structures. The ultimate purpose of the network analysis is the development of methods to visualize associations between entities. These methods are used to demystify data and reveal otherwise hidden patterns leveraging human capabilities to make sense of completely abstract information.

Network analysis has been used in a wide variety of military and civilian domains and tasks such as:

- law enforcement, criminal investigation, and counter-terrorism [1], [9]
- development of effective organization structures and communication networks
- algorithms used in web search engines
- construction of optimal marketing strategies by choosing the right people to spread information [12]
- building effective vaccination strategies thanks to so-called ”super-spreaders” identification [10], [11], [13], [17]
- analysis of bio-systems or chemical compounds [8], [16] and many others.

The methods for measuring graphs and networks similarity and methods for analysis of networks have a common area of applications in many cases. Thus, it is not surprising that some dependencies between them exist. To better understand why these methods are useful, it is worthwhile to show some utilization. The obvious field of applications is computer science. Pattern recognition and computer vision are the most interesting. It leads to, for example, optical character recognition or biometric identification. In chemistry it is possible to model chemical compounds as graphs. This enables to automate identification of isomers, researches of planarity of molecules, etc. Biologists are interested in intelligent analysis of existing data. Over the last twenty years the development of many breakthrough technologies has allowed researchers to study the activating and inhibiting relationships between biological components. Combined with graphs and networks theory it has enabled to study, for example, protein-protein interactions of any species more efficiently.

The goal of the paper is to explain how we can utilize dependencies between graphs and networks similarity measures and complex networks analysis. The paper is organized as follows:

In section 2, we recall some basic definitions and notations.

In section 3, we talk about the most popular methods for analysis of complex networks.

In section 4, we describe methods for measuring graphs and networks similarity. We

also discuss dependencies between methods presented in the section 3 and ones from this section.

In section 5, we focus on tools used in our work and describe one we have constructed for our researches.

In section 6 we give an experimental analysis using the tool we have constructed.

The paper is closed by a short summary.

2. Definitions and Notations

Networks are commonly modelled by means of simple or directed graphs that consist of sets of nodes representing objects under investigation, joined together in pairs by links if the corresponding nodes are related by some kind of relationship. We focus only on simple graph definition and denote it as graph.

Formally, a graph is a vector $G=\langle V,E\rangle$ where: V, E are sets of graph's vertices and edges, respectively $E\subset\{\{v, v'\}: v, v'\in V\}$. Additionally let's denote $n = |V|$, $m = |E|$. Networks are graphs with values on nodes and edges [6]. So in some cases the use of a graph to represent networks does not provide a complete description of systems under investigation. For instance, if contacts in social networks are represented as a graph, we only know whether individuals are connected, but we cannot model the strength of these connections. However, for further consideration, we use only a formal graph definition.

The path is an alternating sequence of vertices and edges, starting in vertex v_i and ending in vertex v_j. The length of a path is defined as the number of links in it and d_{ij} is the shortest path length. Now we can define diameter D as the longest shortest path i.e. $\max\{d_{ij}\}$. Networks very often are represented by a matrix **A** called adjacency matrix, which in the simplest case is a n x n symmetric matrix. The element A_{ij} of adjacency matrix equals 1 if there is an edge between vertices i and j, and 0 otherwise. The first-neighbourhood of a vertex v_i, denoted by $\Gamma_1(v_i)$, is defined as set of vertices immediately connected with v_i, i.e.:

$$\Gamma_1(v_i) = \{v_j \in V: \{v_i, v_j\} \in E\}.$$

The number of existing edges between the first-neighbourhood of a vertex v_i is denoted by:

$$N(v_i)=|\{v_l, v_k\}: v_l, v_k \in \Gamma_1(v_i) \wedge \{v_l, v_k\} \in E|.$$

The degree k_i of a vertex v_i is the number of first neighbours and $k_i = |\Gamma_1(v_i)|$.

Very important concept is the local clustering coefficient C_i [17] for a vertex v_i which is then given by the proportion of $N(v_i)$ divided by the number of edges that could possible exist between first-neighbourhood of a vertex v_i (every neighbour of v_i is connected to every other neighbour of v_i). Formally $C_i = \frac{2N(v_i)}{(k_i-1)\cdot k_i}$. The clustering coefficient C for the whole network is defined as the average of C_i over all $v_i \in V$.

The degree distribution $P(k)$ of a network is defined as the fraction of nodes in the network with degree k. Formally $P(k) = \frac{n_k}{n}$, where: n_k is the number of nodes with degree k; n is the total number of nodes. Most of the real networks are found to have degree distributions that approximately follow a power law: $P(k) \sim k^{-\gamma}$, where γ is a constant.

Identifying and measuring properties of real networks is a first step towards understanding their topology, structure and dynamics. The next step is to develop a mathematical model, which typically takes a form of an algorithm for generating networks with the same statistical properties. Apparently, networks derived from real data (most often are spontaneously growing) have a low average d_{ij}, power law degree distributions ($P(k) \sim k^{-\gamma}$, where γ is a constant), occurrence of hubs (nodes with much higher degrees than the average node degree), tendency to form clusters and many other interesting features. Three very interesting models, which capture these features, have been introduced recently: *Random Graph, Small World* and *Scale Free*. Figure 2.1 presents networks generated using these models.

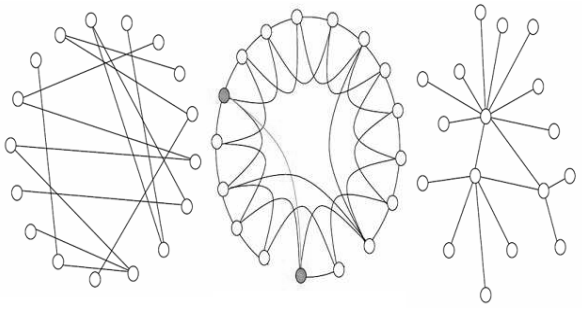

Fig. 2.1. From the left – a random network, a *Small World* network and a *Scale Free* network

The leftmost picture shows Erdős' and Rényi's model of the network presented in [7]. It is unfortunately an inaccurate model of real networks due to the lack of features that the remaining two models have. The middle picture shows an example of Watts' and Strogatz's

Small World network [17]. It is characterized by a high clustering coefficient and a small average shortest path length. A graph is considered small-world if C is significantly higher than C of a random graph constructed on the same vertex set, and if the graph has got approximately the same average shortest path length as its corresponding random graph. The rightmost picture shows an example of Barabási's and Albert's *Scale Free* network [2] which has some additional values in comparison with networks generated using the *Small World* model. It is characterized by a distribution degree that follows a power law. It has been the most accurate model since many empirically observed networks appear to be *Scale Free*, including social networks, Internet, WWW, citation networks, bionetworks, etc.

3. Methods for Analysis of Complex Networks

Centrality measures are the most basic and frequently used methods of analysis of complex networks. They address the question of "What is the most important or central node in a given network?" There is no one measure that is suitable for all applications. We considered five most important centrality measures e.g.:

- *degree centrality*

Gives the highest score of influence to the vertex with the largest number of first-neighbours. The degree centrality is traditionally defined as the degree of a vertex, normalized over the maximum number of neighbours this vertex could have:

$$dc_i = \frac{k_i}{n-1}. \tag{3.1}$$

- *radius centrality*

Chooses the vertex with the smallest value of the longest shortest path starting in each vertex. So if we need to find the most influential node for the most remote nodes it is quite natural and easy to use this measure:

$$rc_i = \frac{1}{\max\limits_{j \in V} d_{ij}} \tag{3.2}$$

- *closeness centrality*

Focuses on the idea of communications between different vertices and the vertex, which is "closer" to all vertices gets the highest score:

$$cc_i = \frac{n-1}{\sum\limits_{j \in V} d_{ij}} \tag{3.3}$$

- *betweenness centrality*

It can be defined as the percent of shortest paths connecting any two vertices that pass through the considered vertex. If $p_{lk}(i)$ is the set of all shortest paths between vertices v_l and v_k passing through vertex v_i and p_{lk} is the set of all shortest paths between vertices v_l and v_k then:

$$bc_i = \frac{2\sum_{l<k} \frac{p_{lk}(i)}{p_{lk}}}{(n-1)\cdot(n-2)}. \tag{3.4}$$

- *eigenvector centrality*

Where degree centrality gives a simple count of the number of connection a vertex has, eigenvector centrality acknowledges that not all connections are equal If we denote the centrality of vertex v_i by ec_i, then we can allow for this effect by making ec_i proportional to the centralities of the v_i's first-neighbours.

$$\begin{gathered} ec_i = \frac{1}{\lambda}\sum_{j=1}^{n} A_{ij} ec_j \\ \Downarrow \\ \vec{ec} = \frac{1}{\lambda} A \vec{ec} \end{gathered} \tag{3.5}$$

So we have $A\vec{ec} - \lambda I \vec{ec} = 0$ and the λ value we can calculate using $\det(A - \lambda I) = 0$. Hence, we see that $\vec{ec}$ is an eigenvector of adjacency matrix with eigenvalue λ.

4. Methods for Measuring Graphs and Networks Similarity

There are many kinds of similarities one may be interested in. The first one is *graph isomorphism*. If two graphs are isomorphic, then they are structurally indistinguishable. Formally two graphs are isomorphic if a bijective function exists between the sets of nodes such that two nodes are connected in one graph, if and only if, their images under the bijection are connected.

Another method is *edit distance*. It is the minimum number of edit operations (node and edge additions and/or removals) required to transform one graph into the other. It is worth noticing that this method is a generalization of isomorphism – two graphs are isomorphic if their edit distance equals zero. Usually different modifications (edit operations) are related to different costs. The cost means how likely the edit is to occur. Then the edit distance problem can be considered as an optimization problem: determine a minimum cost set of modifications to transform one graph into another.

There exists a wide range of similarity algorithms based on the *iterative approach*. The idea is very simple i.e. two nodes of two different graphs are considered similar if the

neighbouring nodes are similar. (see [5] for details).

The last one is the *quantitative nodes similarity* method (this approach is also applicable to edges) which is described in details in [14]. The author do a few calculations. In brief he create a *vector* of matrices describing similarities between nodes (from graph G_A to graph G_B) from different nodes' functions points of view. In our case, these functions are nodes' clustering coefficients or/and centrality measures. Next we normalize these matrices and transform them into single similarity matrix $S(G_A, G_B)$ between nodes of graphs as follows:

$$S(G_A, G_B) = \left[s_{ij} \right]_{n_B \times n_A} \tag{4.1}$$

Having got this matrix it is possible to formulate and solve optimal assignment problem to find the best allocation matrix $X = \left[x_{ij} \right]_{n_B \times n_A}$ of nodes from graph G_A and G_B (see [3], [14], [15] for details). The value of optimal assignment (after some kind of normalization of course) is the measure of similarity between these graphs.

This method is very important because it is possible to use measures for analysis Complex Networks in it (as functions). This is a field where methods for analysis of Complex Networks and methods for measuring graphs and networks similarity depend on each other and we show how this combination can be utilized [3].

5. Functionality and the Architecture of the Constructed Tool for Network Analysis

There is a large number of potential applications that may use network analysis, but currently there are many tools available that deal with network analysis, from the generic to the more specialized and domain-specific ones. Therefore, the choice of a tool is difficult and time-consuming. There are no objective criteria and measures to support the choice.

The Internet-based survey allowed identifying near one hundred tools for network analysis including those for network data visualization. We looked for a software toolkit with utilities available for programming in a common programming language like Java. According to our survey three tools are particularly interesting.

The Java Universal Network/Graph (**JUNG**) is an open source framework that provides a common and extendible language for modelling, analysis, and visualization of data that can be represented as a graph or network. JUNG includes algorithms for statistical analysis, random graph generation, calculating of networks distances, flows and other importance measures. It also provides a visualization framework to construct a tool for visual data exploration.

yFiles is the commercial package for Java and .NET platforms which provides efficient visualization of algorithms. There are a lot of classes for viewing, editing, layouting, and animating networks in the library. Since it is written in Java, yFiles is fit for platform independent applications. It has a graph viewer and supports many functionalities, such as labels for nodes and edges or multiple views of graph. Furthermore, yFiles has some routines exploration and descriptive analysis of networks.

Finally, **Gephi** [4], [19] is open source software for exploring and manipulating graphs and networks. A very flexible and multi-task architecture based on the *NetBeans* platform brings new possibilities to work with complex network data sets and produce curio visual results. Gephi uses a 3D render engine to display large networks in real-time and to significantly speed up the exploration. The GUI consists of workspaces, where users can perform separate activities. Most of the efforts were made to achieve great extensibility of the software. New algorithms can be very easily added to the Gephi software. Sets of nodes and edges can be obtained manually or by using the filters. Because the power of Network Analysis often comes from the ability to assess the values of the position of nodes in the structure of a network, ordering and clustering can be processed according to these values. Graphical modules take a set of nodes as an input and modify the display parameters, like colours or size, to help understand the network structure or content. The current studies of network analysis often touch up the dynamic of networks. Dynamic Networks Analysis offer possibilities to understand the structure transition or diffusion on a network-like virus or information propagation. What is particularly interesting and essential is an exploration of dynamic networks in an easy and intuitive way. It has been incorporated in Gephi since the early beginning. The architecture supports graphs with a varying structure or content.

The final choice of a framework was an uneasy task. Selecting an inappropriate framework may result in time-consuming

implementation or even prohibit performing some kind of analysis in the implemented tool. We decided on Gephi as the most powerful and promising framework for network analysis.

The tool we have built during our researches has been implemented as a set of plugins to the Gephi. We have complied with several principles concerning development of systems that are easy to maintain and extend over time, i.e.: our software is open for extensions and closed for modifications; it is a cross-platform solution; it assures interoperability. We have been able to achieve that thanks to the *Model-View-Controller* (MVC) and *Service Locator* patterns. The first one is a very powerful design pattern, which isolates ”domain logic” (business data) from ”UI logic” (input and presentation layers), permitting independent development, testing and maintenance of each one.

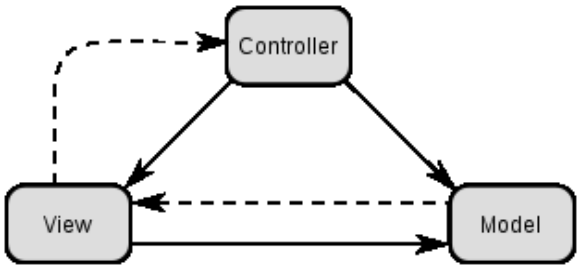

Fig. 5.1. Model-View-Controller pattern

The model represents application data and functions that operate on them. It also informs the view about its state changes, provides business data and gives the controller an access to data. The view propagates user’s demands to the controller and presents the model for him/her. The controller ”translates” user’s actions and propagates them to the model. It also chooses a view to present data basing on the user’s actions, parameters, and results of the model’s data processing.

Service Locator is an implementation of the *Inversion of Control* pattern. It is a technique that allows removing dependencies from the code. We have also used a *Factory* pattern to implement IoC.

Such an organized architecture allows us to develop the code according to SOLID design patterns [18] (abbreviation for *Single responsibility*, *Open-closed*, *Liskov substitution*, *Interface segregation*, *Dependency inversion*) that is five basic patterns of object-oriented programming and design. It makes the code very extensible and scalable.

6. An Experimental Analysis

In this section we show an experimental analysis using a tool, which we have created during our researches.

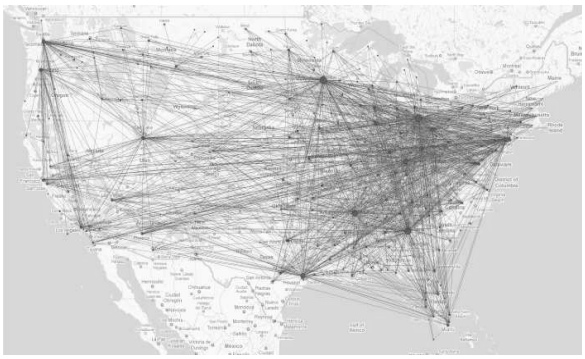

Fig. 6.1. An airlines network in the U.S.

Figure 6.1 shows an airlines network in the U.S. Figure 6.2 presents a random airlines network made using the same set of nodes. It has the same number of edges, but nodes are randomly connected.

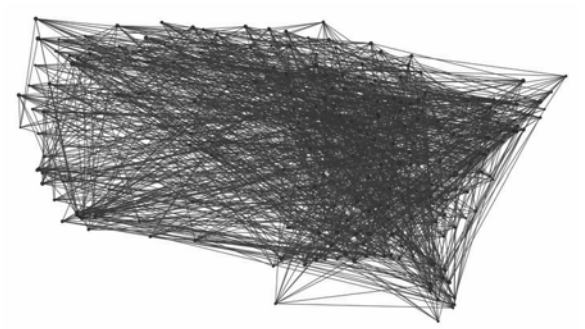

Fig. 6.2. A random airlines network

Both networks have got a diameter equal to four and mean-shortest path length is equal to 2.32 for the real one and 2.53 for the random one.

Figure 6.3 presents the degree distributions for both networks. As we can see, the degree distribution for the real network follows a power law, while the second distribution is rather far from that. It is a result of *hub and spoke* strategy used by the U.S. airlines, what creates a *Scale Free* characteristic of this network. Such networks are resistant to various events that could make some subset of airports non-functional (for instance as a result of terrorism).

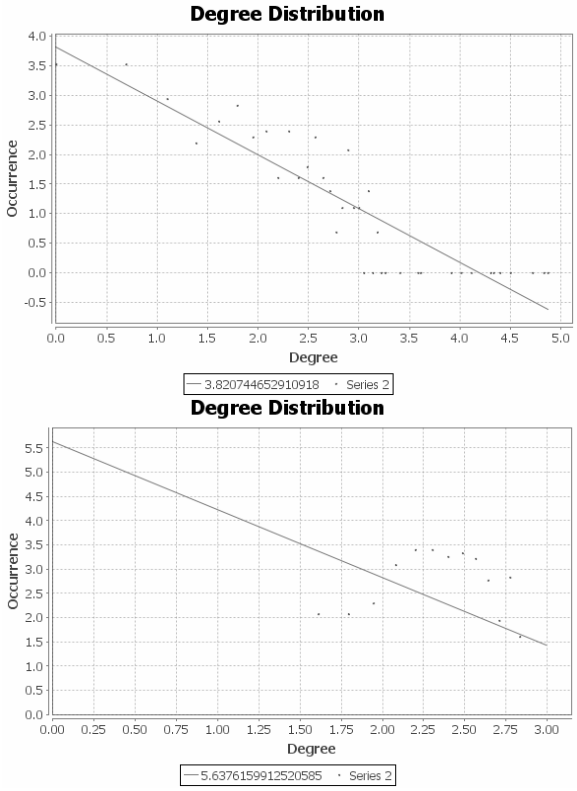

Fig. 6.3. Degree distribution for both networks (top – the airlines network, bottom – the random network)

We also calculate a clustering coefficient and check the similarity of both networks taking this measure into consideration. The x-axis on the chart in Fig 6.4 presents networks (so we also calculate self-similarity) and the y-axis presents values of quantitative nodes similarity measure – the lesser the value, the more similar the networks are.

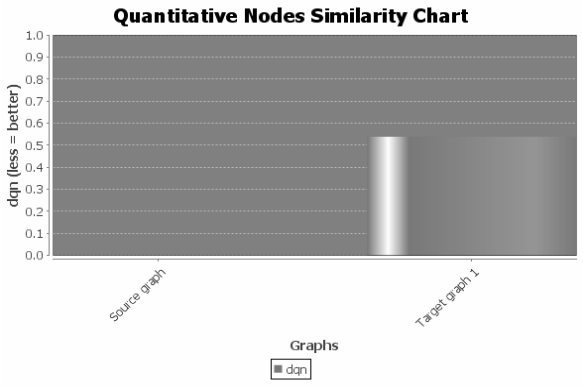

Fig. 6.4. Networks similarity report chart from the clustering coefficient point of view

As we can see the networks are not very similar from the clustering coefficient point of view. Combining this with the fact of very similar mean-shortest path length, we can say that the airlines network has a *Small World* characteristic, which means that thanks to the small number of long-range connections, this network is very efficient from the communication point of view.

7. Summary

Presented ideas can be used to analyze the dynamics of evolving systems which can be modelled using a sequence of graphs or networks. Nodes and edges of such graphs or networks are described by so-called dynamic attributes. Such an attribute, from the implementation point of view, is an interval tree, which contains time intervals that include related values. By means of such an attribute it is possible to describe, for instance, how a degree of some nodes change: $\langle [1,2],0\rangle, \langle (2,3],5\rangle, \langle (4,10),6\rangle$, etc. Dynamic graphs or networks can be queried in order to get a "snapshot", i.e. "static" graphs or networks for particular time intervals. In this way dynamic attributes can be transformed into their static equivalents (the way of setting values of attributes for demanded time intervals can be customized, for example: it could be defined that for overlapped intervals average should be the estimator). Having such snapshots it is possible to create vectors, which are related to concrete nodes and then the nodes similarity method can be used to identify some abnormal states of the system (e.g. increase of terrorism activity).

There are many other possible applications because of the fact that social networks have become a huge topic in the last couple of years. Surely we can expect many wonderful discoveries in this field in the near future.

8. Bibliography

[1] R. Antkiewicz, M. Chmielewski, R. Kasprzyk, A. Najgebauer, Z. Tarapata, "The prediction of terrorist threat on basis of semantic associations and complex network evolution", *Proceedings of Military Communications and Information Systems Conference*, 2007.

[2] A.-L. Barabási, R. Albert, "Emergence of scaling in random networks", *Science*, Vol. 286, 509−512 (1999).

[3] C. Bartosiak, *Analiza związku między metodami wyznaczania podobieństwa grafów i sieci oraz analizy sieci złożonych*, praca inżynierska, promotor – Z. Tarapata, konsultant – R. Kasprzyk, Wydział Cybernetyki, Wojskowa Akademia Techniczna, Warszawa, 2009.

[4] M. Bastian, S. Heymann, M. Jacomy, ”Gephi: an open source software for exploring and manipulating networks”, *International AAAI Conference on Weblogs and Social Media*, 2009.
[5] V.D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P.V. Dooren, ”A measure of similarity between graph vertices applications to synonym extraction and web searching”, *SIAM Review*, Vol. 46 (4), 647-666 (2004).
[6] T. H. Cormen, C.E. Leiserson, R. L. Rivest, C. Stein, *Wprowadzenie do algorytmów*, Wydawnictwa Naukowo-Techniczne, Warszawa, 2004.
[7] P. Erdős, A. Rényi, ”On random graphs I”, *Publ. Math. Debrecen*, Vol. 6, 290−297 (1959).
[8] M. Hattori, Y. Okuno, S. Goto, M. Kanehisa, ”Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways”, *Journal of the American Chemical Society*, Vol. 125, 11853−11865 (2003).
[9] R. Kasprzyk, ”Fault and attack resistance of complex networks”, *X International Workshop for Candidates for a Doctor's Degree OWD*, Wisła, Poland, 2008.
[10] R. Kasprzyk, ”The vaccination against epidemic spreading in complex networks”, *Biuletyn Instytutu Systemów Informatycznych*, Vol. 3, 39−43 (2009).
[11] R. Kasprzyk, B. Lipiński, K. Wilkos, M. Wilkos, C. Bartosiak, ”CARE – Creative Application to Remedy Epidemics”, *Biuletyn Instytutu Systemów Informatycznych*, Vol. 3, 45−52 (2009).
[12] D. Kempe, J. M. Kleinberg, E. Tardos, ”Maximizing the spread of influence through a social network”, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.
[13] R. Pastor-Satorras, A. Vespignani, ”Epidemic spreading in scale-free networks”, *Phys. Rev. Lett.*, Vol. 86, 3200−3203 (2001).
[14] Z. Tarapata, ”Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem”, *Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics* (*MMAR'2007*), 1149−1155, Szczecin, Poland, 2007.
[15] Z. Tarapata, R. Kasprzyk: An application of multicriteria weighted graph similarity method to social networks analyzing, *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, 20−22.07.2009, Athens (Greece), IEEE Computer Society, 366−368, 2009.
[16] Y. Wang, F. Makedon, J. Ford, H. Huang, ”A bipartite graph matching framework for finding correspondences between structural elements in two proteins”, *Engineering in Medicine and Biology Society*, Vol. 2, 2972−2975 (2004).
[17] D.J. Watts, S. Strogatz, ”Collective dynamics of 'small-world' networks”, *Nature*, Vol. 393, 440−442 (1998).
[18] http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod
[19] http://gephi.org/

Wykorzystanie metod badania podobieństwa grafów i sieci do analizy sieci złożonych

C. BARTOSIAK, R. KASPRZYK, Z. TARAPATA

W artykule zaproponowano koncepcję wykorzystania metod badania podobieństwa grafów i sieci do analizy sieci złożonych. Omówiono podstawowe modele sieci złożonych oraz metody badania podobieństwa grafów i sieci. Następnie przedstawiono opis popularnych środowisk do analizy grafów i sieci oraz autorskie narzędzie do badania podobieństwa grafów i sieci. Przedstawiono praktyczny przykład wykorzystania zbudowanej aplikacji potwierdzający jej użyteczność w analizie sieci złożonych.

Słowa kluczowe: grafowe miary podobieństwa, sieci złożone, gephi.