Algorithm for the Transportation Network Nodes Aggregation Using Fuzzy Logic

Aleksander Król
The Silesian University of Technology, Poland

In many situations, the model of the transportation network contains a very large number of nodes, so the algorithms operating on such a model may be too time-consuming. Therefore there is a need to simplify the model by reducing the number of nodes. The simplest approach using the physical neighbourhood of the nodes and then aggregation of nearby nodes may be insufficient, because it does not take into account the different roles played by the nodes subject to the merger. Some of them have local significance and can be aggregated without disturbing the traffic flows across the whole network. For other nodes traffic flows associated with geographically distant nodes can be much larger than the local flows. Nodes of this kind should not be aggregated with their neighbours. This paper presents an algorithm for grouping the transportation network nodes using fuzzy logic, which processes the qualitative characteristics of nodes.

Keywords: transportation network, nodes aggregation, fuzzy logic.

1. INTRODUCTION

A detailed model of the real transportation network consists of a graph with a large number of vertices and edges. From a practical point of view, the network models can be divided with respect to the size into small (up to 100 vertices), medium (from 100 to 1000 vertices), and large (greater than 1000 vertices) [3]. In the case of medium and large models even simultaneous visualization of all vertices and edges is problematic. And even for small models, there may be problems with the analysis and the performance efficiency of some algorithms [2]. At the moment, an issue of great practical importance is the problem of logistics centres location. Most varieties of this problem is NP - hard class, which for the transportation networks with more than 100 vertices remain unresolved [5].

Thereby in many situations there is a need to simplify the model to facilitate the analysis of the phenomena occurring in the transportation networks [4] or for reducing processing time for algorithms operating on the model.

2. THE METHOD

In the simplest approach, in which only the information about the position of the transportation network nodes is taken into account, the traditional clustering algorithms can be applied. Then a group of nodes is replaced by the group centroid [6]. An obvious disadvantage of this method is the lack of use of the information about the transportation demands described by the traffic flows between pairs of nodes.

Many methods require the knowledge about the complete graph structure and in this way search for clusters of vertices that can be joined [1, 11]. The techniques used may be of general application in all areas, where the natural representation of the data are the graphs [8]. The presented work focuses on the use of the information contained in the origin - destination matrix which describes traffic flows between pairs of nodes. Such approach is motivated by the original purpose of the described algorithm - it will be the pre-processor for the procedure which applies the artificial intelligence methods to design transportation networks [9]. It must therefore be assumed that in the moment when the algorithm starts the transportation network topology is not yet known.

3. DESCRIPTION OF THE ALGORITHM

The algorithm for aggregation of the transportation network nodes, taking into account the traffic flows between pairs of nodes, is carried out in two stages:

- creation of the matrix of the nodes affinity,
- matching nodes into groups on the basis of the affinity value.

3.1. CREATION OF THE AFFINITY MATRIX

At the first stage the symmetric matrix of the nodes affinity |P| is calculated. The elements of the matrix take the values from the interval [0, 1]. The affinity value p_{ij} close to one suggests to join the *i*–th and *j*–th into one group, the value close to zero suggests the nodes should not be in the same group.

Determination of the affinity between a pair of nodes requires the calculation of the four quantities, which characterize the pair:

a. geographical distance, expressed as a fraction of the maximum extent of the studied area *d*),
b. total mutual traffic flow between this pair of nodes (p_w),
c. total individual traffic flows calculated independently for each node (p_i and p_j),
d. total external flow of the pair with all other nodes recalculated per single node (p_z).

It was assumed that the affinity value of the pair of nodes is determined by the following, descriptive conditions:

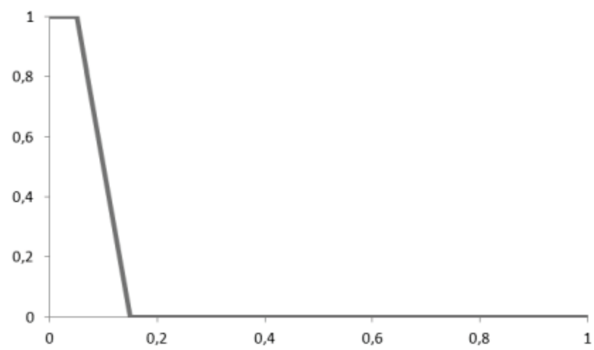
1. the nodes aspiring to take part in the same group should rather be close to each other and the mutual traffic flow should be more important than the external one (unless that individual traffic flows vary considerably, then the role of one of them can be skipped),
2. if the nodes are located quite far from each other, then they rather should not to be located in the same group (regardless of any other factors),
3. if the external flow for a pair of nodes is significantly more important than the mutual, and individual flows are similar, then the nodes rather should not be joined into one group (their individual roles from the perspective of the other nodes are important and should not be ignored).

These colloquial statements contain some imprecise expressions of relationship: rather close, more important, significantly more important. The mathematical formalism that allows for handling of such statements is provided by the fuzzy logic. In this approach, a binary logical expression, determining whether an item belongs or not to a set is replaced by a continuous function, that determines the degree of an element membership in a set. The membership function takes values from the range [0, 1], where 0 means that the element is certainly not in a set, and 1 means the total membership [10].

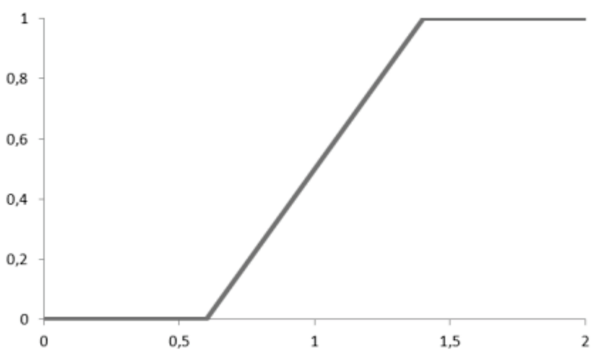
Basing on these concepts, a system of decision making, using descriptive terms, similar to those listed above can be build. The knowledge of such a system is contained in the fuzzy rules (implications) acting on fuzzy (imprecise) quantities. Typical fuzzy inference process consists of five steps:

1. fuzzyfication – for all the input parameters the degrees of membership are calculated with the use of given membership functions,
2. values calculation of the fuzzy logic expressions, which are the antecedents of the fuzzy rules,
3. execution of the fuzzy implications – the successor of the implications is always a fuzzy set, and as a result its membership function is "clipped" to the value calculated for the antecedents,
4. aggregation of all the successors of implications (using the maximum or sum)
5. sharpen – calculation of a single value, which is the result of the process.

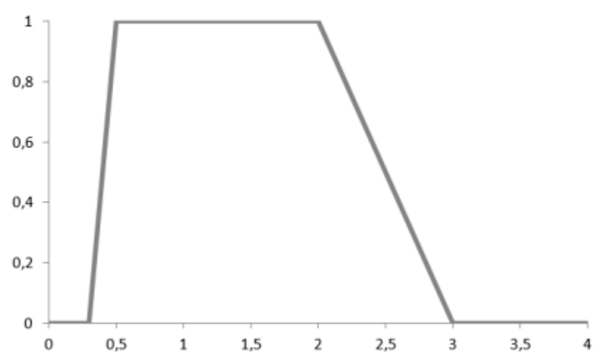
To the relationships established above the appropriate membership functions are assigned. They consist of simple line segments. The course of each of these functions has been chosen empirically. They are shown in Figure 1 (a – e).



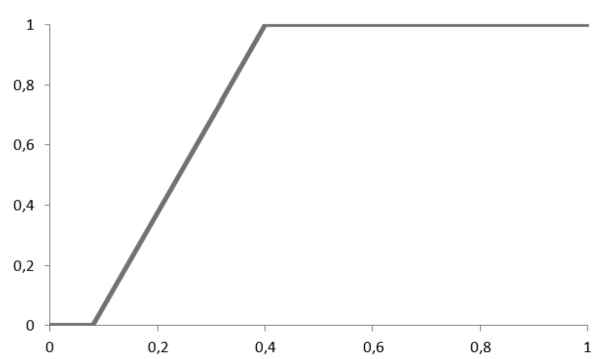
a) relationship “close” (CLS), on the x-axis: the distance of two nodes as a fraction of the extent of the area



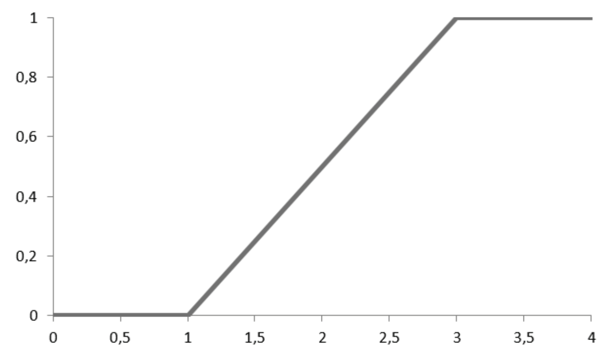
b) relationship “more important” (**MIMP**), on the x-axis: the ratio of the two arguments



c) relationship “similar” (**SML**), on the x-axis: the ratio of the two arguments



d) relationship “quite far” (**QFR**), on the x-axis: the distance of two nodes as a fraction of the extent of the area



e) relationship “much more important” (**MMINP**), on the x-axis: the ratio of two arguments

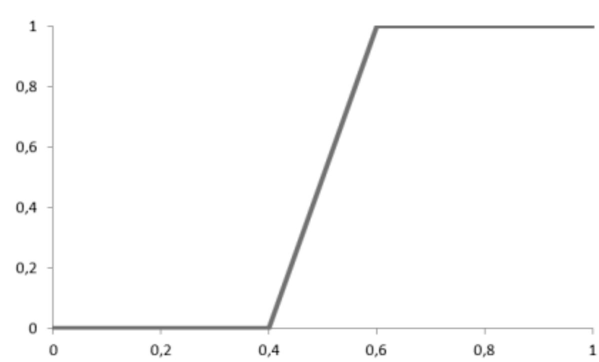
Fig. 1. Membership functions assigned to the described relationships.

Using the above described relationships the colloquial descriptive conditions can be expressed in the form of fuzzy rules:

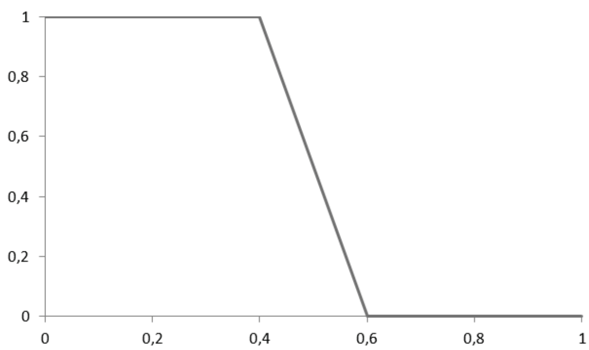
1. **if** CLS(d) **AND** (MIMP(p_w / p_z) **OR NOT** SML(p_i / p_j)) **then** rather group
2. **if** QFR(d) **then** rather not group,
3. **if** MMIMP(p_z / p_w) **AND** SML(p_i / p_j) **then** rather not group.

In the role of fuzzy operators of conjunction, alternative and negation, the functions of minimum, maximum and complement to unity are used as typical. As a result of the operators action the values are calculated for each of the antecedents of the implications.

As the successors of implications the terms "rather" and "rather not" are used. They correspond with the membership functions, which are shown in Figure 2 (a – b).



a) relationship “rather”, determines the degree to which an argument can be treated as "1"; on the x-axis: the value of the implication antecedent



b) relationship "rather not", the degree to which an argument can be treated as "0"; on the x-axis: the value of the implication antecedent

Fig. 2. Functions of the implications successors.

characteristics are summarized in Table 1 is examined.

Table 1. Characteristics of the sample pairs of nodes.

Parameter	**Symbol**	**Value**
Distance	d	0.13
individual flow of i – th node	p_{i}	302000
individual flow of j – th node	p_{i}	400000
mutual flow $i-j$	p_{w}	100000
external flow	p_{z}	77100
calculated affinity	$\boldsymbol{P_{\mathrm{ii}}}$	**0.56**

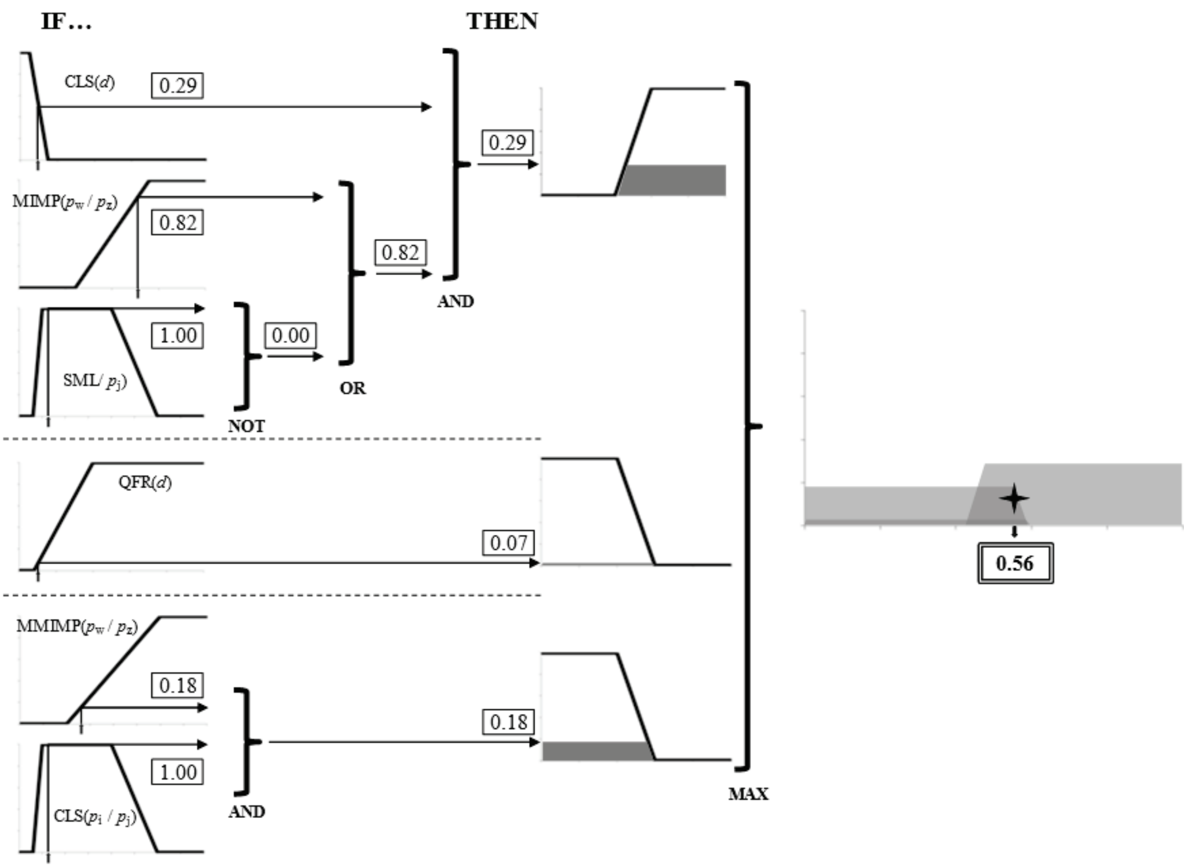


Fig. 3. An example of calculation of the affinity value for pair of nodes.

For the aggregation of the results of all three fuzzy implications the function of maximum is used. Then sharpen is made using the method of centre of gravity – the abscissa of the centre of gravity of the obtained by the aggregation figure is the value of the searched affinity between two nodes.

The above described process of the affinity calculation is illustrated in the example (Figure 3). The relationship of pair of nodes whose

3.2. AGGREGATION OF NODES

Having an affinity matrix it is possible to begin to collect the nodes into groups. Unfortunately, affinity defined as described above does not behave as distance, therefore a set of nodes with the affinity function is not a metric space. In this case, the typical clustering algorithms can't be used [7]. It is therefore proposed to use the original method, to join some of the nodes in the groups taking into account the value of the affinity of pair of nodes. The number of groups is not pre-established, so the described method has the advantage over the majority of classical methods.

The key structure is an array that assigns to each node a number of the group to which the node is attached. At the beginning of the aggregation algorithm the number of groups is zero, and the array is not filled out.

In each step of the algorithm for the first node not attached to any of the existing groups a new group is created. Then, a next yet free node with the highest affinity greater than a certain threshold is searched for. If such node is not found a singleton group is formed, which means that the node does not meet the criteria for including it in any group. Otherwise, the candidate to join the new group is found and the relationship between the candidate and the group is examined:

- according to the rules described another potential neighbour of the candidate is searched for,
- if searching fails the process of extending the group is terminated,
- if searching is successful, the affinity of the found node with all existing members of the group is examined,
- if the affinity is sufficient, the candidate is attached to the group and the testing procedure is called recursively for the neighbour of the candidate (the neighbour becomes the new candidate),
- if its affinity with any member of the group is smaller than the preset threshold, the process of extending the group is terminated, and depending on the result of the comparison of the affinity between the candidate and the group with the affinity between the candidate and the neighbour, the candidate is attached to the group or not.

The algorithm terminates when to each node is assigned a number of the group to which it belongs. The nodes belonging to each group are replaced with its focal point, the coordinates of which are weighted averages of the coordinates of the nodes group. The weights equal the total traffic flows for each node.

The above mentioned affinity threshold which determines the classification of a node as a potential neighbour is the only control parameter at this algorithm phase. Some tests were performed for a few selected values of this parameter and high algorithm resistance to its changes was found. Finally, the threshold was set at 0.5. The block diagram of the algorithm is shown in Figure 4.

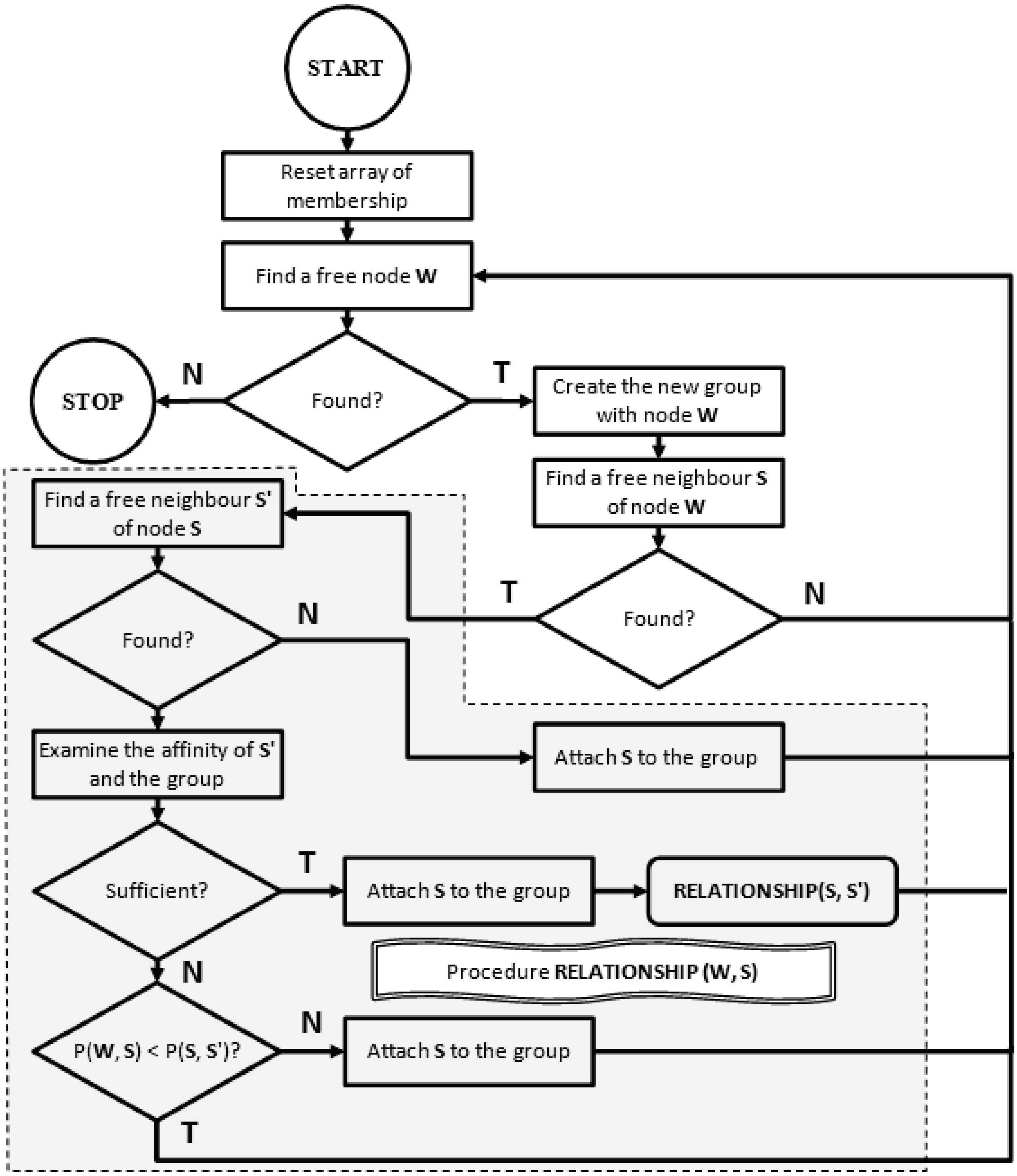


Fig. 4. Block diagram of the recursive procedure that identifies the group of nodes.

4. RESULTS

The input data are put in the text files containing the origin - destination matrix, which describes the traffic flows and then the coordinates of the transportation network nodes. The presented procedure was applied to several test data sets. The obtained results are presented in graphical form, which additionally shows traffic flows between pairs of nodes. Each such flow is shown as a gray band whose width is proportional to the total traffic volume between nodes. The focal point of each group is presented in the form of a circle linked to all the nodes of the group. Nodes that are not grouped with any of its neighbours form a singleton group, in which case the focal point overlaps the node.

First, the algorithm has been tested for the two sets of data with a small size (a network of 12 nodes). Data were designed purposely so that it is possible to group the nodes intuitively and then compare the result obtained by the algorithm (Fig. 5).

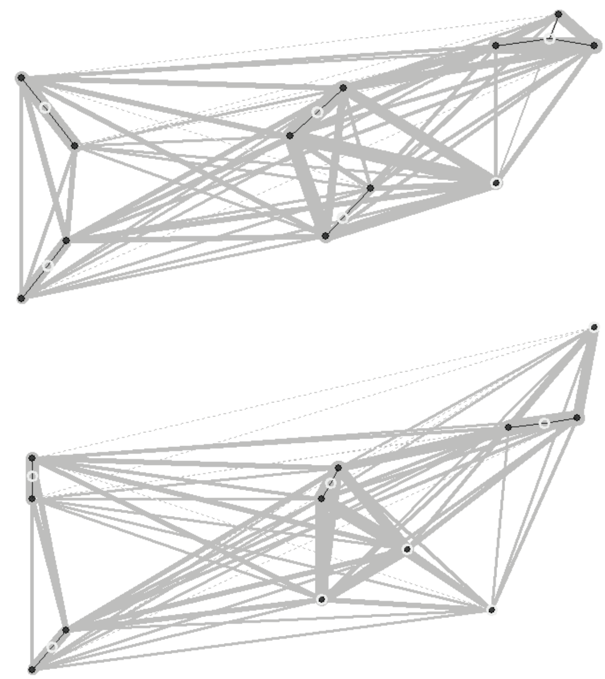


Fig. 5. The results for the two sets of data of small size.

In the next phase of tests the algorithm behaviour for the nearby nodes, but of global importance in the transportation network, was examined. Figure 6 shows such a situation - the node 9 is not aggregated with nodes 7 and 11, because in spite of the geographical proximity the traffic flows with the rest of the network nodes are much larger than the local flows.

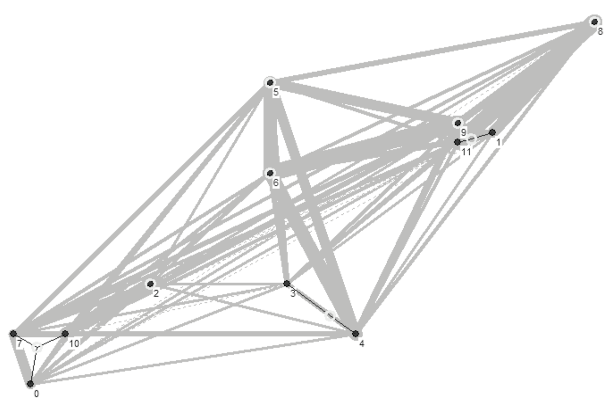


Fig. 6. The data contains the node of global importance.

In subsequent phases of tests, size of the data was increased - the networks of 24 and 100 nodes were used as inputs. For these cases the origin - destination matrices were generated randomly, with the assumption that the traffic flows for a pair of close nodes should be generally higher than for pairs of distant nodes. The results are shown in Figures 7 and 8. In the second case, due to the clarity, the visualization of traffic flows is abandoned.

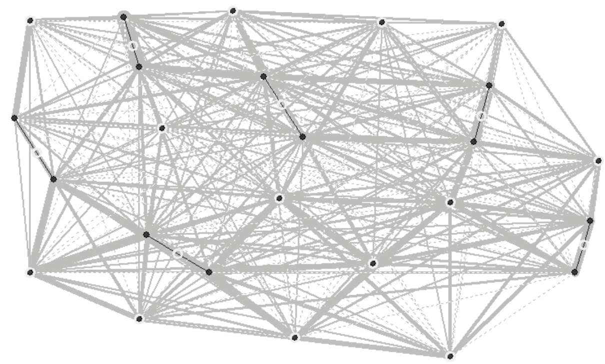


Fig. 7. Results for network of 25 nodes.

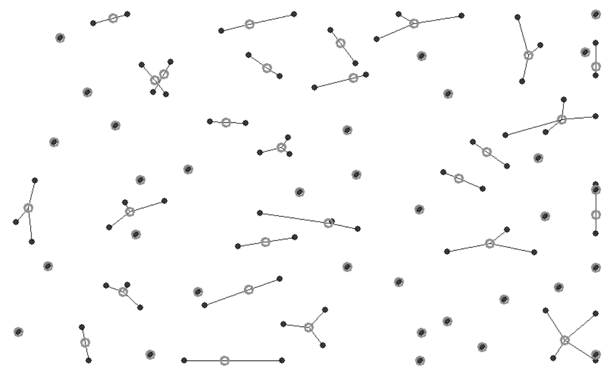


Fig. 8. Results for network of 100 nodes.

In one of subsequent tests the behaviour of the algorithm for non-standard data was examined - an abstract network of regular shape was given as input (Fig. 9). As expected, no multipiece group was obtained.

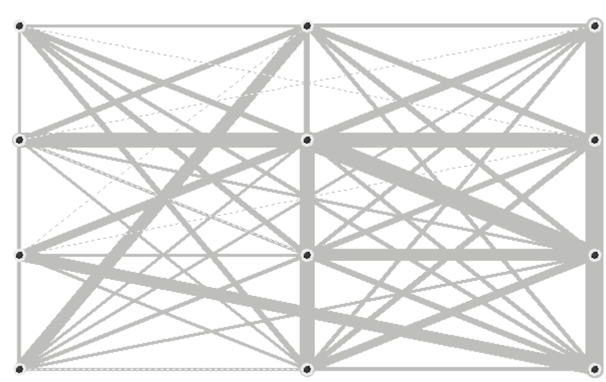


Fig. 9 Results for network with regularly spaced nodes.

The performance at all the examined cases was about fractions of a second, the most time-consuming procedure was the visualization of the results.

5. CONCLUSION

The algorithm presented at the paper can be applied in many studies where the subject is a complex transportation network. By reducing the number of nodes it is possible to significantly speed up the processing. Simplifying the network topology allows also for easier analysis and better understanding of the phenomena occurring in it.

Described ideas can also serve as a basis for the construction of an algorithm which determines the location of the network hubs.

REFERENCES

[1] Bauer D., Daigle J. N., Iliadis I., Scotton P.: Topology aggregation for combined additive and restrictive metrics, Computer Networks 50 2006, pp. 3284–3299.

[2] Bjorke J. T., Nilsen S., Varga M.: Visualization of network structure by the application of hypernodes, International Journal of Approximate Reasoning 51 2010, pp. 275–293.

[3] Börner K., Sanyal S., Vespignani A.: Network science, in: B. Cronin (Ed.), Annual Review of Information Science and Technology, vol. 41, Information Today, Inc./American Society for Information Science and Technology, Medford, NJ 2007, pp. 537–607.

[4] Dupuy G., Stransky V., Cities and highway networks in Europe, Journal of Transport Geography Vol. 4, No. 2 1996, pp. 107-121.

[5] Gavriliouk E. O.: Aggregation in hub location problems, Computers & Operations Research 36 2009, pp. 3136 – 3142.

[6] Hartigan J. A., Wong M. A.: A K-Means Clustering Algorithm, Applied Statistics, Vol. 28, No. 1 1979, pp. 100-108.

[7] Hoeppner F., Klawonn F., Kruse R., Runkler T.: Fuzzy cluster analysis. Methods for Classification, Data Analysis and Image Recognition, John Wiley & Sons, Chichester 1999.

[8] Kashan A. H. et al. : A particle swarm optimizer for grouping problems, Inform. Sci. 2013, in pront, www.dx.doi.org/10.1016/j.ins.2012.10.036.

[9] Król A., Pamuła T.: Using a genetic algorithm for the design of an optimal transport network, Probl. Transp. 2009 vol. 4 z. 4, pp. 107-113.

[10] Piegat A.: Modelowanie i sterowanie rozmyte, Akademicka Oficyna Wydawnicza EXIT, Warszawa 1999.

[11] Schaeffer S. E.: Graph clustering, Computer Science Review 1 2007, pp. 27 – 64.

Aleksander Król
The Silesian University of Technology, Poland
aleksander.krol@polsl.pl