Control and Cybernetics

vol. **47** (2018) No. 2

Properties of an α-clique approach to obtaining the hub and spoke structure in optimization of transportation systems*

by

Barbara Mażbic-Kulma, Jan W. Owsiński[1,], Jarosław Stańczak[1,], Aleksy Barski and Krzysztof Sęp[1,]

[1]Faculty of Computer Science, Warsaw School of Information Technology
ul. Newelska 6, 01-447 Warsaw, Poland
kulma@wit.edu.pl
[2]Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
{owsinski, stanczak, alex, sep}@ibspan.waw.pl

Abstract: The paper is devoted to the analysis of a graph transformation, pertinent for the transport and logistic systems and their planning and management. Namely, we consider, for a given graph, representing some existing transport or logistic system, its transformation to a (non-equivalent) so-called "hub-and-spoke" structure, known from both literature and practice of transportation and logistics. This structure is supposed to bring benefits in terms of functioning and economic performance of the respective systems. The transformation into the "hub-and-spoke" is not only non-equivalent (regarding the original graph of the system), but is also, in general, non unique. The structure sought is composed of two kinds of elements - nodes of the graph (stations, airports, havens, etc.), namely: the subgraph of hubs, which, in principle, ought to constitute a complete sub-graph (a clique), and the "spokes", i.e. the subsets of nodes, each of which is connected in the ultimate structure only with one of the hubs. The paper proposes a relaxation of the hub-and-spoke structure by allowing the hub subgraph not to be complete, but at least connected, with a definite "degree of completeness" (alpha), from where the name of "alpha-clique". It is shown how such structures can be obtained and what are the resulting benefits for various assumptions, regarding such structures. The benefits are measured here with travel times. The desired structures are sought with an evolutionary algorithm. It is shown on an academic example how the results vary and how the conclusions, relevant for practical purposes, can be drawn from such analyses, done with the methods here presented.

Keywords: transport, transport systems, graphs, hub and spoke, evolutionary algorithm, time-wise profitability

*Submitted: September 2018; Accepted: November 2018

1. Introduction

Optimization of transport systems constitutes an important challenge nowadays, both for the logistics and transport companies and for the specialists, cooperating with them. It is possible to build faster transport means, roads of bigger capacity etc., but this requires huge financial expenses. Improvement of the system without big investments sometimes is possible by changing only the operation mode. The *hub and spoke* structure of the transport system, which we consider in this paper, is one of such ideas (see O'Kelly, 1987; O'Kelly and Bryan, 2002). Hub and spoke is one of the structures, derived and possibly implemented for an existing transport system, which can be obtained by analyzing the corresponding graphs. Improvement of some of the important system parameters, or at least a simplification of the structure, leads to implementation of the *hub and spoke* approach. The *hub and spoke* is a class of graph structures having definite properties – described later on – which can, under certain conditions, be obtained from the initial, problem-defined, structure. Here we are not dealing with an equivalence between the initial and the *hub and spoke* structures, but with a transformation, which ought to preserve definite key properties of the initial structure, while securing a possibly significant improvement in other properties. The *hub and spoke* structure is widely used in telecommunications (Klincewicz, 1998), postal systems (Cetiner, Sepil and Süral, 2010), and also in transportation (Campbell and O'Kelly, 2012). The hub and spoke structure applied to a graph of connections is intended to resolve the problem of separating some tightly bounded structures in a graph, corresponding to some real system, especially regarding the transportation networks (Coyle, Bardi and Novack, 1994). We propose in the present paper to apply an evolutionary method to obtain the *hub and spoke* structure that will improve the functioning of the respective transport system.

There exist also other evolutionary approaches to the hub and spoke problem, see, e.g. Eghbali Zarch, Abedzadeh and Setak (2013). The application of the *hub and spoke* structure for a given graph of connections makes it possible to concentrate the flows of transported persons or goods. For the best results, the hub subgraph should be a clique. Yet, in practical cases it is generally often impossible to find such a clique in the original graph, and so we propose a relaxation of this condition and introduce the notion of an **α-clique** of vertices featuring cheap, fast, frequent or high-capacity connections (depending on the modeled transportation system), while the spoke vertices are connected only to the relevant hub vertices, see Fig. 1 for an instance. The idea of *hub and spoke* structure is meant to enable the elimination of many bilateral connections between majority of vertices. Instead, only connections among the hub vertices and the local connections between the individual hub vertices and their corresponding spoke vertices are required. Appropriate choice of several transit nodes and local connections, forming the *hub and spoke* structure, can improve the efficiency of the respective transport system, reducing costs and increasing service quality. The graph of connections, after concentration, when turned into the *hub*

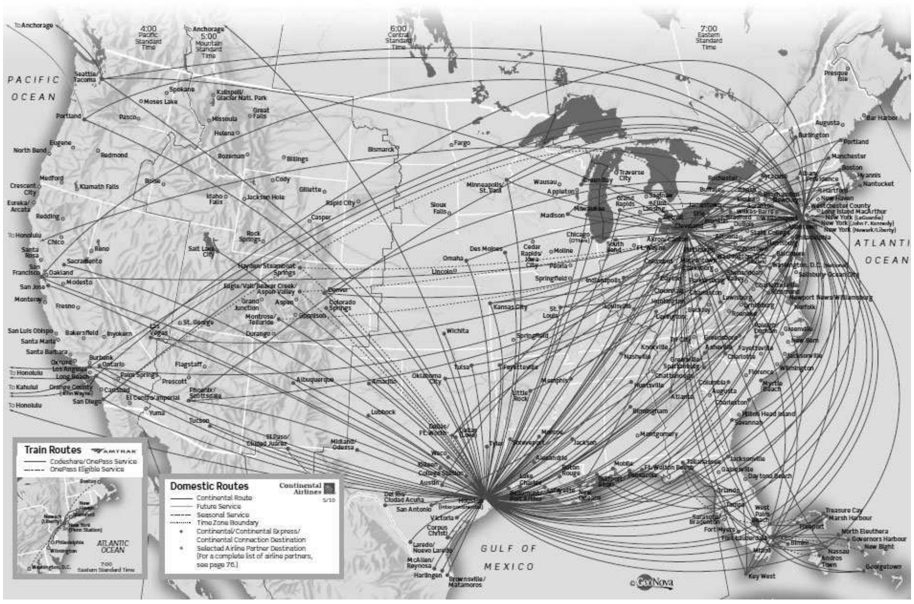

Figure 1. Main hubs in a route map for a major US airline. (Source: Continental Airlines Route Maps)

and spoke structure, reduces also the complexity of the management problem. The advantages of the properly designed hub-and-spoke transport structure may be as follows: more frequent connections among points, lower average times of journeys (including all kinds of waiting times), lower costs of transport, lower number of required transport means to service all connections (elimination of serious capacity slacks on low-demand connections, while maintaining connection frequencies), higher facility of synchronizing, timetabling, etc. with respect to local connections.

The *hub and spoke* method of graph transition is useful in the instances, where direct connections among spokes within one hub cluster are negligible and of importance are only long distance connections with other clusters (hubs and their spokes). In that case all local transfers (within the same cluster) between spokes are performed through the local hub nodes (no direct local connections among spokes of the same hub are necessary nor allowed).

It is possible to use several basic approaches to transform the unstructured graph of connections into the hub and spoke structure. In this paper we use two of them:

- the method, which searches for the minimum number of hubs, constituting at least a connected subgraph, with all remaining nodes connected to their hubs,
- the number of hubs is predetermined, with possibility of direct determi-

nation of hub nodes, while the (rest of) hubs and the spoke nodes are selected by the solving method (the number of hubs, predetermined in this approach, should, of course, be bigger than the minimum value, mentioned in the previous point).

A special case of the approaches mentioned is constituted by the method with the minimum number of hubs. It may be useful when creating communication hubs is very difficult or expensive. It then appears necessary to pick up as low number of hubs as possible, while preserving graph connectivity.

The hub and spoke method has been developed on the basis of O'Kelly (1987) and O'Kelly and Bryan (2002), where similar structures and their applications are described. Computational hardness of graph transformations and lack of efficient, dedicated algorithms, motivated us to use the evolutionary algorithm (Cowen, 1998). In our work, the evolutionary algorithm is responsible for the selection of the optimized configuration of spoke nodes attached to their communication hubs and the best candidates for hubs, if they are not predefined by the user. A short introduction to the evolutionary method and the discussion on the obtained results for the established conditions, regarding benefits in terms of time, after the transformation of the transportation system, are presented further in this paper.

2. Basic concepts

2.1. Preliminaries

The notions given below are based on Wilson (1996).

Thus, a ***graph*** is a pair G=(V, E), where V is a non-empty set of *vertices* and E is a set of *edges*. Each edge is represented by a pair of vertices $\{v_1, v_2\}$ with $v_1 \neq v_2$. Two vertices in a graph G=(V, E) are called ***incident*** if for v_i, $v_j \in V$ there is $\{v_i, v_j\} \in E$ or $v_i = v_j$. Each vertex is ***incident*** to itself. A *subgraph* of graph G=(V, E) is a graph G'=(V', E'), where $V' \subseteq V$, $V' \neq \varnothing$ and $E' \subseteq E$ such that for all $e \in E$ and $e = \{v_1, v_2\}$ if $v_1, v_2 \in V'$ then $e \in E'$. A ***degree*** of a vertex is the number of edges, to which this vertex belongs. A graph G=(V, E) is a ***complete graph***, if for each pair of vertices there is an edge $e \in E$ between them. A ***clique*** (a *complete subgraph*) Q=(V_q, E_q) in a graph G=(V, E) is a graph such that $V_q \subseteq V$ and $E_q \subseteq E$ and $Card(V_q)$=1 or each pair of vertices $v_1, v_2 \in V_q$ fulfils the condition $\{v_1, v_2\} \in E_q$ (Cormen et al., 2009). Each subgraph of a clique is a clique. A ***neigbourhood matrix*** of a graph G=(E, V) with $Card(V)$=n is a square binary matrix $n \times n$ with rows and columns corresponding to vertices. There is 1 in the a_{ij} cell of the neighborhood matrix if vertices v_i and v_j are connecetd, 0 in the opposite case.

2.2. An α-clique

Let:

- A=(V', E') be a subgraph of a graph G=(V, E), with $V' \subseteq V$, $E' \subseteq E$, k=$Card(V')$

- k_i be the number of vertices $v_j \in V'$ such that v_i, $v_j \in E'$.

1. For k=1 the subgraph A of graph G is an ***α-clique(α).***
2. For k>1 the subgraph A of graph G is an ***α-clique*** (α) if for all vertices $v_i \in V'$ the condition $\alpha = \frac{k_i+1}{k}$ is fulfilled, where $\alpha \in (0, 1]$.

In other words, an ***α-clique*** of a graph is its such subgraph that each vertex of this subgraph is connected with not less than the proportion α of the vertices of this subgraph. Further on, we will use the term of ***α-clique*** meaning ***α-clique(α)*** for an earlier established α. More information on ***α-clique*** and its properties can be found in Mażbic-Kulma et al. (2008) and Potrzebowski, Stańczak and Sęp (2007, 2008).

There is, however, another issue to explain, this issue being connected with the construction of a greedy algorithm for solving the possible maximum *α-clique*(γ) problem. Namely, a subgraph of an *α-clique*(γ) is not necessarily an *α-clique*(γ). This is due to the simple fact that $\frac{x+1}{x+2} \geqslant \frac{x}{x+1}$. As a result of this fact, a greedy approach, in which we would try to find an improved solution that is close to an already found one, by adding one or more new vertices to it, could fail.

Let *α-clique* A=(V', E') be a graph with α>0.5; thus, for all vertices v_i belonging to *α-clique*(α), k_i+1>0.5 k.

The set theory implies that if α>0.5, then for each pair of vertices the sets of vertices incident with them have a non-empty intersection, so an *α-clique*(α) with α>0.5 constitutes a connected graph. If $\alpha \leqslant 0.5$, the obtained subgraph may be disconnected.

Now, referring to some other similar proposals, in distinction from the *k-plex,* proposed by S. Seidman (1978), and considered in Pattillo, Youssef and Butenko (2013) as the *s-plex,* in the case of an *α-clique* the difference between the minimum vertex degree in an *α-clique* and the degree of a vertex in a corresponding complete graph (a clique) is changing according to the number of vertices in the graph. In an *α-clique,* the α parameter is constant, but the minimum degree is changing. The advantage of such an approach is some sort of constant graph structure, but there are also some inconveniences. The most important of them is the fact that not every subgraph of an *α-clique*(α) has to be an *α-clique*(α) for the same α.

According to Pattillo, Youssef and Butenko (2013) there are few other approaches to the clique relaxation. Let $d_g(v_1, v_2)$ be the shortest length of path between vertices v_1 and v_2, diam(G)=max $d_g(v, u)$ for all $(v, u) \in V$, δ (G) - the minimum degree of the vertices in G, $\kappa(G)$ - the minimum number of edges whose deletion yields

1. S is called s-clique if for all $v_1, v_2 \in V$ $d_g(v_1, v_2) \leqslant s$
2. S is called s-club if $diam(G) \leqslant s$
3. S is called s-plex if $\delta(S) \leqslant Card(S) - s$
4. S is called s-defective clique if S contains at least Card(S)(Card(S)-1)/2 edges
5. S is called k-core if $\delta(S) \geqslant k$.

The most relevant to our work is the (λ, γ) *quasi-clique*, introduced in Brunato, Hoos and Battiti (2008). Thus, S is a (λ, γ) if *quasi-clique* $\delta(G(S)) \geqslant \lambda(Card(S) - 1)$ and $\rho(G(S)) \geqslant \gamma$. $\delta(G(S))$ is the minimum vertex degree, $\rho(G(S))$ is the ratio of the number of edges to the total number of possible edges (Pattillo, Youssef and Butenko, 2013). The (λ, γ) *quasi-clique*, introduced in Brunato, Hoos and Battiti (2008) is a generalization of the α-*clique*. For $\gamma = 0$, the (λ, γ) *quasi-clique* is an α-*clique* with $\alpha = \lambda$.

3. The idea of generalized hub and spoke

A ***hub and spoke*** structure is a graph H_s=($G_h \cup G_s$, E) where the subset G_h is a fully connected subgraph (a clique) with the relevant subset of the set *E,* each vertex of the subset G_s has degree 1 and is connected exactly with one vertex from the subset G_h (thus forming a spoke), see O'Kelly (1987); Mażbic-Kulma et al. (2008). In the quite frequent case of the sparse graphs of connections, the requirement for the hub subgraph of being a clique in the hub and spoke structure, cannot be fulfilled. This very strong constraint, imposed on the final structure of the connection graph, makes it useless for some of the practical cases. Thus, we decided to weaken this constraint by introducing the earlier described α-clique instead of a clique as the hub subgraph. Of course, it would be better, if α were possibly close to 1 (shorter connections with maximum of two interchange nodes), but in the case of very sparse graphs it is admissible to reduce this requirement to just that of obtaining a connected graph of hubs. In the cases with very sparse graphs the transformation may, in general, turn out to be useless, but the connectedness of a graph constitutes quite a natural limitation to the possibility of the transformation.

A ***generalized hub and spoke*** structure is a graph $H_s = (G_h \cup G_s, E)$ where the subset G_h is at least a connected graph* with the relevant subset of the set *E,*, where each vertex of the subset G_s has degree 1 and is connected exactly with one vertex from the subset G_h (thus forming a spoke), see Potrzebowski, Sta nczak and Sęp (2008). We propose an evolutionary algorithm that transforms the connection graph into an instance of the generalized (and also the standard, when necessary) *hub and spoke* structure according to problem-specific restrictions. Depending on the problem requirements, some of the hub nodes or all of them may be imposed, and, on the top of this, the EA (evolutionary algorithm) method maximizes the strength of connections within the obtained α-clique of hubs and tries to derive structures with desired properties (like similar sizes of the derived clusters – hubs and their spokes).

This structure can be used in transport and logistic models, where direct connections between the spoke nodes, attached to respective hubs, are not very important and, thus, are not really necessary. The hub and spoke structure can be derived using one of two possible options. The first one uses a predetermined,

*The subgraph of hubs should be as close to a clique as possible (an instance of α-clique with α close to 1), but in the case of sparse input graph it should just be a connected graph to preserve its functionality.

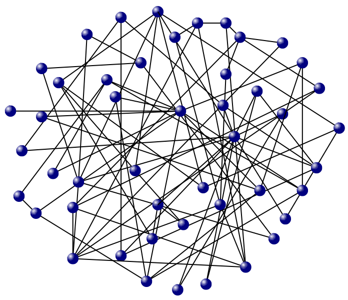

Figure 2. A sparse source graph

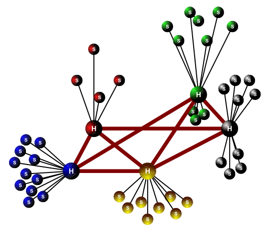

Figure 3. The generalized hub and spoke

by some expert, number of communication hubs, with the possibility of directly determining, which nodes should become hubs, or of selecting them by the solving method. In the second option the method tries to find the minimum number of hubs, which constitute at least a connected subgraph, with all the remaining nodes connected to their hubs. It must be noticed that the number of hubs used in the first option must be bigger than the minimum value, mentioned for the second option.

4. The method to find the hub and spoke structure for the given connection graph

4.1. The preliminaries

The evolutionary algorithms, considered to be the useful tools for solving difficult problems, are often used in graph problems, such as graph coloring, TSP, graph partitioning, maximum clique search, etc. (Chen, Wang and Okazaki, 2008; Marchiori, 1998; Talbi and Bessiere, 1991), because exact algorithms have too high computational complexity. It seems fully justified, then, to use the evolutionary algorithm here, in the considered graph transformation problem.

The standard version an evolutionary algorithm is shown in Algorithm 1. Nowadays, this common scheme is rather treated as a frame for building more efficient, specialized evolutionary methods (Michalewicz, 1996), often called *the memetic algorithms* (**ME**) (see Moscato, 1989). The evolutionary or memetic algorithms, specialised for the particular solved problem, require the introduction of efficient encoding of solutions, invention of specialized heuristic and random genetic operators and, finally, the fitness function.

Algorithm 1 The standard evolutionary algorithm

Require: Input data
Ensure: Output data
  Random initialization of the population of solutions.
  while stop condition is not satisfied **do**
    Reproduction and modification of solutions using genetic operators.
    Valuation of obtained solutions.
    Selection of individuals for the next generation.
  end while

The problem encoding, i.e. the representation of the individuals (EA population members, called also solutions or agents sometimes) depends on the solved problem. In the presented approach, the information on the transformed graph is stored in an array of data, describing all connections among the graph nodes (the neighborhood matrix). Each solution contains the variable length arrays of vertices (spokes) attached to their hub. Hubs are also stored in the variable length array of hubs. This method of encoding makes it easier and faster to modify the selected clusters and to evaluate the graph parameters in clusters, but also makes it difficult to perform crossover-like operations. In the here presented solution we decided to give up the use of the crossover operator, which, in the permutation-type problems, does not bring new solutions by building cells from other individuals, but only exchanges information about node attachments. Instead, we use several, also "intelligent", methods of solution transformation, which work efficiently and do not produce inadmissible solutions (which would be a typical effect of applying the crossover operator).

A method of selecting and executing the specialized operators in all iterations of the algorithm is required in order to apply these operators. It is assumed in the approach used in Sta nczak (2003) that an operator generating good results should be selected for use with a bigger probability and affect more frequently the population than the other ones. The method of computing the respective quality factors is based on reinforcement learning (Sutton and Barto, 1998) (as used in machine learning).

The fitness function in the EA is closely connected with the problem specific quality function, meant to evaluate the quality of solutions. The fitness function evaluates the members of the population. It is a modified (scaled, translated, etc.) problem quality function, prepared for computational use in the EA. The quality function is responsible for obtaining the proper graph structure. In the considered problems, the quality functions are usually the heuristic formulae, obtained on the basis of experiments. It is common that they contain a penalty part for the potential invalid or improper structure of the obtained solutions.

4.2. The evolutionary method to uncover the *hub and spoke* structure in the transportation system

As already mentioned, the hub nodes can be explicitly assigned or only their number may be imposed and then specific nodes are selected by the EA, according to connections and weights of the transformed graph. As it was previously indicated, the subgraph of *hubs* is an α-clique with as big value of α as possible – ideally, hubs should constitute a complete subgraph, but when connections between nodes are very sparse or are determined as existing junction nodes (for instance: railway stations), it is admitted that the subgraph of hubs constitute simply a connected graph. The *spokes* constitute groups of nodes connected only with their hubs.

The representation of a member of the population contains: the table of selected by EA or imposed hubs with lists of attached spokes, the vector of real numbers, describing the knowledge, related to genetic operators, and the index number of the operator chosen to modify the solution in the current iteration.

For the *hub and spoke* structure with predetermined number of hub nodes the quality function promotes solutions where a rather small subgraph of hubs is (almost) fully connected and the sets of spokes attached to their hubs have medium sizes:

$$\max Q = \frac{1}{m} \sum_{i=1}^{n} \left(k_i - \left| \frac{k-n}{n} - k_i \right| + \frac{h_i}{n} \right) \tag{1}$$

n – predetermined number of hubs in the solution, m – number of connected subgraphs in hub subgraph, k_i – number of nodes (spokes) attached to the i^{th} hub, k – number of nodes in the whole graph, h_i – number of connections between hub i and other hubs.

The fitness function (1) promotes the spoke subgraphs, ideally of the size almost equal to the average number of spokes in hub subgraph (in the case of k_i equal $(k-n)/n$ - average number of spokes, the fragment $|(k-n)/n - k_i|$ of the formula (1) is equal 0), assuring connectivity of hub subgraph ($1/m$ – for connected subgraph m should be 1), maximizing the number of connections among hubs (h_i/n) - ideally equal 1.

In this case, the set of genetic operators consists of: *mutation* – exchange of randomly chosen nodes in different sets of spokes, *relocation* of a randomly chosen node to a different set of spokes, and *exchange* of a randomly selected hub for a randomly selected spoke – this operator is inactive when the hub nodes are explicitly assigned. When one node (spoke) is to be moved to the spoke cluster associated with another hub as a result of action of one of genetic operators, it must first be checked whether this node is connected with the newly considered hub. If it is not connected, the operation is canceled and the solution is not modified, because we do not allow the EA to create inadmissible solutions. We also allow to repeat the selected operator a randomly established number of times to increase its efficiency - in this way we implicitly introduce the multiple versions of genetic operators.

A problem arises when the predetermined number of hub nodes is lower than the minimum value assuring that all spoke nodes are attached to their hub nodes. This problem can be solved in two ways. The first one allows the final result to contain unattached nodes. The second increases the number of hub nodes so as to obtain the connected graph of the transportation system. These methods can be implemented using modified forms of quality function (1) with the penalty part for unattached spoke nodes or additional hub nodes.

The *hub and spoke* structure with the minimum size of the hub subgraph is a special case of the *hub and spoke* structure. It is computationally more difficult to solve. The problem encoding is similar the case described above, but the optimized fitness function (2) is different:

$$\min Q = n \cdot m \tag{2}$$

n – number of hubs in the solution evaluated, m – number of connected subgraphs in hub subgraph.

Fitness function (2) promotes the smallest set of connected hub nodes with all spokes attached to their hubs. The genetic operators are in this case the same as in the previous case, but are supplemented by an additional one: *concatenation* – an attempt to concatenate two sets of spokes (this operator tries to minimize the number of the hub nodes).

5. Results of computer simulations

We used as testing examples graphs randomly generated by the yEd† application. The results presented here are obtained from the analysis of a graph with 16 vertices and 100 edges. The methods, described earlier, were used to obtain the transformation of the described test graph to the hub and spoke structure. As it turned out, the method, which finds the minimum number of hubs, yielded a solution with just one hub, which is beyond doubt the smallest number of hubs possible to cover the source graph. Other results come from the method with the given number of hubs. The results are presented in the following sections.

5.1. Results obtained for the minimum number of hub nodes

The evolutionary method that computes the minimum number of hubs is prepared mainly to finding the lower limit on the feasible number of hubs for the approach with the imposed number of hubs. The solution, obtained using this method, is rather useless for practical purposes, due to the possibility of appearance of long distances to spokes and overloading of the only, in the presented example, or, in general, rather small number of hubs.

†yEd is a freely available application for manipulating graphs.

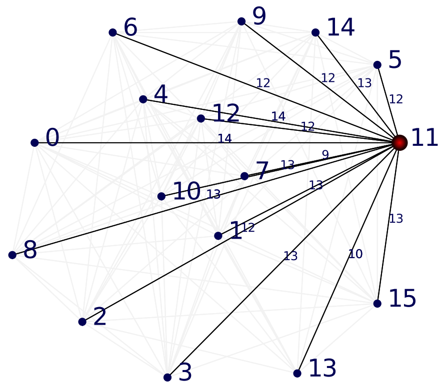

Figure 4. The minimum number of hubs found by EA

The hub and spoke structure obtained with the minimum number of hubs is shown in Fig. 4. As it can be seen, the discussed graph contains a vertex connected with all other vertices. Thus, the evolutionary algorithm has chosen this specific structure as the best solution, based on only one hub vertex (11^{th}) with the rest of nodes being the spokes. In this and the further figures, the bigger dots represent hub(s), the smaller represent spokes, the bigger numbers describe the numbers of vertices, while the smaller represent the flows between nodes (assuming that between every pair of connected nodes in the input graph there is a transfer of one unit of some good per time unit). The thicker lines represent connections between hubs, the thinner ones represent connections between hubs and spokes, the grayed ones being the unused edges of the input graph.

5.2. The problem with the imposed number of the hub nodes

The considered graph with 16 vertices is rather dense, as it contains 100 edges out of 120 possible ones. There is a hub and spoke structure shown in Fig. 5, obtained for the imposed number of two hubs. The results obtained for the imposed numbers of 3, 4, and 5 hubs are presented, respectively, in Fig. 6, Fig. 7 and Fig. 8. In all those cases the subgraph of hubs constitutes a clique, which, as it was said earlier, is not, in principle, necessary in our generalized hub and spoke structure, but if it is achievable, it is advantageous to obtain such a structure and the here applied evolutionary method successfully found it. The situation changes in Fig. 9, where the subgraph of hubs constitutes an α-clique with α equal 5/6 - hub 10 is not connected with hub 1.

5.3. The analysis of the obtained results

The improvement of transfer times of the transformed logistic network, compared to the input graph, depends on many factors: the structure and size of the input graph, the time range (or distance) between vertices, the shuttling

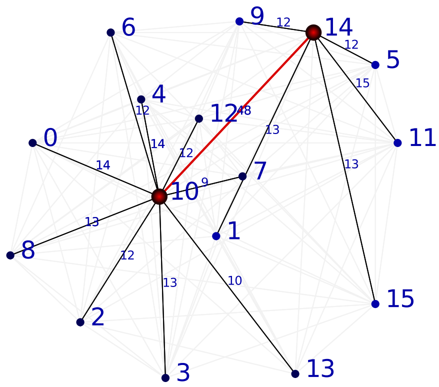

Figure 5. Results obtained for 2 hubs

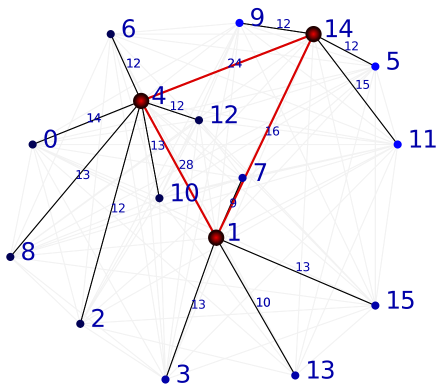

Figure 6. Results obtained for 3 hubs

frequency, and the number of chosen hubs in the output graph. As the number of possible factors that can influence the results is high, we have made several assumptions to simplify this situation. For the considered graph we assumed constant time distances between vertices, equal frequencies of movements of the required transport means and full utilization of their capacity in the input graph and at least the same in the transformed graph. In addition, the capacity of edges, connecting hubs in the transformed graph, is sufficient for the increased level of traffic. The data presented in Fig. 10 show particular results for the ratio of average times of journey before and after graph transformation. Results greater than one represent shorter transfer times after transformation and indicate time-wise better solutions. The results obtained are rather in line with expectations. Transformation of the connection graph into the hub and spoke version is profitable in these situations, in which connections are rather rare or take place with low frequencies. In the case of frequent connections it is im-

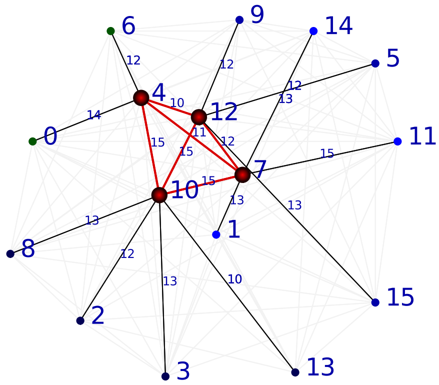

Figure 7. Results obtained for 4 hubs

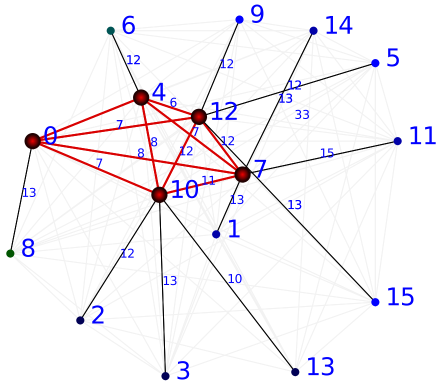

Figure 8. Results obtained for 5 hubs

possible to improve the transportation system by turning it into the hub and spoke structure. In such situation there also arises the problem of the need to build connections with very high bandwidth (capacity) among hubs. The curves of the obtained time improvements for several cases of hub numbers, presented in Fig. 10, were prepared using average values of time improvements for all possible connections.

As it can be seen, the improvements in the average travel time depend on the number of hubs and on frequency of transfers. The curves for different numbers of hubs do not cross in one point, but have different points of crossing the line, corresponding to the value of 1, and different points of crossings between each other. Generally, the biggest values of time improvement can be achieved for the solution with one hub for the widest range of the frequency of transfers (0 - 0.6). It must be noted that in practical cases travel times may be significantly longer in this case due to longer distances from the spokes to the only hub.

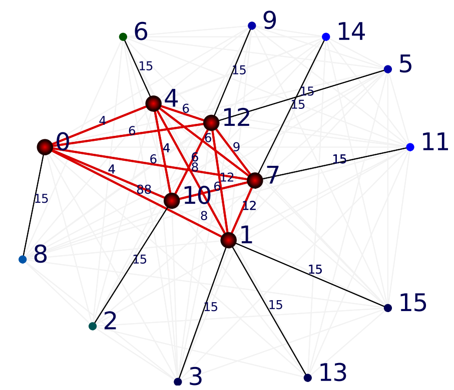

Figure 9. Results obtained for 6 hubs

For frequency of transfers lower than 0.3, the time improvement monotonically decreases with increasing number of hubs, but for frequencies between 0.3 and 0.6 such monotonicity does not appear, with the worst results obtained for 3 and 4 hubs, and better results for lower and higher numbers of them. This phenomenon is probably caused by the number of changes of transport means in the hubs. For small numbers of hubs (1, 2) the average numbers of changes is small – mostly one change is necessary to reach the destination. For bigger numbers of hubs (5 and more) an important part of graph nodes (in this case) are hubs and again the average number of changes is small. The biggest number of changes is observed for graphs with 3 and 4 hubs, which causes the smallest improvement for those cases. This may be an indication, for more realistic graphs, of existence of hub number subsets that are distinctly more or less interesting from the point of view of the qualities of the hub-and-spoke or similar solutions.

6. Conclusions

The here presented and analyzed model is an obvious simplification of the real situations, but it provides the possibility of examining some typical trends and dependences regarding the obtained hub and spoke structures - the preliminary method meant to optimise the public transport network. An improvement of transport times in a logistic network cannot always be easily obtained. The quality of the achieved solution depends on many factors, including the time range (distance) between points and the frequency of shuttling. In this paper we presented the exemplary results, obtained for the graph with 16 vertices, for different imposed numbers of hubs, their best locations having been computed by a specialized EA. The results presented were obtained, therefore, for a rather small graph with a simplified set of parameters, these parameters of the exemplary graph chosen for better visualisation of results. The software developed

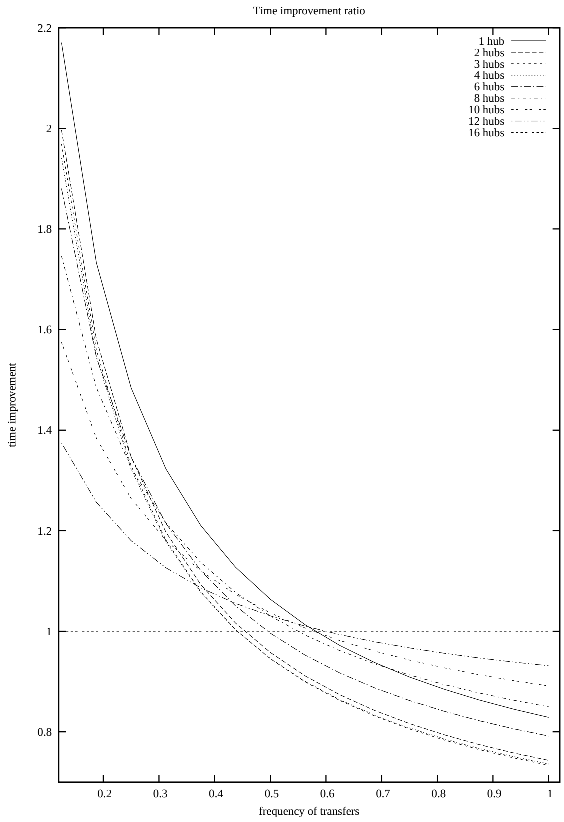

Figure 10. Improvement of transfer times for different number of hubs and frequencies of transfers

is, however, universal and can deal with bigger problems, described by more diverse data. Altogether, the methods here presented appear to be useful for designing and improving the real-life logistic and transportation systems.

References

Brunato, M., Hoos, H., Battiti, R. (2008) On effectively finding maximal quasi-cliques in graphs. In: V. Maniezzo, R. Battiti, J. Watson, (Eds.), *Proc. 2nd Learning and Intelligent Optimization Workshop, LION 2*, **LNCS**, **5313**. Springer Verlag.

Campbell, J.F., O'Kelly, M. (2012) Twenty-Five Years of Hub Location Research. *Transportation Science*, **46**(2), 153–169.

Çetiner, S., Sepil, C., Süral, H. (2010) Hubbing and routing in postal delivery systems. *Annals of Operations Research*, 181, 109–124.

Chen, Z.-Q., Wang, R.-L., Okazaki, K. (2008) An Efficient Genetic Algorithm Based Approach for the Minimum Graph Bisection Problem. *IJCSNS Int. Journal of Computer Science and Network Security*, **8**, 6, 118 – 124.

Cormen, T., Leiserson C., Rivest R., Stein. C. (2009) *Introduction to Algorithms.* MIT.

Cowen, L. (1998) *Approximation Algorithms.* Johns Hopkins University.

Coyle, J.J., Bardi E.J, Novack R.A. (1994) *Transportation.* Fourth Edition, New York: West Publishing Company.

Eghbali Zarch M., Abedzadeh M. and Setak M. (2013) Differential evolution algorithm for multi-commodity and multi-level of service hub covering location problem. *International Journal of Industrial Engineering Computations*, 4, 127–138.

O'Kelly, M. (1987) A quadratic integer program for the location of interacting hub facilities. *European Journal of Operation Research*, 32, 392-404.

O'Kelly, M., Bryan D. (2002) Interfacility interaction in models of hubs and spoke networks. *Journal of Regional Science*, **42**(1), 145-165.

Klincewicz, J.G. (1998) Hub location in backbone/tributary network design: a review. *Location Science*, 6, 307-335.

Marchiori, E. (1998) A Simple Heuristic Based Genetic Algorithm for the Maximum Clique Problem. *Proceedings of the 1998 ACM Symposium on Applied Computing*, 366-373.

Mażbic- Kulma, B., Potrzebowski, H., Stańczak, J., Sęp, K., (2008) Evolutionary approach to solve hub-and-spoke problem using α-cliques. *Evolutionary Computation and Global Optimization, Prace nauk.* PW, Warszawa, 121–130.

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, Berlin-Heidelberg.

Moscato, P. (1989) *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms.* Caltech Concurrent Computation Program (Report 826).

Pattillo J., Youssef N., Butenko S. (2013) On clique relaxation models in network analysis. *European Journal of Operational Research*, 226, 9-18.

Potrzebowski H., Stańczak J., Sęp K. (2007) *Separable decomposition of graph using α-cliques.* In: M. Kurzyński, E. Puchała , M. Woźniak, A. Żołnierek, (Eds.) *Computer recognition systems 2. In: Advances in Soft Computing*, Springer, Berlin-Heidelberg, 386-393.

Potrzebowski, H., Stańczak, J., Sęp, K. *Evolutionary approach to solve hub-and-spoke problem using α-cliques.* 2008. Evolutionary Computation and Global Optimization, Prace naukowe PW, Warszawa, pp. 121-130.

Seidman, S. B. (1978) A graph-theoretic generalization of the clique concept. *J. Mathematical Sociology*, **6**, 139-154 Gordon and Breach Science Publishers, Ltd.

Stańczak, J. (2003) Biologically inspired methods for control of evolutionary algorithms. *Control and Cybernetics*, **32**(2), 411–433.

Stańczak, J., Potrzebowski, H., Sęp, K. (2011) Evolutionary approach to obtain graph covering by densely connected subgraphs. *Control and Cybernetics*, **41**, 3, 80–107.

Sutton, R.S., Barto, A.G. (1998) *Reinforcement Learning: An Introduction.* MIT Press.

Talbi, E.-G, Bessiere, P. (1991) A parallel genetic algorithm for the graph partitioning problem. *Proceedings of the 5^{th} International Conference on Supercomputing*, ACM. NY. 312–320.

Wilson, R.J. (1996) *Introduction to Graph Theory* Addison Wesley Longman.