Zeszyty Naukowe Wydziału Informatycznych Technik Zarządzania
Wyższej Szkoły Informatyki Stosowanej i Zarządzania
„Współczesne Problemy Zarządzania”
Nr 1/2014

THE PARK AND RIDE PROBLEM. THE GRAPH APPROACH

Barbara Mażbic-Kulma*, Jan W. Owsiński*, Jarosław Stańczak*, Aleksy Barski, Krzysztof Sęp***

* Wyższa Szkoła Informatyki Stosowanej i Zarządzania,
01-447 Warszawa, ul. Newelska 6

** Instytut Badań Systemowych PAN,
01-447 Warszawa, ul. Newelska 6

Abstract: In this paper we consider the Park and Ride node location problem in a considered city. Our approach is composed of two steps. First – setting down the set of candidate points for the Park and Ride nodes by selecting a number of communication hubs. Second – picking from the set of hubs a smaller set of a Park and Ride points, located near communication hubs. In the first step we are using two approaches: the evolutionary algorithm and the one based on hypergraph transversal. In the second step we are using an exhaustive search method or the evolutionary approach, depending on the size of the problem, to find the final solution to the Park and Ride points.

Keywords: Park and Ride, evolutionary algorithm, hub and spoke, kernel and shell, graph, hypergraph, transwersal.

1. Introduction

Dynamic growth of urban agglomerations leads to new challenges. One of such essential problems is the communication and growing traffic in the city. Another is pollution increase. In a large city one of the most important sources of pollution are exhaust gases. It is a well-known fact that the public transportation produces less pollution than private transportation, and, in addition, causes less congestion, in general. Based on these observations the idea arose of the *Park and Ride* systems (P&R or sometimes P+R). Generally, the concept of such system is simple; let commuters get to the city, but not closer to the centre than some place in the suburbs or in the surroundings, by their cars, from there let them go further by public transportation. However simple this may look, it certainly is not. People are not just the machines programmed for traveling from their homes to their places of work. They have their own preferences, their private matters, and so their choices are not so ob-

vious. According to the fact the people travel hither and yon, the *Park and Ride* system should provide a convenient possibility of a change. In the current paper we will not consider improvement of urban public transport, but concentrate on the location of the *Park and Ride* facilities problem.

Some of the cities in Europe and in the United States have already brought on the park and ride system. For example there are nine P&R locations offering a total of 3,636 parking spaces in Phoenix in the USA all in the direct vicinity of the Valley Metro Rail (http://www.valleymetro.org/park_and_rides/mlr). There are also, as shown in Fig. 1, five park and ride sites in York in the UK.

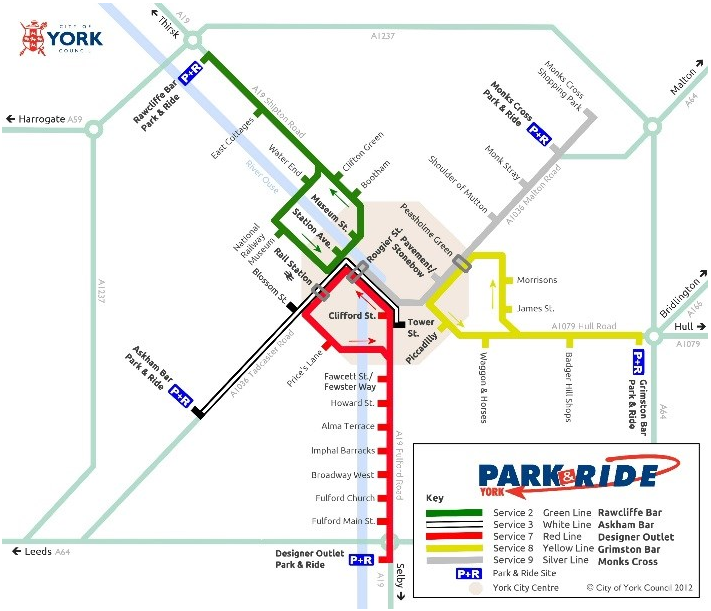


Fig. 1. The P&R system in York in The United Kingdom
source: http://www.itravelyork.info/images/content-images/Buses/York_Park_and_Ride_map.jpg

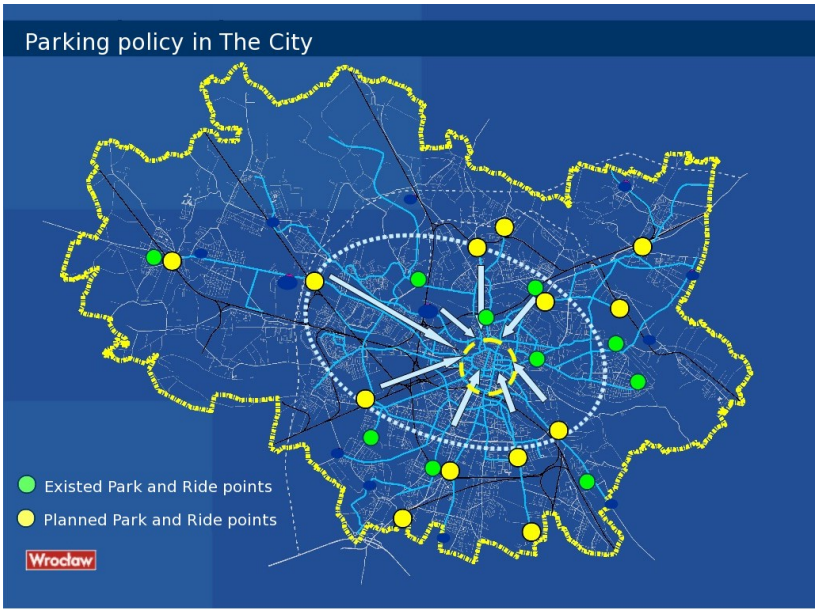


Fig. 2.The planned P&R system in Wrocław in Poland
source : http://www.tuwroclaw.com/wiadomosci,bez-parkowania-na-wlodkowica-i-sw-antoniego,wia5-3266-3621.html

Even though this looks like a step in the right direction, it is still too little to secure a significant improvement for the city traffic. Another question is whether the park and ride facilities should only be located at the border of the city or some of them should also be situated closer to the centre. The approach with park and ride sites inside the city is used in Poland in Wrocław (see Fig. 2).

Another version of the concept is implemented in Cambridge, where all P&R facilities are located in the suburbs (Figs. 3 and 4). What appears to be very important is the connection to the public transportation and the quality of the latter.

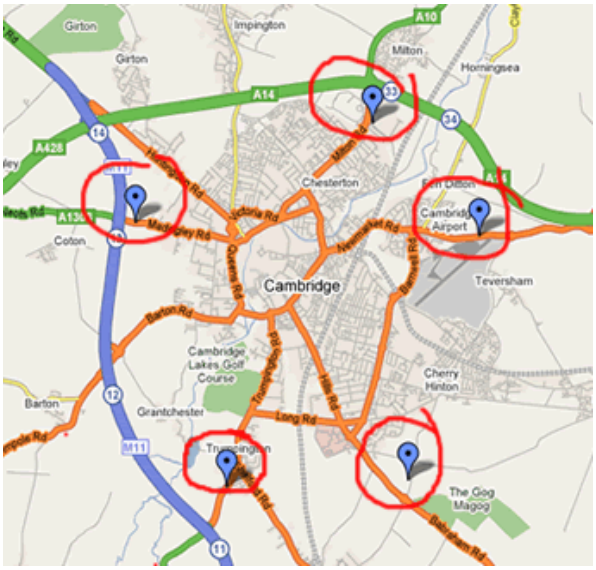


Fig. 3. The P&R system in Cambridge
source: http://www.parkandride.net/html/facilities/images/cambridge_map1.gif

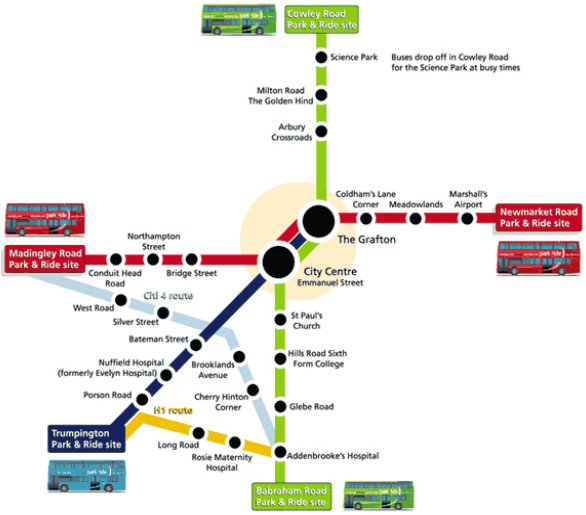


Fig. 4. The P&R and public transportation in Cambridge
source: http://www.mkvale.it/fotografie/uk/cambridge/park_and_ride_cambridge_map.gif

2. Graphs and their relevant structures

Since we shall be modelling the entirety of the P&R problem through the graph representation and then solving it as a problem defined on graphs, we start with the basic notions from graph theory, here given following Wilson (1996).

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 a pair of vertices $\{v_1, v_2\}$ with $v_1 \neq v_2$.

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., 2004). Each subgraph of a clique is a clique.

An ***α–clique*** (Mażbic-Kulma et al., 2008; Potrzebowski et al., 2006; Potrzebowski et al., 2006-2; Potrzebowski et al., 2007; Potrzebowski et al., 2008) can be defined as follows: let $A=(V', E')$ be a subgraph of graph $G = (V, E)$, $V' \subseteq V$, $E' \subseteq E$, $k=Card(V')$ and let k_i be a number of vertices $v_j \in V'$ 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 $v_i \in V'$ fulfill the condition $\alpha = (k_i+1)/k$, where $\alpha \in (0, 1]$.

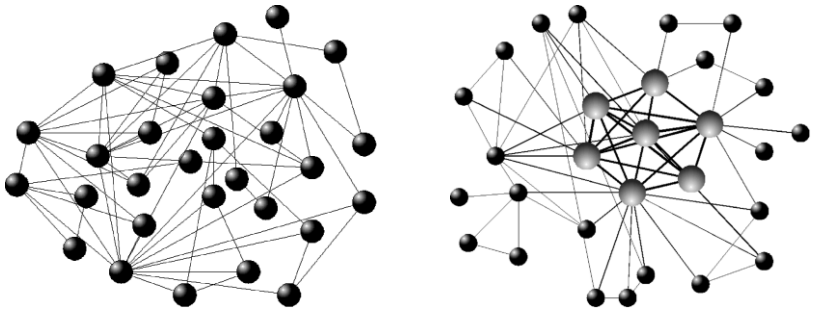
Further on we will use the notion of ***α-clique*** in the sense of ***α-clique***(α) for an earlier established α. A subgraph of an *α-clique* may not be an *α-clique* for the established α.

A ***kernel and shell structure*** in a graph $G(V, E)$ is composed of two graphs:

- ***kernel*** – a subgraph, which constitutes a group of strongly connected vertices $K(V_k, E_k)$, depending on needs, possibilities, or on the input graph structure, it can be a clique (ideally), an α-clique, or at least a connected subgraph;
- ***shell*** – a graph $S(V_s, E_s)$ where $V_s=V-V_k$ and $E_s=E-E_k$, depending on the particular requirements, it can be an α-clique (including its kernel node) or a tree with the kernel node as the root and the shell nodes as leaves.

For logistic and transportation modeling we propose to use evolutionary methods to obtain solutions. These methods may transform the transportation connections graph into an instance of the *kernel and shell* structure, leading to the *hub-and-spoke* or *α-clique* structures according to problem-specific restrictions. An ***α-clique*** structure in graph of connections (Fig. 5b) is also an instance of a more general *kernel and shell* form. It consists of several peripheral (shell) α-cliques G_α featuring the desired values of α, connected with central (kernel) α-clique G_c of strongly connected nodes, ideally with $\alpha \approx 1$. In the case of sparse graph of connections the

requirement of $\alpha \approx 1$ for the kernel subgraph can be reduced to demanding that it be a connected graph to keep its functionality. Depending on the particular problem requirements the kernel nodes or the number of kernel nodes can be imposed, or the evolutionary method can propose the best candidates, taking into account the imposed α parameters on the shell α-cliques, corresponding to the strength of connections within the derived α-cliques. In both cases the evolutionary algorithm (EA) maximizes the strength of connections within the obtained α-cliques and tries to derive structures with desired properties.



a) b)

Fig. 5 a) a source graph and b) the α-clique *kernel and shell* structure obtained.

A ***hub and spoke*** structure (Fig. 6b) is a graph $H_s=(G_h \cup G_s, E)$ where the subset G_h corresponds to at least a connected graph (of hubs) with the relevant subset of set E, each vertex of subset G_s (of spokes) has degree 1 and is connected exactly with one vertex from subset G_h, Mażbic-Kulma et al. (2009), Potrzebowski et al. (2008). The hub and spoke is a particular case of a kernel and shell structure. This structure can be used in such models, where direct connections between nodes-"spokes" attached to respective hub nodes are not very important and direct connections are not necessary.

The hub and spoke structure can be derived using one of three possible approaches. The first method uses a predetermined by some expert number of communication hubs, with the possibility of directly determining which nodes should become hubs or selecting them by the solving method. The second approach tries to find the minimum number of hubs which constitute at least a connected subgraph with all remaining nodes connected to their hubs. It must be noticed that the number of hubs used in the first approach must be bigger than this minimum value. The third approach assumes that the number of communication hubs is determined indirectly by the imposed program parameters, mainly the parameter α (the hub subgraph must constitute an α-clique with imposed value of α).

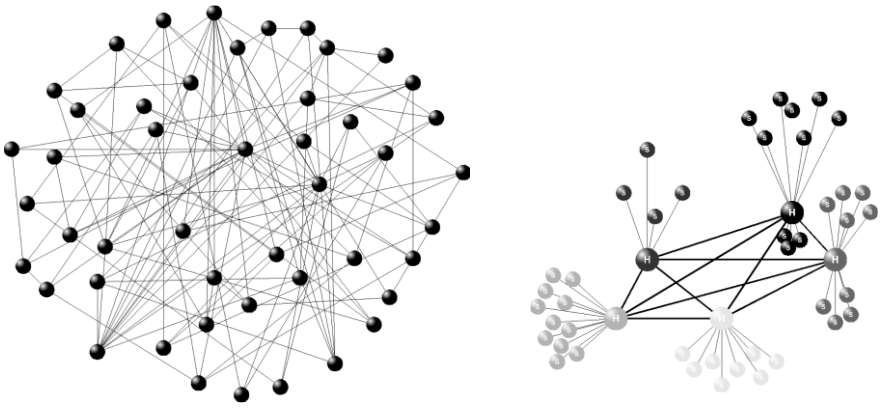
A **hypergraph** is a pair $H=(X, F)$, where X is a non-empty, finite set of

vertices, F is a non-empty family of different subsets of set X fulfilling the condition:

$$\bigcup_{f \in F} f = X \tag{1}$$

F is a set of edges.

Note that an edge can contain any number of vertices (even one), which makes a difference between hypergraphs and graphs. In fact, hypergraphs can be viewed as a direct generalization of graphs.



a) b)

Fig. 6 a) a source graph and b) the hub and spoke structure obtained.

Degree of a vertex is the number of edges to which this vertex belongs.

A **transversal** *Tr* of a hypergraph $H=(X, F)$ is a subset of vertices, which has the property $\forall e \in E: e \cap T = \emptyset$. In other words, a transversal is a set of vertices that covers (or blocks, hence the other name: *blocking set*) all the edges.

A transversal T_{mi} is called minimum in the sense of inclusion if there exists no proper subset of T_{mi} being a transversal.

A transversal is called minimum in the sense of cardinality if there exists no other transversal having less elements.

The minimum transversal problem is *NP*. All known exact algorithms solving this problem are of exponential complexity so they are of no use for cases of bigger graphs, this being the reason to use the approximating algorithms.

3. The problem representation and the methods of solution

We assume that the structures we introduced are helpful in solving the P+R problem, or they can even directly represent such solutions, since edges correspond to transport-wise connections, and nodes represent transport nodes. Thus, by finding these structures, we obtain the solution or the basis for determining the solution.

As mentioned in the introduction, we present here three algorithms for a minimum vertex cover in a hypergraph i.e. a minimum transversal in a hypergraph. Below, the algorithms are described in turn:

1. exact-backtracking
2. Lovász–Johnson–Chvatal (LJC)
3. MSBT

3.1. Exact algorithms - backtracking

This algorithm finds the minimum cover but its worst-case number of steps to find a minimum transversal is $O(2^{|X|})$, and the expected number of steps is:

$$E(X) = \sum_{k=0}^{|X|} \binom{n}{k} 2^{-2^k} \tag{2}$$

Thus, it is practically useless in solving the real problems.

Algorithm 1. The exact backtracking algorithm

Require: C - the set of vertices and v - the first vertex to be removed from C, in the beginning,

$C = X$ and $v = 0$;

for all $w \in C$ such that $w \geq v$ **do**

if C - $\{w\}$ is a transversal and C - $\{w\}$ wasn't already visited **then**

Set $C_{min} := C$ if C has less elements than C_{min} (the current best solution);

Call the procedure recursively with parameters $C := C$ - $\{w\}$, $v := w$;

end if

end for

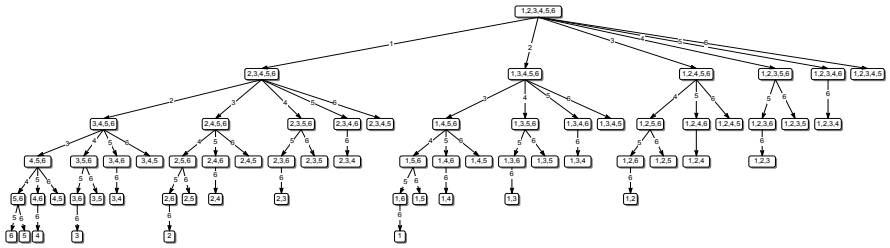


Fig. 7 Backtracking tree for six vertices

3.2. The approximating algorithm of *Lovász – Johnson – Chvatal (LJC)*

The *Lovász–Johnson–Chvatal* (LJC) algorithm is a greedy algorithm – in each step it chooses the vertex that covers the biggest number of edges. Using the LJC algorithm we assume that the set of vertices should contain these vertices which cover the biggest number of edges.

Let $deg(x)$ be a degree of vertex x and $F(x)$ be a set of edges to which vertex x belongs. The *Lovász–Johnson–Chvatal* algorithm for hypergraph $H=(X, F)$ works as follows:

Algorithm 2. The Lovász–Johnson–Chvatal algorithm

```
procedure LJC(H)
begin
Tr:=∅;
V:=X;
E:=F;
while (E⨿∅) do
        begin
                choose vertex v:∀z ∈ Vdeg(z) ⩽ deg(v)
                V:=V\{v};
                E:=E\F(v);
                Tr:=Tr∪{v};
        end;
end;
```

Using the greedy LJC algorithm we obtain the transversal, which can have a proper subset being a transversal. If n is the cardinality of the vertex set, t_{opt} is the cardinality of the smallest transversal, and t_a is the cardinality of the vertex set obtained using the LJC algorithm, then the following condition is satisfied :

$$t_{opt} \leqslant t_a \leqslant t_{opt}(1 + lg_2 n). \quad (3)$$

The *Lovász–Johnson–Chvatal* algorithm finds the transversal of a hypergraph, for some hypergraphs it can be a minimum transversal in the sense of cardinality but there exist hypergraphs for which the LJC algorithm finds the transversal that is not minimum in the sense of inclusion.

3.3. The approximation algorithm *MSBT*

The *MSBT* algorithm seeks the vertices with the lowest degree and removes them from the set of vertices. If without a removed vertex the reduct is not a transversal, then the removed vertex should be added to the transversal under construction, and the edges incident with this vertex are eliminated from the hypergraph – they are deemed to be covered. If without the removed vertex the reduct is a transversal, then we have to search in it for all the vertices with the following property: in the reduct there is at least one edge covered by exactly that precise vertex. We remove these vertices, and all the edges covered by them from the reduct, and the vertices are added to the transversal. This procedure is repeated until all edges are removed i.e. until all edges are covered.

Let $H = (X, F)$ be a hypergraph for which a minimum transversal is sought; X – the set of its vertices, F – the set of its edges.

Let $m(x)$ be a vertex in X incident upon the minimum number of edges; if there are several such vertices we pick any of them.

Let $Z(X, F)$ be a set of vertices in the reduct of the hypergraph $H(X, F)$; the elements of $Z(X, F)$ do not belong to any edge. Let $F(x)$ be a set of all edges incident with the vertex x. We carry out the *MSBT* algorithm as follows:

Algorithm 3. The MSBT algorithm

```
procedure MSBT(H)
begin
Tr:=∅;
V:=X;
Q:=X;
E:=F;

while V⊓∅ and E⊓∅ do
    begin
        k:=m(V);
        V:=V\{k};
        if V is not transversal of hypergraph (Q, E) do
           begin
                 Tr:=Tr∪{k};
```

```
                E:=E\F(x);
                V:=V\Z(V, E);
            end;
        else
            begin
                for each edge covers by exact one vertex v
                Tr:=Tr∪{v};
                E:=E\F(v);
                V:=V\{v};
            end;
        Q:=V;
    end;
end;
```

The transversals obtained from the *MSBT* algorithms are minimum in the sense of inclusion.

4. The park and ride problem

Consider a public transportation network. We analyse it as a graph where each stop corresponds to only one vertex in the graph. The case of the edges is more complicated. There are at least two approaches:

1. Two vertices are adjacent if there is a direct connection between relevant stops without any stops in the middle.
2. Two vertices are adjacent if they belong to at least one common public transportation line.

There is of course one other problem to solve, that of defining the weight of each edge. We propose to assign the number of rides between due vertices.

4.1. The Park and Ride candidates

4.1.1. The minimum transversal approach

Consider a hypergraph, where each vertex corresponds to only one stop and each edge corresponds to a single public transport segment, is the collection of all vertices belonging to this line.

According to the composite algorithm where:

Tr1 := transversal obtained by the LJC algorithm

Tr2 := transversal obtained by the MSBT algorithm

we can use just one of these two sets as a set of possible candidates for P&R points.

For the city of Warsaw[1] we obtained with these two approaches the results that are presented in Figs. 8 and 9. The differences, which can be seen, are not only just associated with the properties of the methods, but, at the same time, they bear an important interpretative value (the implicit criteria of selection, which are embedded in the respective methods).

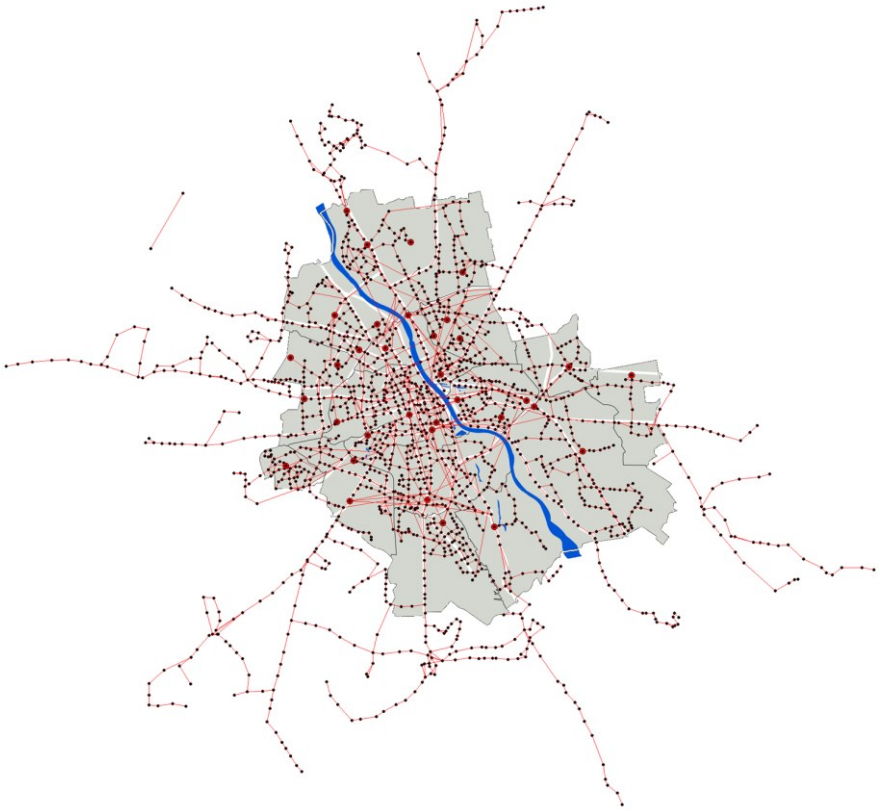


Fig. 8 Results obtained with the LJC algorithm.

[1] The results shown here in the figures were obtained with consideration of only bus and streetcar lines, without taking into account the underground and the fast city railway (SKM). The latter, though, even if accounting for a significant passenger flow, correspond to just a margin of the entire public transport network of Warsaw.

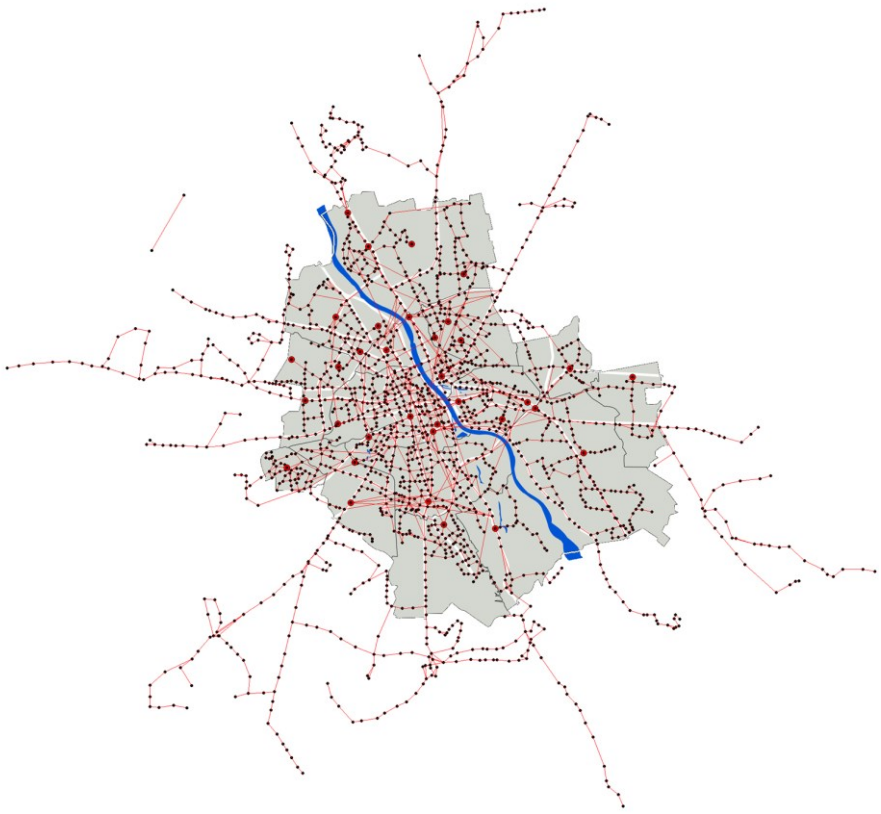


Fig. 9 Results obtained with the MSBT algorithm.

4.1.2. The evolutionary approach based on the hub and spoke structure with predetermined number of kernel nodes

Evolutionary algorithms (EAs) are often used to solve difficult graph problems such as graph coloring, TSP, graph partitioning, maximum clique search, etc. (Chen et al., 2008; Marchiori, 1998; Talbi and Bessiere, 1991; Yu et al., 2003); thus, it seems fully justified to use the evolutionary algorithm in the present graph transformation problem.

The standard evolutionary algorithm works in the manner shown in Algorithm 4., but this simple scheme requires many problem specific improvements to work efficiently (Michalewicz, 1996).

The adjustment of the evolutionary algorithm to the solved problem requires proper encoding of solutions, development of specialized genetic operators proper for the analyzed data structure and the solved problem and, finally, the fitness function (or another manner of evaluating solutions) to be optimized by the algorithm.

Algorithm 4. The standard evolutionary algorithm

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

4.2. Park and ride selection method

The hub nodes, selected by the evolutionary method, become candidates for P+R locations. As the number of hubs is rather small, the hubs of the P+R can be selected using simple selection of hubs with the best values of quality function given in the formula (4).

Commutation cost:

$$\min_{y_p} \sum_{p=1}^{n} \left[\alpha S(x_p)\left(1 - r(x_p)y_p\right) + \beta\left(t_s(x_p - x_0) - y_p t_k(x_p - x_0)\right) + \gamma\left(c_s(x_p - x_0) - y_p c_k\right)\right]$$
$$\sum_{p=1}^{n} y_p \leqslant m \tag{4}$$

where:

$S(x_p)$ – average traffic intensity

$t_s(x_p\text{-}x_0)$ – commutation time from x_0 to x_p by car,

$t_k(x_p\text{-}x_0)$ – commutation time from x_0 to x_p by public transportation,

$c_s(x_p\text{-}x_0)$ – commutation cost from x_0 to x_p by car,

$c_k(x_p\text{-}x_0)$ – commutation cost from x_0 to x_p by public transportation,

$r(x_p)$ – quantity of the traffic reduction by x_p ,

y_p- decision variable, its value is 1 when x_p is P&R and 0 if x_p is not,

α, β, γ - coefficients of proportionality >0,

n – number of hubs,

m – the maximum number of possible P&R to realize, determined by possessed financial means.

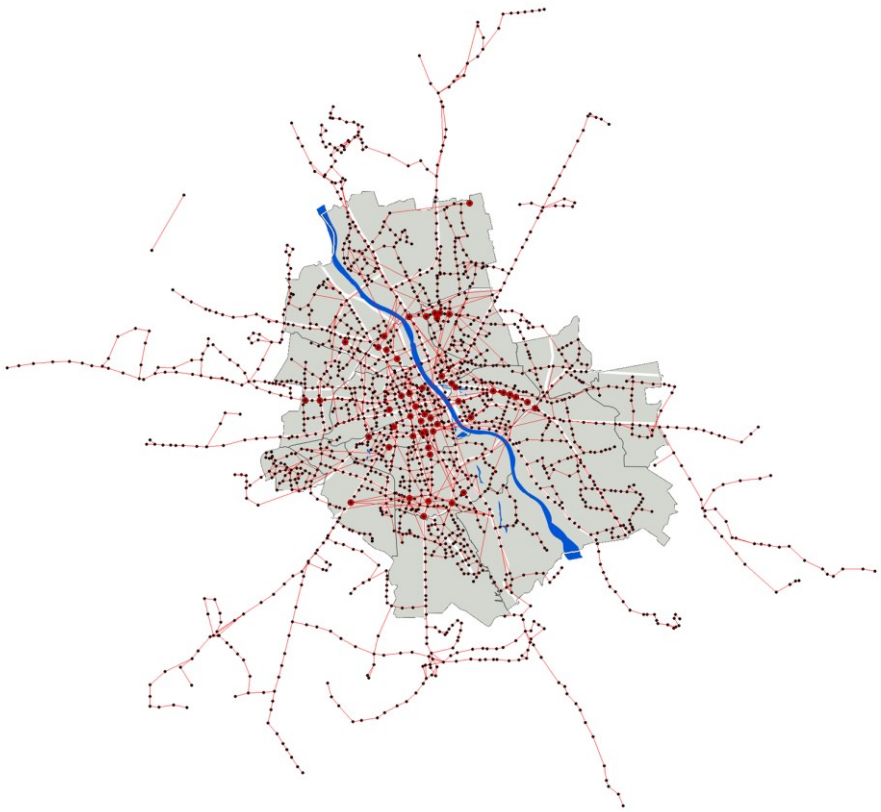


Fig. 10 Results obtained with the evolutionary algorithm, assuming 50 P&R candidates

The analyzed data have been acquired from www.ztm.waw.pl. The ZTM schedules has not contain easy to process SKM railroad schedule so as the underground schedule as well. For the preliminary analysis we took then just a schedule

containing buses and trams. In the formula (5) we propose the quality function however for further computation there are other data not so easy to obtain needed.

5. Conclusions

The approach presented appears as a promising methodology for supporting the planning of further development of public transportation and the P&R structure. The LJC and MSBT algorithm provided similar solutions because it happens for the sparse hypergraphs. As a different approach the evolutionary algorithm based on the kernel and shell selects points with the bigger traffic capacity. According to this obtained results should be treated as a preprocessing results or as a decision support solution.

References

Berge C. (1989) *Hypergraphs, Combinatorics of Finite Sets North-Holland*.

Chen, Z. Q., Wang, R. L. and Okazaki, K. (2008) An Efficient Genetic Algorithm Based Approach for the Minimum Graph Bisection Problem, *IJCSNS International Journal of Computer Science and Network Security*, Vol. 8 No.6, pp. 118 – 124.

Chvatal V. (1979) A greedy heuristic for the set-covering problem, *Mathematics and Operations Research* **4**.

Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., Stein (2009) *Introduction to algorithms*, MIT.

Johnson D. S. (1974) Approximation Algorithms for Combinatorial Problems, *Journal of Computer and System Sciences* **9**, 1974.

Kułaga P., Sapiecha P., Sęp K. (2005) *Approximation Algorithm for the Argument Reduction Problem Advances in soft computing Computer recognition systems CORES'05*, Springer Verlag Berlin, 2005.

Lovasz L. On the ratio of optimal integral and fractional covers (1975) *Discrete Mathematics* **13** , 1975.

Marchiori E. (1998) *A Simple Heuristic Based Genetic Algorithm for the Maximum Clique Problem*, Proceedings of the 1998 ACM symposium on Applied Computing, pp. 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 naukowe PW, Warszawa, pp. 121-130.

Mażbic- Kulma B, Potrzebowski H., Stańczak J., Sęp K.(2009) *Evolutionary approach to find kernel and shell structure of a connection graph*, Total Logistic Management AGH,Cracow, pp. 37-50.

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

Min H. and Gou Z. (2004): The location of hub-seaports in the global supply chain network using a cooperative competition strategy. *Integrated supply management* **1**(1), pp. 51-63.

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

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

Potrzebowski H., Stańczak J., Sęp K. (2005) *Evolutionary method in grouping of units, Proceedings of the 4th International Conference on Recognition Systems CORES'05*, Springer-Verlag, Berlin-Heidelberg.

Potrzebowski H., Stańczak J., Sęp K. (2006) *Evolutionary Algorithm to find graph covering subsets using α-cliques, Evolutionary Computation and Global Optimization*, Prace naukowe PW, Warszawa, pp. 351-358.

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

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

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

Talbi E.-G. and Bessiere P. (1991) *A parallel genetic algorithm for the graph partitioning problem*, Proceedings of the 5th International conference on Supercomputing, ACM, New York, pp. 312-320.

Wilson R.J. (1996) *Introduction to graph theory,* Addison Wesley Longman.

Yu T.L., Goldberg D.E., Yassine A., Yassine C.A. (2003) Genetic algorithm design inspired by organizational theory, Genetic and Evolutionary Computation Conference Chicago, Illinois, USA, Springer-Verlag, Heidelberg, *Lecture Notes in Computer Science* **2724**, 2003.