

Solving the k -Centre Problem as a method for supporting the Park and Ride facilities location decision

B. Prokop
 Faculty of Computer Science
 Warsaw School
 of Information Technology
 ul. Newelska 6
 01-447 Warsaw, Poland
 Email: prokop@wit.edu.pl

J. W. Owsiniński, K. Sęp
 Systems Research Institute
 Polish Academy of Sciences
 ul. Newelska 6
 01-447 Warsaw, Poland
 Email: {owsinski, sep}@ibspan.waw.pl

P. Sapięcha
 The Faculty of Electronics
 and Information Technology
 Warsaw University of Technology
 ul. Nowowiejska 15/19
 00-665 Warsaw, Poland
 Email: sapięcha@tele.pw.edu.pl

Abstract—In this article we analyze the problem of optimal location of transportation hubs in Warsaw, namely the Park and Ride location problem (*P&RP*). We take into account the expected travel time using public transport between particular points of the trip. In the currently existing *P&R* system we have 14 hub locations, and in this case the maximum travel time exceeds 50 minutes. The *P&R* problem can be reduced to the centers location problem (in our particular approach - the dominating set problem, *DS*), which is an *NP* hard problem. In order to determine the optimal locations for *P&R* two methods: the greedy and the tabu search algorithms were chosen and implemented. According to the computational experiments for the travel time restriction to 50 minutes, we obtain the *DS* composed of 3 hubs, in contrast to the existing 14 elements. The analysis of the *P&R* location in time domain is presented in this article in the context of further development of the Warsaw public transportation network, which seems to be interesting.

I. INTRODUCTION

THIS article is devoted to the **Park & Ride** facilities location problem (*P&RP*) for the case of public transportation network of Warsaw [1], [9], [19]. Data used in the analysis of this problem were obtained from the official website of the **Public Transport Authority** in Warsaw [22].

In the previous approach to solve *P&RP* [18], the transport network graph was modeled as follows. The set of vertices represented the collection of bus stops and there was an edge between a pair of vertices if and only if there was a possibility of getting from one stop to another by bus, without any transfers. This model was, obviously, too simple and impractical: only one, and the real transportation mode was taken into consideration (buses), real travel times were irrelevant (e.g. there existed some edges that represented travel times exceeding 90 minutes).

In this article we propose a much more precise model. The graph is modeled with application of real-world information about expected travel times between pairs of stops (including all modes of transit as well as transfers). This model takes also into consideration a rather common situation in which a pair of stops does not share any line but distance between

them is very short (for example less than 50 meters). In the new model such vertices should be merged together. Using the **Open Trip Planner** (*OTP*) open-source software package the estimated travel time distances were computed.

Our research consists in applying the vertex domination methods in graphs to a real-life public transportation network.

II. BASIC MATHEMATICAL DEFINITIONS

This section provides some basic notation, following [8], [6]. A **graph** is a representation of a set of objects, where some pairs of objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect pairs of vertices are called edges. More formally, a **graph** is an ordered pair $G = (V, E)$ comprising a set V of **vertices** or **nodes** together with a set E of **edges**, which are 2-element subsets of V (E is a subset of $V \times V$). An **undirected** graph is the one in which edges have no orientation. The edge (a, b) is then identical to the edge (b, a) . A vertex v is adjacent to u if and only if $(v, u) \in E$. Let $N(v) = \{u \in V : (v, u) \in E\}$ be an open **neighborhood** for a given vertex v .

A **dominating set** for a graph G is a subset D of V such that every vertex not in D is adjacent to at least one member of D [7], [11], [12]. This problem is strongly related to a problem well known in computational geometry, the **art gallery problem**. The domination number $\gamma(G)$ is the number of vertices in a smallest dominating set for G . The **k -dominating set problem** concerns testing whether $\gamma(G) = k$ for a given graph G and a natural number k ; it is a classical ***NP*-complete** decision problem in computational complexity theory [15]. According to the theorem of Ore [5], if $G = (V, E)$ is a graph without isolated vertices, then the complement of a minimal dominating set of G is also a dominating set of G . This implies that every such graph has two disjoint dominating sets and hence, $\gamma(G) \leq \frac{1}{2} \text{Card}(V)$.

A **hypergraph** is a generalization of a graph in which an edge can connect any number of vertices [3]. Formally, a hypergraph H is a pair $H = (X, F)$, where X is a set of elements called vertices, and F is a set of non-empty subsets of X called **hyperedges**. Let F be a subset of $P(X) \setminus \{\emptyset\}$, where $P(X)$ is the power set of X and $F(x) = \{f \in F : x \in f\}$ for $x \in X$. A hypergraph is also called a **set system** or a family of sets drawn from the universal set X . The **rank** of hypergraph H is the size of the largest hyperedge in H . A **set covering** of a hypergraph $H = (X, F)$ is a subfamily C of F , such that the union of hyperedges from C equals the universe of vertices. A **transversal** (or **hitting set**) of a hypergraph $H = (X, F)$ is a subset T of X that has a nonempty intersection with every edge. The notions of hitting set and set covering are equivalent. The **decision versions** of the **hitting set** and **set covering** problems are **NP-complete**. The **greedy algorithm** for set covering chooses the sets according to one rule: at each stage, choose the hyperedge that contains the largest number of uncovered elements. This algorithm actually achieves an **approximation ratio** $\frac{Card(C)}{Card(Opt)}$ (Opt is an optimal set covering) of $h(rank)$, where $h(n)$ is the n^{th} harmonic number. This value is approximately given by: $\mathcal{O}((1 + \log(Card(V))))$. We can construct a dual algorithm for a hitting set problem, for which the performance ratio is: $\mathcal{O}((1 + \log(Card(F))))$.

Algorithm 1 The Greedy set covering method

input: hypergraph $H = (X, F)$;

output: set covering C ;

$U := X$; $C := \emptyset$;

```

while ( $C \neq X$ ) do {
    select  $S$  from  $F$  such that
        maximizes  $Card(S \cap U)$ ;
     $U := U \setminus S$ ;
     $C := C \cup \{S\}$ ;
}

```

return: C ;

It is interesting that there exists a pair of **polynomial-time reductions** between the **minimum dominating set problem** and the **minimum set covering problem**. These reductions show that an efficient algorithm for the minimum dominating set problem would provide an efficient algorithm for the set covering problem and vice versa. According to the above presented facts, the greedy algorithm provides a factor $1 + \log(Card(V))$ approximation of a minimum dominating set. Let us consider a reduction from the dominating set problem to the set covering problem. For any given graph $G = (V, E)$ with $V = \{1, 2, \dots, n\}$, construct a hypergraph $H = (X, F)$ as follows: the universe X is V , and the family of hyperedges F is $\{F_1, F_2, \dots, F_n\}$ such that F_v consists of the vertex v and all vertices adjacent to v in G . Hence, if D is a dominating set for G , then $S = \{S_v : v \in D\}$ is a feasible solution of the set covering problem, with $Card(C) = Card(D)$. Conversely, if $S = \{S_v : v \in D\}$ is a

feasible solution of the set covering problem, then D is a dominating set for G , with $Card(D) = Card(C)$. Hence, the greedy algorithm provides a factor $1 + \log(Card(V))$ approximation of a minimum dominating set.

Problems of finding the best location of facilities in networks or graphs abound in practical situations, such as determining locations for factories, assembly plants or warehouses, as well as in airline crew scheduling. One of the well known facility location problems is the **vertex k -center problem**, where given n cities and distances between all pairs of cities, the aim is to choose k cities called **centers** so that the largest distance of any city to its nearest center is minimal. Let $G = (V, E)$ be a complete undirected graph with edge costs satisfying the triangle inequality (for a given **metric** $d : E \rightarrow R$), and k be a positive integer not greater than $Card(V)$. For any subset S of V and a vertex $v \in V$, define: $d(v, S)$ to be the length of the shortest edge from v to any vertex in S . The **vertex k -center problem** is to find such a subset S of V , where $Card(S) \leq k$, which minimizes: $max(d(v, S))$ for $v \in V$. The vertex k -center problem is **NP-hard**. In this paper we solve the k -center problem as a series of a minimum dominating set problems.

A set of different approaches (like: tabu search, variable neighborhood search) to solve the k -center problem was given by authors [17]. Parallely, the various greedy methods were proposed in the following publications [13], [14].

III. PROBLEM DEFINITION

This article is devoted the $P\&R$ facilities location problem ($P\&R$) in public transportation network of Warsaw. The problem is to find a set of stops such that in the worst case scenario using the public transport each trip will take no more than assumed k minutes. In this paper we propose the application of two methods: the *greedy algorithm* and the *Tabu Search*.

The k -Park&Ride Problem

Input: Given $G = (V, E)$ —transport network,

$d : E \rightarrow R^+$ —an average distance between two vertices,

$k \in R^+$ - time limit;

Output: Find $S \subseteq V$ such that:

(*) for each $v \in V : \min_{s \in S} d(v, s) \leq k$,

(**) $minCard(S)$.

IV. OUR APPROACH

In this chapter the data characteristics and the solution construction method are presented.

A. Data

The dataset that we are working on consists of coordinates of all public transportation stops in the city of Warsaw and complete schedules for different mass transit modes (buses, trams, metro and passenger trains). There are about **4000 stops** within the city limits and **317 different routes** each with 1-4 unique trips.

The data containing stops, routes and schedules is made available by the Public Transport Authority in Warsaw. However, the files are not shared in any standardized format and thus require parsing to be suitable for processing. Several software tools have been developed by the authors in order to work with this data (parsers, graph builders). They usually consist of over 4 million connections divided in multiple sections. The first step is to parse the file into *JSON* format. The second one is to convert it to the *GTFIS* using our tools. *GTFIS* is the required format to use *OTP* implementation of the **Raptor** algorithm [2] to find the shortest paths in multi-modal transportation network with a schedule. In our case, over 16 million requests had to be sent to instances of *OTP* servers in order to fill the adjacency matrix of the graph.

B. Methods

We model the network in a time oriented manner, using average travel times between each pair of stops as the weights of the edges in the **graph G**. In order to achieve this, we have converted the data to *Google Transit Feed Specification* and used *Open Trip Planner (OTP)* to calculate the shortest trips between all stops in a given time lapse. The *OTP* application is an open-source tool for journey planning in multi-modal transit networks.

Given a complete directed graph G , in which the weight w of the edges represents the average travel time between the stops, we choose the parameter k measured in minutes as a cut-off value and construct a new **graph H** such that:

$$V(H) = V(G),$$

$$E(H) = \{e \mid e \in E(G), w(e) \leq k\}.$$

We then use H as an input to the *minimal Dominating Set* algorithms in order to approximate the smallest subset of stops that allows us to reach all other stops within the given time bounds. Information about the density of Warsaw's mass transit network with respect to different values of k is presented in Table 1.

TABLE I
DENSITY AND NUMBER OF EDGES WITH RESPECT TO PARAMETER k

k [min]	Edges	Density
15	679296	≈ 0.04
30	3863959	≈ 0.23
45	9121623	≈ 0.55
53	11916338	≈ 0.72

Our Approach to The k -Park&Ride Problem

- 1) Construct a complete directed graph $G=(V, E)$, where $d(v, u)$ is the average travel time for $(v, u) \in E$ (obtained from the application of the *OTP* package),
- 2) Construct a graph H based on graph G with respect to the time limit k ,
- 3) Find a *minimal Dominating Set* for a given graph H using the Tabu Search or a greedy method.

Two *minimal Dominating Set* approximation methods were used in our computations. We applied the greedy algorithm and the metaheuristic tabu search method. Both approaches were implemented by us in *Python* programming language with extensive use of *NumPy* library for fast matrix operations.

V. RESULTS

We prepared computation experiments based on typical PC (Intel i5 2.7 Ghz 8GB Ram) using the following software: OSX El Capitan 10.11.14 (clients and computations), Ubuntu 14.04 (server OTP), Python 3.5.0, iPython 4.0.0, NumPy 1.10.1, python-geojson, environment PyCharm, iPython Notebook, Vizualization: MapBox and Ruby on Rails,

We achieved the following results in terms of the time complexity:

- 1) The Greedy Set Covering in dependency of k :

k [min]	15	30	45	53
Comp.time	≈ 16.4	≈ 5.3	≈ 4.61	≈ 3.52

- 2) The OTP - 16 000 000 queries (≈ 111 hours).
- 3) The Tabu Search in dependency of k (time for 100 iterations):

k [min]	15	30	45	53
Comp.time	≈ 4.1	≈ 4.2	≈ 4.5	≈ 5.0

The Park and Ride system in Warsaw consists currently of 14 parkings and they form a dominating set under our model when the time trip is $k \geq 53$ minutes. According to the computational results based on the new model these locations ensure that the longest journey from them to any stop will not exceed 53 minutes (they form a dominating set when $k \geq 53$). Under the same constraint (maximum 53 minutes travel time) we have found dominating sets of sizes 5 (greedy) and 3 (tabu search). It is interesting to note that when the maximum of travel time is limited to 30 minutes, the cardinality of the calculated dominating sets, as well as the locations of parkings are similar to those of the already existing facilities in Warsaw. This might suggest that small improvements/changes to the existing park and ride system might be very beneficial in terms of maximum journey distance (cutting it down from 53 to 30 minutes).

TABLE II
CARDINALITY OF DOMINATING SETS FOUND BY DIFFERENT METHODS

Algorithm	k (min)			
	15	30	45	53
Greedy	69	15	8	5
Tabu search	62	12	5	3

However, we can achieve domination in this network by selecting only 3 nodes (see Figure 2).

VI. CONCLUSIONS

In this article we describe the analysis of the optimal location of transport hubs in Warsaw, in the context of the Park

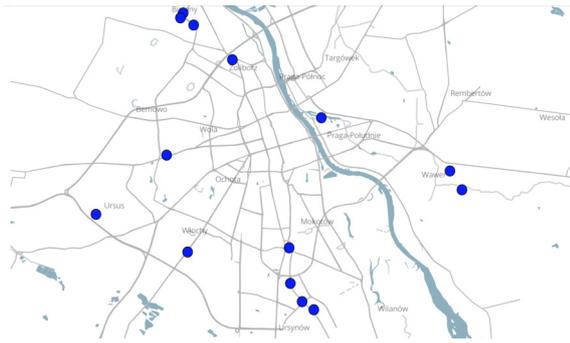


Fig. 1. Existing Park and Ride facilities



Fig. 4. Dominating set of cardinality 5 for $k = 53$ (greedy algorithm)



Fig. 2. Dominating set of cardinality 3 for $k = 53$ (tabu search)

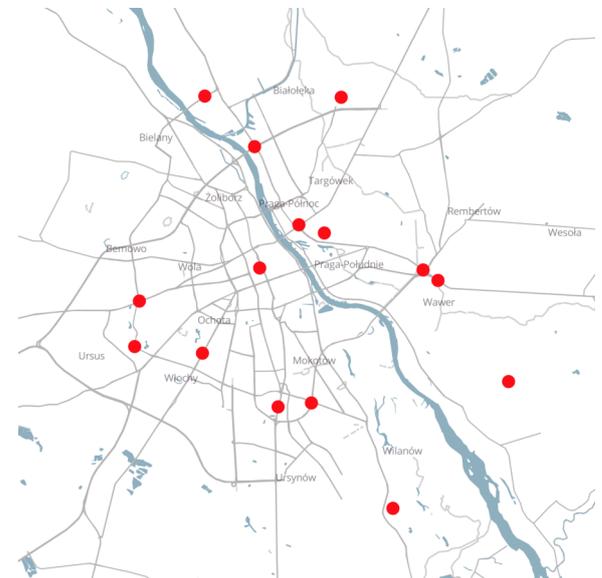


Fig. 5. Dominating set of cardinality 15 for $k = 30$ (greedy algorithm)



Fig. 3. Dominating set of cardinality 12 for $k = 30$ (tabu search)

and Ride location problem. We took into account the expected travel time between two particular points. This problem was reduced to the problem of determining the centers in the sense of the dominating set. Data for our research were obtained and collected from the website of the public transport authority. In order to establish the format, the data was converted to *JSON* a format, and then to the *GTFS* one. Using the *OTP* package we computed the expected journey times between all pairs of stops in Warsaw public transport. In the currently existing *P&R* system we have 14 locations, and in this case the maximum travel time exceeds 50 minutes. In order to determine the optimal locations for *P&R* two methods: the greedy and the tabu search algorithms have been implemented (in *Python*). The result obtained imply the dominating set consisting of 3 stops, in the contrast to the existing 14 *P&R* parks. For the

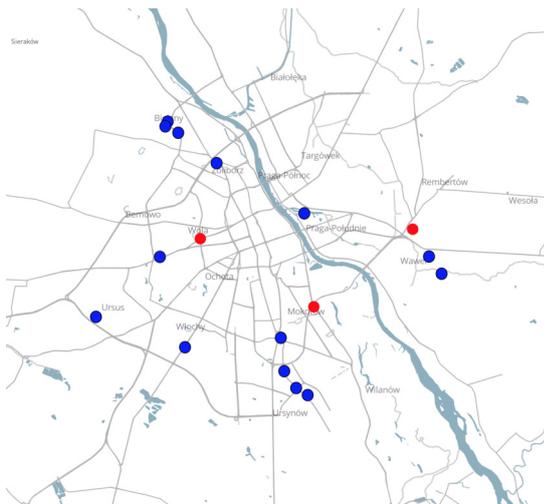


Fig. 6. Dominating set of cardinality 3 for $k = 53$ (red dots) and existing $P&R$ (blue dots) (tabu search)

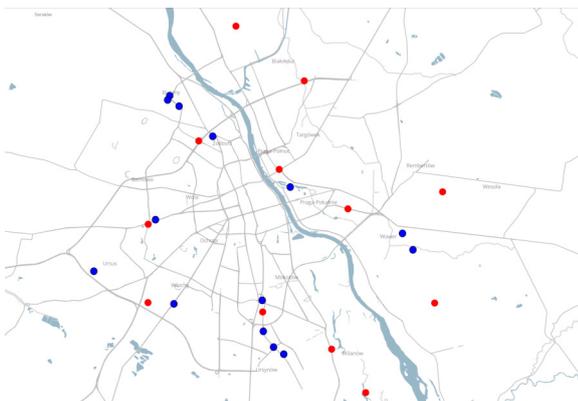


Fig. 7. Dominating set of cardinality 12 for $k = 30$ (red dots) and existing $P&R$ (blue dots) (tabu search)

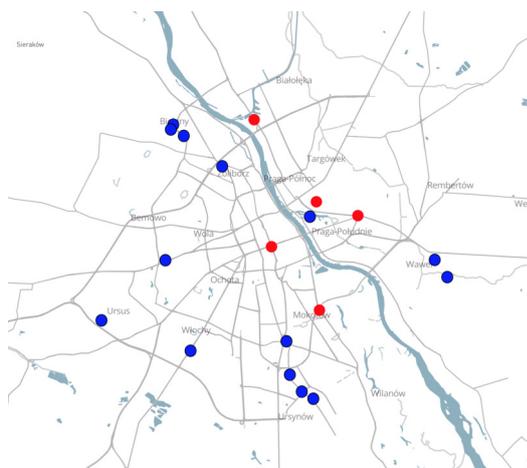


Fig. 8. Dominating set of cardinality 5 for $k = 53$ (red dots) and existing $P&R$ (blue dots) (greedy algorithm)

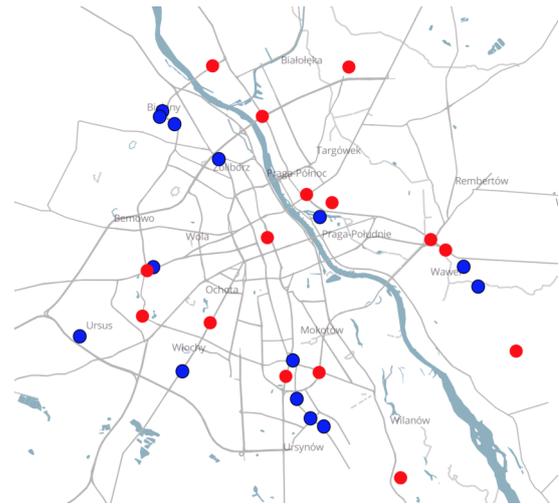


Fig. 9. Dominating set of cardinality 15 for $k = 30$ (red dots) and existing $P&R$ (blue dots) (greedy algorithm)

assumed 30 minutes time limit, the result consists of 12 stops. This analysis seems to be interesting in the context of further development of the $P&R$ system. To summarize, the aim of the $P&R$ system is not only to discharge the traffic directed to the city center, but also to enable passengers to conveniently travel to different locations. Unfortunately, we can not say this about the existing solution.

REFERENCES

- [1] Aros-Vera F., Marianov V., Mitchell J., *p-Hub approach for the optimal park-and-ride facility location problem*, European Journal of Operational Research 226, 2013
- [2] Bast H., Delling D., Goldberg A., Müller-Hannemann M., Pajor T., Sanders P., Wagner D., Werneck R. F., *Route Planning in Transportation Networks*, Cornell University Library, arXiv:1504.05140, 2015
- [3] Berge C., *Hypergraphs: Combinatorics of Finite Sets*, Elsevier, 254 pp., 1984
- [4] Bunke H., Wang P. S. P., *Graph classification and clustering based on vector space embedding*, World Scientific Publishing Co. Pte. Ltd., Singapore, Series in Machine Perception and Artificial Intelligence, Vol. 77, 2010
- [5] Chartrand G., Lesniak L., Zhan P. *Graphs & Digraphs*, CRC Press, 2010
- [6] Cohen R., Havin S., *Complex Networks: Structure, Robustness and Function*, Cambridge University Press, Cambridge, 2010
- [7] Ding-Zhu Du, Peng-Jun Wan, *Connected Dominating Set: Theory and Applications*, Springer Optimization and Its Applications, 2012
- [8] Erciyes K., *Complex Networks: An Algorithmic Perspective*, CRC Press, 320 pp., 2014
- [9] Farhan B., Murray A., *Siting Park and Ride facilities using a multi-objective spatial optimization model*, Computers & Operations Research 35, 2008
- [10] Fortunato S., *Community detection in graphs*, Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY, 2010
- [11] Haynes T., Hedetniemi S., Slater P., *Fundamentals of Domination in Graphs*, Marcel Dekker Inc., 1998
- [12] Henning M., Yeo A., *Total Domination in Graph*, Springer Monographs in Mathematics, 2013
- [13] Hochbaum D. S., Shmoys D. B., *A best possible heuristic for the k -center problem*, Mathematics of Operations Research, 10:180-184, 1985.
- [14] Hochbaum D. S., *Approximation Algorithms for NP-hard Problems*, PWS Publishing Company, 596 pp., 1995

- [15] Garey M. R., Johnson D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 344 pp., New York 1979
- [16] Glover F., Laguna M., *Tabu search*, Springer Verlag, Handbook of Combinatorial Optimization, pp. 3261-3362, 2013
- [17] Mladenović N., Labbe M., Hansen P., *Solving the p-center problem with tabu search and variable neighborhood search*, Networks 42, 1: 48-64, 2003.
- [18] Owsiniński J.W., Stańczak J., Barski A., Sep K., Sapiecha P., *Graph based approach to the minimum hub problem in transportation network*, proceedings of FEDCSIS 2015, pp. 1641-1648, 2015
- [19] Schöbel A., *Optimization in Public Transportation*, Springer Verlag, Optimization and Its Applications, 2006
- [20] Takes F. W., *Algorithms for Analyzing and Mining Real-World Graphs*, PhD thesis was employed at Leiden University, 2014
- [21] Vazirani V. V., *Approximation Algorithms*, Springer Verlag, Berlin Heidelberg, 2003
- [22] The source of data, *Warsaw public transport timetable*, <http://www.ztm.waw.pl>, 2016