

# AN IMPROVED APPROXIMATION ALGORITHM FOR OPTIMAL ROUTES GENERATION IN PUBLIC TRANSPORT NETWORK

Jolanta Koszelew<sup>1</sup>

<sup>1</sup>Faculty of Computer Science, Białystok University of Technology, Białystok, Poland

**Abstract:** This paper presents a new version of Routes Generation Matrix Algorithm, called Routes Generation Matrix Improved Algorithm (RGMIA), for determining routes with optimal travel time in public transport network. The method was implemented and tested on the real public transport network in Warsaw city. This network was completed with walk links and therefore resultant routes are more practical and can perform various users' preferences. Effectiveness of the improved method was compared in two aspects: time complexity and quality of results, with another two algorithms - previous version of Routes Generation Matrix Algorithm (RGMA) and Routes Generation Genetic Algorithm (RGGGA). RGMA and RGGGA algorithms were described in previous author's papers [9,10].

**Keywords:** public transport network, time-dependent shortest path, optimal routes, genetic algorithm

## 1. Introduction

The shortest path problem is a core model that lies at the heart of network optimization. It assumes that weight link in traditional network is static, but is not true in many fields such as Intelligent Transportation Systems (ITS) [16]. The optimal path problems in variable-time network break through the limit of traditional shortest path problems and become foundation theory in ITS. The new real problems make the optimal path computing to be more difficult than finding the shortest paths in networks with static and deterministic links, meanwhile algorithms for a scheduled transportation network are time-dependent.

A public transportation route planner is a kind of ITS and provide information about available public transport journeys. Users of such system determine source and destination point of the travel, the start time, their preferences and system returns

---

Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 5, pp. 5-17, 2010.

as a result, information about optimal routes. In practice, public transport users' preferences may be various, but the most important of them are: a minimal travel time and a minimal number of changes (from one vehicle to another). Finding routes with minimal number of changes is not a difficult problem, but generating routes with minimal time of realization, on the base of dynamic timetables, is much more complexity task.

Moreover, standard algorithms considered graphs with one kind of links (undirected or directed) which have no parallel arcs. Graph which models a public transport network includes two kinds of edges: directed links which represent connections between transport stops and undirected arcs correspond to walk link between transport stops.

Additionally, with each node in a graph which represents a transportation network, is concerned detail information about: timetables, coordinates of transport stops, etc. This information is necessarily to determine weights of links during realization of the algorithm.

Besides, standard shortest path algorithms generate only one optimal path, but methods used in journey planners must return few alternative optimal paths. These four differences between standard shortest path problem and routing problem in public transportation network cause that time complexity of algorithms which solve this problem may be very high.

Many algorithms has been developed for networks whose edge weights are not static but change with time but most of them take into consideration a network with only one kind of link, without parallel links and returns only one route. Cooke and Halsey [5] modified Bellman's [3] "single-source with possibly negative weights" algorithm to find the shortest path between any two vertices in a time-dependent network. Dreyfus [6] made a modification to the standard Dijkstra algorithm to cope with the time-dependent shortest path problem. Orda and Rom [13] discussed how to convert the cost of discrete and continuous time networks into a simpler model and still used traditional shortest path algorithms for the time-dependent networks. Chabini [4] presented an algorithm for the problem that time is discrete and edge weights are time-dependent. Other algorithms deal with finding the minimum cost path in the continuous time model. Sung [14] et al. gave a similar result using a different version of cost and improved the algorithm's efficiency. Ahuja [1] proved that finding the general minimum cost path in a time-dependent network is NP-complete and special approximation method must be used to solve this problem.

The RGMA [9] and RGGGA [10] are approximation methods which generates  $k$  routes with optimal travel time. Like the  $k$ -shortest paths algorithm [11], since these methods generate multiple "better" paths, the user can have more choices from where

he or she can select based on different preferences such as total amount of fares, convenience, preferred routes and so on.

RGMA realizes label-setting strategy [2] for construct optimal routes and uses special matrices which are applied as heuristics in this algorithm. RGA algorithm is a genetic algorithm [12] which starts with a population of randomly generated initial set of routes and tries to improve individuals in population by repetitive application of selection and crossover operators. Both algorithm was implemented and tested on realistic data - from Bialystok city public transport network [10]. Computer experiments had shown that genetic algorithm (RGA) generates routes as good as matrix based algorithms (RGMA), but significantly faster.

In this paper author of RGMA presents a new improved version of this method called Routes Generation Matrix Improved Algorithm (RGMIA) which has lower time-complexity than RGMA and generates more optimal routes than RGMA and RGA. In this paper next section includes definition of optimal routes generation problem and description of public transport network model. In the third section author presents common idea of RGMA and RGMIA and illustrates it by a simple example. Section 4 is concerned on detail description of differences between RGMIA and RGMA. Subject of Section 5 is the comparison of effectiveness of RGMIA, RGMA and RGA methods in two aspects: time complexity and quality of results. This comparison is based on experimental results which were performed on realistic data. The paper ends section with some remarks about future work on possibilities of further improving of RGMIA.

## **2. Network Model and Problem Definition**

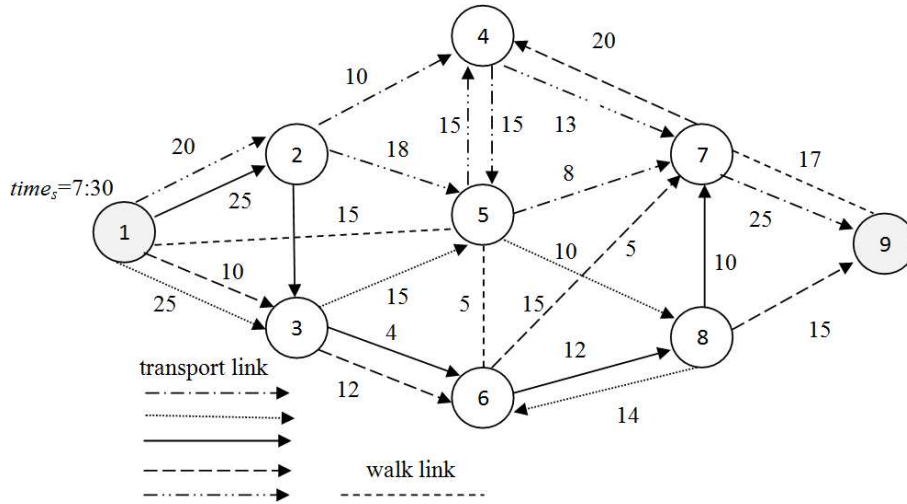
A public transportation network in our model is represented as a bimodal weighted graph  $G = \langle V, E \rangle$  [8], where  $V$  is a set of nodes,  $E$  is a set of edges. Each node in  $G$  corresponds to a certain transport stop (bus, tram or metro stop, etc.), shortly named stop. We assume that stops are represented with numbers from 1 to  $n$ . The directed edge  $(i, j, l, t) \in E$  is an element of the set  $E$ , the line number  $l$  connects the stop number  $i$  as a source point and the stop number  $j$  as a destination. One directed edge called transport link corresponds to one possibility of the connection between two stops. Each edge has a weight  $t$  which is equal to the travel time (in minutes) between nodes  $i$  and  $j$  which can be determined on the base of timetables. A set of edges is bimodal because it includes, besides directed links, undirected walk links. The undirected edge  $\{i, j, t\} \in E$  is an element of the set  $E$ , if walk time in minutes between  $i$  and  $j$  stops is not greater than  $limit_w$  parameter. The value of  $limit_w$

parameter has a big influence on the number of network links (density of graph). The  $t$  value for undirected edge  $\{i, j\}$  is equal to walk time in minutes between  $i$  and  $j$  stops. We assume, for simplification, that a walk time is determined as an Euclidian distance between stops.

A graph representation of public transportation network is shown in Fig. 1. It is a very simple example of the network which includes only nine stops. In the real world the number of nodes is equal to 3500 for the city with about 1 million of inhabitants.

Formal definition of our problem is the following. At the input we have: graph of transportation network,  $timetable(l)$  - times of departures for each stops and line  $l$ , source point of the travel ( $o$ ), destination point of the travel ( $d$ ), starting time of the travel ( $time_o$ ), number of the resultant paths ( $k$ ), maximum number of changes ( $max_t$ ) and limit for walk links ( $limit_w$ ). At the output we want to have the set of resultant routes, containing at most  $k$  quasi-optimal paths with minimal time of realization (in minutes) with at most  $max_t$  changes.  $max_t$  parameter takes into account only transfers from one vehicle to another (not from vehicle to walk or inversely).

Weight of transport link  $(i, j, l, t)$  is strongly dependent on the starting time parameter ( $time_o$ ) and  $timetable(l)$  which can be changed during the realization of the algorithm. The  $t$  value of  $(i, j, l, t)$  link is equal to the result of subtraction: time of arrival for line  $l$  to the stop  $j$  and start time for stop  $i - time_i$ .



**Fig. 1.** Representation of a simple transportation network (different styles of lines mark different transport links; dot lines mark walk links)

### 3. The Common Idea of RGMA and RGMIA

RGMA and RGMIA algorithms of determining paths with the minimal travel time are based on label-setting method of the shortest paths generation. In order to find optimal paths, for  $k \geq 1$ , it is important to choose different routes throughout the network [15]. It can be realized by labeling nodes and edges or by removal of a node or an edge. Because it is easier to implement the labeling algorithms than the path deletion algorithms for the transportation network, the algorithm described in this section is based on the label-setting technique [2].

The idea of both methods is the same. Before first iteration of the algorithm we must labeled each node in the network. The initial value of label (marked as  $et$ ) of the node  $o$  is equal to  $k$  ( $et(o) = k$ ) and 0 for other nodes. In first step of the method we find the closest node  $u$  to the start point  $o$ . Node  $u$  is the closest to node  $o$  iff  $H(o) = u$ , where  $H$  is an heuristic which determines the choice of closest node. This heuristic is different in RGMA and RGMIA. The label of the closest node  $u$  is increasing at that moment. Next, we add to the graph  $G$  new arcs: from  $o$  to each node  $v$  which incidences with node  $u$ . The weight  $t_{ov}$  of the new arc is equal to  $t_{ou} + t_{uv}$ . Next step executes as the same way as the first step. The algorithm stops, when the label of the end node  $d$  is equal to  $k$  or there is no nodes closest to  $o$ . We have  $k$  (or less than  $k$  if there is no  $k$  paths from  $o$  to  $d$  in network) paths from  $o$  to  $d$  as a result of the method.

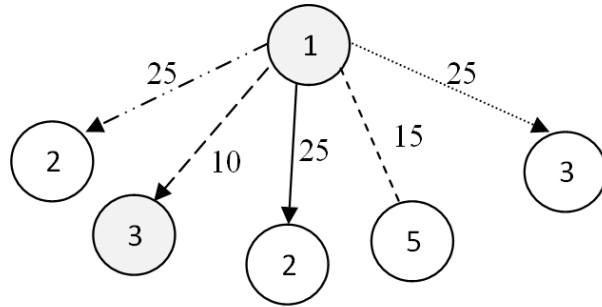
The common idea of RGMA and RGMIA is written in the psedocode form presented bellow. Line number six in this pseudocode realizes different heuristics  $H$  for RGMA and RGMIA which are detail described in Section 4.

```

Pseudocode CommonIdea(G,o,d,timetables,k)
1: Begin
2:   for i:=1 to n do et(i):=0;
3:   et(o):=k;
4:   while et(o)<k do
5:     Begin
6:       u:=H(o);
7:       if u not exist then break;
8:       if u=d then return route from o to d;
9:       add new arcs (o,v,t) to G for each node v which incidences with u
10:      et(u):=et(u)+1;
11:    End
12: End;

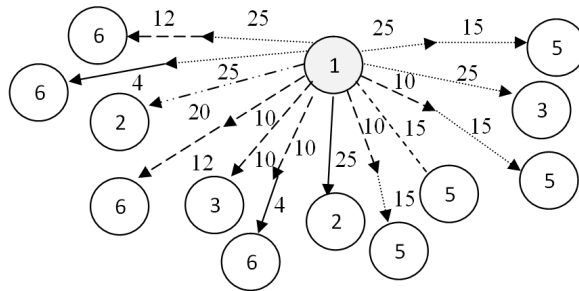
```

In Fig. 2 and Fig. 3 we illustrate first step of realization of common idea of RGMA and RGMIA on the example network presented in Fig. 1. In this presentation we assume that  $u = H(o)$  iff there is a link from  $o$  to  $u$  and  $et(u)$  is minimal in first order and travel time from  $o$  to  $u$  is minimal in second.



**Fig.2.** Choice of the closest node in first step of pseudocode CommonIdea realized on network presented in Fig. 1.

If start node is equal to 1 and destination node is equal to 9 then the closest node in network presented in Fig. 1 is equal to 3, because of this node has a minimal value of travel time from node 1. Narrowly, there are five links between nodes 1 and 9. Four transport links:  $(1,2,20)$ ,  $(1,2,25)$ ,  $(1,3,10)$ ,  $(1,3,25)$  and one walk link  $(1,2,20)$  (the number of line is omitted for simplification). The algorithm chooses node 3 as the closest node because transport link  $(1,3,10)$  has the smallest travel time.



**Fig.3.** Addition new arcs to the network in second step of RGMIA-RGMA realized on network in Fig. 1

In the second step of the algorithm seven new arcs are added to the graph:  $(1, 5, 39), (1, 5, 35), (1, 6, 14), (1, 6, 22), (1, 6, 29), (1, 6, 39)$ . The travel time  $t$  for new link  $(1, u, t)$  is equal to the sum of travel time from 1 to 3 travel time from 3 to  $u$ . There are twelve links beginning in node 1 in the graph now.

The basic difference between RGMA and RGMIA rests on the heuristic  $H$  which determines conditions of the closest node choosing. We detail describe this difference in the next section.

#### 4. Differences between RGMA and RGMIA

To detail present differences between RGMA and RGMIA we must define special matrices used in definition of  $H$  heuristic in both methods:

1.  $Q = q[i, j]_{i, j=1..n}$  - minimal number of changes matrix.  $q[i, j]$  element is equal to the number of minimal changes in route from stop  $i$  to stop  $j$ . The algorithm which determines  $Q$  matrix is detail presented in [8].
2.  $D = d[i, j]_{i, j=1..n}$  - minimal number of stops matrix.  $d[i, j]$  element is equal to the minimal number of stops in route from stop  $i$  to stop  $j$ . We can calculate  $D$  matrix using standard Breath First Search algorithm for each variant of start stop  $i$  in the network.
3.  $T_{rh} = t_{rh}[i, j]_{i, j=1..n}$  - minimal travel-time in rush hours matrix.  $t_{rh}[i, j]$  element is equal to the minimal travel time in rush hours in route from stop  $i$  to stop  $j$ . Rush hours are specific for a given city (from 7:00 a.m. to 10:00 a.m. and from 03:00 p.m. to 07:00 p.m. for example in Warsaw). We can obtain  $T_{rh}$  matrix on the base of fragment of timetables which concerned on rush hours.
4.  $T_{oh} = t_{oh}[i, j]_{i, j=1..n}$  - minimal travel-time outside of rush hours matrix.  $t_{oh}[i, j]$  element is equal to the minimal travel time outside of roush hours in route from stop  $i$  to stop  $j$ . We can determine  $T_{oh}$  matrix on the base of fragment of timetables which concerned on hours outside rush.

In practical implementation of RGMIA it is possible to determine more than two kinds of minimal travel-time matrix, dividing twenty-four hours into parts, taking into consideration an intensity of traffic - specific for a given city. It's very important that we can determine each of above matrix only one time - before first execution of RGMA or RGMIA. Therefore the time-complexity of algorithms which calculate these special matrices doesn't have an influence on time-complexity of RGMA and RGMIA.

Now, we can define  $H$  heuristics for our methods:  $H_{RGMA}$  and  $H_{RGMIA}$ .  
 $H_{RGMA}(s) = u$  iff:

1.  $\{o, u, t\} \in E$  and  $et(u)$  is minimal in first order and
2. value of  $t + D[o, u]$  is minimal or differs from minimal not greater than  $\epsilon$  ( $\epsilon$  is a parameter of closeness to  $o$  given on the input of the algorithm) in second order and
3.  $Q[o, u]$  is minimal in third order.

$H_{RGMA}(s) = u$  iff:

1.  $\{o, u, t\} \in E$  and  $et(u)$  is minimal in first order and
2. value  $t + T_{rh}[o, u]$  is minimal or differs from minimal not greater than  $\epsilon$  if parameter  $time_o$  belongs to rush hours or value  $t + T_{oh}[o, u]$  is minimal or differs from minimal not greater than  $\epsilon$  if parameter  $time_o$  doesn't belong to rush hours) in second order and
3.  $Q[o, u]$  is minimal in third order.

Intuitively  $H_{RGMA}$  heuristic may be a better heuristic than  $H_{RGMA}$  because of  $T_{rh}$  and  $T_{oh}$  matrices are time-dependent and give information about lower bound of travel time not only for first link in route but for a whole route. Matrix  $D$  used in  $H_{RGMA}$  is time-independent and therefore we can choose worse (then in RGMA) closest node in each step of RGMA. This intuitively observation was confirmed by experimental results on real data.

## 5. Experimental Results

There were a number of computer tests conducted on real data of transportation network in Warsaw city. This network consists of about 4200 stops, connected by about 240 bus, tram and metro lines. Values of common parameters for RGMA, RGMA and RGA were following:  $max_t = 5$ ,  $k = 3$ ,  $limit_w = 15$ . The value of  $limit_w$  is very important because it influences the density of network. The bigger value of  $limit_w$ , the more possibilities of walk links in a network. Density of network is of a key importance for time-complexity of algorithms. The parameter of closeness  $\epsilon$  in RGMA and RGMA was equal to 5 minutes. This parameter also has an influence on quality of results and time-complexity of these methods. The bigger value of  $\epsilon$ , the more possibilities of choice of closest node. We performed three kinds of tests. We examined routes from the centre of the city to the periphery of the city (set  $C - P$ ), routes from the periphery of the city to the centre of the city (set  $P - C$ ) and routes from the periphery of the city to the periphery of the city (set  $P - P$ ). Each of these sets includes 50 specification of first ( $o$ ) and last ( $d$ ) stops in the route which are difficult cases for each algorithm. First matter is a long distance from  $o$  to  $d$  (in  $P - P$



set), the second is a high density of the network in  $o$  or  $d$  localization (in  $C - P$  and  $P - C$  sets).

Selected results of tests for 10 chosen specification of  $o$  and  $d$  for each examined set of routes, generated by RGMIA, RGMA and RGGGA, are presented in Tab. 1, 2, 3. For each of algorithm we show in these tables:  $o$  and  $d$  specification, minimal travel-time for  $k$  resultant routes (RGMIA-t, RGGGA-t, RGMA-t), number of changes for route with minimal travel-time (RGMIA-ch, RGGGA-ch, RGMA-ch).

**Table 1.** The results for routes from set  $P - C$ ;  $time_s = 7 : 30$ ;  $o =$  Skolimowska 02

$d$ -destination stop	RGMIA-t	RGGGA-t	RGMA-t	RGMIA-ch	RGGGA-ch	RGMA-ch
Pl.na Rozdrożu 01(al. Ujazd.)	57	59	61	2	2	2
Pl.Konstytucji 04(Piękna)	62	63	71	2	2	2
GUS 08(Wawelska)	60	62	66	3	1	1
Dw. Centr.17(Świętokrz.)	64	66	69	2	1	1
Mennica 01(Grzybowska)	79	80	81	3	3	3
Ordynacka 02(Nowy Świat)	62	66	64	2	2	2
Siedmiogrodzka 02(Grzybowska)	78	80	82	3	3	3
Młynarska 04(Młynarska)	71	79	81	3	3	3
Nowy Świat 04(Świętokrzyska)	65	67	69	2	2	2
Emilii Platter(PLN. Dw. Central)	71	73	73	3	3	3

**Table 2.** The results for routes from set  $C - P$ ;  $time_s = 15 : 30$ ;  $o =$  Mennica 01 (Grzybowska)

$d$ -destination stop	RGMIA-t	RGGGA-t	RGMA-t	RGMIA-ch	RGGGA-ch	RGMA-ch
Plantanowa(Młochów)	104	112	124	3	4	4
Ogodowa 01(Głusków)	82	83	97	3	3	3
Kępa Okrzewska 01	64	84	88	3	4	4
Dziechciniec-Sklep02	103	86	88	3	3	3
Wąska(Józefów)	81	81	84	2	3	3
Struga 02(Marki)	63	64	63	2	2	2
Stokrotki 02(Nieporęt)	75	78	83	3	2	2
Orzechowa 02(Łopuszańska)	35	37	44	3	3	2
Długa 02(Dawidy)	67	73	84	3	3	3
3 Maja(Legionowo)	66	68	71	3	2	3

All results presented in above tables are confirmation of good quality of routes of RGMIA algorithm because the values of travel-time for the best and worst route are significantly less than for other comparable method. Generally, for  $P - C$  set, in

**Table 3.** The results for routes from set  $P - P$ ;  $time_s = 7 : 30$ ;  $o =$  Struga 01(Marki)

$d$ -destination stop	RGMA-t	RGMA-t	RGGA-t	RGMA-ch	RGMA-ch	RGGA-ch
Plantowa(Pruszków)	76	88	91	3	4	4
Ogrogowa 01(Głusków)	117	137	149	4	4	4
Kępa Okrzewska 01	103	123	138	5	5	5
Dziechciniec-Sklep 02	99	149	181	3	4	5
Mennica 01(Grzybowska)	79	80	81	3	3	3
Wąska(Józefów)	96	108	111	3	4	4
Stokrotki 02(Nieporęt)	77	84	93	0	0	4
Orzechowa 02(Łopuszańska)	75	78	81	4	5	5
Długa 02(Dawidy)	105	106	120	5	4	4
3 Maja(Legionowo)	82	87	96	3	4	4

29 cases of  $o - d$  specification RGMA generates the best routes, in 17 cases RGGA was the best and only in 4 cases RGMA resultant routes were the best. For  $P - C$  set, in 26 of cases of  $o - d$  specification RGMA generates the best routes, in 21 of cases RGGA was the best and only in 3 cases RGMA resultant routes were the best. For  $P - C$  set, in 27 of cases of  $o - d$  specification RGMA generates the best routes, in 23 of cases RGGA was the best and only in 0 cases RGMA resultant routes were the best.

The last experiment was focused on comparison of time complexity of algorithms. The results are presented in Fig. 4.

In this experiment we tested examples of routes with a minimal number of stops, between 22 and 47. On the horizontal axis there are points representing the minimal number of stops on a route. These values were computed as a result of standard *BFS* graph search method and they are correlated with difficulty of the route. On the vertical axis there is marked time of execution in ms (processor Pentium 3.0 GHz). Each possible route with a given number of the minimal number of bus stops was tested by two algorithms at starting time at 7:30 a.m., weekday. The executing time of algorithms was averaged over every tested routes. We can see that RGMA performs in significantly shorter time than RGMA and insignificantly than RGGA, specially for routes with minimal number of stops greater than 35.

We can conclude on the base of our experiments that RGMA returns results better then RGMA and even RGGA and is significantly faster than its previous version.

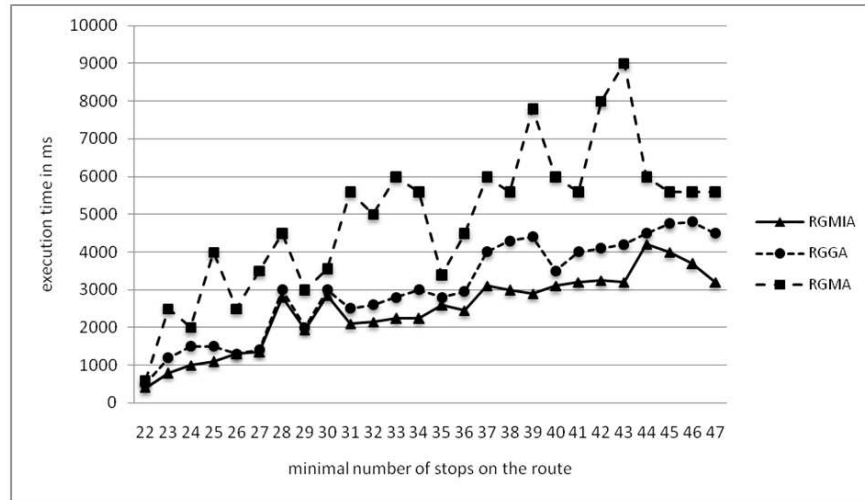


Fig. 4. The comparison of time-complexity of RGMIA, RGMA and RGGA

## 6. Conclusions

The author's motivation was to try to improve the RGMA in two aspects: the quality of resultant routes and time-complexity[10]. Computer experiments have shown that RGMIA - improved version of RGMA performs much more better than RGMA and significantly faster and is unexpectedly better than RGGA in both examined aspects.

Future work will be concentrated on testing RGMIA and RGGA on another transport networks for big metropolises which have different size, density and topology than network for Warsaw topology, such as Gornoslaski Okrag Przemyslowy (Silesian Industrial Region). The transport network of this region is very special because it consists of many big cities (hubs) connected by very rare fragment of network. If tests show poor performance of RGMIA or/and RGGA the new heuristics must be added to the algorithm. The proposal of improvement which can be considered includes to the algorithm information about geographic location of start and destination stops.

## References

- [1] Ahuja, R. K., Orlin, J. B., Pallotino, S., Scutella, M.G.: Dynamic shortest path minimizing travel times and costs, *Networks*, 41 (4), 2003, pp. 197-205.

- [2] Ahuja R. K., Magnanti T. L., Orlin, J. B.: *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 2008.
- [3] Bellman R. E.: On a Routing Problem, *Journal Quarterly of Applied Mathematics* 16, 1958, pp. 87-90.
- [4] Chabini I.: Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time, *Journal Transportation Research Records*, 1998, pp. 170-175.
- [5] Cooke K.L., Halsey, E.: The shortest route through a network with time-dependent intermodal transit times, *Journal Math. Anal.Appl.*, 14, 1996, pp. 493-498.
- [6] Dreyfus, S.E.: An Appraisal of Some Shortest-path Algorithms, In *Journal Operations Research*, vol.17, 1969, pp. 395-412.
- [7] Hansen P.: Bicriterion path problems. In *Multicriteria decision making: theory and applications*, *Lecture Notes in Economics and Mathematical Systems* 177, Eds. G. Fandel, T. Gal. Heidelberg, Springer-Verlag 1980, pp. 236-245.
- [8] Koszelew, J.: The Theoretical Framework of Optimization of Public Transport Travel, In: *Proceedings of 6th International Conference on Computer Information Systems and Industrial Management Applications: CISIM 2007*, IEEE Computer Society, 2007, pp. 65-70.
- [9] Koszelew, J.: Approximation method to route generation in public transportation network, *Polish Journal Environment Studies*, Vol.17, Nr 4C, 2008, pp. 418-422.
- [10] Koszelew, J., Piwonska, A.: A new genetic algorithm for optimal routes generation in public transport network, *Proceedings of 13th International Conference on System Modelling Control: SMC'2009*, Lodz University of Technology, 2009.
- [11] Lawler, E. L.: A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem, *Management Science* 18, 1972, pp. 401-405.
- [12] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs* WNT, Warsaw, 1996.
- [13] Orda, A. and Rom, R.: Shortest path and minimum - delay algorithms in networks with time-dependent edge-length, In *Journal Assoc. Computer Mach.*, 37(3), 1990, pp. 607-625.
- [14] Sung, K., Bell, M.G.H., Seong, M. and Park, S.: Shortest paths in a network with time-dependent flow speeds, *European Journal Operational Research*, 121(1), 2000, pp. 32-39.
- [15] WU, Q., Hartley J. K.: Accommodating User Preferences in the Optimization of Public Transport Travel, *International Journal of Simulation Systems*, Science

and Technology: Applied Modeling and Simulation, Vol.5, No 3-4, 2004, pp. 12-25.

- [16] Wikipedia, the free encyclopedia [[http://en.wikipedia.org/wiki/Intelligent\\_transportation\\_system](http://en.wikipedia.org/wiki/Intelligent_transportation_system)].

## **POPRAWIONA WERSJA PEWNEGO APROKSYMACYJNEGO ALGORYTMU GENERUJĄCEGO OPTYMALNE TRASY W SIECI TRANSPORTU PUBLICZNEGO**

**Streszczenie** Artykuł zawiera opis poprawionej wersji algorytmu generującego optymalne trasy w sieci transportu publicznego uzupełnionej o linki piesze, nazywanego przez autora Routes Generation Matrix Improved Algorithm (RGMIA). Trasy generowane przez RGMIA są optymalne pod względem czasu realizacji i mogą zawierać odcinki piesze, co sprawia, że wynikowe ścieżki są bardziej praktyczne i mogą spełniać określone preferencje użytkowników środków transportu. Algorytm został zaimplementowany i przetestowany na danych realnej sieci transportowej. Efektywność poprawionej metody została porównana w dwóch aspektach: złożoności czasowej i jakości wynikowych tras, z poprzednią wersją algorytmu nazwaną Routes Generation Matrix Algorithm (RGMA) oraz z metodą genetyczną Routes Generation Genetic Algorithm (RGGGA). Algorytmy RGMA oraz RGGGA zostały opisane w poprzednich artykułach autora [9,10].

**Słowa kluczowe:** sieć transportu publicznego, problem najkrótszych ścieżek zmiennych w czasie, optymalne trasy, algorytm genetyczny

Artykuł zrealizowano w ramach pracy badawczej S/WI/3/08