MATEC Web of Conferences **81**, 02005 (2016) DOI: 10.1051/matecconf/20168102005
ICTTE 2016

Maneuvering Route Safety Planning for Military Vehicles during Wartime

Tan Zhao, Jincai Huang, Guangquan Cheng, Chao Chen and Jianmai Shi

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, P. R. China.

Abstract. Maneuvering route planning for the military vehicles during wartime is a noteworthy problem in the field of military transportation. Existing research paid little attention to security of vehicles, which we consider as a non-ignorable aspect with the military background. Based on scenarios, we build a k-th shortest paths planning model aiming at vehicles safety improving. A genetic algorithm with specific encoding rules is proposed. And two groups of computational experiments based on the real road network of Fujian, China are performed. The results of the experiments show this is an effective algorithm to improve the vehicles' safety during maneuvering.

1 Introduction

Vehicle route planning is a traditional problem in communication and transportation area. With regard to the military application, it is indeed an intractable problem for commanders to plan the military vehicles' maneuvering routes during wartime, on account of the effects of weathers, road statuses, operations of enemies, etc. The operations of enemies, especially, turn this problem completely different from a normal route planning problem, considering the enemies' capability to destroy the traveling vehicles, which interrupts the course of action (COA) significantly. We can present the Military Vehicles Route Planning during Wartime (MVRPW) problem as follows: design a set of routes for a fleet of i identical military vehicles required to transfer to j preliminary positions in preparation for the forthcoming mission. On the premise that all vehicles arrive on schedule, our scheme shall ensure vehicles' security as possible, in all considered scenarios.

With respect to route planning for military vehicles, Guan, L. [1] put forward a two-stage algorithm to get the optimal vehicles distribution for multitask using types of military vehicles with the highest efficiency. Zhong, Liu, et al. [2] presented a multiple destinations route planning algorithm based on dynamic programming considering the target values. Their focus is the efficiency of vehicles' allocation. YiePinedo, Ruben D. [3] studied a problem of routing several vehicles from multiple origins to multiple destinations in an unsafe environment, and presented solution to allocate zone-limited escorts to minimize the average threat level. The same as what they did, vehicles' security was stressed in our paper. To ensure the course of action not breaking, avoiding the enemies' attack and ensuring the maneuvering action as safe as possible is our main purpose.

In view of our military background, we are supposed to provide a feasible scheme at the very beginning of a mission, rather than determine the next road in every intersection for vehicles. Thus, a model having each vehicle's path specified is required. An effective solution is the k-th shortest path technique, which is able to give the shortest k paths from sources to destinations intactly for each vehicle. The applications of this technique can be seen in the emergency evacuation field. It is first seen in the dissertation of Campos and Netto [4]. They applied the k-th shortest path to their model, aiming at a larger traffic capacity and shorter clearance time. Alexander and Smith [5] combined the k-th shortest path with the M/G/c/c queuing theory in their model to deal with congestion and time delay. He, Yunyue, *et al*. [6] proposed a k-th-shortest-path-based technique that uses explicit congestion control to optimize evacuation routing and police resource allocation. Their objection is to minimize the overall clearance time in city emergency evacuation. Noticing that few researchers apply this technique in military field, we attempt to use the k-th shortest path tech in our military vehicles route planning problem.

With an eye to the complexity in battlefield, kinds of situations exist and we may not know what situation comes next. It is a great challenge that all situations are supposed to be considered in one scheme. Inspired by work of Zhang, Ying, *et al*. [7], we will consider situations as discrete scenarios in this paper. Scenario describes the future situation may occur, and also presents the uncertainty of reality [8]. A scenario, with specific chance of happening, specifies states of all factors in a situation. Our purpose is to search for an eclectic scheme, not optimal in a certain scenario though, performing robustly in scenarios.

In the rest of this paper, we propose a scenario-based integer programming model with application of k-th shortest path technique for MVRPW problem in section 2; to solve this model, a genetic algorithm with a novel genetic encoding rule is introduced in section 3; and in section 4, we give two computational examples based on a real road network; section 5 concludes the discussion.

2 K-shortest-path-based model for MVRPW

In this section, we introduce the problem description, underlying assumptions and mathematic formulation.

This problem is a type of optimal routing problem (ORP) with consideration of multiple scenarios. The objective of the model is to maximize the overall security, indicated by expectation of the safe arrival of all vehicles in all scenarios. We present a transportation network consisting of origins (e.g. bases or campsites), destinations (i.e. the positions) and related road links, and intersections in this problem. The military vehicles are supposed to maneuver from origins to destinations through this network, with consideration of safe passing probabilities (SPPs, considering enemies' disruption) of each road link respectively in each scenario. Meanwhile, a mission deadline is considered, before which the vehicles shall accomplish the maneuver mission.

It is a tough job to describe a scenario exactly, as so many dimensions or factors should be considered. Recognizing it is not our main work, we can present the scenario in a brief way. While saying a scenario, we suppose that all statuses in all dimensions are set. In other word, a scenario is an intersection of statuses in each dimension. For example, 2 dimensions are considered to describe a scenario. Fig. 1 shows the scenarios we will consider in this example. Besides, the probability of a scenario occurring is assumed to be known. In this way, considering all dimensions, we can present scenarios S as S = {s1, s2, s3, …} with probabilities P = {P1, P2, P3, …}, one to one correspondence.

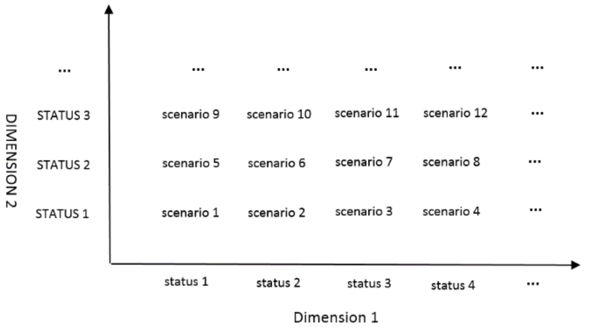

Figure 1. An example to explain the scenario considered, 2 dimensions.

2.1 Assumptions

To highlight the nature of the problem and clarify the boundaries of the problem concerned, some rational assumptions are proposed.

1) The SPPs vary based on scenarios.

Without describing the scenarios comprehensively though, it is supposed that we can still get the safe passing probability (i.e. SPP) of each road link in each scenario via simulation technique and the analysis of practice data. As scenarios are different, the SPP of each road link varies.

2) The travelling time and capacity of each road link are set constant.

Taking no account of road destruction, the travelling time shows little different when scenario changes. What is noteworthy, the capacity of each road link is a value set by the commander rather than an objective value. Considering the practical bearing capacity of a road is meaningless in the case of confrontation. We suppose that the commander will integrate all factors into account and set clear numbers as road capacity values. Yang, Ping et al [9] figured that uniform traffic flow in each road brings a rise of elusiveness of the whole road network. Thus the value of each road capacity shall not be set large.

3) Each destination can accommodate 1 vehicle at most.

Military vehicles, such as missile launchers and movable radars, are usually able to execute assignments independently. To ensure the safety of vehicles and to reduce losses after arrival, sending vehicles to separate terminals is a bright choice. In this paper, we suppose that each destination can accommodate 1 vehicle at most.

2.2 Notation

We propose integer programming to describe the MVRPW problem. The notation used in the problem formulation is introduced below:

Indices. i an index for military vehicles, $i= 1,2,...,I$, I is the total number of vehicles;

j an index for destinations, $j= 1,2,...,J$, J is the total number of destinations ;

k an index for k-th-shortest paths, $k= 1,2,...,K_{ij}$, K_{ij} is the total number of shortest paths vehicle i can travel to destination j, set or get via limitation;

s an index for scenarios, $s=1,2, ...,S$, S is the total number of scenarios considered;

l an index for road links, $l= 1,2,...,L$, L is the total number of road links;

n an index for road nodes, $n= 1,2,...,N$, N is the total number of nodes in the transportation network;

Parameters. λ_l the capacity of road link l;

t_l the travelling time of road link l;

R_{ijk} the set of road links, which the k-th shortest path vehicle i travelling to destination j covers;

P_s the probability scenario s occurs, $\sum_{s=1}^{S} P_s = 1$;

p_{sl} the SPP of road link l in scenario s;

P_{sijk} the SPP of the k-th shortest path vehicle i travelling to destination j in scenario s;

T_0 time allowed for maneuver(i.e. deadline);

Decision variables. x_{ijk} 0–1 variable. 1 for vehicle i travels to destination j in the k-th shortest path, 0 for otherwise.

2.3 Formulation

We formulate the problem as follows:

$$\text{(MVRPW) max } Z=\sum_{s} P_s \sum_{i}\sum_{j}\sum_{k} x_{ijk} P_{sijk} \tag{1}$$

Subject to:

$$x_{ijk} \sum_{l\in R_{ijk}} t_l \le T_0 \text{ , } \forall i.j,k \tag{2}$$

$$\sum_{j}\sum_{k} x_{ijk} = 1 \text{ , } \forall i \in I \tag{3}$$

$$\sum_{i}\sum_{k} x_{ijk} \le 1 \text{ , } \forall j \in J \tag{4}$$

$$\sum_{l\in R_{ijk}} x_{ijk} \le \lambda_l \text{ , } \forall i,j,k,l \tag{5}$$

$$P_{sijk} = \prod_{l\in R_{ijk}} p_{sl} \tag{6}$$

$$x_{ijk} \in \{0,1\} \text{ , } \forall i,j,k \tag{7}$$

The objective function of the MVRPW formulation is the sum of expectation of the safe arrival of all vehicles in all scenarios, indicating the robustness of the scheme. Constraints (2) ensure that all vehicles arrive on schedule. Constraints (3) represent that each vehicle arrive at a destination. Constraints (4) ensure that each destination can accommodate 1 vehicle at most, as noted in assumption 3). Constraints (5) show that the traffic flow on road link l shall not exceed the set capacity, as assumption 2) implies. Constraints (6) are based on the definition of P_{sijk}, representing the safe passing probability of the k-th shortest path in which vehicle i travels to destination j in scenario s. Constraints (7) indicate that the decision variables x_{ijk} are 0–1 variables. This is a typical Integer Programming Problem (IPP). We try to solve it with a genetic algorithm.

3 Genetic algorithm for MVRPW

In this section, genetic operators which are used in GA for MVRPW will be explained.

The first application of GA to transportation/distribution problem is the study of Michalewicz, Z [10] in 1991. They used matrix-based representation to represent transportation tree. They used $|K|$ and $|J|$ to represent the number of sources and depots respectively, the dimension of matrix is $|K|\cdot|J|$. While Gen, M. and Cheng, R [11] applied *Prüfer* number to represent transportation tree in 2000, it successfully reduced the digits to represent a transportation tree with $|K|$ sources and $|J|$ depots to $|K|+|J|$-2. Altiparmak, Fulya, et al. [12] developed a priority-based encoding from research of While Gen, M. and Cheng, R in 2006. In priority-based encoding, the position of a gene is used to represent a node (source/depot in transportation network), and the value is used to represent the priority of corresponding node for constructing a tree among candidates. A similar encoding rule is applied in our paper.

3.1 Encoding algorithm

In our encoding rule, I identical vehicles are requested to maneuver to J destinations ($J \ge I$). Here, we set J as D. If we set the sum of origins as O ($O \ge 1$, and for one origin can accommodate more than one vehicle, $I \ge O$), the length of our chromosome is set $O + D$. The front O genes represent the origin nodes, while the rear D genes the destination nodes. The gene values represent priorities of origins and destinations to obtain transportation tree. The transportation tree corresponding with a given chromosome is generated by sequential arc appending between origins and destinations. At each step, only one arc is added to tree， selecting an origin/destination with the highest priority and connecting it to a destination/origin through a k-th shortest path considering the biggest *SPP*, with the capacity of road link and deadline time in account. A length-equal operator is given to each chromosome, called *surplus*. The *surplus* shows the remaining capability to send/accommodate vehicles of an origin/destination. For each time a military vehicle is sent/accommodated, the corresponding value of *surplus* shall be reduced by 1. When a value of *surplus* reduces to 0, showing this origin/destination has no ability to send/accommodate a vehicle more, we set the corresponding gene as 0.

An example showing how the selecting procedure works is given in Fig. 2. In this example, 2 origins and 5 destinations are considered. There are 1 and 2 military vehicles waiting to maneuver from the two origins separately. With time and road capacity constraints considered, Table. 1 shows the decoding algorithm for the priority-based encoding in detail.

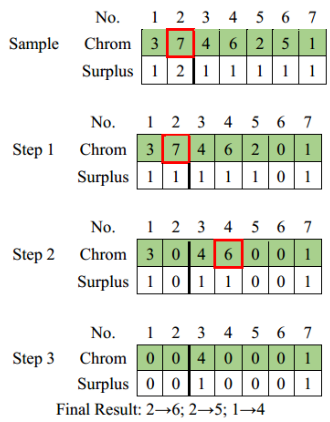

Figure 2. An example to show the selecting procedure.

Table. 1. Pseudo-code of the decoding algorithm.

Procedure: Decoding of the chromosome for transportation tree

Input:	O	:	set of origins;
	D	:	set of destinations;
	a_o	:	capacity of origin o, $\forall o \in O$;
	b_d	:	accommodation of destination j, b_d =1, $\forall d \in D$;
	λ	:	an integer matrix, showing the road capacity between two adjacent nodes;
	c	:	chromosome;
	Time	:	deadline time;

DOI: 10.1051/matecconf/20168102005

T_{odk} : time cost from origin o to destination d through the k-shortest path, $\forall o,d,k$;

R_{odk} : the route from origin o to destination d through the k-shortest path, $\forall o,d,k$;

SPP : the set of safe passing probabilities of all k-shortest paths from origins to destinations, with scenarios already considered in.

P_s : probability scenario s will happen, $\forall s \in S$

Output: M : an O-by-J matrix. //If $M_{od} = k$, a vehicle is arranged through the k-shortest path from origin o to destination d; if M_{od} =0, no vehicle is arranged from o to d.

Index : *Index* shows the advantage of the chromosome. The bigger *Index* is, the more robust the scheme is.

Step 1. M_{od} ←0 , $\forall o \in O, \forall d \in D$, *surplus*←[a, b], *Index*←0;//initialize

Step 2.*mark*←arg max$\{c(i) \mid i \in O+D\}$;//select a node

Step 3.**if** $mark \in O$,**then** o^*←*mark*;//select a origin

Select an available path to a destination with the highest *SPP.*

$$d^*,k^* \leftarrow \arg\max\{ \sum_S P_s \cdot SPP_{o^*dk} \mid T_{o^*dk} \le Time ,$$

$$\prod_{(w,v)\in R_{o^*dk}} \lambda_{wv} > 0 , c(d) \neq 0, \forall d,k \};$$

else d^*← *mark*;//select a destination

Select an available path from an origin with the highest *SPP.*

$$o^*,k^* \leftarrow \arg\max\{ \sum_S P_s \cdot SPP_{od^*k} \mid T_{od^*k} \le Time ,$$

$$\prod_{(w,v)\in R_{od^*k}} \lambda_{wv} > 0 , c(o) \neq 0, \forall o,k \};$$

Step 4.**if** d^*=**none**， **then return**; //If a vehicle from origin o^* cannot finds an available destination, this chromosome is infeasible.

else $M_{o^*d^*}$ ← k^*; //assign available path

Update capacity of origins and destinations, and λ .

surplus(o)*←*surplus(o*)* -1,

surplus(d)*←*surplus(d*)* -1;

$\lambda_{wv} \leftarrow \lambda_{wv}$ -1, $\forall (w,v) \in R_{o^*d^*k^*}$;

Step 5.**if** *surplus(i)* = 0, **then** *c(i)* ←0, $\forall i \in O+D$

Step 6.**if** *surplus(o)* = 0, $\forall o \in O$,

then *Index*←sum(SPP_{odk}, $\forall \{o,d,k \mid M_{od}=k\}$) and **return**,

else goto Step 2.

3.2 Genetic operators

3.2.1 Selection

In this GA, initial population is randomly generated. To get non-worse results for every generation, we adopt an elitist preservation strategy. In this strategy, we evaluate the SPPs of all chromosomes in population, and set them as the fitness of chromosomes. A fixed proportion, named *gap*, is given as the selected portion. The worst *gap* of the population, measured by fitness, will be selected. Thus an amount of *gap**population chromosomes are selected to take part in the following genetic operating, while an amount of (1-*gap*)*population of advantageous chromosomes are preserved.

9	5	1	3	7	4	2	10	8	6
10	5	4	6	3	8	7	2	1	9
After cross:									
9	5	1	6	3	8	7	10	*	*
10	5	*	3	7	4	2	*	1	9
Partial Mapping:									
9	5	1	6	3	8	7	10	4	2
10	5	8	3	7	4	2	6	1	9

Figure 3. A partial mapping crossover example.

3.2.2 Crossover

The crossover is done to explore new solution space. We employ a partial mapping crossover (PMX) operator. In this operator, we have two parents crossover. Two random positions are selected, and the segments between the two positions are interchanged. Then if repetitive values exist in the same chromosome, we keep the exchanged parts and non-redundant segments, and use the mapping crossover to eliminate the conflicts. An example is shown in Figure 3, assuming length of chromosome as 10, and the two random numbers as 4 and 7. In this figure, conflicted positions are marked by stars.

3.2.3 Mutation

Mutation is used to prevent the premature convergence and explore new solution space. The mutation operator is modifying the gene within a chromosome with a mutation chance. While mutating, two random positions of the chosen chromosome are selected, and the values of the two positions will be exchanged.

4 Computational examples

In this part, we apply the GA for MVRPW in the road network of Fujian Province, China, to evaluate how this algorithm performs. A road network of 200 nodes are selected, i.e. nodes 1-200 in Figure 4. The dataset is from an open resource [13]. The k-th shortest paths data are calculated by a commercial GIS software, ArcGIS, with application of a related algorithm [14]. Other related parameters, including S, P_s, p_{sl} and λ, are either set or generated with probabilities. We coded the algorithm in MATLAB R2013a, and executed the programs on a 32-bit Windows XP computer with 2.93GHz Dual-Core CPU and 3.0 GB of physical RAM. Two groups of experiments are performed as followed.

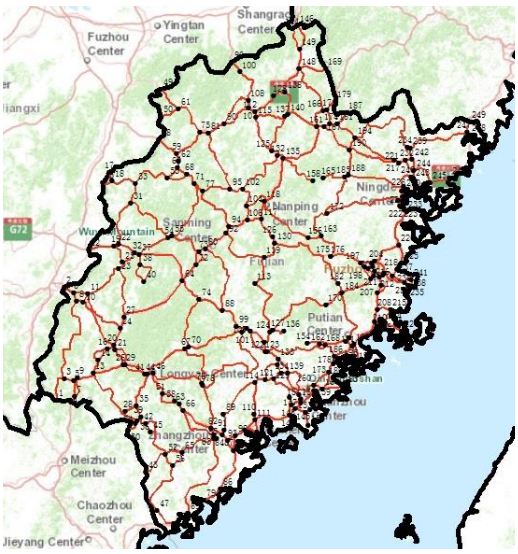

Figure 4. The Road Network of Fujian, China.

The speed of vehicles are all set 40km/h. Robustness of a scheme is indicated by the average SPP of vehicles. In experiment group I, 7 vehicles are arranged to maneuver from 3 origins to 26 destinations, with 10 shortest paths of each origin-destination pair considered; in experiment group II, 12 vehicles are arranged to maneuver from 6 origins to 30 destinations, with 20 shortest paths considered. 20 random scenarios are considered in both experiments. Thus, experiment in group II is around 4 times the k-th shortest paths scale of in group I. In the GA, the value of population is set 200, and the stopping criterion is if no better result appears in a continuous 15 generations. To evaluate the performances of our algorithm, 10 repetitions are carried out respectively in each experiment. Other experimental data and performances of the GA are listed in Table 2.

Table 2. Performances of the Algorithm.

Experiment	No.	Capacity of Road Links	Time permitted (h)	Proportion of Unavailable Paths (%)	Optimization Results			Computing Time(s)		
					max	min	average	max	min	average
I	1	[2,3;0.6,0.4] *	10	34.87	0.1392	0.1379	0.1390	437.3	264.4	342.4
	2	2	12	5.64	0.1592	0.1465	0.1567	571.0	373.7	514.5
	3	[2,3;0.6,0.4]	12	5.64	0.1592	0.1435	0.1555	948.8	374.2	676.7
II	1	2	8	43.11	0.0939	0.0930	0.0933	663.7	345.5	386.4
	2	2	10	26.47	0.0980	0.0936	0.0957	867.3	403.1	695.2
	3	[2,3;0.6,0.4]	10	26.47	0.0999	0.0938	0.0986	732.1	329.3	586.4
	4	[2,3;0.6,0.4]	12	5.08	0.1003	0.0935	0.0981	1096.2	621.4	915.9
	5	3	12	5.08	0.1008	0.0911	0.0962	1307.4	717.1	965.8

*[2,3;0.6,0.4] means element values are 2 with 60% probability or 3 with 40% probability.

According to Table 2, the algorithm can handle both the groups of experiments. The gap between the minimum and the maximum of optimization results varies. The biggest gap is 10.94%, appearing in experiment I-3. By this token, it is a relatively stable algorithm. As the constraints loosening, the maximum of optimization results increases steadily, indicating that better results are searched out. However, looser constraints brings a larger solution space, which results in a greater probability of local optimal solution. This can be observed from the enlarging gap between the minimum and the maximum. On account of that initial solutions are randomly generated, and the randomness in the genetic process as well, time cost in computing differs greatly. Time consumed is between 4.5 minutes to 22 minutes, which is an acceptable time cost in practical route planning.

Table 3. Changes of K-th shortest paths selected.

Part 1:Permitted Time=12h

Capacity	K-th Shortest Path Selected											
[1,2;0.1,0.9]	8	20	2	5	1	10	3	4	2	12	1	1
2	17	20	2	5	1	1	3	4	2	12	1	1
[2,3;0.6,0.4]	17	20	2	5	1	1	3	4	2	1	1	1
3	3	20	2	5	5	1	1	4	2	1	1	1
inf *	5	13	2	1	1	1	3	4	2	1	1	1

Part 2:Road Capacity=2

Permitted Time	K-th Shortest Path Selected											
8h	3	20	2	5	5	1	3	4	2	2	1	1
9h	17	20	2	5	1	1	1	4	2	2	1	1
10h	17	20	2	5	5	1	3	2	2	2	1	1
11h	17	20	11	5	1	1	3	4	2	2	1	1
12h	17	20	2	5	1	1	3	4	2	12	1	1

* *inf* means no constraint is set here.

To observe the changes of *k*-th shortest paths selected of each vehicle, we expand experiment group II. Listed in Table 3, we set the permitted travelling time and the road capacities constant respectively, to see how the shortest paths selected change. In part 1, permitted travelling time is set 12 hours. With the road capacity grows, an obvious trend is that vehicles tend to select the shortest path to travel. As in each scenario, SPP of each road link is set randomly, path with fewer road link is safer in common. And because no ultra-long road link exists in this road network, path with fewer road link is usually shorter as well. This explains the trend of selecting shortest paths with loosening road capacity constraint. While setting road capacity constant and varying the permitted time in Part 2, we get an opposite trend. More vehicles maneuver through a longer path. In that longer time is permitted, safer paths with longer distances get unlocked. This is why more vehicles are arranged to maneuver in a path with a long distance.

5 Conclusion

In this paper, we present a MVRPW model to describe the safety planning problem of military vehicles maneuvering during wartime. This integration optimization model allows decision makers to get a robust scheme to improve the security of vehicles with the consideration of scenarios. A GA algorithm with a novel encoding rule dealing with the selection problem of *k*-th shortest paths is presented. Two groups of computational examples based on a real road network are conducted to verify this algorithm. These experiments indicate that this algorithm is effective and relatively stable.

In the future, in order to search for better and more stable results, we will test other heuristic algorithms in the MVRPW model. Moreover, to improve the security of maneuvering military vehicles further, optimal allocation of protective resources will be considered in this problem.

Acknowledgment

This project was supported partially by National Natural Science Foundation of China (71471174).

References

1. L. Guan. "The Research of the Optimal Distribution for the Multitask using Types of Vehicles in the Military Road Transportation." International Conference on Transportation Engineering 2007 ASCE:539-544, (2012).
2. L. Zhong, et al. "A Task Assignment Algorithm for Multiple Aerial Vehicles to Attack Targets With Dynamic Values." IEEE Transactions on Intelligent Transportation Systems **14**.1:236-248, (2013).
3. R. D. YiePinedo. "Route optimization while improving safety using escort vehicles." Dissertations & Theses – Gradworks, (2013).
4. V.B.G. Campos, P.A. da Silva, P.O.B. Netto. "Evacuation transportation planning: A method of identify optimal independent routes." Urban Transport V, (2000):555-564.
5. A. Stepanov and M. G. Smith. "Multi-objective evacuation routing in transportation networks." European Journal of Operational Research, **198**.2:435-446, (2009).
6. Y. He, et al. "K-Shortest-Path-Based Evacuation Routing with Police Resource Allocation in City Transportation Networks." Plos One **10**.7, (2015).
7. Y. Zhang, et al. "A metaheuristic approach to the reliable location routing problem under disruptions." Transportation Research Part E Logistics & Transportation Review **83**:90-110, (2015).
8. S. T. Zhang, Y. J. Dou, and Q. S. Zhao. "Evaluation of Capability of Weapon System of Systems Based on Multi-Scenario." Advanced Materials Research 926-930:3806-3811, (2014).
9. P. Yang, W. D. Liu, and L. I. Ming yu. "An optimization model of transportation scheme in preoperational preparations of conventional missile." Fire Control & Command Control, (2007).
10. Z. Michalewicz. "A Non-Standard Genetic Algorithm for the Nonlinear Transportation Problem." Orsa Journal on Computing **3**:307-316, (1991).
11. M. Gen, R. Cheng. "Genetic algorithms and Engineering Optimization." New York: Wiley, (2000)
12. F. Altiparmak, et al. "A genetic algorithm approach for multi-objective optimization of supply chain networks." Computers & Industrial Engineering **51**.1:196-215, (2006).
13. Openstreetmap. Available from: http://www.openstreetmap.org.
14. H. Aljazzar, and S. Leue. "K *: A heuristic search algorithm for finding the k shortest paths." Artificial Intelligence **175**.18(2011):2129-2154.