Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 2865149, 10 pages
https://doi.org/10.1155/2018/2865149

Research Article

Route Planning for Military Ground Vehicles in Road Networks under Uncertain Battlefield Environment

Tan Zhao,[1] Jincai Huang ,[1] Jianmai Shi ,[1,2] and Chao Chen [1]

[1] *Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan 410073, China*
[2] *School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China*

Correspondence should be addressed to Jianmai Shi; jianmaishi@gmail.com

Received 16 September 2017; Accepted 19 December 2017; Published 21 January 2018

Academic Editor: Luigi Dell'Olio

Route planning for military ground vehicles in the uncertain battlefield is a special kind of route planning problem, as the military vehicles face a great of uncertain and unpredicted attacks. This paper models these uncertainties in the road network by a set of discrete scenarios. A kth shortest-path method is introduced to find intact routes from the origin to the destination for each vehicle. A binary integer programming is presented to formulate the problem. As the combination of the uncertainties results in a huge number of scenarios, we employed the sample average approximation method to obtain a robust solution for the problem. The solution approach is illustrated and tested through three road networks with different scales. The computational results show that, for networks of small scale, our method can provide a good solution with a sample of small size, while, for the large network, with sample of small size, this method usually leads to a suboptimal solution, but a good solution can still be obtained as the sample size grows bigger. In addition, variation trend of the deviation with different sample size indicates that a sample of larger size can bring more stability to the results.

1. Introduction

During wartime, Military Ground Vehicles (MGVs) usually have to maneuver out of their rear bases or concealed places (origins), pass through a complex road network, and arrive at the predetermined areas (destinations) to complete their operational missions, as in Figure 1. For example, mobile missile launchers depart from the military bases and move to the missile launch sites to complete the launch mission. Generally, origins are well fortified and disguised, which can ensure the safety of the MGVs. However, when the MGVs depart from the origins and move on the road network, they may be detected by the enemy's reconnaissance sensors. Because of the lack of fortification and camouflage resources, MGVs are exposed to man-made attacks during their movement in the road network. Such attacks, mainly arising from ground assault and air raid, are likely to occur frequently in battlefield. In fact, according to South Asia Terrorism Portal (SATP) [1], there were 309 attacks to the North Atlantic Treaty Organization (NATO) supply transport in Pakistan from 2008 to 2014. The number of deaths due to such attacks attains to 143. The problem of maneuvering in networks prone to attacks is prominently important due to the numerous zones that are currently at war around the world. According to the data given by the Global Security Organization (GSO) [2] there were 35 major ongoing wars and more than 25 minor conflicts around the world by the end of 2015. All these regions are highly prone to attacks where enemies perform attacks to military staff and vehicles travelling in the road networks. Thus, properly planning routes for MGVs in uncertain battlefield road networks, so as to reduce the risk of being attacked, is an important issue worth studying in the military field.

A classical problem related to MGVs' route planning is the Convoy Movement Problem (CMP) [3–5]. In the CMP, there is a set of convoys with a specific origin-and-destination (OD) pair. The objective is to find the routes for every convoy that minimize the total overall travel time while adhering to some strategic constraints [6]. Time is the most important factor in this problem since present-day military

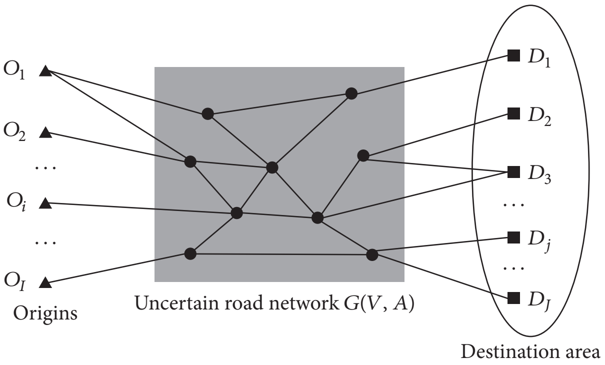


Figure 1: An illustration of MGVs' maneuvering.

engagements emphasize the need for mobility more than ever [3]. However, if we consider route planning associated with a mission, MGVs need not accomplish the maneuver mission as early as possible but must reach destinations before the mission deadline. Thus, in this paper we will consider the routing time of each MGV as a strong constraint rather than the objective for costs. Given the deadline of the mission, all MGVs involved in the maneuvering are requested to arrive at their destinations in a time no longer then this timeline.

Another important feature of the studied MGV routing problem is that the destination for each MGV is not deterministic and there are a number of candidate destinations for the MGVs to choose, which is quite different from the one-to-one OD pairs in CMP. Destination location is a critical decision for MGVs, which can be coordinately optimized with the routes. This is a practical requirement in military operations; for example, there usually are a set of potential launch areas for the mobile missile launchers to complete their mission and the final launch destination has to be determined according to the situation of the battlefield and the road network. Thus, the MGV routing problem includes two main parts: for each vehicle, select a destination among the candidate locations and a route among the kth-shortest paths between its origin and destination.

The emergency evacuation problem is concerned with shelter selection and route planning for evacuates, which displaces people from evacuation sources (areas suffered natural or human-made disasters) and transports them to evacuation destinations (safe shelters) [7]. The evacuation problem is related to our research in the location and routing decisions, but most of the work in this field needs not to consider the uncertain attacks during the movement in the road network. A comprehensive review of this topic can be seen in the work of Altay and Green III [8] and Galindo and Batta [9].

In the evacuation problem, there are usually a couple of candidate shelters and a given road network with deterministic delivery capacity in each road arc. The decision maker has to determine the suitable shelters and choose paths for all evacuees to optimize the evacuation cost and efficiency. Yamada [10] modeled the city road network as an undirected graph, and by solving a shortest-path problem on this graph, they obtained the optimal evacuation plan for assigning each resident in the city to one of the places for refuge. Cova and Johnson [11] focused on traffic delays occurring at intersections. To reduce these delays, they presented a network flow model for identifying optimal lane-based evacuation routing plans in a complex road network. Besides network-flow-based approach, the kth-shortest-path method is another method widely employed in the evacuation problem. Campos et al. [12] first presented this method for vehicle flow allocation in emergency transportation planning. Their goal is to send a greater number of vehicles in a minimum time outside a region under menace of some catastrophic event. Stepanov and Smith [13] applied the $M/G/c/c$ state dependent queuing models to cope with congestion and time delays on arcs. In their model, the set of feasible and potential egress routes is defined with the kth-shortest-path algorithm. He et al. [7] presented a k-shortest-path-based evacuation routing method with consideration of police resource allocated in transportation network. They proposed a nonlinear mixed integer programming to minimize overall evacuation clearance time. In general, the main purpose of the evacuation is to minimize the overall travelling time, and the uncertainty in the evacuation problem usually arises from the congestions and time delay occurring at intersections or arcs, which is different from the uncertain attacks considered in this paper.

To the best of our knowledge, existing studies rarely considered the arcs' uncertainty that arcs may be disrupted or travelling vehicles may be attacked during moving on the arc. In wartime, it is important for MGVs, such as missile launchers and movable radars, to quickly and safely arrive at their destination for completing the mission. In this work, we model the road uncertainty by a set of discrete scenarios. Each scenario, occurring with a specific probability, specifies a subset of arcs that become potentially insecure, on which MGVs travelling may be destroyed by intentional attacks. The optimization objective is to maximize the expectation that all MGVs accomplish the maneuver mission securely.

The main contributions of this paper are as follows.

We employ the k-shortest-path method to describe the scenario-based route planning problem. A binary integer programming (BIP) model is developed for the MGVs' route planning in battlefield road network, considering arcs' insecurity and the uncertainty in travelling time.

In each scenario, the instance of the problem is deterministic and can be solved by CPLEX solver [14]. To cope with the huge number of scenarios caused by uncertainties in the road network, we adapt the sample average approximation (SAA) method to address the computational complexity and propose a solution approach based on SAA.

Three road networks with different scales are presented to test the performance of the SAA-based algorithm. Sensitivity analysis on the sample size is conducted to search for a balance between efficiency and accuracy.

The rest of this paper is organized as follows. First we present the formulation of the problem in Section 2; next we introduce the SAA background and describe the solving process based on SAA in Section 3; then the computational experiments are presented and results are discussed in Section 4; the conclusion is given in the final section.

2. Model Formulation

In this section, we present the problem description in detail, the underlying assumptions, and the model formulation.

We define the problem on an undirected road network graph $G = (V, A)$, where V is the node set and A is the set of arcs. There are a number of I origins for the MGVs, and a set of J potential destinations. The total number of MGVs in all origins is W, which satisfies $I \leqslant W \leqslant J$. The security of arcs is assumed to be independent. Considering a route L, containing arcs $l_1, l_2, \ldots, l_n$, with secure probability (i.e., the probability a vehicle travels through the arc safely) $p_1, p_2, \ldots, p_n$, respectively, the secure probability of route L can be formulated as

$$P_L = \prod_{l=1}^{n} p_l. \quad (1)$$

We are supposed to select a route for each MGV so that the sum of all routes' P_L is maximized. We employ the kth-shortest-path method to help find the possible routes for each OD pair in the road network, based on which the routing problem of MGV is transformed into MGV allocation problem. Through allocation of MGVs to different kth-shortest paths of OD pairs, their route and destinations can be optimized to maximize the overall security.

2.1. Assumptions. In order to facilitate the formulation of the problem, several underlying assumptions are proposed.

There are two statuses for each arc in the road network: secure and insecure. For an arc in the secure status, MGVs can safely travel through the arc, while, for an arc in the insecure status, there is a probability that the MGV is attacked.

Each road arc is associated with a capacity limiting the maximum number of MGVs that can travel through. We consider the capacity of each arc is limited in the whole maneuvering process. Unlike the hypothesis in CMP [3] that no pair of convoys is allowed to occupy the same part of the network at the same time, we employ the vehicle hiding description presented by Yang et al. [15] as the hypothesis. In their work, they set $v_{ij}(P)$ as standardized value of the flow between node i and j in a given transport program P, and the information entropy between node i and j is expressed as

$$I\left(v_{ij}(P)\right) = -\sum_{i}\sum_{j \neq i} v_{ij}(P) \ln v_{ij}(P). \quad (2)$$

It is observed that the information entropy $I(v_{ij}(P))$ increases along with the $v_{ij}(P)$ decreasing. If all vehicles are distributed to different roads, we can obtain higher network information entropy, which also provides a better performance in elusiveness. Therefore, the capacity of each arc is assumed to be artificially limited in the entire maneuver process in our paper.

The MGVs considered are identical and are able to execute missions independently. To ensure the safety of vehicles and reduce losses during the moving process, sending vehicles to separate terminals is a bright choice. In this paper, we suppose that each destination can accommodate one vehicle at most.

As destinations are also assumed to be homogenous in this paper, the main issue we need to focus on is to select a route to an alternative destination for each vehicle.

2.2. Notations. To describe the problem more clearly, the notations used in the model are presented as follows.

Indices

i: index of origins, and $i = 1, 2, \ldots, I$, where I is the total number of origins;

j: index of alternative destination, and $j = 1, 2, \ldots, J$, where J is the total number of destinations;

k: index of kth-shortest path, and for each pair of OD $k = 1, 2, \ldots, K_{ij}$, where K_{ij} is the total number of shortest paths considered between origin i and destination j;

l: index of arc s, and $l = 1, 2, \ldots, L$, where L is the total number of arcs involved in the road network;

s: index of scenarios, and $s = 1, 2, \ldots, S$, where S is the total number of scenarios considered.

Parameters

w_i: the number of MGVs dispatched from origin i;

p_s: the probability that scenario s occurs;

q_l: the probability that arc l is in insecure state;

p_{sl}: the probability that a vehicle travels through arc l safely in scenario s, where the arc l can be either secure or insecure;

θ_l: the probability a vehicle travels through the insecure arc l safely;

a_{sl}: 1 if arc l is in secure state in scenario s, 0 otherwise;

P_{ijks}: the secure probability of the kth-shortest path from origin i to destination j in scenario s;

R_{ijk}: a set of arcs, composed of all arcs in the kth-shortest path from origin i to destination j;

t_{sl}: time consumed for travelling through arc l in scenario s, which is obtained considering distance of the arc and the vehicle velocity in the specific scenario;

T_e: time length of the mission, that is, the time allowed to finish the movement;

α_{lijk}: 1 if arc l is included in the kth-shortest path from origin i to destination j, 0 for otherwise;

λ_l: the capacity of arc l.

Decision Variable

x_{ijk}: 0-1 variable. 1 for dispatching a vehicle from origin i to destination j through the kth-shortest path, 0 for otherwise.

As each scenario specifies a subset of arcs which become insecure, given scenario s (in which a_{sl}, $\forall l$ are known), the probability that it occurs can be obtained through the following function:

$$p_s = \prod_{l \in L: a_{sl}=0} q_l \cdot \prod_{l \in L: a_{sl}=1} (1 - q_l). \tag{3}$$

2.3. Formulation. We formulated the problem as follows:

$$\max \quad Z = \sum_s p_s \sum_i \sum_j \sum_k x_{ijk} P_{ijks} \tag{4}$$

$$\text{s.t.} \quad \sum_j \sum_k x_{ijk} = w_i, \quad \forall i, \tag{5}$$

$$\sum_i \sum_k x_{ijk} \le 1, \quad \forall j, \tag{6}$$

$$x_{ijk} \sum_{l \in R_{ijk}} t_{sl} \le T_e, \quad \forall i, j, k, \tag{7}$$

$$p_{sl} = \begin{cases} 1 & \text{if } a_{sl} = 1 \\ \theta_l & \text{if } a_{sl} = 0 \end{cases} \tag{8}$$

$$P_{ijks} = \prod_{l \in R_{ijk}} p_{sl} \quad \forall i, j, k \tag{9}$$

$$\sum_i \sum_j \sum_k \alpha_{lijk} x_{ijk} \le \lambda_l \quad \forall l \tag{10}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j, k. \tag{11}$$

The objective function of the problem is to maximum the sum of expectations of all vehicles travelling securely to their destinations. Constraints (5) ensure that all vehicles at origin i are dispatched. Constraints (6) restrict that the number of vehicles in each destination should be no more than one. Constraints (7) present that in any scenario, each vehicle's moving time should be less than the time length restricted by the mission. In our paper, the time consumed in each arc, t_{sl}, is assumed to different under different scenario. Constraints (8) show that if an arc is in secure status, the MGV can pass through it safely with probability 1, and the safely travelling probability turns to be θ_l if the arc is insecure. Constraints (9) are the variants of formula (1) in an appropriate form. Constraints (10) show that the flow in each arc is limited. Constraints (11) state that decision variables, x_{ijk}, are all 0-1 variables.

It is observed that in a specified scenario, the problem is a deterministic BIP, which can be solved by CPLEX solver. If there are r arcs involved in a problem and each arc can either be in secure status or insecure status, 2^r scenarios need to be considered. Thus, the number of scenarios grows exponentially with the number of arcs. It would be a tough work to get the optimal solution with all scenarios considered. To cope with this problem, we introduce the SAA method to solve the problem in the next section.

3. Solving Methodology

In this section, we present the solution approach based on the SAA method.

3.1. Problem Description Based on SAA. The SAA method is a Monte Carlo simulation based approach to stochastic discrete optimization problems [16]. In this method, the expected value of the objective function is approximated by solving the problem for a sample of scenarios. The sample size of the SAA problem is much smaller than the number of scenarios in the *true* problem and as the sample size increases, SAA converges exponentially fast to the true problem [16]. The SAA method has been applied in many stochastic programs lacking exact algorithms, for example, the network design [17–19] and empty container repositioning [20, 21].

The SAA method repeats the sampling and solving process for several times to select a best solution. In each sample, N scenarios are randomly generated, and thus the objective function (4) of the true problem is approximated with

$$\widehat{Z}_N(x) \coloneqq \max \frac{1}{N} \sum_{n=1}^{N} \sum_i \sum_j \sum_k x_{ijk} P_{ijk}^n. \tag{12}$$

For each scenario in a sample, both the time, t_l, and the secure probability, p_l, of each arc are fixed. The true problem (4)–(11) in each scenario can be rewritten as

$$\max \quad Z = \sum_i \sum_j \sum_k x_{ijk} P_{ijk} \tag{13}$$

ST (5)-(6), (10)-(11), and

$$x_{ijk} \sum_{l \in R_{ijk}} t_l \le T_0, \quad \forall i, j, k, \tag{14}$$

$$P_{ijk} = \prod_{l \in R_{ijk}} p_l, \quad \forall i, j, k. \tag{15}$$

This problem is a deterministic BIP, which can be solved by CPLEX solver. However, since the time, t_{sl}, varies from scenario to scenario, a solution suiting a scenario well may be an infeasible solution to another scenario. The solution we selected should be universally applicable. A solving process to this matter is given in Procedure 1.

First, we solve the problem of each scenario and get the optimal solutions separately. The solutions recurring for many times are thought to be of better performances than these hardly recurring, which are optimal in just some extreme scenarios. With all scenarios considered, we keep several solutions with the most recurring times. (In fact, in the experiment it is observed that the first three or four solutions consist of best solutions to more than a half of all scenarios.) From these solutions, the solution which is feasible in all scenarios and has the largest value of formula (12) is selected as the solution to the sample, denoted by x.

The process of sampling N scenarios and solving to obtain $\widehat{Z}_N$ and x_N is repeated M times. Calculate the average of the

```
Procedure: Selecting a solution for a sample
(1) Input the $N$ randomly-generated scenarios, and initialize the solution $x = \emptyset$
(2) Solving the problem of each scenario with CPLEX, record the best solution to each scenario
(3) Counting the solutions, sort the diverse solutions in non-increasing order of counts
(4) Choose the first $R$ diverse solutions ($R$ is the number of solutions to check)
(5) For solution $x_r$ $(r = 1, 2, \dots, R)$, calculate the $\widehat{Z}_N(x_r)$ in formula (12)
(6) Check solution $x_r$ in all scenarios, if $x_r$ is not feasible overall, update $\widehat{Z}_N(x_r) = 0$
(7) Choose $x_{r^*}$ with the biggest $\widehat{Z}_N(x_{r^*})$, and let $x = x_{r^*}$
(8) Return $x$
```

Procedure 1: The solving process in a sample.

```
(1) Choose initial sample sizes $N$ and $N'$, a decision rule for determining repeating times $M$ and a rule for increasing the sample
    sizes $N$ and $N'$.
(2) For repetition $m$ $(m = 1, 2, \dots, M)$, generate a sample of size $N$, and solve the SAA problem to get the objective value $\widehat{Z}_N^m(x)$,
    and the corresponding solution $x_m$.
(3) Calculate the sample average $\overline{Z}_N^M$ and the corresponding deviation $\widehat{\sigma}_{\overline{Z}_N^M}$.
(4) For repetition $m$ $(m = 1, 2, \dots, M)$, generate a sample of size $N'$ and calculate $\widehat{Z}_{N'}(x_m)$.
(5) Select a solution with the maximum objective value $x^* = \arg\max\{\widehat{Z}_{N'}(x) : x \in x_1, x_2, \dots, x_M\}$, $\widehat{Z}_{N'} = \widehat{Z}_{N'}(x^*)$, and calculate
    the corresponding deviation $\widehat{\sigma}_{\widehat{Z}_{N'}}$.
(6) Calculate the two gaps, gap and $\widehat{\sigma}_{\overline{Z}_N^M+\widehat{Z}_{N'}}$.
(7) If convergence criterions are met then stop; if not, change $N$ and $N'$, and repeat from step (2).
```

Algorithm 1: SAA algorithm.

optimal objective function values of the M samples, namely, the value of the sample average, as

$$\overline{Z}_N^M := \frac{1}{M}\sum_{m=1}^{M}\widehat{Z}_N^m(x). \tag{16}$$

It is noteworthy that the sample size N is a tradeoff between computational complexity and accuracy. A larger value of N brings a more accurate estimate of the true problem, but an increase in computing complexity as well. The size N can be selected dynamically. The SAA solving process starts with a smaller size, and then we can regulate the sample size based on the acquired accuracy and the computational complexity.

The selected solution, given a candidate solution $\overline{x}$, for example, a feasible solution from the optimal solutions of M samples, needs to be evaluated. The evaluation of objective function is conducted by taking sample size of N' $(N' \gg N)$. That is, a number of N' scenarios are randomly generated for the evaluation. Let

$$\widehat{Z}_{N'}(\overline{x}) := \frac{1}{N'}\sum_{n=1}^{N'}\sum_{i}\sum_{j}\sum_{k}\overline{x}_{ijk}P_{ijk}^{n}. \tag{17}$$

Since N' is much larger than N, the objective value obtained $\widehat{Z}_{N'}(\overline{x})$ is considered as an estimator of the true objective function (3) [19]. In a common SAA problem, the quality of the solution is evaluated by calculating the optimality gap and the variance of optimality gap. In our problem, as the involved expectations are all numbers around 1, to highlight the dispersion of the result, we replace the variance of optimality gap with the deviation of optimality gap. Here, we define the optimality gap as

$$\text{gap} := \overline{Z}_N^M - \widehat{Z}_{N'}(\overline{x}). \tag{18}$$

And we define the deviation of optimality gap as

$$\widehat{\sigma}_{\overline{Z}_N^M+\widehat{Z}_{N'}} := \widehat{\sigma}_{\overline{Z}_N^M} + \widehat{\sigma}_{\widehat{Z}_{N'}}, \tag{19}$$

where

$$\begin{aligned}\widehat{\sigma}_{\overline{Z}_N^M} &:= \sqrt{\frac{1}{M(M-1)}\sum_{m=1}^{M}\left(\widehat{Z}_N^m(x) - \overline{Z}_N^M\right)^2},\\ \widehat{\sigma}_{\widehat{Z}_{N'}} &:= \sqrt{\frac{1}{N'(N'-1)}\sum_{n=1}^{N'}\left(Z_n(\overline{x}) - \widehat{Z}_{N'}(\overline{x})\right)^2}.\end{aligned} \tag{20}$$

3.2. SAA Algorithm. Based on description above, the SAA algorithm process is presented in Algorithm 1.

4. Experiment Design and Computational Results Analysis

4.1. Experiment Design. In this part, we examine the performance of the algorithm on different dataset. We selected three road networks: a small-scale designed network with 13 nodes

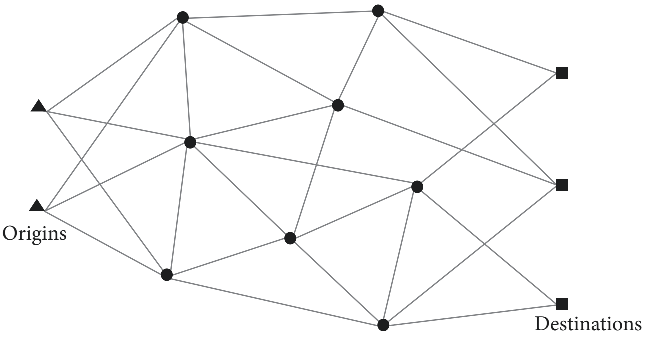


Figure 2: Network 1: the small-scale random network.

Table 1: The problems to be solved in the 3 networks.

Number	Network	Amount of nodes	Involved nodes	Involved arcs	Number of origins (I)	Number of destinations (J)	Number of MGVs dispatched (w_i)	Road capacity (λ_l)
(1)	Random	13	13	27	2	3	[1, 1][a]	2
(2)	Changsha City	46	37	57	3	5	[1, 2, 1]	2
(3)	Fujian Province	226	112	147	3	15	[2, 3, 2]	2

[a] Values in square brackets stand for the number of vehicles dispatched from the origins separately.

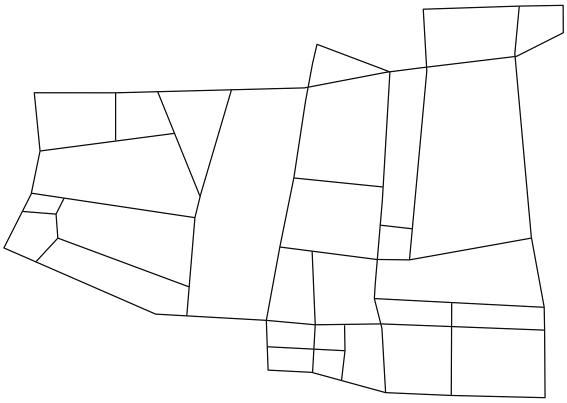


Figure 3: Network 2: a network from Changsha, China.

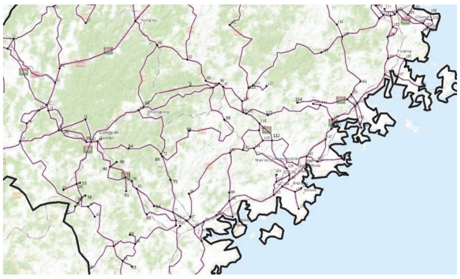


Figure 4: Network 3: the arterial traffic network of Fujian Province of China.

and 27 arcs (Figure 2), a middle scale practical network with 46 nodes and 77 arcs in Hunan Province of China (Figure 3), and a large scale practical network with 226 node and 239 arcs in southeast of China (Figure 4). The lengths of the arcs in Network 1 were randomly generated, while in Network 2 and Network 3 the lengths were calculated by the commercial software, ArcGIS, based on the practical data of the two regions.

The kth-shortest-path datasets were derived by the methods in [22–24]. Other key parameters are given as follows:

θ is a constant and is set to be 0.2 in all experiments;

the value of t_{sl} is set to be the road length divided by the velocity, which varies in a range around ±10% of a given velocity as the scenario varies;

q_l is randomly generated.

The data for all the experiments can be found in the supplementary material noted as dataset. Table 1 shows a brief look of the problems considered in 3 networks, respectively. For the fulfillment of SAA, the value of M is set to be 10 and keeps constant for all sample sizes. The sample size N ranges from 100 through 1000, with an interval of 100. And the value of N' is set to be 10,000. The algorithms were coded in MATLAB by invoking CPLEX v.12.6. And all the programs were executed on a 64-bit computer with a 3.2 GHz CPU and 8.0 GB of physical RAM.

4.2. Computational Results and Analysis. The computational results for the three networks are presented in Tables 2, 3, and 4, respectively. We give the minimum, maximum, average, and corresponding deviation of each sample size, as the results of the SAA process. The calculating time of each SAA process is also listed as the CPU time. It can be seen that the calculating time increases as the sample size grows, where the growth rate keeps almost constant for each network separately. This is an acceptable result, since that we viewed the working time of each module of the program and found the scenario generating and CPLEX solving parts, both of

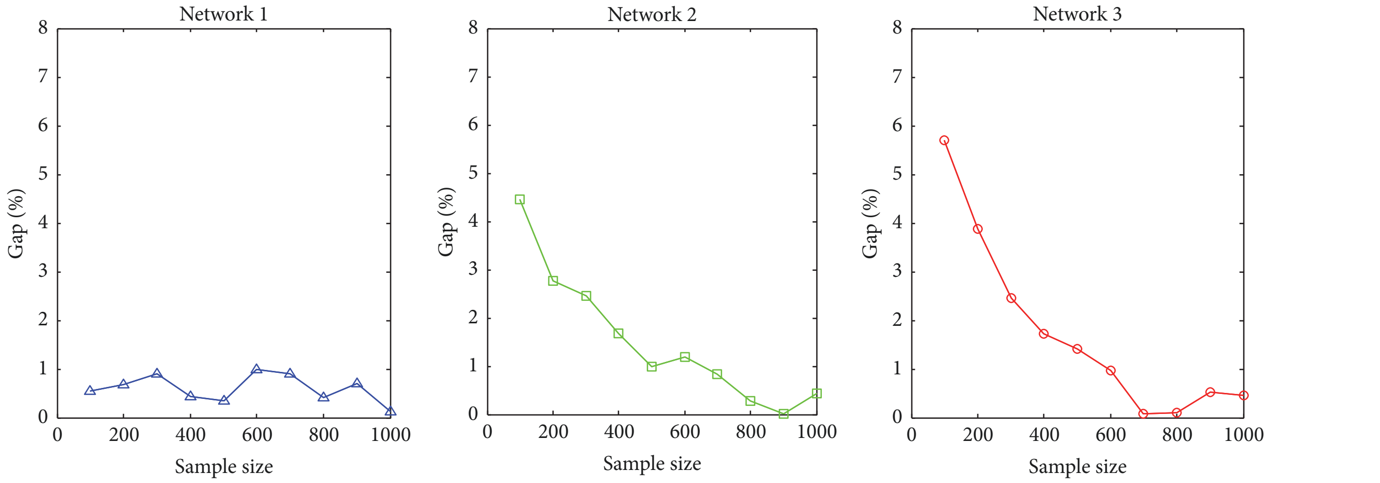


FIGURE 5: The line graph of sample size and *gap* (%).

TABLE 2: SAA results for test Network 1.

Sample Size (N)	$Z_{\min}$	$Z_{\max}$	$\overline{Z}_N^M$	$\widehat{\sigma}_{\overline{Z}_N^M}$	CPU time[a] (s)	$\widehat{Z}_{N'}(\overline{x})$	$\widehat{\sigma}_{\widehat{Z}_{N'}}$	gap	gap (%)[b]	$\widehat{\sigma}_{\overline{Z}_N^M+\widehat{Z}_{N'}}$
	SAA process					Evaluation		Optimality gap		
100	0.532	0.5864	0.5579	$5.3275e-03$	0.83	0.5548	$6.7234e-03$	$3.0553e-03$	0.5477	$1.2051e-02$
200	0.5456	0.5731	0.5586	$3.8670e-03$	1.54	0.5548	$6.7919e-03$	$3.7395e-03$	0.6695	$1.0659e-02$
300	0.5305	0.5783	0.5498	$5.4558e-03$	2.28	0.5548	$6.7253e-03$	$-4.983e-03$	0.9063	$1.2181e-02$
400	0.5438	0.5589	0.5525	$1.7122e-03$	3.01	0.5550	$6.7581e-03$	$-2.4441e-03$	0.4423	$8.4703e-03$
500	0.5450	0.5684	0.5567	$3.0237e-03$	3.76	0.5548	$6.836e-03$	$1.967e-03$	0.3533	$9.8597e-03$
600	0.5397	0.5595	0.5495	$2.0991e-03$	4.49	0.5549	$6.8159e-03$	$-5.4032e-03$	0.9833	$8.9149e-03$
700	0.5493	0.5695	0.5600	$2.6198e-03$	5.20	0.5549	$6.8249e-03$	$5.0636e-03$	0.9043	$9.4447e-03$
800	0.5501	0.5656	0.5565	$1.7425e-03$	5.94	0.5542	$6.7327e-03$	$2.2589e-03$	0.4059	$8.4752e-03$
900	0.5497	0.5651	0.5588	$1.3669e-03$	6.73	0.5550	$6.8224e-03$	$3.9135e-03$	0.7003	$8.1893e-03$
1000	0.5417	0.5630	0.5542	$2.5303e-03$	7.39	0.5549	$6.6988e-03$	$-6.5813e-03$	0.1187	$9.2291e-03$

[a]The CPU time presents the total time cost in the SAA sampling and solving process. [b]gap (%) = $(|\text{gap}|/\overline{Z}_N^M) * 100\%$.

which require repeating N (the value of sample size) times and take the most of the time (more than 95% of the whole). The evaluation part shows the results of employing the selected solution of the SAA process to a much bigger sample of size N'. And the gaps between these results are given in the optimality gap part. Several discussions about the optimality gap and the deviations are given as in Tables 2, 3, and 4.

Figure 5 shows the trends of *gap* (%) for the 3 networks while the sample size increases. Here the index, *gap* (%), means the closeness between the SAA results and evaluation results, which can be viewed as the estimator of the true problem. For Network 1, the *gaps* (%) are all below 1% and fluctuate irregularly, while, for Network 2 and Network 3, the values start from 4.46% and 5.69%, respectively, as the sample size is 100 and decrease distinctly as the sample size increases. Although there are several fluctuations in both of them, the downtrends are obvious. These differences between Network 1 and Network 2/3 arise from the solutions selected in each network. In fact, the solution selected for Network 1 stays constant from sample size 100 through 1000. The fluctuation is caused by the uncertainty of sampling. However, the solutions selected for Network 2 and Network 3 vary as the sample size increases. Better solution brings smaller optimality gap, which indicates that better solution to the true problem is obtained.

Figure 6 presents comparison for the trends of SAA results and evaluation results. We can see the evaluation results of Network 1 stay relative constant, while the SAA results fluctuate around the evaluation results in a quite small range. For Network 2, the SAA results have a distinct downtrend as the sample size increases, while the trend of the evaluation results is opposite. The gap between the two values is first below 1% when sample size is 500, after which the SAA results fluctuate around the evaluation results. The smallest gap between the two values reaches 0.013% when sample size is 900. The discussion for Network 2 can be applied to Network 3 as well, in which these trends are more obvious. The SAA results rise and evaluation results fall distinctly as the sample size increases from 100 through 600. From 600 through 1000, the SAA results fluctuate. And the

Table 3: SAA results for test Network 2.

Sample size (N)	SAA process				CPU time (s)	Evaluation		Optimality gap		
	$Z_{\min}$	$Z_{\max}$	$\overline{Z}_N^M$	$\widehat{\sigma}_{\overline{Z}_N^M}$		$\widehat{Z}_{N'}(\overline{x})$	$\widehat{\sigma}_{\widehat{Z}_{N'}}$	*gap*	*gap* (%)	$\widehat{\sigma}_{\overline{Z}_N^M+\widehat{Z}_{N'}}$
100	0.6988	0.7319	0.7175	3.7930e – 03	3.2	0.6855	1.5881e – 02	3.1969e – 02	4.4557	1.9674e – 02
200	0.6934	0.7197	0.7058	2.9367e – 03	5.97	0.6862	1.5822e – 02	1.9587e – 02	2.7752	1.8760e – 02
300	0.6943	0.7108	0.7035	1.6460e – 03	8.65	0.6862	1.5815e – 02	1.7336e – 02	2.4642	1.7461e – 02
400	0.6923	0.7169	0.7036	2.8679e – 03	11.45	0.6918	1.2582e – 02	1.1795e – 02	1.6764	1.5449e – 02
500	0.6872	0.712	0.6996	2.5119e – 03	14.17	0.6927	1.2499e – 02	6.9041e – 03	0.9869	1.5011e – 02
600	0.6929	0.7099	0.7013	1.6643e – 03	17.03	0.6929	1.2475e – 02	8.3798e – 03	1.195	1.4140e – 02
700	0.6922	0.7033	0.6998	1.2746e – 03	19.75	0.694	1.2445e – 02	5.8449e – 03	0.8352	1.3719e – 02
800	0.7004	0.7077	0.7035	7.1520e – 04	22.41	0.7015	1.0357e – 02	2.0178e – 03	0.2868	1.1072e – 02
900	0.6947	0.7098	0.7016	1.9050e – 03	25.22	0.7015	1.0322e – 02	9.2599e – 05	0.0132	1.2227e – 02
1000	0.6959	0.7039	0.6995	9.1399e – 04	27.99	0.7025	1.0217e – 02	−2.9942e – 03	0.4281	1.1131e – 02

Table 4: SAA results for test Network 3.

Sample size (N)	SAA process				CPU time (s)	Evaluation		Optimality gap		
	$Z_{\min}$	$Z_{\max}$	$\overline{Z}_N^M$	$\widehat{\sigma}_{\overline{Z}_N^M}$		$\widehat{Z}_{N'}(\overline{x})$	$\widehat{\sigma}_{\widehat{Z}_{N'}}$	*gap*	*gap* (%)	$\widehat{\sigma}_{\overline{Z}_N^M+\widehat{Z}_{N'}}$
100	1.0532	1.1621	1.0922	1.0115e – 02	36.02	1.0301	2.2434e – 02	6.2161e – 02	5.6911	3.255e – 02
200	1.0519	1.1191	1.0744	7.0514e – 03	64.02	1.0328	2.2288e – 02	4.1606e – 02	3.8724	2.934e – 02
300	1.034	1.1179	1.0641	7.8205e – 03	92.09	1.038	1.8193e – 02	2.6097e – 02	2.4525	2.6013e – 02
400	1.0535	1.1072	1.0683	4.8197e – 03	120.2	1.0499	1.6410e – 02	1.8438e – 02	1.7259	2.1229e – 02
500	1.0517	1.07	1.0626	2.2199e – 03	148.23	1.0476	1.6670e – 02	1.5009e – 02	1.4125	1.889e – 02
600	1.0344	1.0796	1.047	4.2979e – 03	176.4	1.0571	1.3658e – 02	−1.009e – 02	0.9638	1.7956e – 02
700	1.0381	1.0881	1.0628	5.6404e – 03	204.6	1.0619	1.3634e – 02	8.4620e – 04	0.0796	1.9275e – 02
800	1.0412	1.0829	1.0546	4.0003e – 03	232.48	1.0557	1.3209e – 02	−1.1158e – 03	0.1058	1.7209e – 02
900	1.0439	1.0745	1.054	2.7739e – 03	260.69	1.0595	1.3410e – 02	−5.4381e – 03	0.5159	1.6184e – 02
1000	1.0432	1.0601	1.0535	1.7549e – 03	288.78	1.0584	1.3530e – 02	−4.8823e – 03	0.4634	1.5284e – 02

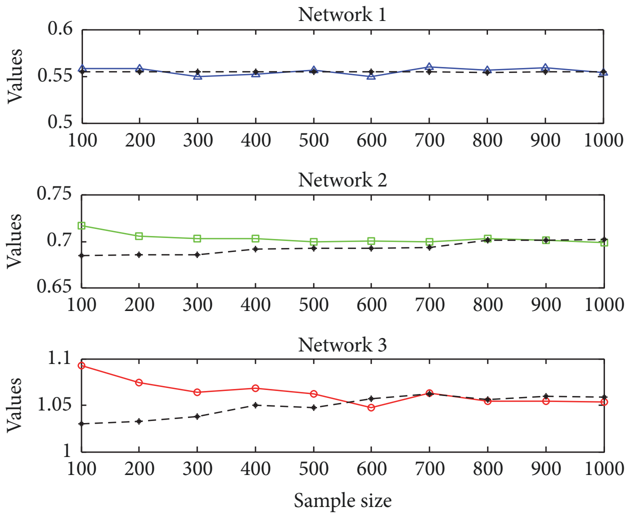


Figure 6: The comparison between SAA results (the colored solid lines) and evaluation results (the black dashed lines) with sample size varying.

smallest gap between the two is 0.08%, when the sample size is 700.

The results of Network 1 indicate that, for a network of small scale, a good solution can be obtained with a considerably small sample size. But for networks of bigger scales, such as Network 2 and Network 3, as the number of arcs considered in the problem increases, the number of possible scenarios increases. A sample of small size cannot reflect the panorama of the problem. In our algorithm, 10 samples are generated and solved, and the sample solution with the biggest expectation value is selected, while, for a sample of small size, a suboptimal solution may have a better performance than the "good" solution (as we cannot tell whether it is the optimal one) does. This explains why for networks of bigger scales, suboptimal solutions are usually selected when the sample size is small. But if the sample size is big enough, the good solutions can still be obtained constantly.

Figure 7 presents the deviation trends with the sample size increasing. All the deviations of the three networks have downward trends, which are especially evident in Network 2

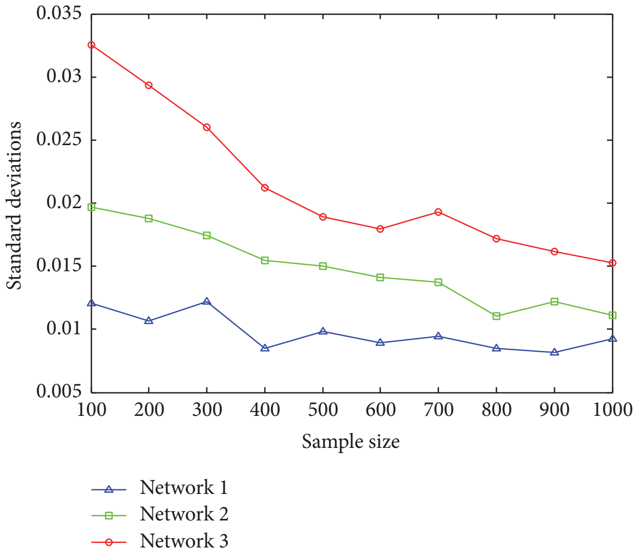


Figure 7: The line graph of sample size and deviations of 3 networks.

and Network 3. The deviation consists of the sampling deviation and the evaluation deviation. The evaluation deviation reflects the quality of a given solution. As all solutions selected in sample sizes are identical in Network 1, the evaluation deviations vary a little. The variation of the deviation is mainly caused by the sampling deviations, which manifests volatility. For networks 2 and 3, of each sample size, the evaluation deviation is much bigger than the sampling deviation. And as the selected solutions vary, the evaluation deviations have evident downtrends. In fact, the sampling deviations of Network 2 and 3 are fluctuating, of which the overall trends are drops as well. As the longest time spent on sampling and calculating is just 288.78 s (for Network 3, of sample size 1000), the time spent on calculating is relatively short and grows linearly with the growth of the sample size, we can solve the problem with bigger sample size to get a good solution of higher accuracy to reflect the true problem better.

5. Conclusion

In this paper, we addressed the problem MGVs maneuver in unsecure road networks in battlefield environment. We describe the uncertainty of the network based on discrete scenarios and introduce the kth-shortest-path method to calculate the possible paths between origins and destinations. The problem is formulated into a binary integer programming model. Due to the huge number of scenarios generated by the combination of uncertain arcs, we proposed a SAA method to solve the model. Experiments based on three networks with different scales were conducted to test the efficiency of the solution approach. Computational results show that bigger sample brings relevant progress of the approximation and an acceptable increasing in computational complexity.

In our paper, we did not consider the fortification or camouflage resource for the maneuver, and we only need to decide routes for all vehicles. In some cases, there are protective resources available in the MGV moving process, which can be allocated to some road arcs and reduce the risk for being attacked. Therefore, it is a valuable extension for integrating the protective resources in the MGV routing problem in future works.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research is supported by the National Natural Science Foundation of China (no. 71771215 and no. 71471174).

Supplementary Materials

1. Dataset of Network-1. 1.1. AdjacentData: here we give the distances between each pair of adjacent nodes. If there is no direct connection between a pair of adjacent, the distance is set infinite and will not be listed here. *(Supplementary Materials)*

References

[1] SATP, NATO related attacks in Pakistan, http://www.satp.org/satporgtp/countries/pakistan/database/natoattack.htm.

[2] GSO, The World at War, http://www.globalsecurity.org/military/world/war/index.html.

[3] A. L. Tuson and S. A. Harrison, “Problem difficulty of real instances of convoy planning,” *Journal of the Operational Research Society*, vol. 56, no. 7, pp. 763–775, 2005.

[4] P. N. Ram Kumar and T. T. Narendran, “Integer programming formulation for convoy movement problem,” *International Journal of Intelligent Defence Support Systems*, vol. 1, no. 3, pp. 177–188, 2008.

[5] K. S. Krishna and P. N. Ram Kumar, “On the amenability and suitability of ant colony algorithms for convoy movement problem,” *Procedia—Social and Behavioral Sciences*, vol. 189, pp. 3–16, 2015.

[6] Y. Pinedo and D. Ruben, *Route Optimization While Improving Safety Using Escort Vehicles*, State University of New York at Buffalo, ProQuest Dissertations Publishing, 2013.

[7] Y. He, Z. Liu, J. Shi, Y. Wang, J. Zhang, and J. Liu, “K-shortest-path-based evacuation routing with police resource allocation in city transportation networks,” *PLoS ONE*, vol. 10, no. 7, Article ID 131962, 2015.

[8] N. Altay and W. G. I. Green III, “OR/MS research in disaster operations management,” *European Journal of Operational Research*, vol. 175, no. 1, pp. 475–493, 2006.

[9] G. Galindo and R. Batta, “Review of recent developments in OR/MS research in disaster operations management,” *European Journal of Operational Research*, vol. 230, no. 2, pp. 201–211, 2013.

[10] T. Yamada, “A network flow approach to a city emergency evacuation planning,” *International Journal of Systems Science*, vol. 27, no. 10, pp. 931–936, 1996.

[11] T. J. Cova and J. P. Johnson, “A network flow model for lane-based evacuation routing,” *Transportation Research Part A: Policy and Practice*, vol. 37, no. 7, pp. 579–604, 2003.

[12] V. B. G. Campos, P. A. L. Da Silva, and P. O. B. Netto, “Evacuation transportation planning: a method of identify optimal independent routes,” *Urban Transport V*, vol. 44, pp. 555–564, 1999.

[13] A. Stepanov and J. M. Smith, “Multi-objective evacuation routing in transportation networks,” *European Journal of Operational Research*, vol. 198, no. 2, pp. 435–446, 2009.

[14] CPLEX Optimizer, https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer.

[15] P. Yang, W. D. Liu, and M. Li, “An optimization model of transportation scheme in preoperational preparations of conventional missile,” *Fire Control & Command Control*, vol. 32, no. 2, pp. 41–43, 2007.

[16] S. Ahmed and A. Shapiro, “The sample average approximation method for stochastic programs with integer recourse,” *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[17] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro, “A stochastic programming approach for supply chain network design under uncertainty,” *European Journal of Operational Research*, vol. 167, no. 1, pp. 96–115, 2005.

[18] G. R. Patil and S. V. Ukkusuri, “Sample average approximation technique for flexible network design problem,” *Journal of Computing in Civil Engineering*, vol. 25, no. 3, pp. 254–262, 2011.

[19] B. Ayvaz, B. Bolat, and N. Aydin, “Stochastic reverse logistics network design for waste of electrical and electronic equipment,” *Resources, Conservation & Recycling*, vol. 104, pp. 391–404, 2015.

[20] Y. Long, L. H. Lee, and E. P. Chew, “The sample average approximation method for empty container repositioning with uncertainties,” *European Journal of Operational Research*, vol. 222, no. 1, pp. 65–75, 2012.

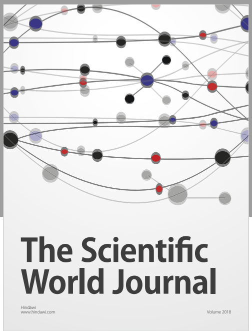
[21] Y. Long, E. P. Chew, and L. H. Lee, “Sample average approximation under non-i.i.d. sampling for stochastic empty container repositioning problem,” *OR Spectrum*, vol. 37, no. 2, pp. 389–405, 2015.

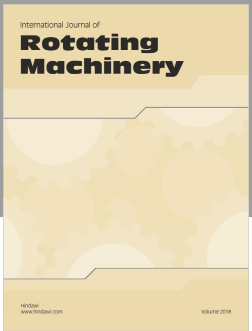
[22] J. Y. Yen, “Finding the K shortest loopless paths in a network,” *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[23] D. Eppstein, “Finding the k shortest paths,” *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1998.

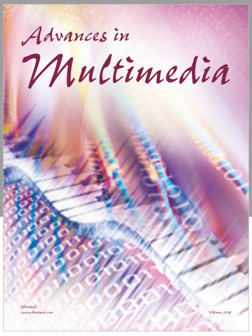
[24] H. Aljazzar and S. Leue, “K^*: a heuristic search algorithm for finding the *k* shortest paths,” *Artificial Intelligence*, vol. 175, no. 18, pp. 2129–2154, 2011.

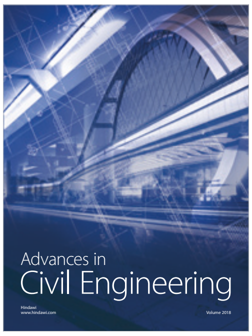
Journal of
Engineering

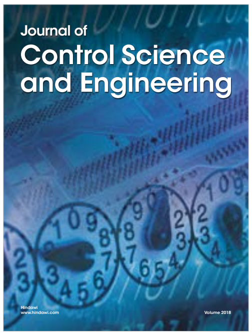

The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

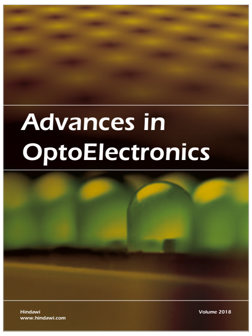

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

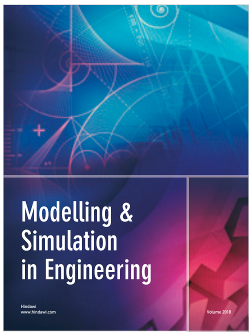
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

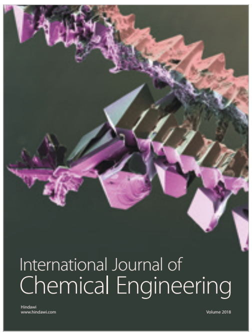

Advances in
OptoElectronics

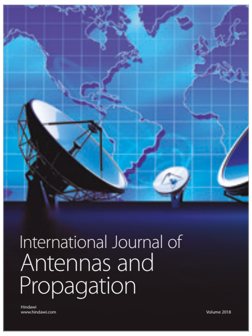
VLSI Design

International Journal of
Navigation and
Observation

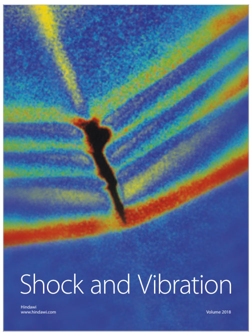

Modelling &
Simulation
in Engineering

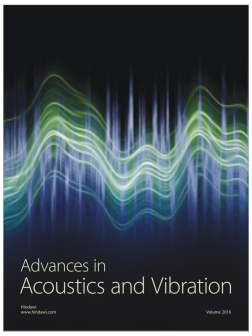
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration