

Research Article

A Study of Network Violator Interception Based on a Reliable Game Model

Shi An, Jianxun Cui, Meng Zhao, and Jian Wang

School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin 150090, China

Correspondence should be addressed to Jianxun Cui; cuijianxun@hit.edu.cn

Received 29 March 2014; Revised 28 May 2014; Accepted 28 May 2014; Published 15 June 2014

Academic Editor: Hamid Reza Karimi

Copyright © 2014 Shi An et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study focuses on planning interceptor locations in a general transportation network to maximize the expected benefits from catching violators mixing in public traveler flow. Two reliability-related characteristics are also integrated into the planning model to make it more practical. One is that each interceptor (e.g., a sensor or a checkpoint) has a failure probability. The second is the existence of a “game” between the interceptor planner and violators. A nonlinear nonconvex binary integer programming model is presented. We develop a simulated annealing (SA) algorithm to solve this model, and numerical experiments are conducted to illustrate the computational efficiency of the proposed algorithm. We also analyze the sensitivity of the disruption probability of interceptors to optimal objective function values and discuss how to determine the values of these parameters in a violator route choice model.

1. Introduction

There currently exist many types of violators in daily life networks (such as ground transportation networks, commercial airline networks, and communication networks), any of whom may pose a serious threat to public safety. A typical example is that of the drunk drivers who frequently emerge in an urban road network [1–4]. This is a difficult, real, and common problem troubling many local governments in many countries. Car accidents caused by drunk driving claim the lives of more than 15,000 people in the US every year and injure many more. More intuitively, as concluded by authorities [5], (1) one-third of all Americans will be involved in drunk driving accidents at some point in their lifetimes; (2) someone is injured in an alcohol-related car accident approximately every 60 seconds; (3) someone is killed in an alcohol-related car accident approximately every 40 minutes; and (4) drunk driving car accidents cost the US an average of \$114 billion annually. This phenomenon also exists in China, where drunk driving and consequent traffic accidents are even more prevalent and serious. Eliminating drunk driving has long been a major focus of traffic safety professionals at the federal, state, and local levels. Terrorists, another type of violators, aim to attack airline networks

[6–9]. They mix in with the public flow of passengers and watch for an opportunity to launch an attack. As further examples, viruses and illegal data may exist within the public information flow in a local intranet or a world-wide internet. Viruses, in particular, may steal or violate a public user's private information or cause disruption to public internet users. Means of intercepting and eliminating illegal viruses is an important subject in the field of network security.

A common problem faced by decision makers who want to protect networks is how to catch violators mixing in the public flow as soon as possible by setting up interceptors (which could be DUI checkpoints for catching drunk drivers, or a virtual network police force that screens for viruses and illegal information). A network cannot have unlimited numbers of interceptors due to budgetary or logistical limits (e.g., a DUI checkpoint for drunk driving at one location will cause delays because vehicles need to stop and be checked, limiting the number of checkpoints that can be established). Furthermore, an unlimited number of interceptors is not necessary. One can always attempt to find an optimal interceptor layout with less than a given number of interceptors to obtain the largest expected benefit. This interceptor location problem (ILP) is very similar to “network interdiction” problems. The deterministic network interdiction problem has been

studied by Zenklusen [10], Granata et al. [11], Rad and Kakhki [12], and Yates and Sanjeevi [13] with military applications and with applications to the interdiction of illegal drugs and precursor chemicals. In this problem, given a capacitated network, a source s and a sink t , the aim is to interdict arcs in the network to minimize the maximum flow from s to t , subject to constraints on the number of prohibited arcs. The stochastic version of this problem studied by Cormican et al. [14] and Ramirez-Marquez and Rocco S. [15] assumes that interdictions on arcs do not always succeed but may be either completely unsuccessful, partially successful, or completely successful.

All network interdiction problems are rooted in the well-known max-flow-min cut theorem. These problems consider violators a type of commodity and attempt to break arcs to leave as few as possible arc-disjoint paths remaining for violators to use. However, for the ILP, we treat violators as a minority mixing in with public travelers, who are the majority. We attempt to discriminate violators from innocent public travelers as soon as possible, while they share the same network at the same time. ILP is not a max-flow-min problem in nature, as the purpose of network interdiction problems is to intercept as many violators as possible regardless of how long the violator will travel in the network or how many public travelers will be threatened by the violator. Additionally, in practice, we implement a “game” between interceptor planners and violators. After planners instantiate a specific interceptor layout, violators will adjust their route choices as much as possible to avoid being caught by interceptors. Thus, we expect that a robust solution to the interceptor layout will achieve equilibrium between planners and violators.

For the ILP, which is different from network interdiction research, we propose a reliable interceptor location problem (RILP). We assume that, in a general network, two types of flow exist: one is the public traveler flow, composed of innocent travelers, and the other is the flow of violators, who are mixed in with the public traveler flow and pose a threat to public travelers. We aim to effectively identify violators from the public flow by setting up interceptors along routes of flow. The installed interceptors are assumed to have a given probability of failure at capturing violators as they pass by. Additionally, “game” behaviors between planners and violators are integrated into the proposed model. To the best of our knowledge, our attempt is the first attempt to study ILP in a reliable and gamified manner.

This paper is organized as follows. Section 2 presents the model formulation. Section 3 discusses the algorithms for this proposed model. Section 4 presents some numerical tests to investigate the computational performance of the proposed model and algorithms. Section 5 concludes the paper and discusses future research directions.

2. Model Formulation

To formulate RILP, a network should first be classified into a series of origin-destination (OD) pairs between which daily traffic demand (including violators) is generated. We use w to represent the set of all OD pairs, φ_r to represent the set

of available routes between a specified OD pair $r \in w$ and φ to represent the set of all possible routes between OD pairs. Let ϕ_i be the set of candidate locations along route $i \in \varphi_r$ for interceptors and let ϕ be the set of all candidate interceptor locations. D_r represents the expected number of violators generated in OD pair $r \in w$ during any specified time interval.

For any OD flow, an interceptor will check passengers if and only if the flow passes the interceptor. Any violators in the flow will be caught. In this case, we say that the flow is covered by the interceptor or that we have *flow coverage*. The benefit obtained from setting up an interceptor along a specific route, which we denote by *path coverage*, depends on not only the inspected passenger volume but also the lengths of the covered OD paths. For simplicity, this study adopts a vehicle-mile coverage measure [16] such that the path coverage benefit for an OD path is proportional to both its traffic volume and the covered length.

The existence of violators will pose a safety threat to the traveling public. The magnitude of the threat is dependent on the public traveler flow encountered and the length of the path traveled by the violators before their interception. For a specific route i , setting up an interceptor at candidate location $j \in \phi_i$ results in a corresponding potential benefit b_{ij} stemming from the public’s protection from violators on that route. Therefore, in a network with link set A , the *path coverage* benefit b_{ij} can be defined as follows:

$$b_{ij} = \sum_{a \in A} \delta_{ij}^a f_a L_a, \quad \forall i \in \varphi, \quad \forall j \in \phi_i, \quad (1)$$

where L_a and f_a are, respectively, the length and the public traveler flow on link $a \in A$ and parameter δ_{ij}^a is defined as follows:

$$\delta_{ij}^a = \begin{cases} 1, & \text{link } a \text{ is on path } i \text{ and downstream of location } j \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In practice, due to the performance limit of interceptors, violators may not always be caught when they pass by interceptors. Even when interceptors are fully competent to discriminate violators from the public flow, we should not assume that all passing passenger flows are checked by interceptors because it is not practical to stop the flow to check every passenger (i.e., checking for drunk drivers at a DUI checkpoint will cause traffic delays if the sampling rate is high). This is similar to the concept of “sensor failure” that is widely studied in the literature [17–21]. In this paper, we define this failure as “interceptor failure.” Complementary to the introduction of interceptor failure, the expected intercepting benefit from setting up an interceptor at candidate location j along route i is related to the “head level” of that interceptor. Supposing that there are n_i interceptors installed on route i . We see that once the locations with installations on i are given (i.e., $\{j_1^i, j_2^i, \dots, j_{n_i}^i\}$ ordered from upstream to downstream), their head levels are defined as follows.

Definition 1 (head level). An interceptor at location j_s^i is the level- s head interceptor along route i .

The primal decision variables $x = \{x_j\}$ determine where to install DUI sites, where

$$x_j = \begin{cases} 1, & \text{a interceptor is installed at candidate location } j \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Given x , the auxiliary variables $h = \{h_{ij}^m\}$ decide how interceptors are assigned to paths, where

$$h_{ij}^m = \begin{cases} 1, & \text{a interceptor is installed at candidate location } j \\ & \text{along route } i \text{ and assigned to head level } m \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Assume that each interceptor fails independently with an identical probability $0 \leq p < 1$. The *path coverage* benefit from setting up interceptors at location j along route i is b_{ij} . As interceptor j functions to catch violators with probability $(1-p)p^{m-1}$, its expected contribution to catching violators is $(1-p)p^{m-1}h_{ij}^mb_{ij}$.

Additionally, each route $i \in \varphi_r$ has a probability α_i of being selected by violators, such that $\sum_{i \in \varphi_r} \alpha_i = 1, \forall r \in w$. This route selecting probability should not be defined statically but should instead be related to the layout of interceptors. For example, locating a DUI checkpoint at a certain location one night may induce drunk drivers to avoid this location later. To account for this “game” between interceptor planners and violators, we assume that violators will select a route between a specific OD pair according to a Logit model, in which the utility of violators to select a specific route is determined by two aspects: the numbers of installed interceptors along all possible OD routes and the travel lengths of these routes. We assume that violators always try to find a route that possesses as few interceptors as possible and a short travel distance. We define α_i as follows:

$$\alpha_i = \frac{\exp(\beta_1 p^{n_i} - \beta_2 L_i)}{\sum_{k \in \varphi_r} \exp(\beta_1 p^{n_k} - \beta_2 L_k)}, \quad \forall r \in w, \forall i \in \varphi_r, \quad (5)$$

where $n_i = \sum_{j \in \phi_i} \sum_{m=1}^{R_i} h_{ij}^m$ is the number of installed interceptors along route i , L_i is the travel length of route i , and β_1, β_2 are two coefficients corresponding to interceptor number and travel length, respectively, in the utility function.

Taking into account the discussion above, the expected intercepting benefits across an exponential number of failure scenarios and the game behavior between planners and

violators can be consolidated into a compact expression, and RILP can be written as follows:

$$(RILP) \quad \max_{x,h} \sum_{r \in w} \sum_{i \in \varphi_r} \sum_{j \in \phi_i} \sum_{m=1}^{R_i} D_r \alpha_i b_{ij} h_{ij}^m p^{m-1} (1-p) \quad (6)$$

$$\text{s.t.} \quad \sum_{j \in \phi} x_j \leq N \quad (7)$$

$$\sum_{m=1}^{R_i} h_{ij}^m = x_j; \quad \forall i \in \varphi, \forall j \in \phi_i \quad (8)$$

$$\sum_{j \in \phi_i} h_{ij}^1 \leq 1; \quad \forall i \in \varphi \quad (9a)$$

$$\sum_{j \in \phi_i} h_{ij}^m \leq \sum_{j \in \phi_i} h_{ij}^{m-1}; \quad \forall i \in \varphi, \forall m \in [2, R_i] \quad (9b)$$

$$R_i = \min(|\phi_i|, N); \quad \forall i \in \varphi \quad (10)$$

$$x_j, h_{ij}^m \in \{0, 1\}; \quad \forall i \in \varphi, \forall j \in \phi_i, \quad (11)$$

$$\forall m \in [1, R_i].$$

Objective function (6) aims to maximize the total expected benefit from all interceptors. Constraint (7) enforces the budget limit, which is that no more than N interceptors are allowed to be installed. Constraint (8) ensures that each installed interceptor is assigned to each of its corresponding paths at one and only one head level. Constraint (9a) indicates that no more than one head interceptor is assigned to each path at each level. Constraint (9b) implies that, for each path i , all the implemented head assignment levels $\{m \mid \sum_{j \in \phi_i} h_{ij}^m = 1\}$ start from 1 and form a consecutive sequence. Constraint (10) is the definition of the head level number along a specific route. Constraint (11) defines the binary variables.

3. Solution Algorithms

Solving the model of RILP is relatively difficult, as it is a non-linear, nonconvex binary integer programming model, the objective function of which contains exponential, fractional forms of decision variables. We will introduce simulated annealing (SA) to solve this model.

SA is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. SA is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more efficient than exhaustive enumeration, provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than to find the best possible solution. The algorithm is well

Step 1. Initialization.

(1.1) Randomly select N candidate locations to initialize an initial feasible solution x^0 . For an arbitrary route i , determine h^0 according to interceptor layout x^0 , then compute objective function value $z(x^0)$ according to (6). Let $x = x^0$.

(1.2) Set Markov length ML, initial temperature T_0 , error ϵ , and let $T = T_0$.

(1.3) Set the outer iteration counter $n = 1$.

Step 2. For the given T , perform the following:

(2.1) Set the inner iteration counter to $k = 1$.

(2.2) Randomly select an element from x (selected candidate locations) and $\phi - x$ (unselected candidate locations), respectively, and exchange them to produce a new solution \hat{x} .

(2.3) Compute the objective function value $z(\hat{x})$ according to (6).

(2.4) Set $\Delta z = z(\hat{x}) - z(x)$, if $\Delta z > 0$, $x = \hat{x}$; else let $x = \hat{x}$ with probability $p(\Delta z) = \exp(\Delta z/T)$.

(2.5) Check whether $k = ML$. If $k = ML$, go to Step 3; else $k = k + 1$, return to (2.2).

Step 3. Determine whether to stop

(3.1) Check whether $T < \epsilon$. If $T < \epsilon$, stop; go to (3.2)

(3.2) Let $T = \eta T$, $n = n + 1$, and return to Step 2. (where η decides the speed of annealing)

ALGORITHM 1: SA on RILP.

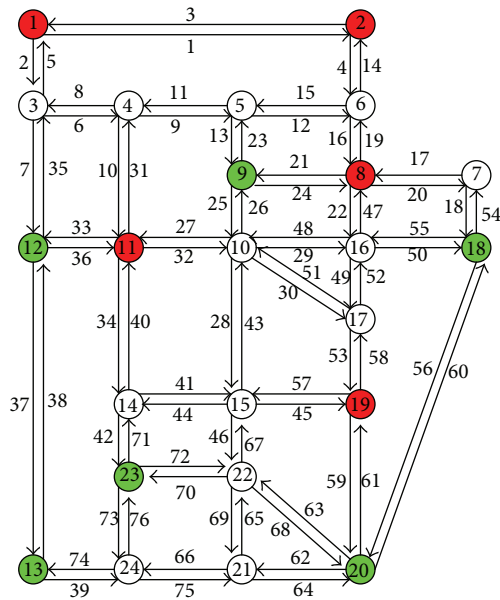


FIGURE 1: Sioux Falls network.

described by Liu [22]. A brief summary of the algorithm is given below.

(1) *Initialization.* An initial solution is generated randomly from the feasible region. The initial temperature should be high enough to allow all candidate solutions to be accepted.

(2) *Markov Length.* The iteration number M is used in each temperature. This number should be set appropriately high such that the objective function values reach a Boltzmann distribution.

(3) *Cooling Schedule.* The cooling schedule is the rate at which the temperature is reduced. In this paper, 0.8 is used for the first 12th temperature reductions. A cooling schedule of 0.8 means the temperature of the next stage is 0.8 times the

current temperature. A cooling schedule of 0.5 is used after the 12th temperature reduction.

(4) *Step Size.* Step size at each move should be decreased along with the reduction in temperature. Feasible solutions at lower temperatures are close to the optimal solution. When the temperature is low, a stochastic search tends to be a deterministic search. If the step size is too large, at low temperature, some feasible solutions will be rejected, thereby wasting computation time.

(5) *Neighboring Solutions.* Neighboring solutions are the set of feasible solutions that can be generated from the current solution. Each feasible solution can be directly reached from the current solution by a move and the resulting neighboring solution.

(6) *Stopping Criteria.* The algorithm stops when the number of temperature transitions reaches a prespecified number, or when the temperature is reduced to a threshold or when the neighboring solution was not improved after a given period of time.

The algorithm of SA on RILP is summarized in Algorithm 1.

4. Case Study

In this section, we consider a special type of violator, drunk drivers who blend in with public travelers in an urban road network and aim to find a reliable layout of interceptors (DUI checkpoints) to catch them and consequently protect public innocent travelers. Several sets of numerical experiments are performed to test the computational performance of the proposed model and solution algorithm. All algorithms are coded in MATLAB R2010b and run on a desktop computer with 2.70 GHz CPU and 4.00 GB memory.

The proposed models are applied to the Sioux Falls network, which includes 24 nodes and 76 links, as shown in Figure 1. Every node represents a zone. The nodes with

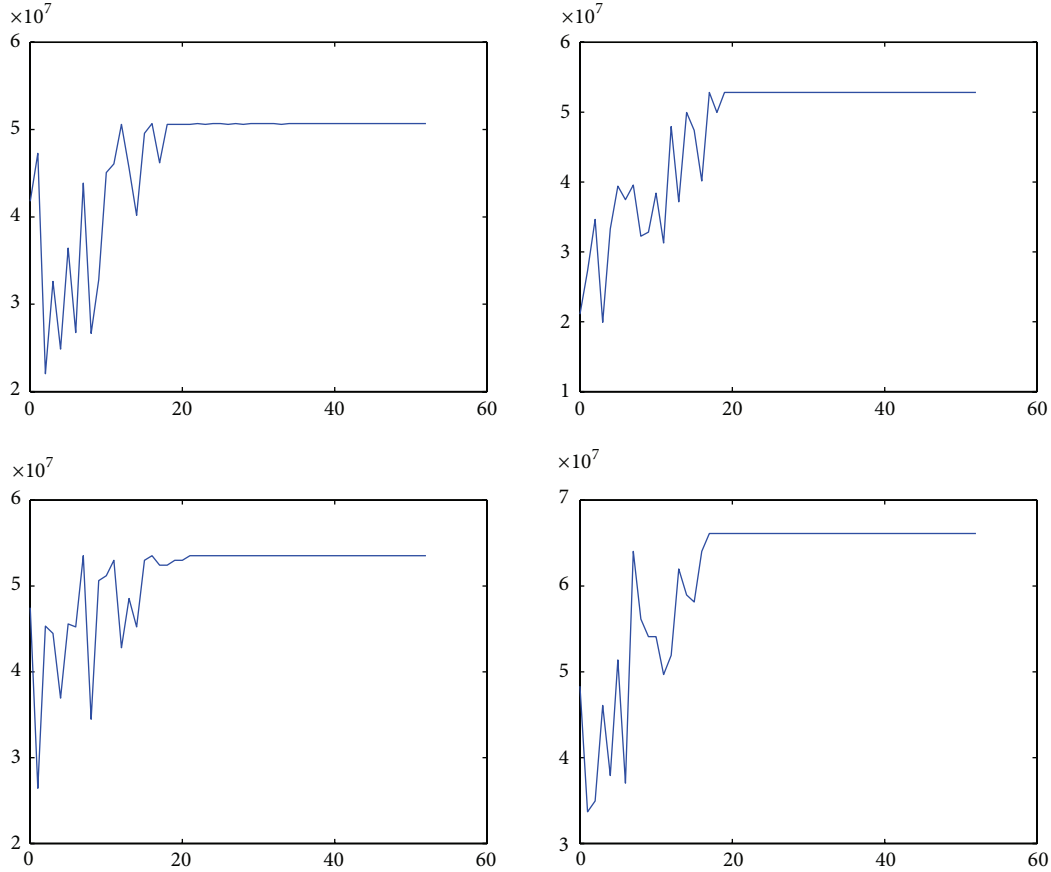


FIGURE 2: Objective function values generated by SA with $N = 5$, $T_0 = 500$, Markov length = 100, $\eta = 0.8$, $\varepsilon = 0.005$.

red/green colors are assumed to be the origins/destinations of drunk drivers. All OD trip data and network data, including the link lengths, link free flow travel times, link capacity, link characteristics (i.e., BPR function), and link indices, are downloaded from Bar-Gera [23]. Using network data and OD trip data, the traffic assignment of user equilibrium (UE) criteria [24] is implemented to obtain the link flow $f_a, \forall a \in A$. We then use f_a to calculate b_{ij} with formula (1).

The difficulty of implementing RILP lies in the exponential number of routes between each OD pair. This will be exacerbated for larger networks. However, it can reasonably be assumed that not every route will be used by drunk drivers. Based on this fact, we select a feasible and reasonable number of routes between each OD pair for drunk drivers. We only use those routes whose probabilities of being taken by drunk drivers are no less than a certain small value (e.g., 0.1). The probability that each route between a given OD pair is taken by drunk drivers is computed by the following Logit-like formula:

$$p_i = \frac{\exp(\sum_{m \in \varphi_r} t_m/t_i)}{\sum_{n \in \varphi_r} \exp(\sum_{m \in \varphi_r} t_m/t_n)}, \quad \forall r \in w, \forall i \in \varphi_r, \quad (12)$$

where t_i , t_m , and t_n are the travel times of routes i , m , and n between OD pair r , respectively, which can be obtained by the implementation of UE. The loop-less k -shortest path

algorithm presented by Yen [25] can find an arbitrary number of ranked shortest paths between each OD pair. We use that algorithm to successively find and add a new route to the set of k paths until the selected probability, computed by formula (12), is smaller than a predefined small value.

In our example, there are a total of 30 OD pairs (with 5 origins and 6 destinations) for generating and absorbing drunk drivers. The “demand” of drunk drivers between each OD pair is randomized between 1 and 100 in Excel 2010. The total number of routes generated by the k -shortest path algorithm is 99. In this example, all 24 nodes in the Sioux Falls network are assumed to be candidate locations for interceptors.

4.1. Solution Algorithm Performance. To analyze the performance of the SA algorithm in solving our RILP model, we run a series of instances for $p = 0.1$, $\beta_1 = 1000$, $\beta_2 = 2.5$, $N \in \{5, 10\}$, $T_0 = 500$, $Markov.Length = 100$, $\eta = 0.8$, and $\varepsilon = 0.005$. In each instance, the initial solution for the interceptor layout is randomized by the MATLAB function “Randperm”. The results are shown in Figures 2 and 3. As observed, the best objective function value always stabilizes after several iterations of SA in each run, and the near-optimal result is very similar across all runs for a specified N ($N = 5$ or $N = 10$). This indicates that the proposed SA algorithm for RILP is nonsensitive to the initial solution

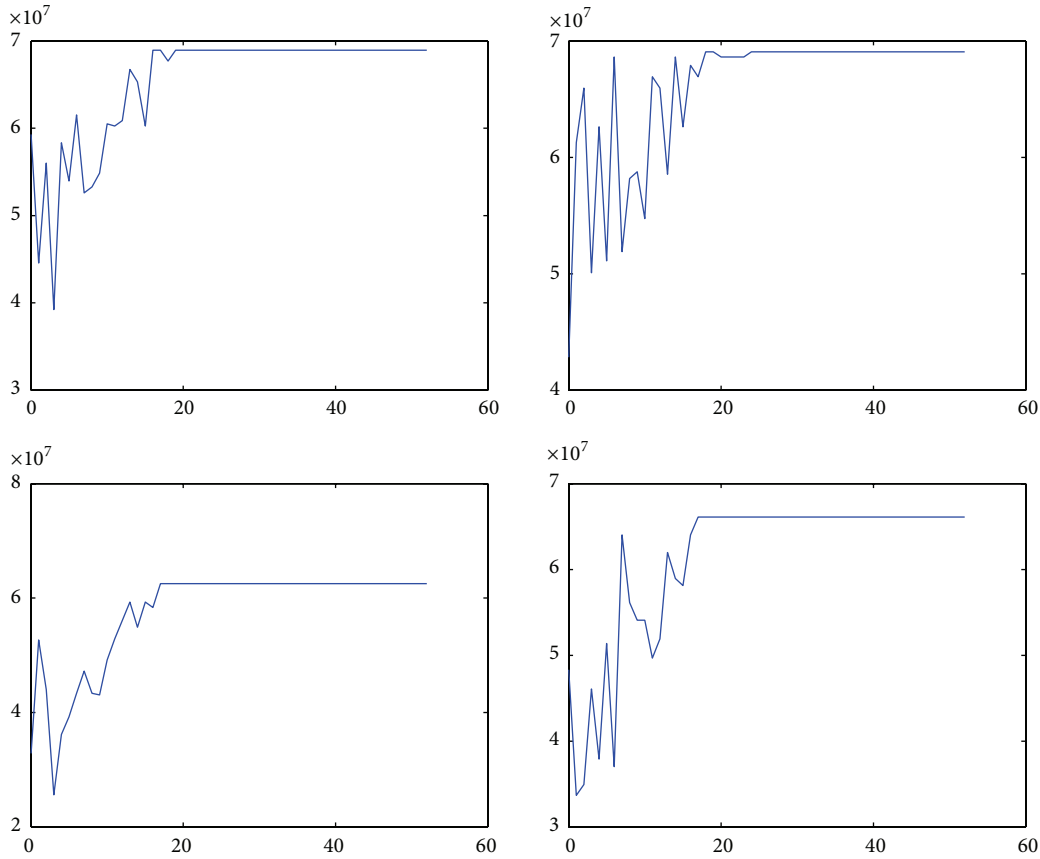


FIGURE 3: Objective function values generated by SA with $N = 10$, $T_0 = 500$, Markov length = 100, $\eta = 0.8$, $\varepsilon = 0.005$.

and performs stably. Additionally, each run almost stabilizes at close to 20 iterations, and in each iteration 100 (because $Markov_length = 100$) computations are necessary to obtain objective function values. Thus, SA uses approximately 22×100 computations to approach an optimal solution, while an enumeration algorithm may cost $C_{24}^5 = 42504$ computations. Therefore, the proposed SA algorithm is more effective at solving RILP than enumeration.

4.2. Sensitivity Analysis of Disruption Probability p . In this set of experiments, we aim to investigate changes in objective values and interceptor deployment solutions under different disruption scenarios.

Using four different disruption probability (p) values, we use SA to solve RILP with N values from 1 to 24. For each N , we implement SA 4 times and use the average objective value as the final result. Figure 4 shows the benefits obtained for each p over varying numbers of installed interceptors. Overall, objective values approach stability when the number of installed interceptors is more than 5 under each disruption scenario. We call $N = 5$ the “critical point”. This is a reasonable result because we assume there are only 5 potential origins for violators, and if these 5 origins all contain installed interceptors, all of the first nodes of each violator route will be covered. Before the number of installed interceptors achieved at the “critical point”, the potential improvement of

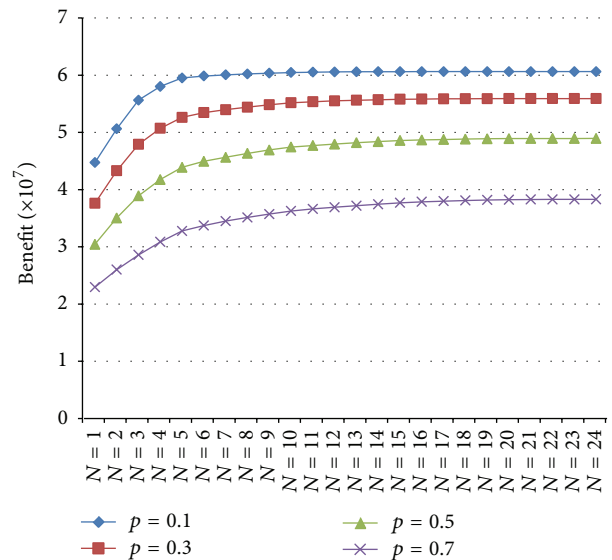


FIGURE 4: Objective values under different disruption probabilities p .

the objective value from adding a new interceptor will be much greater than that after the critical point. Under each disruption scenario, the curve of objective values increases very steeply before the “critical point”, then increases at a flatter trajectory after $N = 5$. We can see that the larger

TABLE 1: Interceptor deployment solutions under four disruption scenarios.

	Selected candidate interceptor
$N = 3$	
$p = 0.1$	8, 10, 15
$p = 0.3$	8, 10, 15
$p = 0.5$	9, 10, 15
$p = 0.7$	9, 10, 15
$N = 4$	
$p = 0.1$	8, 9, 10, 15
$p = 0.3$	8, 9, 10, 15
$p = 0.5$	8, 9, 10, 15
$p = 0.7$	9, 10, 15, 16
$N = 5$	
$p = 0.1$	8, 9, 10, 15, 17
$p = 0.3$	8, 9, 10, 15, 17
$p = 0.5$	8, 9, 10, 15, 17
$p = 0.7$	8, 9, 10, 15, 16

the disruption probability, the sharper the increase in the objective value before the “critical point” and the slower the increase after the “critical point”. Figure 4 also indicates that while N is determined, the decrease in objective values is not linear across different disruption probabilities, although we attempt to linearly increase the probability. The difference in objective values between any two neighbor disruption scenarios is exacerbated sequentially.

Table 1 presents interceptor solutions under four different disruption scenarios (with only $N = 3, 4, 5$ shown as examples). We can see that when the disruption probability varies, the “optimal” interceptor solution also changes.

4.3. Additional Discussions. In practice, β_1 represents the resultant benefit from a one unit increase in the probability of a drunk driver avoiding an interceptor, and β_2 represents the cost incurred by increasing the travel distance by one unit. One should be careful in determining the values of these two parameters because they will cause significant differences in how drunk drivers select routes and, consequently, differences in the optimal interceptor layout. If a government increases the amount of attention it pays to eliminating drunk driving, β_1 should be very large (e.g., in China, drunk driving leads to high fines and also may lead to suspended driver’s licenses and jail time).

5. Conclusion

This paper studies an interceptor location problem in an arc-node network. The interceptor identifies violators from network traffic and reduces the safety risk for the general public. We propose a reliable interceptor location model (RILP) to account for interceptor disruption and the “game” played between interceptor planners and violators. The RILP is a nonlinear, nonconvex binary integer programming model that is difficult to solve. We develop a simulated annealing (SA) algorithm to solve the RILP, and numerical case studies

are conducted to test the algorithm. Various managerial insights are also drawn from the numerical results. We find that the proposed algorithm is able to effectively and efficiently solve the RILP. Sensitivity analysis of the disruption probability shows that (i) the total benefit of the system decreases as the disruption probability increases and (ii) there exists a “critical point” for the number of installed interceptors. We further discussed the physical meanings of two parameters in a Logit model for the selection of routes by drunk drivers and how to determine these parameters.

In this study, we consider a heuristic algorithm, SA. Because of the uncertainty inherent in heuristic methods, future studies should aim to develop global optimal algorithms to solve this nonlinear programming model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Project no. 71203045), Heilongjiang Natural Science Foundation (Project no. E201318), and the Fundamental Research Funds for the Central Universities (Grant no. HIT.KISTP.201421). This work was performed at the Key Laboratory of Advanced Materials & Intelligent Control Technology on Transportation Safety, Ministry of Communications, China.

References

- [1] B. Hubicka, H. Laurell, and H. Bergman, “Criminal and alcohol problems among Swedish drunk drivers—predictors of DUI relapse,” *International Journal of Law and Psychiatry*, vol. 31, no. 6, pp. 471–478, 2008.
- [2] S. Pavanello, R. Snenghi, A. Nalesso, D. Sartore, S. D. Ferrara, and M. Montisci, “Alcohol drinking, mean corpuscular volume of erythrocytes, and alcohol metabolic genotypes in drunk drivers,” *Alcohol*, vol. 46, no. 1, pp. 61–68, 2012.
- [3] M. Portman, A. Penttilä, J. Haukka et al., “Profile of a drunk driver and risk factors for drunk driving. Findings in roadside testing in the province of Uusimaa in Finland 1990–2008,” *Forensic Science International*, vol. 231, no. 1–3, pp. 20–27, 2013.
- [4] S. Velmurugan, S. Padma, E. Madhu, S. Anuradha, and S. Gangopadhyay, “A study of factors influencing the severity of road crashes involving drunk drivers and non drunk drivers,” *Research in Transportation Economics*, vol. 38, no. 1, pp. 78–83, 2013.
- [5] National Highway Traffic Safety Administration (NHTSA), *Traffic Safety Facts: Alcohol-Impaired Driving*, US Department of Transportation, Washington, DC, USA, 2012, <http://www-nrd.nhtsa.dot.gov/Pubs/811606.pdf>.
- [6] C. F. Greer and K. D. Moreland, “United Airlines and American Airlines’ online crisis communication following the September 11 terrorist attacks,” *Public Relations Review*, vol. 29, no. 4, pp. 427–441, 2003.
- [7] V. S. Guzhva and N. Pagiavlas, “US Commercial airline performance after September 11, 2001: decomposing the effect of the

- terrorist attack from macroeconomic influences," *Journal of Air Transport Management*, vol. 10, no. 5, pp. 327–332, 2004.
- [8] H. Ito and D. Lee, "Assessing the impact of the September 11 terrorist attacks on U.S. airline demand," *Journal of Economics and Business*, vol. 57, no. 1, pp. 75–95, 2005.
- [9] K. V. Smith, C. Mansfield, and H. A. Klaiber, "Terrorist threats, information disclosures, and consumer sovereignty," *Information Economics and Policy*, vol. 25, no. 4, pp. 225–234, 2013.
- [10] R. Zenklusen, "Network flow interdiction on planar graphs," *Discrete Applied Mathematics*, vol. 158, no. 13, pp. 1441–1455, 2010.
- [11] D. Granata, G. Steeger, and S. Rebennack, "Network interdiction via a critical disruption path: branch-and-price algorithms," *Computers & Operations Research*, vol. 40, no. 11, pp. 2689–2702, 2013.
- [12] M. A. Rad and H. T. Kakhki, "Maximum dynamic network flow interdiction problem: new formulation and solution procedures," *Computers and Industrial Engineering*, vol. 65, no. 4, pp. 531–536, 2013.
- [13] J. Yates and S. Sanjeevi, "A length-based, multiple-resource formulation for shortest path network interdiction problems in the transportation sector," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 107–119, 2013.
- [14] K. J. Cormican, D. P. Morton, and R. K. Wood, "Stochastic network interdiction," *Operations Research*, vol. 46, no. 2, pp. 184–197, 1998.
- [15] J. E. Ramirez-Marquez and C. M. Rocco S., "Stochastic network interdiction optimization via capacitated network reliability modeling and probabilistic solution discovery," *Reliability Engineering and System Safety*, vol. 94, no. 5, pp. 913–921, 2009.
- [16] P. B. Mirchandani, M. Gentili, and Y. He, "Location of vehicle identification sensors to monitor travel-time performance," *IET Intelligent Transport Systems*, vol. 3, no. 3, pp. 289–303, 2009.
- [17] X. Li and Y. F. Ouyang, "Reliable sensor deployment for network traffic surveillance," *Transportation Research B: Methodological*, vol. 45, no. 1, pp. 218–231, 2011.
- [18] Y. F. Ouyang, X. Li, C. P. L. Barkan, A. Kawprasert, and Y. C. Lai, "Optimal locations of railroad wayside defect detection installations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 24, no. 5, pp. 309–319, 2009.
- [19] F. O. Hounkpevi and E. E. Yaz, "Robust minimum variance linear state estimators for multiple sensors with different failure rates," *Automatica*, vol. 43, no. 7, pp. 1274–1280, 2007.
- [20] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: the expected failure cost case," *Transportation Science*, vol. 39, no. 3, pp. 400–416, 2005.
- [21] X. Li and Y. Ouyang, "A continuum approximation approach to reliable facility location design under correlated probabilistic disruptions," *Transportation Research B: Methodological*, vol. 44, no. 4, pp. 535–548, 2010.
- [22] C. Q. Liu, *Advanced Traffic Planning*, People's Traffic Press, 2001.
- [23] H. Bar-Gera, "Transportation network test problems," 2001, <http://www.bgu.ac.il/~bargera/tntp/>.
- [24] Y. Sheffi, *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice Hall, England Cliffs, NJ, USA, 1985.
- [25] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, pp. 712–716, 1971.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

