Security Routing Games with Multivehicle Chinese Postman Problem

Dorit S. Hochbaum and Cheng Lyu
Department of Industrial Engineering and Operations Research, University of California, Berkeley

Fernando Ordóñez
Department of Industrial Engineering, Universidad de Chile, Santiago

Key in the efforts to deter and prevent nuclear terrorism is the ability to detect the presence of possible nuclear threats in a given area. Resources capable of detecting such threats are limited, expensive, and only capable of scanning a certain total area in a given amount of time. This limit on the ability to detect nuclear threats makes imperative the development of efficient deployment strategies of the detection resources. In this work, we propose a Stackelberg game-based model to determine the optimal patrolling strategy of security assets over a network in the presence of a strategic adversary that seeks to place a nuclear threat on edges of the network. To efficiently solve this model, we introduce a novel decomposition of the problem which requires the solution of a multivehicle rural Chinese postman problem (CPP). Our theoretical contributions present hardness and approximation results for the k-vehicle rural CPP. Our computational results demonstrate the benefit of this decomposition for the nuclear threat detection security problem. **NETWORKS, Vol. 64(3), 181–191 2014**

Keywords: Chinese postman; vehicle routing; adversarial games; security games; approximation algorithm

1. INTRODUCTION

In this era of heightened threats of nuclear terrorism, security forces face a difficult challenge: limited security resources must be deployed effectively to protect the homeland against many different threats. Effective algorithmic methods lead to a more effective use of expensive and sophisticated detection equipment, and reduce the requirements for man-power, by producing automatic alerts in suspect cases that merit additional investigation.

In this work, we focus on the problem of detecting possible nuclear threats in a given area represented by a network. An approach to detect the presence of Special Nuclear Material (SNM) in an area is to use a highly sensitive sensor mounted on a specially designed truck to patrol the area and contrast the radiation readings found during the patrols with the radiation background. Significant differences from the radiation background could indicate the presence of a threat and would help focus more detailed inspections. The trucks are mounted with multiple detectors and their cost is very high. Consequently, the number of trucks available is too small to allow to patrol all target areas of interest. The efficiency with which SNMs can be detected in an area strongly depends on the patrolling or inspection visits policies to this area. For example, if a deterministic route is used to patrol for changes in radiation, and it becomes known to the attacker, then the attacker needs only to wait for the sensor to pass a certain location to attack or move through that location undetected. This example also illustrates that it is critical to consider the strategic nature of the attacker when planning the patrolling strategies. Therefore, we consider the problem of constructing efficient patrolling strategies over a network that is being monitored for changes in background radiation, taking into account that the patrolling strategy can be observed by a potential adversary.

Recent work in game theory has represented problems, where the defender takes a strategic attacker into account in determining its optimal strategy as Stackelberg game models [16, 21, 23]. Stackelberg games are models of interaction between agents where one can commit to a strategy before the rest and have been widely studied in domains involving defenders and adversaries [4, 25]. In a Stackelberg model, the defender (security forces) acts first and the attacker (terrorist) can observe these actions and, using this information, then decides how and when to attack. This work has lead to real-world deployments of Stackelberg-based security models for

Received December 2012; accepted August 2014
Correspondence to: D. S. Hochbaum; e-mail: hochbaum@ieor.berkeley.edu
Contract grant sponsor: NSF; Contract grant numbers: CMMI-1200592, CBET-0736232 (to D. S. H)
Contract grant sponsor: Conicyt; Contract grant number: ACT87 (to F. O)
Contract grant sponsor: Instituto Milenio Sistemas Complejos de Ingeniería; Contract grant numbers: ICM.FIC P05-004, FBO-16 (to F. O)
DOI 10.1002/net.21563
Published online 15 September 2014 in Wiley Online Library (wileyonlinelibrary.com).

the LAX Airport, Federal Air Marshal Service [16], and US Coast Guard [24]. These existing security games consider the targets as independent. We adapt these Stackelberg game models for the problem of patrolling a network to detect SNMs. Since the prospective targets are within road network, detecting the presence of SNMs by a traveling vehicle means that rather than point-wise detection, the detector trucks traverse street blocks and once the truck enters a certain street block it can detect radiation changes within the block. As such, we consider the targets graph representation as edges in a graph. The targets in this domain are dependent since their geographical positions impose an added burden on the resources (detection vehicles in this case). When a vehicle is inspecting geographically dispersed targets, valuable capacity of the resource is consumed while traversing the network from one target location to the next.

Other security detection plans that optimize the accuracy of detection of SNMs were given in [13, 12]. That plan was not set to take into account adversarial game issues as resource availability was not a consideration in that context. Instead it addressed the problem of providing comprehensive detection capability to a large city by mounting relatively inexpensive detectors on vehicles such as taxi cabs, and collecting the information along the routes traversed by those taxi cabs, without dictating what the routes should be. Another work related to security is on video tracking of tagged objects, [10]. This work also does not consider limits on security resources, and therefore, is not concerned with adversarial aspects of the problem.

Previous work that addresses the effect of geographical locations on the defender strategy in the context of adversarial games, either leads to computationally intensive approaches that are impractical for real world problems, or this type of dependency is so difficult to address that the developed methods can be applied only for very specific target configurations, again impractical for our purposes [1, 14, 17, 24]. Solving games of the type and magnitude that occur when geographical considerations are incorporated is well beyond the capabilities of current algorithms. We, therefore, tackle the problem, by decomposing the construction of the proposed strategy into two phases:

Phase 1: Determines the optimal defender strategy as if the targets were independent. We represent the nuclear detection patrolling problem as a Stackelberg security game, where the defenders view targets independently without considering the geographical effect on defender strategies. This phase develops mixed strategies for defending the targets based only on the relative importance of their rewards. The output of this stage will be a frequency with which each target is protected, assigning a probability of protection to each target. This represents the optimal defender strategy for independent targets.

Phase 2: Determines defender patrols that match the optimal outcome of the adversarial game (from phase 1) given limited resources (k defenders with limited capacity) and a maximum set of targets within a certain area. We show that determining the optimal solution for this problem is the NP-hard k-vehicle Rural Postman Problem (k-RPP). We devise a $(2.5 - \frac{1}{k})$-approximation algorithm for this problem that works in any setting (including nonmetric setting). This approximation algorithm is used to construct maximal reward patrolling routes for k defenders that visit the edges sampled according to the optimal defender strategy from phase 1.

1.1. Security Games

We now provide a brief background on Stackelberg security games based on [16, 23]. These security game models represent the interaction between the defender and the attacker as a Stackelberg game. In this game, the defender is the leader and acts first, selecting an action i from a set of possible actions X with probability x_i. The attacker decides its action after the defender has committed to a strategy, hence is referred to as the follower, and taking into account the defender's strategy decides an action j from a set of actions Q with probability q_j. We use vectors $\mathbf{x} = (x_i)_{i\in X}$ and $\mathbf{q} = (q_j)_{j\in Q}$ to denote the complete strategies of the leader and follower, respectively. Let e be a vector of all 1's. Then, if the rewards for the defender (attacker) of the combined actions $i \in X$ and $j \in Q$ are denoted by R_{ij} (respectively C_{ij}), the bilevel program that determines the optimal strategy $\mathbf{x}$ for the leader can be denoted by

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{q}} \sum_{i\in X, j\in Q} R_{ij} x_i q_j \\ &\quad \text{s.t. } e^T\mathbf{x} = 1,\ \mathbf{x} \geq 0 \\ &\qquad q_j = \underset{e^T\mathbf{q}=1,\mathbf{q}\geq 0}{\operatorname{argmax}} \sum_{i\in X, k\in Q} C_{ik} x_i q_k. \end{aligned}$$

This problem finds the leader's mixed strategy that maximizes its revenue assuming the adversary breaks ties in favor of the leader (the strong Stackelberg equilibrium solution). We note that although the follower has more information when making a decision, the leader in a Stackelberg game can only improve from the Nash equilibria solution. This is because the leader can always play the Nash equilibrium solution, forcing the follower to also play Nash. Hence, the leader will only deviate if it is to its advantage. For example, suppose the defender has to decide in which of n possible entry points to place a detector of nuclear material and subsequently the attacker, knowing the frequency with which a detector is placed in each entry point, then decides which of the n entry points attack. In this case, both sets of actions equal $X = Q = \{1 \dots n\}$.

For every fixed leader strategy $\mathbf{x}$, the inner optimization solved by the attacker is a linear optimization problem over the simplex. In this case, since there is always an optimal extreme point and all extreme points correspond to pure strategies, there is no loss of generality in considering only pure strategies for the adversary. We can then express the leader's optimization problem as a mixed integer programming problem using the integrality of the adversary's strategies. For this we need auxiliary variables d to represent the reward obtained by the defender and a to represent the optimal reward obtained by the attacker. The problem is expressed as follows:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{q},d,a} \ & d \\ \text{s.t.} \ & \sum_{i\in X} R_{ij}x_i \geq d - M(1-q_j)\ j \in Q \\ & \sum_{i\in X} C_{ij}x_i \leq a\ j \in Q \\ & \sum_{i\in X} C_{ij}x_i \geq a - M(1-q_j)\ j \in Q \\ & e^T\mathbf{x} = 1,\ \mathbf{x} \geq 0 \\ & e^T\mathbf{q} = 1,\ \mathbf{q} \in \{0,1\}^{|Q|} \end{aligned} \tag{1}$$

where M is a large constant and the first set of constraints is a linealization of the defender's reward given the attacker's optimal pure strategy. The second set of constraints enforces that a is an upper bound on the attacker's reward for any action and the third makes a the optimal reward value of the attacker for the action q_j selected.

Bayesian formulations of this security game have been developed in [23, 26] that considers multiple security resources and allows for uncertainty over adversary types, rationality, and observation capabilities. Efficient algorithms are developed to solve these Bayesian Stackelberg games. These efficient algorithms exploit structure present in the security domain, such as: (1) the rewards (for every player) take only two values, depending on whether attacker and defender meet or not at a target. If they meet, an attack is prevented. (2) The network structure that exists between targets [15, 16, 18]. Additional work has considered the development of algorithms when there is uncertainty due to how the optimal defender strategy is implemented or how it is observed by the attacker [26].

2. THE CHINESE POSTMAN PROBLEM, THE *K*-CHINESE POSTMAN PROBLEM AND THE RURAL POSTMAN PROBLEM

We are concerned here with edge traversal, and therefore, we assume that the given graph, or network, representing the city where the detection of threats is to be deployed, is represented by an undirected graph $G = (V, E)$. Although the directed version and mixed version are of interest as well, their discussion is beyond the scope of this article.

The set of targets to be protected and traversed form a subset of edges in a graph. As such the Chinese Postman Problem (CPP) concerning the shortest tour that traverses each edge at least once is relevant here. The CPP is a classical problem of finding a tour, in a given graph $G(V, E)$, that starts from the central depot (post office), traverses each edge (street) in E at least once, then returns to the same depot node with the total distance traveled minimized. The problem was originally proposed in 1962 by Guan [11] in Chinese Mathematics; thus the name. CPP is polynomial time solvable on undirected graphs, or on directed graphs [6], but is NP-hard on mixed graphs, that contain both (undirected) edges and (directed) arcs, [20].

There are several versions of the k-CPP problem. The problem involves k postmen (referred to henceforth as vehicles) that have to cover all edges in a graph $G(V, E)$. Depending on how the objective function is defined the problem may or may not be hard. In [22], Pearn considers the following definition of k-CPP: For a graph $G = (V, E)$ with edge weights (distances), the k-CPP is to find a set of k tours so that: (1) Each route begins and ends at a common node called the central depot, (2) Each edge in E must be serviced by exactly one vehicle, (3) All the k vehicles must be involved in the delivery service, and (4) The total distance traveled is minimized. Pearn, [22], showed that this k-CPP is solved in polynomial time, by reducing it to a specific CPP.

The k-CPP problem, where each vehicle has a bounded capacity in terms of the total edge distances it can traverse, is to find up to k routes (that could be paths rather than closed tours) that jointly traverse all edges of E at least once. Even this simple decision problem is NP-complete with an easy reduction from k-bin packing. Our interest is in the problem where there is a common depot and each vehicle has to complete a tour.

Another variant of interest of CPP is the RPP [19]. In that problem there is a subset of edges $E' \subset E$ that must be traversed. Removing the requirement that all edges must be traversed turns the CPP into the NP-complete RPP. This is easy to see by a reduction from the Traveling Salesman Problem (TSP). The multivehicle RPP is obviously hard as well. We will consider the version of this problem where each vehicle has a bounded capacity ℓ. The respective decision problem is formally defined as follows:

k-RPP(ℓ) problem:

Input: An undirected graph, $G = (V, E)$ with edge weights d_{ij} for all $[i, j] \in E$, and a subset $E' \subseteq E$, $k > 1$, a depot node v, a positive number ℓ.

Question: Are there k tours $E_1, ..., E_k$, each including v, so that $d(E_i) \leq \ell$ and so that $E' \subseteq \cup_{i=1}^{k} E_i$?

Here, the notation $d(E_i)$ refers to the sum of edges weights in the tour E_i. In the optimization version of this problem, one seeks to minimize the value of the capacity ℓ so that there is a feasible solution to the decision problem k-RPP(ℓ).

To our knowledge, the most relevant result to this latter problem in the literature is that of [2], that seeks to minimize the capacity for a k-RPP (with no depot). The authors devise a 7-approximation algorithm for this problem, where there is no depot and the vehicles are not required to complete tours, but are required instead to traverse paths that cover the subset of edges of interest. The objective is to minimize the length of the maximum path. Conversely, the optimization version of the decision problem k-RPP(ℓ) is closely related to the min-max k-RPP (see, e.g., [3]). Consider any instance of the optimization version of k-RPP(ℓ), either the problem is infeasible, or it should have the same optimum as the corresponding min-max k-RPP. However, to our knowledge, no approximation algorithms are devised for the min-max k-RPP.

3. AN APPROXIMATION ALGORITHM FOR *K*-RPP(ℓ)

3.1. A 1.5-Approximation Algorithm for RPP

We provide here a 1.5 approximation algorithm for the RPP which is an easy extension of Christofides' 1.5-approximation for TSP on graph weights that satisfy the triangle inequality (metric graphs), [5]. Existence of a 1.5-approximation algorithm for RPP based on an extension of Christofides' algorithm was mentioned by [7], without proof. We provide here the proof and demonstate that the triangle inequality assumption, made in [5], is unnecessary—the result holds for any graph weights. Christofides' algorithm identifies first a minimum spanning tree, and then extends it to a Eulerian graph and TSP tour.

An instance of RPP includes a graph $G = (V, E)$ with edge weights d_{ij} for all $[i,j] \in E$, and a subset of edges $E' \subset E$ that must be covered by a tour. We denote an optimal solution to the RPP problem, by RPP*, and the sum of edge weights in RPP* is denoted by $d(\text{RPP}^*)$. The concept of a spanning tree in Christofides' algorithm is replaced, for the RPP, by a minimum cost collection of edges that contain E' and form a connected subgraph of G. Such connected subgraph exists, with total weight that is only less than the cost of the optimal RPP tour, $d(\text{RPP}^*)$, since the set of edges of the optimal RPP tour RPP* induces a connected subgraph containing E'.

Let $C_1, \dots C_q$ be the set of connected components induced by the edges of E' in G. Therefore, $\{C_1, \dots C_q\}$ forms a partition of E'. Define a complete graph $H = (V(H), E(H))$ on q nodes where node k represents the component C_k. Let $\bar{d}_{ij}$ be the length of the shortest path $\bar{p}_{ij} = [i, i_1, \dots, i_r, j]$ between i and j in G. Let the weight of an edge in H between node $k \in V(H)$ and node $k' \in V(H)$ be the shortest path distance between the two components: $d'_{kk'} = \min_{i \in C_k, j \in C_{k'}} \bar{d}_{ij}$. Let $T^{(C)}$ be a minimum spanning tree in H, of total length $d(T^{(C)})$.

Theorem 3.1. *The total weight of the edges in E' and the spanning tree $T^{(C)}$ is less than the weight of the edges in RPP*:*

$$d(E') + d(T^{(C)}) \leq d(RPP^*).$$

Proof. Consider the optimal tour RPP* in the order of edges visited. As a "first" edge in the tour, pick, arbitrarily, an edge of E' in the tour RPP* from component C_{i_1}, e_{i_1}. Let the truncated sequence of edges in the tour RPP*, starting with edge e_{i_1} be:

$$[e_{i_1}, \bar{P}^{i_1,i_1}, e'_{i_1}, P^{i_1,i_2}, e_{i_2}, \bar{P}^{i_2,i_2}, e'_{i_2}, \dots, P^{i_{j-1},i_j}, e_{i_j}, \bar{P}^{i_j,i_j}, e'_{i_j}, \dots, P^{i_{q-1},i_q}, e_{i_q}]$$

This sequence is truncated once all q components have been visited, in the sense that this section of the tour traversed at least one edge in each component. The notation is interpreted as follows: e_{i_j} denotes the first edge of component C_{i_j} visited in the sequence; $\bar{P}^{i_j,i_j}$ denotes the sequence of edges on the tour that traverses, in addition to edges of E, only edges in $\cup_{k=1}^{j} C_{i_k}$; e'_{i_j} denotes the last edge in the sequence, in components $\cup_{k=1}^{j} C_{i_k}$ prior to the first visit to the next component $C_{i_{j+1}}$. It could be the case that $\bar{P}^{i_j,i_j} = \emptyset$ and, in that case $e_{i_j} = e'_{i_j}$. That is, the first and last edges are the same.

Note that the sections of the tour $P^{i_1,i_2}, \dots, P^{i_{q-1},i_q}$ consist of edges of E only and induce a spanning tree in the graph H, since each connect components $\cup_{k=1}^{j-1} C_{i_k}$ to component C_{i_j} via the path in E, P^{i_{j-1},i_j}. Therefore, the total length of $\cup_{j=1}^{q} P^{i_{j-1},i_j}$ is greater or equal to $d(T^{(C)})$. Since the tour RPP* also traverses all edges of E' at least once, the stated result follows. ∎

Let V' be the set of nodes of G induced by the edges of E'. The set of edges $E' \cup T^{(C)}$ is a connected, not necessarily simple, subgraph of G. Consider the degrees of the nodes induced by this multigraph, counting the multiplicity of an edge in the degree. Then only nodes of V' can have odd degree, as nodes of $V \backslash V'$ can only be internal nodes in the shortest paths added in the graph H, and as such their degrees must be even.

Let the subset of the nodes of odd degree be $V'_{\text{odd}} \subseteq V'$, and $|V'_{\text{odd}}| = p$. Define a complete graph H'_{odd} that has for each pair of nodes in V'_{odd} an edge connecting them of cost of the shortest path in G between the respective nodes. We find in H'_{odd} a minimum cost perfect matching M^*_{odd}.

Lemma 3.1. *The weight of the edges in M^*_{odd} is less than half the weight of the edges in RPP*,*

$$d(M_{odd}) \leq \frac{1}{2} d(RPP^*).$$

Proof. The proof here is analogous to the respective proof for TSP: Consider the tour RPP* restricted to the set of nodes of V'_{odd} visited in the order $v_{j_1}, v_{j_2}, \dots, v_{j_p}$ and replace the tour section between each pair of nodes $v_{j_\ell}, v_{j_{\ell+1}}$ by the shortest path between the respective nodes. Since the number of nodes of odd degree p is even, the collection of these paths sections form two disjoint feasible matchings on the nodes of V'_{odd}, one on the edges of H'_{odd}. One is of weight $\sum_{\ell=1}^{\frac{p}{2}} \bar{d}_{j_{2\ell-1},j_{2\ell}}$, and the other of weight $\sum_{\ell=1}^{\frac{p}{2}} \bar{d}_{j_{2\ell},j_{2\ell+1}}$. Thus, $d(\text{RPP}^*) \geq 2d(M^*_{\text{odd}})$. ∎

Combining the results of Theorem 3.1 with Lemma 3.1 gives an Eulerian graph that includes all edges of E' of length at most 1.5 times the optimum $d(\text{RPP}^*)$.

3.2. A $(2.5 - \frac{1}{k})$-Approximation Algorithm for k-RPP(ℓ)

It is assumed that the k-RPP(ℓ) has a feasible solution. This requires the existence of a vehicle capacity, ℓ, that is

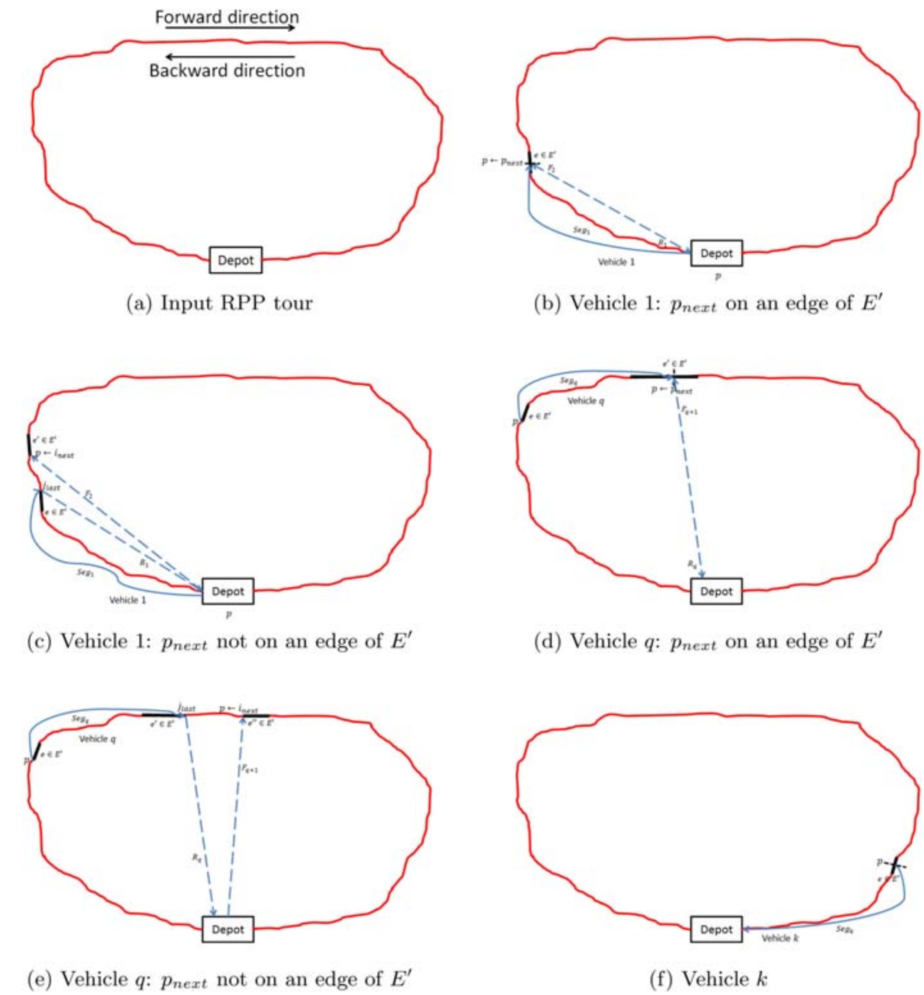


FIG. 1. The $(2.5 - \frac{1}{k})$-approximation algorithm for k-RPP(ℓ). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

large enough so that any single edge e of E', e can be traversed with a vehicle that arrives along a shortest path from depot to e, traverses it, and returns to the depot via a shortest path. Therefore, from any point (not necessarily an endpoint) on any edge of E' there is a path of length less than $\frac{1}{2}\ell$ to the depot.

Let ℓ^* be the smallest capacity for which k-RPP(ℓ) has a feasible solution, i.e. there are k tours, each including the depot and each of length at most ℓ^*, covering all the edges of E'. Let $\ell^{\max}$ be the maximum of the shortest path distance between any point on any edge of E' and the depot. The following fact follows from the discussion earlier:

Fact 3.1. $\ell^{\max} \leq \frac{1}{2}\ell^*$

Recall that $d(\text{RPP}^*)$ is the length of the shortest RPP tour (single tour, or $k=1$) that covers all edges of E'. Then, the following is obvious:

Fact 3.2. *The minimum vehicle capacity value, ℓ^*, required to cover all the edges of E' with k vehicles satisfies,*

$$\ell^* \geq \frac{d(RPP^*)}{k}.$$

3.2.1. The Algorithm. The approximation algorithm for k-RPP(ℓ) takes as input a feasible and approximate RPP tour that covers all the edges of E'. The idea of the algorithm is to partition the approximate RPP tour into k segments, and add to each one the necessary trip from the depot and back to the depot, so that the length of each tour of the k resulting tours is at most a kth of the length of the tour plus $2\ell^{\max}$. The challenge is that edges of $E \backslash E'$ maybe further away than $\ell^{\max}$ from the depot. It is, therefore, necessary to construct a partition so that the "start" edge of each segment and the "end" edge of each segment are edges in E'. Also, note that the first segment starts from the depot, and therefore, does not require to arrive at the first edge from the depot; similarly, the last, kth, segment arrives at the depot, and therefore, does not require the addition of a path to the depot.

Let the 1.5-approximate RPP tour, from Section 3.1, be denoted by t_{RPP}, of length $T^H = d(t_{\text{RPP}})$. Since the tour t_{RPP} is an undirected cycle, we consider the clockwise and counterclockwise directions from the depot (Fig. 1a) and refer to the clockwise direction as the forward direction and the counterclockwise direction as the backward direction. Advancing distance L along the tour from point p means going forward from p to a point a distance L away. Similarly, retracting distance L along the tour from point p means going backward from p to a point a counterclockwise distance L away.

Define p as the point (not necessarily an endpoint of an edge) on t_{RPP} for the next vehicle to first reach from the depot to traverse along t_{RPP}. Initially p is the depot node. Define p_{next} as the point (not necessarily an endpoint of an edge) on t_{RPP} for the vehicle to finish traversing along t_{RPP} and return to the depot. Denote the depot node as node 0. Define $L = \frac{T^H - 2\ell^{\max}}{k}$.

Procedure 2.5- approximate k-RPP $(G, E', t_{\mathrm{RPP}}, k)$ **:**

Step 1: $q=1$. Let p_{next} be the point on the tour t_{RPP} at a forward distance $L' = L + \ell^{\max}$ from the depot p.
Step 2: If p_{next} is on an edge of E', set Seg_q equal to the tour segment from p to p_{next}. Set $p \leftarrow p_{\mathrm{next}}$; go to step 5.
Step 3: Else, retract from p_{next} backward toward p until the first edge in E' is reached and its endpoint j_{last}. If no such edge exists then $\mathrm{Seg}_q = \emptyset$ (the qth tour is empty) and go to step 4. Else, set $p_{\mathrm{next}} \leftarrow j_{\mathrm{last}}$. Let Seg_q be the tour segment from p to p_{next}.
Step 4: Advance from p_{next} to the first edge encountered in E' and its endpoint i_{next}. If no such edge, STOP. Set $p \leftarrow i_{\mathrm{next}}$.
Step 5: Let R_q be the shortest (return) path from p_{next} to depot. Let F_{q+1} be the shortest (forward) path from the depot to p. Set $q \leftarrow q+1$. If $q=k$ go to step 7.
Step 6: Else, advance distance L from p. Let p_{next} be the point on the tour t_{RPP} at a forward distance L from point p. Go to step 2.
Step 7: $q=k$. The depot is at a distance at most $L' = L + \ell^{\max} = \frac{T^H + (k-2)\ell^{\max}}{k}$ from p. Seg_k is the tour segment from p to the depot. STOP.

Note that in step 2, if p_{next} is on an edge of E', then the return and forward paths in step 5, are identical, since in this case $p = p_{\mathrm{next}}$.

If the algorithm reaches STOP in step 3, then there are $q<k$ tours that cover all the edges of E' within the designated capacity, that will be shown to be within a factor of 2.5 of the optimum. The output set of tours are,

$$k - \mathrm{tours}(H) = \{(\mathrm{Seg}_1, R_1), (F_2, \mathrm{Seg}_2, R_2), \ldots, (F_{k-1}, \mathrm{Seg}_{k-1}, R_{k-1}), (F_k, \mathrm{Seg}_k)\}.$$

A figure illustration of the above 2.5-approximate k-RPP procedure is displayed in Figure 1. The input 1.5-approximate RPP tour t_{RPP} is illustrated in Figure 1a. For vehicle 1, if p_{next} is on an edge of E', the process of the algorithm (Step 1 $\rightarrow$ Step 2 $\rightarrow$ Step 5) is displayed in Figure 1b; if p_{next} is not on an edge of E', the process of the algorithm (Step 1 $\rightarrow$ Step 3 $\rightarrow$ Step 4) is shown in Figure 1c. Then for any vehicle q ($2 \leq q \leq k-1$), similarly, if p_{next} is on an edge of E', the process of the algorithm (Step 6 $\rightarrow$ Step 2 $\rightarrow$ Step 5) is illustrated in Figure 1d; if p_{next} is not on an edge of E', the process of the algorithm (Step 6 $\rightarrow$ Step 3 $\rightarrow$ Step 4) is demonstrated in Figure 1e. Finally, for vehicle k, the process of the algorithm (**Step 7**) is shown in Figure 1f.

Notice that some tour splitting procedures similar to the above that obtain $e + 1 - (1/k)$-approximation starting from e-approximation were proposed by [8, 9] for different arc routing problems (e.g., for the min-max k stacker crane problem and min-max k-CPP).

3.2.2. Correctness Proof

Theorem 3.1. P*rocedure* 2.5-*Approximate k-RPP outputs a feasible solution to k-RPP(ℓ).*

Proof. By construction, all edges of E' that are covered in the sections traversed of t_{RPP} are also covered by the set of tours $k - \mathrm{tours}(H)$. It remains to show that all edges of E' in t_{RPP} are traversed within the distances prescribed.

The total length of the segments along t_{RPP} covered by the first $k-1$ tours is at least

$$\begin{aligned}\ell^{\max} + L + (k-2)L &= \ell^{\max} + (k-1)L \\ &= \ell^{\max} + (k-1)\frac{T^H - 2\ell^{\max}}{k} \\ &= \frac{(k-1)T^H - (k-2)\ell^{\max}}{k}.\end{aligned}$$

Therefore, the total length of the remaining uncovered edges along the input tour for the last vehicle k to cover is at most,

$$\begin{aligned}T^H - (\ell^{\max} + (k-1)L) &= T^H - \frac{(k-1)T^H - (k-2)\ell^{\max}}{k} \\ &= \frac{T^H + (k-2)\ell^{\max}}{k} = L'.\end{aligned}$$

Since L' is the maximum distance permitted for Seg_k, this completes the proof. ∎

Lemma 3.2. *The length of each tour in the set $k - tours(H)$ is at most $\frac{T^H + 2(k-1)\ell^{\max}}{k}$.*

Proof. The first tour and the kth tour are both of length at most $L' + \ell^{\max}$ since they include a segment of length L' and a return/forward, to/from, the depot, of length $\ell^{\max}$. Tours 2 through $k-1$ are of length at most L for the segment, plus $2\ell^{\max}$ for the return and forward to and from the depot. Therefore, the total length traversed by each vehicle is at most,

$$L' + \ell^{\max} = L + 2\ell^{\max} = \frac{T^H + 2(k-1)\ell^{\max}}{k}.$$ ∎

Theorem 3.2. *The k tours delivered, $k - tours(H)$, each have capacity at most $(2.5 - \frac{1}{k})\ell^*$.*

Proof.

$$\begin{aligned}
\frac{T^H + 2(k-1)\ell^{\max}}{k} &\le \frac{1.5d(RPP^*) + 2(k-1)\ell^{\max}}{k} \\
\text{(By Fact 3.2)} &\le \frac{1.5k\ell^* + 2(k-1)\ell^{\max}}{k} \\
\text{(By Fact 3.1)} &\le \frac{1.5k\ell^* + (k-1)\ell^*}{k} \\
&= \left(2.5 - \frac{1}{k}\right)\ell^*. \qquad (2)
\end{aligned}$$

■

3.3. An Alternative Heuristic Procedure for k-RPP(ℓ)

For the sake of comparison with the k-RPP 2.5-approximation algorithm, we devised a simple heuristic based on the nearest neighbor greedy algorithm. From each position, we seek the next edge in E' not covered as of yet, that is nearest. The tour proceeds to the next such edge along the shortest path in the graph.

4. THE STACKELBERG NETWORK PATROLLING PROBLEM

In this section, we specialize the Stackelberg security game model to the problem of patrolling a network where a strategic adversary is attacking edges. We then describe how this problem is decomposed separating it into a security game problem with independent targets that is used for Phase I.

In general terms, we consider a security game where the defender patrols a network $G = (V, E)$ by using k vehicles to travel routes on the network and conduct surveillance activities on some of the edges traversed. The set of possible actions for the defender X then corresponds to groups of k feasible routes with surveillance activities on some subset of the arcs. A route is feasible if a vehicle can traverse the edges in the route and perform the surveillance activities in the time available for that shift. The adversary then selects one edge to attack. Making the set of possible actions of the attacker the set of edges E. We could add a dummy target to the set of actions of the attacker to represent that the attacker decides not to attack.

We assume that if an attacker targets an edge where the defender is conducting surveillance (that is, the defender performs surveillance on that edge) then the attacker is caught and the attack prevented. Conversely, if there is no surveillance on the edge that is attacked then we consider the attack successful. Furthermore, the payoffs to the defender and attacker only depend on whether the attack was successful or not.

If an attack on edge $j \in E$ is successful then the attacker receives a reward R_j^a and the defender a penalty P_j^d. If the attack is prevented by the defender, then the attacker receives a penalty P_j^a and the defender a reward R_j^d. Note that $R_j^a > P_j^a$ and $R_j^d > P_j^d$ for any edge $j \in E$. Given a defender action $i \in X$ (a group of k feasible routes) and an edge $j \in E$, we write $j \in i$ to represent that edge j was checked by one of the routes of defender action i. Then we have that the rewards of the Stackelberg game satisfy

$$R_{ij} = \begin{cases} R_j^d & j \in i \\ P_j^d & j \notin i \end{cases} \qquad C_{ij} = \begin{cases} P_j^a & j \in i \\ R_j^a & j \notin i. \end{cases}$$

Therefore, the network patrolling problem is given by the optimization problem in (1) with the rewards above and action sets given by $Q = E$ and X the set of feasible routes. This mixed integer program can be solved by solving $|E|$ linear optimization problems. This is because every feasible integer solution is a pure strategy of the follower, which implies that there are at most $|E|$ possible feasible integer solutions. To solve (1), we only need to fix the integer variable $\mathbf{q}$ to each of these possible pure strategies and solve the remaining linear optimization problem in variables $\mathbf{x}, d, a$. The linear problem that achieves the maximum reward gives the optimal solution and the pure strategy that defines this problem, the optimal attacker strategy. Unfortunately the number of variables of the linear programming problem that has to be solved can be very large, since the size of X can be exponential in the number of nodes or edges of the network.

A standard method of addressing the large number of variables of these LPs is to implement a column generation method, where we begin with a small set of defender actions and gradually incorporate variables that improve the objective. A drawback of column generation methods is that it may be necessary to generate a large number of variables, making the method slow. In this article, we follow a different decomposition method that separates the complicating constraints that define a feasible route from the decision to cover or not different targets. We explain this decomposition in the next subsection.

4.1. Phase I

We begin by showing that the defender's and attacker's rewards can be expressed in terms of the frequency with which each target is covered instead of the probability of conducting each action. Given defender-attacker strategies $\mathbf{x}$ and $\mathbf{q}$ we have

$$\begin{aligned}
\sum_{i \in X, j \in Q} R_{ij} x_i q_j &= \sum_{j \in E} q_j \left(\sum_{i: j \in i} R_j^d x_i + \sum_{i: j \notin i} P_j^d x_i \right) \\
&= \sum_{j \in E} q_j \left(R_j^d \sum_{i: j \in i} x_i + P_j^d (1 - \sum_{i: j \in i} x_i) \right) \\
&= \sum_{j \in E} q_j (R_j^d y_j + P_j^d (1 - y_j)). \qquad (3)
\end{aligned}$$

where the second equality comes from $1 = \sum_{i \in X} x_i = \sum_{i: j \in i} x_i + \sum_{i: j \notin i} x_i$ and $y_j(\mathbf{x}) = \sum_{i: j \in i} x_i$ (or simply y_j) denotes the frequency of coverage of target/edge j given the defender strategy $\mathbf{x}$. Note that we can compute $\mathbf{y} = (y_j)_{j\in E}$ from any defender mixed strategy $\mathbf{x}$, however, the converse is not necessarily true if there are constraints on what are feasible defense actions $i \in X$.

Similarly, we can express concisely the attacker's reward by

$$\sum_{i\in X, j\in Q} C_{ij}x_i q_j = \sum_{j\in E} q_j(R_j^a(1-y_j) + P_j^a y_j). \tag{4}$$

Equations (3) and (4) express the rewards of the defender and attacker in terms of the vector of coverage over edges $\mathbf{y}$. These expressions remove the dependency of the attacker and defender rewards on the set X, and suggest a formulation with a much smaller set of variables. For this suppose that every feasible route can visit c edges, which implies $\sum_{j\in E} y_j = kc$ and gives the following problem:

$$\begin{aligned}
\max_{\mathbf{y},\mathbf{q},d,a}\ & d \\
\text{s.t. } & R_j^d y_j + P_j^d(1-y_j) \geq d - M(1-q_j) \quad j \in E \\
& R_j^a(1-y_j) + P_j^a y_j \leq a \quad j \in E \\
& R_j^a(1-y_j) + P_j^a y_j \geq a - M(1-q_j) \quad j \in E \\
& e^T\mathbf{y} = kc,\ 1 \geq \mathbf{y} \geq 0 \\
& e^T\mathbf{q} = 1,\ \mathbf{q} \in \{0,1\}^{|E|}
\end{aligned} \tag{5}$$

The decomposition of problem (1) introduced here removes the constraints $y_j = \sum_{i:j\in i} x_i$ and thus solves problem (5) on Phase I, which relaxes the representation of feasible routes. Problem (5) is equivalent to (1) if a solution on the set of actions X can be recovered from the optimal solution $\mathbf{y}$ found. That is, Problem (5) with additional variables $\mathbf{x}$ that satisfy the constraints $\mathbf{x} \in X$, and $y_j = \sum_{i:j\in i} x_i$ for all $j \in E$ is equivalent to (1). It is possible to find such feasible routes $\mathbf{x}$ from the solution to (5), for example, when $k=1$ and every feasible route can patrol only one edge $c=1$. In general, however, it might not be possible to recover a feasible route from (5) due to the geographic restrictions and capacity constraints in X.

We note that the Stackelberg Network Patrolling Problem could similarly be defined when the adversary targets nodes instead of edges. As long as the defender actions patrol edges, then we can decompose using a k-RPP in Phase II. Here, we assume that a defender action i protects against an attack on node j if action i patrols an edge incident on j. Therefore, the defender reward when the adversary attacks j can be expressed in terms of y_e the frequency of coverage of edge e as: $R_j^d \sum_{e\in\partial(\{j\})} y_e + P_j^d(1 - \sum_{e\in\partial(\{j\})} y_e)$. A similar expression can be written for the adversary reward. Using these expresions we can formulate a problem to determine the optimal frequency of coverage as:

$$\begin{aligned}
\max_{\mathbf{y},\mathbf{q},d,a}\ & d \\
\text{s.t. } & R_j^d \sum\nolimits_{e\in\partial(\{j\})} y_e + P_j^d\left(1 - \sum\nolimits_{e\in\partial(\{j\})} y_e\right) \\
& \qquad \geq d - M(1-q_j) \quad j \in V \\
& R_j^a\left(1 - \sum\nolimits_{e\in\partial(\{j\})} y_e\right) + P_j^a \sum\nolimits_{e\in\partial(\{j\})} y_e \\
& \qquad \leq a \quad j \in V \\
& R_j^a\left(1 - \sum\nolimits_{e\in\partial(\{j\})} y_e\right) + P_j^a \sum\nolimits_{e\in\partial(\{j\})} y_e \\
& \qquad \geq a - M(1-q_j) \quad j \in V \\
& e^T\mathbf{y} = kc,\ 1 \geq \mathbf{y} \geq 0 \\
& e^T\mathbf{q} = 1,\ \mathbf{q} \in \{0,1\}^{|V|}.
\end{aligned}$$

Similar to (5) this problem is equivalent to (1) if we can recover the solution $x \in X$ from $\mathbf{y}$. This can be done, for example, when both k and c equal 1 but it is not true in general.

4.2. Phase II

Phase II builds feasible routes in the set X from the frequency of coverage $\mathbf{y}$ found in Phase I. The procedure is to sample without replacement edges according to a probability distribution proportional to $\mathbf{y}$ until the set of edges sampled cannot be serviced with the available capacity. The approximation algorithm for k-RPP(ℓ) introduced above is used to find a feasible set of routes with the available capacity.

5. COMPUTATIONAL RESULTS

In this section, we compare the performance of the proposed strategy against simple alternate solution strategies. For this comparison, we build a realistic security instance that aims to deter and prevent theft in downtown Santiago. This domain is analogous to the SNM detection problem described above and the data used to construct the instances is from police crime statistics in downtown Santiago. The downtown Santiago network is modeled as an undirected geometric planar graph, where each edge represents a street and each node represents an intersection of some streets. Each node has its X-Y coordinate and the length of an edge is the Euclidean distance between two end nodes of the edge. The graph consists of 119 nodes and 203 edges, as shown in Figure 2.

The attacker in this security game is a thief and we assume that he (she) conducts theft only on nodes. The defender is the police and we assume that defenders patrol and protect edges (e.g., police vehicles travel streets monitoring activity). If an edge is protected by police then the two end nodes of the edge are considered protected. Then, the attack on either node is deterred and prevented. The crime statistics have the number and total amount (in dollars) of thefts that have occurred at every node of the graph over 3 years. Thus, we compute the average amount of a theft (in dollars) on every node. We assume that the reward of the attacker for conducting an attack at an unprotected node is the average amount of theft on that node, while the penalty for the attacker when targeting a protected node is the negation of the total amount of theft on that node. Symmetrically, we assume that

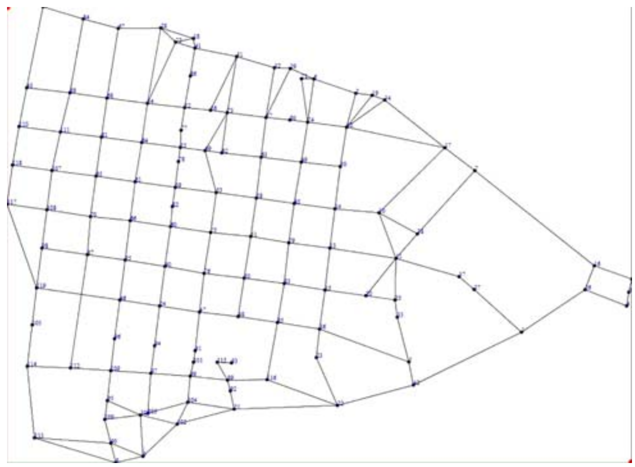


FIG. 2. The downtown network of Santiago, Chile. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the reward for the defender of protecting a node from attack is the total amount of theft on that node, while the penalty for the defender of not protecting a node from attack is the negation of the average amount of theft on that node. Notice that since the defender protects edges, the game is not a zero sum game as the defenders receive the rewards of both end nodes when protecting a certain edge.

We first compute the defender's mixed strategy on this problem instance, according to Phase I of our proposed solution approach. This determines the frequency of coverage on each edge as described in Section 4.1 (we refer to this solution as the Stackelberg probability). Solving this problems also determines the set of nodes that give the maximum reward to the attacker. In our experiment, we run Phase I with $kc = 15$, which generates a defender strategy that sets a large number of nodes (110 out of the total of 119 nodes) with the maximum reward for the attacker.

In Phase II, we implement the 2.5-approximate k-RPP algorithm described in Section 3.2.1. In this phase, k represents the number of available police patrols (vehicles) and the capacity of each patrol is the maximum distance that the vehicle can patrol in 1 day, starting from and returning to the depot (police department). We simulate a time range of 100 days in the experiment. Each day, the attacker uniformly samples a node to attack from the set of nodes with maximum reward. The defender, in turn, selects edges to patrol by sampling without replacement according to the Stackelberg probability. This sampling process is continued while the edges selected can be patrolled with k patrols of limited capacity, which is checked using the k-RPP approximation algorithm. On each day, we consider that the patrol successful if the node attacked is incident to one edge protected by the patrols, otherwise we assume the attack was successful. The utility perceived by the defender (and attacker) depends on whether the patrol was successful or not: with the defender accruing either the reward or the penalty for the attacked node depending on whether it was covered or not. The performance of the complete two-phase strategy is measured by the total net utility (rewards plus penalties) of the defender accumulated over the 100 days.

For each phase, we consider a counterpart for comparison. For Phase I, we consider another mixed strategy where the protection probability is uniform over all edges. With uniform probability, the node that gives the maximum reward to the attacker is unique. For Phase II, we consider another simple nearest-neighbor routing heuristic as is outlined in Section 3.3: for each vehicle, starting from the depot, it always covers the nearest edge that needs protection (in E'). Then starting from this edge, it covers next the nearest edge that needs protection. The vehicle continues this searching procedure until the remaining capacity forces it to go back to the depot.

As a summary, by doing cross combinations, we compare the following four two-phase strategies:

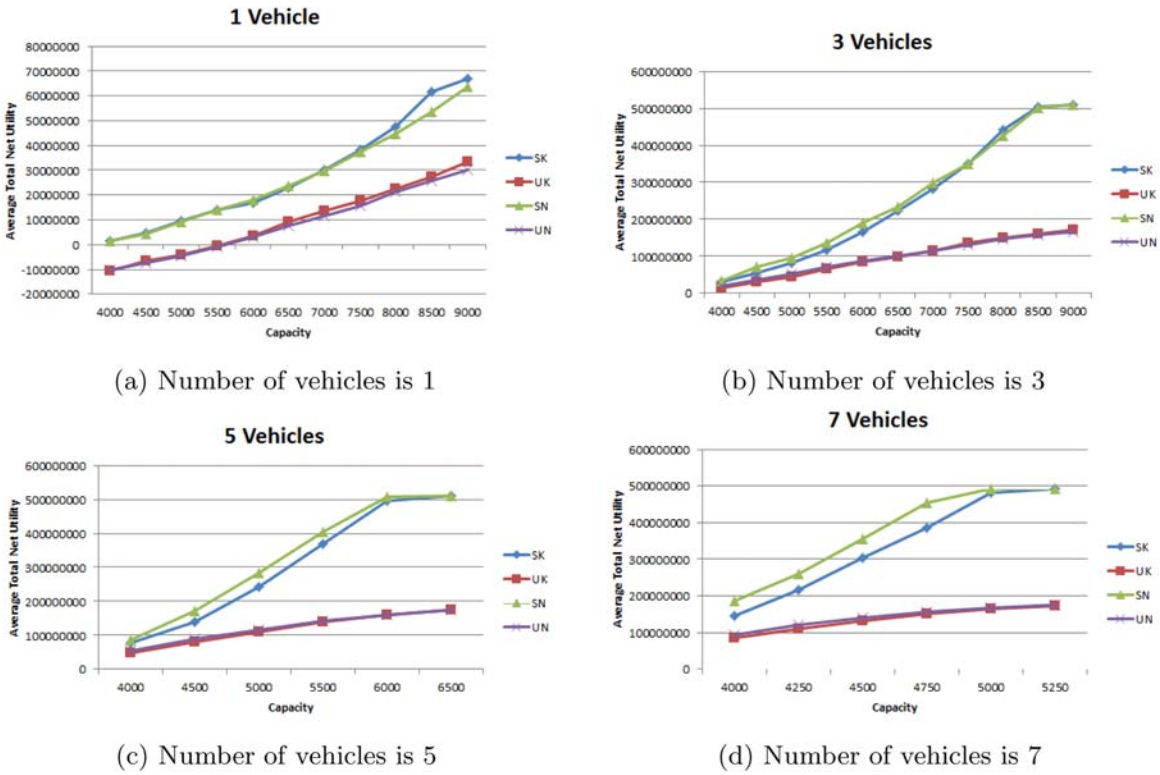
- Combination SK—Phase I: Stackelberg probability. Phase II: k-RPP algorithm.
- Combination UK—Phase I: Uniform probability. Phase II: k-RPP algorithm.
- Combination SN—Phase I: Stackelberg probability. Phase II: Nearest-neighbor algorithm.
- Combination UN—Phase I: Uniform probability. Phase II: Nearest-neighbor algorithm.

For each two-phase strategy, we study the change of the total net utilities as a function of k, the number of available patrols, and as a function of the patrols' capacities. To differentiate these two effects, we explore two setups:

- For a fixed number of patrolling vehicles, we study the change of the total net utilities as a function of the vehicles' capacities.
- For a fixed level of vehicles' capacities, we study the change of the total net utilities as a function of the number of available patrolling vehicles.

The results for the above two cases appear in Figure 3 when the number of patrolling vehicles is kept constant and in Figure 4 for constant capacity. For every combination of the k and the capacity, we run all four two-phase strategies 10 times and report the average total net utility. The performance of the four two-phase strategies are plotted in the same figure for each testing scenario.

The above computational results show that the Stackelberg probability gives better performance than the uniform probability. This follows because the Stackelberg probability is the optimal solution of model (1) in Phase I, which computes the maximum net utility. This is consistent with the previous computational results showing that the Stackelberg probability outperforms the uniform probability (without considering the effect of geographical locations on the defender strategy) [16, 27]. Conversely, the "worst case" 2.5-approximate k-RPP algorithm works well for the case of 1 vehicle ($k = 1$). For the cases of more vehicles, since the problem is defined in a Euclidean plane, a simple heuristic like the nearest-neighbor algorithm would do well, while the "worst case" 2.5-approximate k-RPP algorithm does not improve the solution.



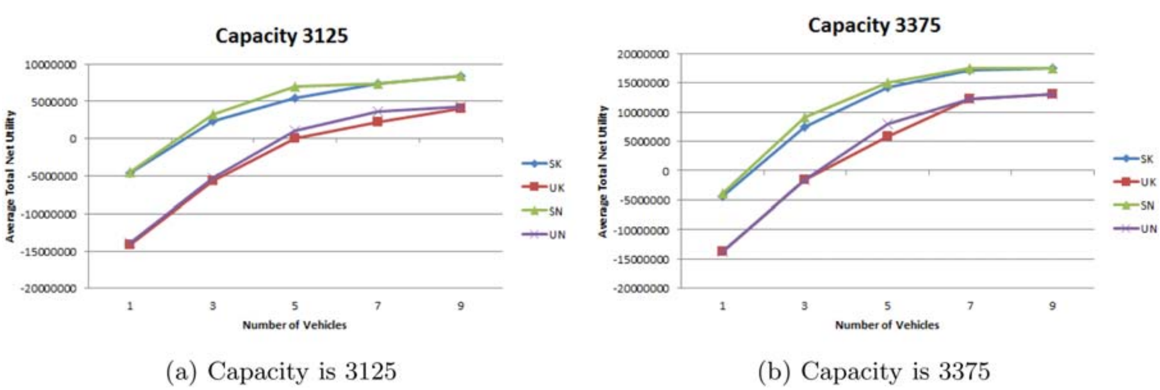
(a) Number of vehicles is 1

(b) Number of vehicles is 3

(c) Number of vehicles is 5

(d) Number of vehicles is 7

FIG. 3. The change of the average total net utility with increasing capacities for a fixed number of vehicles. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]



(a) Capacity is 3125

(b) Capacity is 3375

FIG. 4. The change of the average total net utility with increasing number of vehicles and fixed vehicle capacity. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

6. CONCLUSIONS

We study the security detection problem that addresses the effect of geographical locations on the defender strategy in the context of adversarial games. We propose a novel decomposition of the problem into two phases. In Phase I, we determine the defender's optimal strategy as if the targets are independent. We model the subproblem as a Stackelberg security game. In Phase II, we take into consideration the effect of geographical locations. Given the limited resources, to meet the optimal outcome of the adversarial game from Phase I, we determine a maximum set of targets within a certain area that can be patrolled. This subproblem is the NP-hard k-vehicle RPP. We devise a $(2.5 - \frac{1}{k})$-approximation algorithm for this problem that works in any setting (including the nonmetric setting). We conduct computational experiments to compare the performance of the proposed two-phase strategy with other simple heuristics.

The instance used in the computational study is not a good instance to show the advantage of the 2.5-approximate k-RPP algorithm over simple heuristics. It would be interesting to find a test instance, where the 2.5-approximate k-RPP algorithm is better than simple heuristics. Another question that is worth of further study is the effect of separating the Stackelberg Security Game into these two phases on the optimality of the solution found.

REFERENCES

[1] S. Alpern, A. Morton, and K. Papadaki, Patrolling games, Oper Res 59 (2011), 1246–1257.

[2] E.M. Arkin, R. Hassin, and A. Levin, Approximations for minimum and min-max vehicle routing problems, J Algorithms 59 (2006), 1–18.

[3] E. Benavent, A. Corberán, I. Plana, and J.M. Sanchis, Min-Max K-vehicles windy rural postman problem, Networks 54 (2009), 216–226.

[4] G. Brown, M. Carlyle, J. Salmerón, and K. Wood, Defending critical infrastructure, Interfaces 36 (2006), 530–544.

[5] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem, Technical Report, Carnegie Mellon University, 1976.

[6] J. Edmonds and E.L. Johnson, Matching, euler tours and the chinese postman, Math Program 5 (1973), 88–124.

[7] G.N. Frederickson, Approximation algorithms for some postman problems, J ACM 26 (1979), 538–554.

[8] G.N. Frederickson, M.S. Hecht, and C.E. Kim, Approximation algorithms for some routing problems, 17th Annu Symp Found Comput Sci, Houston, TX, 1976, pp. 216–227.

[9] G.N. Frederickson, M.S. Hecht, and C.E. Kim, Approximation algorithms for some routing problems, SIAM J Comput 7 (1978), 178–193.

[10] B. Fishbain, D.S. Hochbaum, and Y. Yang, Video segmentation and target tracking using Pseudoflow algorithm through joint utilization of intensity and coarse motion data, UC Berkeley manuscript, 2010.

[11] M. Guan, Graphic programming using odd and even points, Chin Math 1 (1962), 67.

[12] D.S. Hochbaum and B. Fishbain, Nuclear threat detection with mobile distributed sensor networks, Ann Oper Res Special Issue: Port Security/Safety, 2010. Available at: http://www.springerlink.com/content/y622t73564377685/0254-5330, pp 1–19, 2009.

[13] D.S. Hochbaum, The multi-sensor nuclear threat detection problem, Operations Research and Cyber-Infrastructure, 47 Operations Research/Computer Science Interfaces Series, Springer US, 2009, pp. 389–399.

[14] M. Jain, V. Conitzer, M. Pechoucek, and M. Tambe, Security scheduling for real-world networks, AAMAS-2013 Conf, Saint Paul, MN, 2013, pp. 215–222.

[15] M. Jain, E. Kardes, C. Kiekintveld, M. Tambe, and F. Ordóñez, Security games with arbitrary schedules: A branch and price approach, AAAI-2010 Conf, Atlanta, GA, 2010, pp. 792–797.

[16] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, F. Ordóñez, and M. Tambe, Software assistants for randomized patrol planning for the LAX airport police and the Federal Air Marshals service, Interfaces 40 (2010), 267–290.

[17] M. Jain, D. Korzhyk, O. Vanek, V. Conitzer, M. Pechoucek, and M. Tambe, A double oracle algorithm for zero-sum security games on graphs, AAMAS-2011 Conf, Taipei, Taiwan, 2011, pp. 327–334.

[18] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez, Computing optimal randomized resource allocations for massive security games, AMAAS-2009 Conf, Budapest, Hungary, 2009, pp. 689–696.

[19] C.S. Orloff, A fundamental problem in vehicle routing, Networks 4 (1974), 35–64.

[20] C.H. Papadimitriou, On the complexity of edge traversing, J ACM (JACM) 23 (1976), 544–554.

[21] P. Paruchuri, J. Pearce, M. Tambe, F. Ordóñez, and S. Kraus, An efficient heuristic approach for security against multiple adversaries, AMAAS-2007 Conf, Honolulu, HI, 2007, pp. 299–306.

[22] W.L. Pearn, Solvable cases of the k-person chinese postman problem, Oper Res Lett 16 (1994), 241–244.

[23] P. Paruchuri, J. Pearce, J. Marecki, M. Tambe, F. Ordóñez, and S. Kraus, Playing games for security: An efficient exact algorithm for solving bayesian games, AMAAS-2008 Conf, Estoril, Portugal, 2008, pp. 895–902.

[24] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer, PROTECT: An application of computational game theory for the security of the ports of the United States, AAAI-Conf 2012, Spotlight Track, Toronto, ON, 2012, pp. 2173–2179.

[25] L.M. Wein and M.P. Atkinson, The last line of defense: Designing radiation detection-interdiction systems to protect cities from a nuclear terrorist attack, IEEE Trans Nucl Sci 54 (2007), 654–669.

[26] Z. Yin, M. Jain, M. Tambe, and F. Ordóñez, Risk-averse strategies for security games with execution and observational uncertainty, Conf Artif Intell (AAAI), San Francisco, CA, 2011, pp. 758–763.

[27] R. Yang, C. Kiekintvled, F. Ordóñez, M. Tambe, and R. John, Improving resource allocation strategies against human adversaries in security games: An extended study, Artif Intell J (AIJ) 195 (2013), 440–469.