AAAI Technical Report SS-12-03
Game Theory for Security, Sustainability and Health

Extending Security Games to Defenders with Constrained Mobility

Ondřej Vaněk, Branislav Bošanský, Michal Jakob, Viliam Lisý and Michal Pěchouček
Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University
Technická 2, 16627 Praha 6, Czech Republic
{vanek, bosansky, jakob, lisy, pechoucek}@agents.fel.cvut.cz

Abstract

A number of real-world security scenarios can be cast as a problem of transiting an area guarded by a mobile patroller, where the transiting agent aims to choose its route so as to minimize the probability of encountering the patrolling agent, and vice versa. We model this problem as a two-player zero-sum game on a graph, termed the *transit game*. In contrast to the existing models of area transit, where one of the players is stationary, we assume both players are mobile. We also explicitly model the limited endurance of the patroller and the notion of a base to which the patroller has to repeatedly return. Noting the prohibitive size of the strategy spaces of both players, we develop single- and double-oracle based algorithms including a novel acceleration scheme, to obtain optimum route selection strategies for both players. We evaluate the developed approach on a range of transit game instances inspired by real-world security problems in the urban and naval security domains.

Introduction

Hostile area transit and patrolling is an important problem relevant to many real-world security scenarios, such as illegal border crossing, smuggling interdiction or transport logistics in insecure regions (Gilpin 2009). For the transiting agent, the problem is to choose such a route to get across the hostile transit area that minimizes the risk that it will be encountered and intercepted by the patrolling agent that moves within the area; the objective of the patrolling agent is the opposite.

Choosing the optimum routes becomes non-trivial if we assume that the agents are aware of and capable of reasoning about each other's objectives and/or whenever the situation happens repeatedly and the agents are able to learn from experience. In both such cases, predictability is disadvantageous as it can be exploited by the opponent agent. Randomized route selection can be used in these cases to make exploitation more difficult by increasing the uncertainty in opponent agent's behavior. Game theory provides a principled way of achieving optimum randomization, taking into account information, preferences and constraints of both agents.

In this paper, we model the area transit and patrolling problem as a zero-sum game, termed *transit game*, between two players – the Evader and the Patroller. Pure strategies of both players correspond to routes in/through the transit area and the utilities are directly related to the probability that the Patroller will intercept the Evader.

Game-theoretic approach has been successfully applied to similar problems in the past (Jain et al. 2010b), resulting in a variety of games reflecting specific assumptions, domain restrictions and player capabilities. In contrast to existing models of area transit, where one of the players is always stationary, our transit game assumes *both* players are mobile; (Vaněk et al. 2010) is the sole exception but see Related work section for differentiation. Unlike existing models, we also allow to consider Patroller's home base and maximum endurance, both features frequently present and required in real-world scenarios. Our objective is to find such a strategy for the players that maximizes the minimum expected utility the players can obtain, which — thanks to the zero-sum property of the transit game — corresponds to a Nash equilibrium of the game.

Unfortunately, the mobility of both players triggers combinatorial explosion in the number of possible strategies and makes the application of standard methods for finding Nash equilibria ineffective. For example, in a rectangular grid graph with 15 nodes, the number of possible strategies exceeds 10^{10}, rendering standard approaches for computing Nash equilibria in normal-form games inapplicable. We therefore employ iterative solution techniques known as *oracle-based algorithms* (Barnhart et al. 1994; McMahan, Gordon, and Blum 2003) which do not require explicit enumeration of strategies for the players. Unfortunately, although the oracle-based approach alleviates the problem to some extent, it requires repeated best response calculation, which is hard in our case. We therefore propose a novel variant of the oracle algorithms, termed *accelerated oracle*, which reduces the need for best response calculation, and thus speeds up the calculation of Nash equilibria.

We evaluate our approach on two classes of transit games, directly inspired by real-world security applications — regular grid graphs suitable for modeling transit through open areas and irregular planar graphs suitable for modeling transit through structured environments. For the former — by employing agent-based simulation of maritime piracy

AgentC (Jakob, Vaněk, and Pěchouček 2011) — we conduct experiments that transcend the game-theoretic framework and allows us to validate the transit game model from a broader perspective and compare its effectiveness to currently deployed solutions in the maritime domain.

Related Work

Game-theoretic framework has been applied to a wide range of strategic problems where one of the players — termed *evader* — wants to minimize the probability of being detected and/or intercepted, while the other player — termed *patroller* — wants to maximize the probability of detecting the first player and/or thwarting its plans.

We can distinguish several classes of such games: *ambush games* (Ruckle et al. 1976) and *interdiction games* (Washburn and Wood 1995) model an immobile intercepting player selecting static ambush points in an area represented by a graph that the evading player tries to transit, recently extended e.g. in (Dickerson et al. 2010; Jain et al. 2011). The mobility of the players is reversed in *search games* (Gal 1980) where the evader is stationary and the searching player is mobile. In hider-seeker games (Flood 1972), both players are mobile; however the players have no explicit restrictions on their trajectories and do not have additional goals.

Close to our model are *infiltration games* (Alpern 1992) in which both the players are mobile and the evader tries to cross a given area between defined entry and exit, however the patrolling player can move freely and does not have a base to return to. We further enrich this model by associating interception probability with each node and edge in the graph (similar concept was used e.g. in (Brooks, Schwier, and Griffin 2009)).

The techniques used to find solution of the transit game are inspired by single- and double-oracle algorithms; the former usually termed *column/constraint generation* or *branch and price* (Barnhart et al. 1994), used among others for solving large-scale games (Jain et al. 2010a), extended to double-oracle approach (McMahan, Gordon, and Blum 2003), used e.g. in (Halvorson, Conitzer, and Parr 2009).

The model of the transit game was first introduced in (Vaněk et al. 2010). We extend the model by (1) allowing the Evader to move freely on an arbitrary graph and by (2) allowing the positioning of the Patroller's base anywhere in the environment. Moreover, we redefine the utility of the game so that the relative direction of player's movement is taken into account, which results into a well-defined, probabilistic interpretation of the game value.

Problem Definition

We formalize the problem of hostile area transit and patrolling as follows: let us have a connected *transit area* with defined *entry* and *exit* zones and a *base* location. There are two players that move in the area: the Evader and the Patroller. The Evader's objective is to get from any location in the entry zones to any location in the exit zones *without* encountering the Patroller. The Patroller's objective is to intercept the Evader's transit by strategically moving through the transit area. In addition, because of its limited endurance, the Patroller has to repeatedly return to the base.

Modeling Assumptions

We make the following assumptions regarding the area transit problem: (1) both players have full knowledge about the topology of the transit area, including the location of Patroller's base and the location of entry and exit zones. (2) The Evader knows the Patroller's capability to detect and intercept the Evader at any position in the transit area (this capability can vary due to environmental reasons). (3) The Evader knows Patroller's maximum endurance (and consequently the limits on the maximum length of Patroller's walks). (4) The players have no information about the location of the other player, unless they meet (at which point the game ends). (5) The Patroller has no information of whether the Evader has already entered the area. The Evader does not know when the Patroller visits the base.

Transit Area Representation

We use discrete representation of space. We represent the transit area as a simple directed graph with loops termed *transit graph* $G(N, E)$, $N = \{1, 2, \ldots, n\}$ denotes a set of nodes represented directly by natural numbers, and $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)\}$ where $i_k \in N \wedge j_k \in N$ is the set of edges defining legal movement of players through the transit area. Every edge has a unit length, i.e., each step (see below) the player can move from one node to any adjacent node. Three special types of nodes are defined on the transit graph:(1) *entry nodes* N_{in} – the Evader can start its path in any node of this type; (2) *exit nodes* N_{out} – the Evader aims to reach any node of this type; (3) *base node* n_b – the node in which all Patroller's walks start and end.

Player Movement

Analogously to space, time is also discretized. The movement of both players happens simultaneously in synchronized *steps*. During each step, a player can move to an adjacent node or stay in the same node. Both players have the same movement speed and all edges take a single step to traverse. Player's movement through the transit graph can be unambiguously represented by a node *walk*, i.e., a sequence of nodes $w = [n_0, n_1, \ldots, n_k]$[1]; we then denote $|w|$ the length of walk w, and $w[j]$ the j-th node on the walk (for $0 \leq j \leq |w| - 1$).

The following will be required for the definition of utilities. For any finite walk w, we define *infinite walk repetition* $w^{\infty}[i] = w[i \bmod |w|]$ and *shifted infinite walk repetition* $w^{\infty \triangleright m} = w[(i - m) \bmod |w|]$. E.g. for $w = [1, 4, 7]$, we have

index	…	$[-2]$	$[-1]$	[0]	[1]	[2]	[3]	[4]	…
w	…	-	-	1	4	7	-	-	…
w^{∞}	…	4	7	1	4	7	1	4	…
$w^{\infty \triangleright 1}$	…	1	4	7	1	4	7	1	…

[1] In a slight abuse of common mathematical notation and to emphasize similarity with the array data structure, we use brackets to denote sequences and to index sequence items.

Encounters We say two walks w_1 and w_2 have:

- a *node encounter* at node $i \in N$ at step t if $i = w_1[t] = w_2[t]$ (being at the same node at the same time step);
- an *edge encounter* at edge $(i, j) \in E$ at step t if $i = w_1[t] = w_2[t] \wedge j = w_1[t+1] = w_2[t+1]$ (traveling the same edge simultaneously in the same direction) or $i = w_1[t] = w_2[t+1] \wedge j = w_1[t+1] = w_2[t]$ (traveling the same edge simultaneously in the opposite directions)
- an *encounter* at *location* $l \in N \cup E$ at step t if w_1 and w_2 either have a node encounter or an edge encounter at l at step t.

The *encounter sequence* of two walks w_1 and w_2 is a sequence $[(l_0, t_0), (l_1, t_1), \ldots, (l_n, t_n)]$ where $\forall i \in \{0 \ldots n\}$ w_1 and w_2 have an encounter at l_i at step t_i and $t_i \leq t_{i+1}$, i.e. the order in which the encounter locations appear on walks w_1 and w_2 is preserved; the *encounter location sequence*, denoted as $w_1 \cap w_2$, is then the encounter sequence without timestep indices but with the ordering preserved – $[l_0, l_1, \ldots, l_n]$.

As an example, let us consider $w_1 = [1, 4, 7, 6, 9, 2, 5, 3]$ and $w_2 = [1, 4, 8, 6, 2, 9]$.

index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
w_1	1	4	7	6	9	2	5	3
w_2	1	4	8	6	2	9	-	-
$w_1 \cap w_2$	1	(1,4)	4		6	(2,9)		

The resulting encounter location sequence $w_1 \cap w_2 = [1, (1, 4), 4, 6, (2, 9)]$.

Interceptions To provide more expressiveness to the transit game model, encounters are assumed to lead to interceptions only with a defined location-specific *interception probability* $p(l)$ $\forall l \in N \cap E$. Interception probability may e.g. reflect the likelihood that the Patroller will be able to detect the Evader and/or will be able to physically apprehend the Evader in a given location. The concept is similar to the concept of sensors with probability of detection in (Brooks, Schwier, and Griffin 2009).

Transit Game

The modeling assumptions allow us to model the transit problem as a normal-form game between two players – the Evader and the Patroller. In the following, we assume the transit game takes place on a transit graph $G = (N, E)$ with entry nodes N_{in}, exit nodes N_{out} and base n_b.

Strategy Spaces The set S_E of all possible pure Evader's strategies is the set of all walks starting in an entry node and ending in an exit node, with the nodes in between not being an entry or exit node, i.e.,

$$S_E = \{[n_0, \ldots, n_m] | n_0 \in N_{\text{in}} \wedge n_m \in N_{\text{out}} \wedge n_i \in N \setminus \{N_{\text{in}} \cup N_{\text{out}}\} \forall i = \{1, \ldots, m-1\}\} \quad (1)$$

Note that because in general Evader's walks are unlimited, the above set can be infinite, however we will limit the Evader to visit every node at most once.

The set S_P of all possible pure Patroller's strategies is the set of all closed walks starting and ending in the base with length not exceeding L_P, i.e.,

$$S_P = \{[n_0, \ldots, n_m] | m \leq L_P \wedge n_i \in N \wedge n_0 = n_b \wedge (n_m, n_b) \in E\} \quad (2)$$

Note that if L_P is so small that it does not allow the Patroller to cross one of the Evader's walks, then the Evader will have a deterministic strategy that guarantees safe transit of the graph. The threshold for L_P under which this is the case depends on the position of Patroller's base and topology of the transit graph. In contrast to the Evader, which performs its walk only once, the Patroller executes its walk repeatedly[2].

We further denote Σ_E and Σ_P the set of all mixed strategies of the Evader and the Patroller, respectively.

Utility Functions The utility function in the game is directly related to the probability of interception. First, given an encounter location sequence I, we can express the probability $\pi(I)$ that the Evader will be intercepted by the Patroller as

$$\pi(I) = \sum_{i=0}^{|I|-1} p(I[i]) \prod_{j=0}^{i-1} (1 - p(I[j])) \quad (3)$$

where

$$p(I[i]) \prod_{j=0}^{i-1} (1 - p(I[j])) \quad (4)$$

is the probability that the Patroller will not intercept the Evader at locations $I[0], \ldots, I[i-1]$ and will intercept it at location $I[i]$.

To calculate the interception probability $\pi(s_E, s_P)$ of a pair of pure strategies $(s_E, s_P) \in S_E \times S_P$, we need to determine all possible encounter location sequences that can result from executing these strategies. Recalling that the Patroller has no knowledge on when the Evader enters the transit area, we have to consider all possible mutual shifts of Evader's and Patroller's walks; however, because Patroller's walk s_P is perpetually repeated, we only need to consider $|s_P|$ shift. The interception probability can therefore be calculated as

$$\pi(s_E, s_P) = \frac{1}{|s_P|} \sum_{i=0}^{|s_P|-1} \pi(I^{\triangleright i}) \quad (5)$$

where

$$I^{\triangleright i} = s_E \cap s_P^{\infty \triangleright i} \quad (6)$$

For a given pure strategy pair $(s_E, s_P) \in S_E \times S_P$, we now define the Patroller's utility $u_P(s_E, s_P)$ as equal to the interception probability, i.e.,

$$u_P(s_E, s_P) = \pi(s_E, s_P) \quad (7)$$

We define the Evader's utility as the opposite value of Patroller's utility, i.e.,

$$u_E(s_E, s_P) = -\pi(s_E, s_P) \quad (8)$$

[2]Patroller does not repeats the walk indefinitely, the game ends once the Evader reaches one of the exit nodes.

Finally, we define the *transit game with deterministic encounters* as a transit game with unit interception probabilities at all nodes and edges; i.e., $p(l) = 1 \; \forall l \in N \cup E$.

Solution

We employ mixed-strategy Nash equilibrium (NE) as a solution concept for the transit game. However, because of the enormous size of the strategy spaces of both players, standard techniques for computing a NE of normal form games, requiring the construction of the full game matrix, are not applicable. We thus employ iterative techniques known as *oracle-based algorithms* (McMahan, Gordon, and Blum 2003). In the following, we first describe the iterative-oracle-based search in general, independent of any particular normal-form game to which it is applied. The specifics of the transit game only need to be considered when formulating specific oracles, used in applying the oracle algorithms to the transit game.

Iterative Oracle-based NE Computation

Instead of searching for the NE of the full normal-form game, oracle-based algorithms iteratively construct and solve a growing succession of (significantly) smaller subgames until they reach a subgame whose NE is also a NE of the full game. Depending on the structure of player's strategy spaces, a NE of the full game maybe found (long) before the full game needs to be constructed and solved. Assuming the computation of NE is significantly faster for the much smaller subgames than for the full game, this may lead to overall significantly faster computation.

Single-Oracle Algorithm Let us consider a two-player normal-form zero-sum game Γ with pure strategy sets S_1 and S_2 for Player 1 and Player 2, respectively, and the corresponding mixed strategy sets Σ_1 and Σ_2. The single-oracle algorithm iteratively constructs a sequence of subgames $[\Gamma^{(0)}, \Gamma^{(1)}, \ldots]$ where each game $\Gamma^{(k)}$ consists of the complete pure strategy set S_1 for Player 1 but only a subset $\hat{S}_2^{(k)} \subseteq S_2$ of the full pure strategy set S_2 for Player 2. In each iteration of the single-oracle algorithm, a Nash equilibrium $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$ of the current subgame $\Gamma^{(k)}$ is first sought using standard linear program approach. Next, a Player's 2 *best-response oracle* $\omega^* : \Sigma_1 \mapsto S_2$ is consulted to obtain a pure best-response strategy $s_2 \in S_2$ of Player 2 against the Player's 1 strategy $\sigma_1 \in \Sigma_1$; if the resulting pure strategy $s_2 \in S_2$ is already in $\hat{S}_2$, the algorithm terminates and the NE (σ_1, σ_2) is the NE of the full game Γ (McMahan, Gordon, and Blum 2003); otherwise, the strategy s_2, now termed *subgame expanding strategy*, is added to $\hat{S}_2$ and the algorithm continues.

In the ideal case, the iterative oracle-based algorithm would only add such pure strategies $s \in S_2$ to the pure strategy subset $\hat{S}_2$ that are in the *support*[3] of the Player's 2 resulting mixed NE strategy of the full game Γ. Best response calculation can be viewed as a heuristic for selecting such subgame expanding pure strategies $s_2 \in S_2$ that the algorithm terminates after as few iterations as possible.

[3]The *support* of a mixed strategy σ_i is a set of pure strategies $\{s_i | \sigma_i(s_i) > 0\}$.

Algorithm 1 Accelerated Single-Oracle Algorithm. Heuristic and best-response oracle denoted as ω and ω^*, respectively.

```
equilibrium_found ← false
Ŝ_2 ← ∅
σ_1 ← uniform distribution over all s_1 ∈ S_1
repeat
    s ← ω(σ_1)
    if s ∉ Ŝ_2 then
        Ŝ_2 ← Ŝ_2 ∪ {s}
    else
        s* ← ω*(σ_1)
        if (s* ∈ Ŝ_2) then
            equilibrium_found ← true
        else
            Ŝ_2 ← Ŝ_2 ∪ {s*}
        end if
    end if
    (σ_1, σ_2) ← compute NE using LP for S_1 and Ŝ_2
until equilibrium_found
```

Accelerated Single-Oracle Algorithm The above reasoning paves the way for using alternative methods of selecting subgame expanding strategies. In principal, there can be two reasons for doing so: (1) best response calculation is too expensive to be invoked in each cycle and (2) the alternative selection method may navigate the space of subgames more effectively, resulting in a lower number of iterations. In either case, we term the oracle that returns a subgame expanding strategy the *subgame expansion oracle* (denoted as ω) and the resulting method, which uses two distinct oracles, the *accelerated oracle* algorithm.

Pseudocode of the accelerated oracle algorithm is given in Algorithm 1. Note that we use the heuristic oracle ω to obtain game expanding strategies; the termination condition still uses the best response as only this ensures that the NE obtained on the current subgame $\Gamma^{(k)}$ is an NE of the full game Γ too.

Theorem 1 *If pure strategy sets of both players are finite, the Algorithm 1 with accelerated oracle finds the Nash equilibrium of the full transit game.*

Proof 1 *At the worst case, the heuristic oracle of Player 2 adds all pure strategies to the Player's 2 strategy subset and the final subgame equals to the full game. If the algorithm terminated* before *all Player's 2 pure strategies were enumerated, then the best response for the current mixed strategy σ_1 of Player 1 had to be already in Player's 2 strategy subset $\hat{S}_2$. However, this is the termination condition of the original oracle algorithm for which (see (McMahan, Gordon, and Blum 2003)) show that it is satisfied only if the NE of the subgame corresponds to the NE of the full game. Hence, this is also true for the accelerated oracle algorithm.*

Double-Oracle Algorithm The double-oracle algorithm (Algorithm 2) uses incrementally expanded strategy subsets

Algorithm 2 Accelerated Double-Oracle Algorithm. Best-response oracles for Player 1 and Player 2 denoted as ω_1^* and ω_2^*, respectively. Heuristic oracle for Player 1 denoted as ω_1.

```
equilibrium_found ← false
Ŝ1 ← { arbitrary strategy s1 ∈ S1 }
Ŝ2 ← { arbitrary strategy s2 ∈ S2 }
repeat
    (σ1, σ2) ← compute NE using LP for Ŝ1 and Ŝ2
    s1 ← ω1(σ2)
    s2 ← ω2*(σ1)
    if (s1 ∈ Ŝ1) ∧ (s2 ∈ Ŝ2) then
        s1* ← ω1*(σ2)
        if s1* ∈ S1 then
            equilibrium_found ← true
        else
            Ŝ1 ← Ŝ1 ∪ {s1*}
        end if
    else
        Ŝ1 ← Ŝ1 ∪ {s1}
        Ŝ2 ← Ŝ2 ∪ {s2}
    end if
until equilibrium_found
```

$\hat{S}_1$ and $\hat{S}_2$ for both players. Termination condition requires that best responses for both players, computed by best response oracula ω_1^* and ω_2^*, are already present in the respective strategy subsets. Analogously to the single-oracle algorithm, distinct subgame expansion oracles can be used for subgame expansion as long as the best response oracula ω_1^* and ω_2^* are used in the termination condition. For the sake of simplicity and because of the way the double-oracle algorithm is employed for solving the transit game, Algorithm 2 describes a variant where the subgame expansion oracle is only employed for Player 1.

Transit Game Oracles

We now describe the implementation of the specific oracles used in solving the transit game. The Evader and the Patroller correspond to Player 1 and Player 2, respectively. Correspondingly, we further use E and P instead of 1 and 2 to index players. The definition of utility (see Equations 5 and 7) — taking into account the relative directions of the paths — prohibits the direct formulation of a best-response as an (mixed-integer) linear program. The oracles thus employ standard search techniques to provide the best response for each player.

Evader's Best-Response Oracle In a given transit game, the Evader's best response oracle ω_E^* provides a pure strategy $s \in S_E$ which is the Evader's pure-strategy best response to Patroller's mixed strategy $\sigma_P \in \Sigma_P$; in other words, such a walk $s_E^* \in S_E$ from any entry node to any exit node which minimizes the Patroller's expected utility

$$s_E^* = \arg\min_{s_E \in S_E} \sum_{s_P \in S_P} u(s_E, s_P) \cdot \sigma_P(s_P) \qquad (9)$$

where $\sigma_P(s_P)$ is the probability that $s_P \in S_P$ is played in σ_P.

The Evader's best-response oracle ω_E^* is implemented as a *best-first-search* algorithm, starting from all entry nodes and expanding the best walk so far. To avoid infinite-length walks, we require that the same node is not visited multiple times. Partial-walk utility for each incomplete walk is computed at each step and the best walk (i.e. the one with the highest partial-walk utility) is chosen for further expansion. If this best walk is complete (i.e. going from an entry node all the way to an exit node), the algorithm terminates and returns the walk. Because of the utility formulation (Equation 8), the Evader's utility of an Evader's walk cannot increase with the walk's expansion[4], thus the first complete walk is guaranteed to be the best response for the Evader.

Evader's Subgame Expansion Oracle Due to the combinatorial explosion in the number of possible Evader's walks, best response calculation for the Evader is expensive. For subgame expansion, we therefore employ a distinct subgame expansion oracle ω_E which determines the subgame expanding strategy by searching for the best response $s \in S_E$ only in the set of Evader's walks of a limited maximum length. The optimal maximum length depends on the structure of the transit graph; as a reasonable estimate, the length of the shortest path between the two most distant entry and exit nodes can be used.

Patroller's Best-Response Oracle Patroller's best-response oracle ω_P^* provides a pure strategy $s^* \in S_P$ which is the Patroller's pure strategy best response to an Evader's mixed strategy $\sigma_E \in \Sigma_E$., i.e., a bounded-length closed walk from Patroller's base maximizing the expected Patroller's utility

$$s_P^* = \arg\max_{s_p \in S_P} \sum_{s_E \in \mathcal{S}_E} u(s_E, s_P) \cdot \sigma_E(s_E) \qquad (10)$$

Computation-wise, the Patroller's best-response oracle ω_P^* is implemented as a branch-and-bound depth-first search.

Evaluation

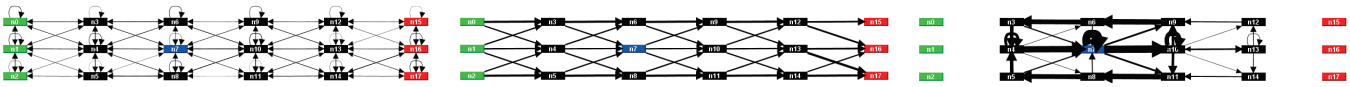
We present the evaluation of our approach on two characteristic and application-relevant classes of transit games. We begin with example solutions and then compare the properties of individual solution algorithms in terms of performance, scalability and convergence. Finally, we validate the game-theoretic strategies on an agent-based simulator of maritime piracy.

Test Problems

We study the properties of the transit game and its solution on two types of graphs, motivated by real-world domains: (1) rectangular grid graphs and (2) planar city graphs. For each graph, we examine the properties of the proposed algorithms, both for a transit game with deterministic and non-deterministic encounters.

A grid graph with cycles and diagonal edges (depicted on the Figure 1a) is a suitable representation of a open transit

[4]Note that the complete-walk utility will be always lower than or equal to the partial-walk utility so we can use the partial-walk utility for the *best-first* estimate.



(a) Regular 6x3 grid graph. (b) Evader's optimal strategy. (c) Patroller's optimal strategy.

Figure 1: Exemplar transit game on a rectangular grid graph with deterministic encounters; three entry nodes placed to the left, three exit nodes placed to the right and Patroller's base positioned in the middle. The final game value is 0.327, i.e. giving the Patroller a chance of 32.7% to intercept the Evader.

areas (such as desert, forest or sea.). We denote the longer and shorter side as *length* and *width*, respectively. In this evaluation, we place the entry and exit locations on the opposite shorter sides of the grid and the base in the middle.

As a representative of planar city graphs, we use a road network graph extracted from GIS data of part of Prague (see Figure 2a).

Example Solutions

The solution of the transit game with deterministic encounters on the rectangular grid graph (of width $w = 3$, length $l = 6$ and base $b = 7$) for the Evader and Patroller is presented in the form of a probability distribution over edges on Figures 1b and 1c, respectively. The weight of an edge is the probability of the respective player traversing that edge if it plays its optimal mixed strategy. Note that, in the transit games with deterministic encounters, the resulting Evader strategy contains only forward edges (i.e. edges $\{(i,j) |\ d(j,o) < d(i,o),\ o \in N_{out}\}$[5]). The solution of the transit game on the city graph is depicted on Figures 2b and 2c.

Note that all described variants of oracle algorithms provide the same solutions on the above test problems. In contrast, standard solution approach working with full strategy sets for both players was not applicable on a standard desktop computer due to the size of the resulting linear program.

Computation Time

Although all described oracle algorithms provide the same solution, the time required to compute the solution differs. We evaluate the following three variants of oracle-based algorithms:

1. *Evader's Single Oracle* (*ESO*) is the standard single-oracle algorithm utilizing only the Evader's best-response oracle for the Evader.
2. *Evader's Accelerated Single Oracle* (*ESO-A*) is the accelerated single-oracle algorithm utilizing the Evader's subgame expansion oracle.
3. *Accelerated Double Oracle* (*DO*) is the accelerated double oracle algorithm utilizing Evader's subgame expansion oracle, and Patroller's best-response oracle.

Note that the single-oracle algorithm utilizing Patroller's best-response oracle could not be used since the size of the set of all possible Evader paths (that are needed to be enumerated completely) quickly exceeds the memory limits.

[5] $d(j,o)$ denotes Chebyshev/L_∞ distance and N_{out} is the set of exit nodes.

Tables 1a and 1b summarize the computation times for the algorithms on both types of graphs and with both deterministic and non-deterministic encounters. The maximum length of Patroller's walk was $L_P = 7$. The Evader's accelerated single oracle *ESO-A* outperforms both the Evader's standard single oracle *ESO* and the double oracle *DO* on all test instances. Interesting difference in performance can be observed when solving games on grid and city graphs. The number of all possible Patroller's walks is 24463 for the grid graph and only 185 for the city graph; this inequality is reflected in runtimes of *DO* (employing Patroller's oracle). On the grid graph, *DO* needs significantly more time to find a NE than the single-oracle-based algorithms. However, on the city graph, *DO* is not penalized by using the Patroller's oracle; in this case, the computation time is highest for *ESO*.

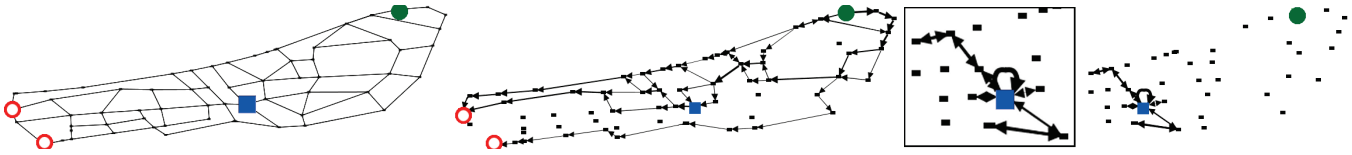
Note the higher number of iterations for *DO* compared to *ESO-A*. Although both *DO* and *ESO-A* employ Evader's heuristic oracle, due to the incomplete enumeration of the Patroller's strategies, *DO* needs more iterations to converge. For the single-oracle algorithms, the time spend on linear programs computing NE of the subgames is higher than for the double-oracle algorithm due to the fact that in the case of single-oracle algorithms, the generated subgames contain all Patroller's strategies (i.e. all closed walks from the base) and the resulting linear programs are significantly larger.

We explore the scalability of the fastest algorithm (*ESO-A*) on set of rectangular grid graphs of various width and length. The runtime of the algorithm depends both on the size of the graph and the maximum length of Patroller's path, which was varied from 6 to 8. See the results in Figure 3. In general, the limits of *ESO-A* were hit because of the Evader's subgame expansion oracle which turned up unable to find a subgame expanding strategy on a grid graph 12x4 with $L_P = 8$.

Convergence and Approximation

We have studied the dynamics of the proposed oracle algorithms with the ambition of utilizing the knowledge of their convergence for trading time for optimality and producing approximate solutions.

The speed of convergence of *ESO-A* and the time needed to obtain a solution within required bounds is depicted on the Figure 4. We denote the difference between the value of the response provided by the oracle and the current subgame value as δ and the final game value as $\mathcal{V}^*$. The y-axis shows the ratio $\epsilon = \delta / \mathcal{V}^*$ which serves as the error upper bound, if we terminate the algorithm at given time (x-axis).



(a) Exemplar planar city graph. Loops and interception probabilities omitted.

(b) Evader's optimal strategy.

(c) Patroller's optimal strategy.

Figure 2: Exemplar transit game on a planar city graph with non-deterministic encounters. The graph contains 66 nodes and 100 edges and has one entry node in the eastern part of the graph (depicted as the full green circle), two exit nodes in the western part of the graph (depicted as empty red circles) and the base (depicted as the blue square) in the middle. The final game value is 0.146, i.e. giving the Patroller a chance of 14.6% to intercept the Evader.

Graph	Alg	Iters	Time	EO	PO	CLP
Grid 4x6	ESO	36	7.2	3.5	1.1	2.5
	ESO-A	33	6.2	2.7	1.1	2.4
	DO	60	91.1	2.7	88.3	0.01
City	ESO	13	76.7	76.6	0.1	0.04
	ESO-A	14	18.7	18.6	0.07	0.04
	DO	16	29.8	28.8	0.5	0.06

(a) Transit game with deterministic encounters.

Graph	Alg	Iters	Time	EO	PO	CLP
Grid 4x6	ESO	29.0	37.5	34.1	1.1	2.3
	ESO-A	30.9	6.9	3.5	1.2	3.2
	DO	51.9	178.6	5.7	172.56	0.09
City	ESO	18.8	177.2	177.0	0.04	0.06
	ESO-A	11.7	29.3	29.3	0.01	0.03
	DO	14.3	37.5	37.0	0.3	0.02

(b) Transit game with non-deterministic encounters. Averaged over 20 samples of random distribution of interception probabilities.

Table 1: Runtime results in seconds for the transit game. EO — Evader's oracle time, PO — Patroller's oracle time, CLP — time to solve all subgames.

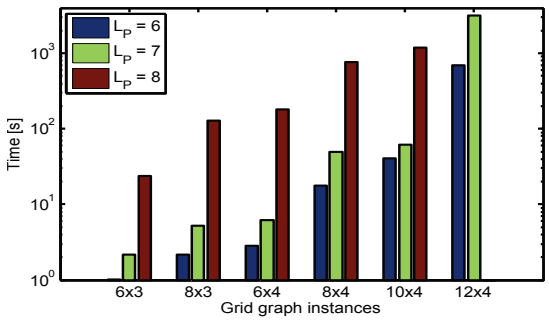


Figure 3: Computation time of *ESO-A* for a transit game with deterministic encounters on grid graphs of various size and varying maximum length of Patroller's walk L_P.

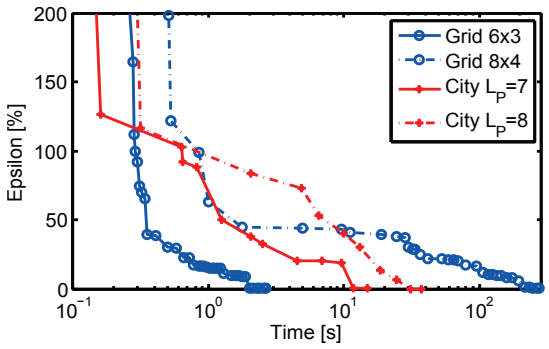


Figure 4: Trade-off between solution accuracy and convergence speed for various test problems for *ESO-A*.

Simulation-based Evaluation

To validate the presented approach outside the very framework of game theory and to provide a bridge towards more applied work on maritime security, we have tested the game-theoretic route selection strategies on an agent-based simulation of maritime traffic (Vaněk et al. 2011). In contrast with the highly abstracted transit game, the simulation represents the domain with a much higher level of detail with near-continuous time and continuous space. We applied the transit game solution from the perspective of a vessel (corresponding to the Evader) that needs to repeatedly transit through the Gulf of Aden area. The pirate (corresponding to the Patroller) was not employing game-theoretic model; instead, in line with the domain knowledge, it was represented as an adaptive agent capable of learning from its past successes and failures in trying to intercept the transiting vessel. The learning capability was implemented using a softmax multi-armed bandit model (Sundaram 2005). We represented the transit area as a 12x4 grid graph with Patroller's base placed on the node closest to the Bosaso harbor, known as a major piracy hub. We computed the optimal Evader's strategy and deployed it in the simulation (Figure 5). We then simulated 10000 transit runs and measured the pirate success ratio after its performance converged.

We compared the game-theoretic solution (denoted as *GT*) to two other transit strategies: the *IRTC* method representing the current transit scheme in which the transport vessels follow a fixed International Recommended Transport Corridor[6] and the *UNIFORM* method choosing randomly with uniform distribution from all possible shortest paths. Results in the Table 2 show that both the randomized strategies significantly outperform the current transiting scheme, moreover, the game-theoretic randomization out-

[6]http://www.cusnc.navy.mil

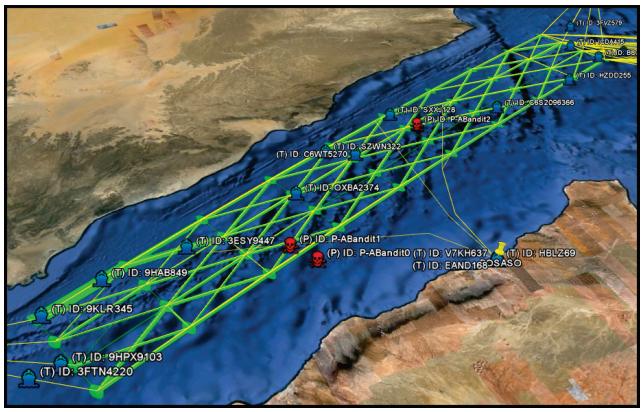


Figure 5: Gulf of Aden transit simulation with a deployed game-theoretic solution. Best viewed in color.

performs the uniform randomization. The relatively small difference between the two randomized strategies is given by the fact that the symmetric and homogeneous nature of the transit game leads to a solution which is close to uniform. Higher differences can be expected if more detail and/or structure is introduced (e.g. the real area in not strictly rectangular) however this is left for future work.

Strategy	IRTC	UNIFORM	GT
Rate [%]	22	9.19	8.53

Table 2: Comparison different strategies in simulation of transiting the Gulf of Aden.

Conclusions

We addressed the problem of a strategic confrontation between two mobile agents — an Evader trying to pass through an area where a Patroller is trying to intercept its transit. In order to find optimum route selection strategies, we have formalized the problem as a transit game, a novel security game model which assumes both players to be mobile.

The huge size of the strategy spaces called for the application of iterative oracle-based algorithms for NE computation. In order to limit the need for best response calculation, which is hard for games on graphs, we have introduced a novel accelerated variant of oracle algorithm which speeds up NE computation without the loss of the optimality guarantee. We have evaluated the approach on two classes of games inspired by real-world security applications and, in one case, were even able to compare the effectiveness of game-theoretic solution with currently deployed solutions.

Acknowledgments

This research was supported by the U.S. Office of Naval Research (grant no. N000140910537) and by the Czech Ministry of Education, Youth and Sports (grant no. LH11051).

References

Alpern, S. 1992. Infiltration Games on Arbitrary Graphs. *Journal of Mathematical Analysis and Applications* 163:286–288.

Barnhart, C.; Johnson, E.; Nemhauser, G.; and Savelsbergh, M. 1994. Branch and Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 46:316–329.

Brooks, R. R.; Schwier, J.; and Griffin, C. 2009. Markovian Search Games in Heterogeneous Spaces. *IEEE Trans. on Systems, Man and Cybernetics Part B* 39(3):626–335.

Dickerson, J.; Simari, G.; Subrahmanian, V.; and Kraus, S. 2010. A Graph-Theoretic Approach to Protect Static and Moving Targets from Adversaries. In *Proceedings of AAMAS*, 299–306.

Flood, M. M. 1972. The Hide and Seek Game of Von Neumann. *Management Science*.

Gal, S. 1980. *Search Games*. Academic Press, New York.

Gilpin, R. 2009. Counting the Costs of Somali Piracy. Technical report, US Institute of Peace.

Halvorson, E.; Conitzer, V.; and Parr, R. 2009. Multi-step Multi-sensor Hider-seeker Games. In *Proceedings of IJCAI*.

Jain, M.; Kardes, E.; Kiekintveld, C.; Ordonez, F.; and Tambe, M. 2010a. Security Games with Arbitrary Schedules: A Branch and Price Approach. In *Proceedings of AAAI*.

Jain, M.; Tsai, J.; Pita, J.; Kiekintveld, C.; Rathi, S.; Tambe, M.; and Ordonez, F. 2010b. Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshals Service. *Interfaces* 40:267–290.

Jain, M.; Vaněk, O.; Korzhyk, D.; Conitzer, V.; Tambe, M.; and Pěchouček, M. 2011. Double oracle algorithm for zero-sum security games on graphs. In *Proceedings of AAMAS*.

Jakob, M.; Vaněk, O.; and Pěchouček, M. 2011. Using agents to improve international maritime transport security. *IEEE Intelligent Systems* 26(1):90–96.

McMahan, H. B.; Gordon, G. J.; and Blum, A. 2003. Planning in the Presence of Cost Functions Controlled by an Adversary. In *Proceedings of ICML*, 536–543.

Ruckle, W.; Fennell, R.; Holmes, P. T.; and Fennemore, C. 1976. Ambushing Random Walks I: Finite Models. *Operations Research* 24:314–324.

Sundaram, R. K. 2005. *Studies in Choice and Welfare*. Springer Berlin Heidelberg.

Vaněk, O.; Bošanský, B.; Jakob, M.; and Pěchouček, M. 2010. Transiting Areas Patrolled by a Mobile Adversary. In *Proceedings of IEEE CIG*.

Vaněk, O.; Jakob, M.; Hrstka, O.; and Pěchouček, M. 2011. Using multi-agent simulation to improve the security of maritime transit. In *Proceedings of MABS*.

Washburn, A., and Wood, K. 1995. Two-person Zero-sum Games for Network Interdiction. *Operations Research* 43(2):243–251.