Transiting Areas Patrolled by a Mobile Adversary

Ondřej Vaněk, Branislav Bošanský, Michal Jakob and Michal Pěchouček

***Abstract*— We study the problem of a mobile agent trying to cross an area patrolled by a mobile adversary. The transiting agent aims to choose its route so as to minimize the probability of hostile encounter; the patroller agent, controlling one or more patrol units, aims at the opposite. We model the problem as a two-player zero-sum game (termed *transit game*) and search for an optimum route selection strategy as a mixed Nash equilibrium of the game. In contrast to existing game-theoretic models of this kind, we explicitly consider the limited endurance of patrols and the notion of bases to which the patrols need to repeatedly return. Noting the prohibitive size of the transit game, we employ two techniques for reducing the complexity of finding Nash equilibria – a compact network-flow-based representation of transit routes and iterative single- and double-oracle algorithms for incremental game matrix construction. We measure the computational time of all the methods on a range of transit game instances. In order to assess the practical relevance of the approach, we apply the transit game model and its solution to the real-world case of ship transit through areas affected by piracy. The results obtained using an agent-based simulation of maritime traffic show that the randomized game-theoretic transit routing strategy results in a lower number of pirate attacks than the currently employed method based on static transit corridors.**

I. INTRODUCTION

A common problem faced by agents operating in areas controlled by an adversarial mobile agent is how to move while avoiding detection and interception. An important variant of this problem arises when an agent needs to safely cross an area from one side to another and the mobile adversary wants to thwart the attempt by finding and intercepting the transiting agent. Real-world examples of such situations include logistics in insecure regions, illegal border crossing and/or smuggling interdiction.

In such cases, a rational transiting agent aims to choose a route for which the probability of encounter is minimal; analogously, a rational patrolling agent aims to choose a route maximizing the probability. The situation can then be modeled as a game and the best route selection strategy can be sought as a solution of the game. Game-theoretic approach has been applied to similar problems in the past, resulting in a family of various games, each reflecting specific assumptions and modeling decisions made regarding the capability of the agents and the properties of the environment (see the analysis in the Related Work section).

In this paper, we model the problem of transiting an area with a mobile adversary as a *zero-sum game* between *two players*: the first player – *the Evader* – chooses a route from one side of the area to the other side; the second player – *the Patroller* – controls one or more mobile patrols for which it chooses a closed-walk starting and ending in a given location in the area, termed *base* . If, following their chosen routes, some patrol and the Evader meet, the Patroller wins; otherwise the Evader wins.

The concept of patrol bases and the closed-walk nature of patrols' routes are salient features of our model, termed *transit game*. Despite their importance to many real-world scenarios, these elements have not been, to our best knowledge, considered in existing game-theoretic work.

Formalizing a strategic conflict as a game is only the first step in addressing the problem. In order to determine a Nash equilibrium of the game, i.e., the route selection strategy that a rational agent should execute, we employ linear programming in a standard construction used for solving two-player zero-sum games. In its basic form, however, the approach becomes quickly intractable due to the prohibitive size of the resulting linear programs when the size of the transit area or the number of patrols increases. To tackle this problem, we employ two complexity reduction techniques: (1) using a compact *network-flow-based representation* of the Evader's strategy space (described in [1] and later used e.g. in [2]) and (2) using an *single-* and *double-oracle iterative algorithms* (presented in [3]) for finding a Nash equilibrium. Employing the two techniques, individually and in concert, and analyzing their computational requirements is the second important contribution of the work.

We do not stop here, however. Taking advantage of our work on transport security [4], [5], we apply the transit game model to a real-world problem of ship transit through areas affected by maritime piracy. Employing an agent-based simulation of maritime traffic, we validate the presented model and evaluate the effectiveness of the game-theoretic approach on a very timely use case.

II. RELATED WORK

As already mentioned, game theory has been applied to model and study strategic behavior in scenarios involving mobile patrols. The models developed share several characteristics – they are usually defined on a discrete space (i.e. graph), consist of one player trying to avoid contact (termed *hider*, *evader*, or *infiltrator*) and the other player (termed *searcher*, *seeker*, *pursuer*, *patroller* or *guard*) trying to discover and intercept the first player. All such games can be expressed in the normal form if the current position and strategies of one player are unknown to the other players. In further text we refer to these type of games as the *patrol games*.

Despite certain similarities, there are also characteristics distinguishing individual classes of patrol games. In *ambush games* [6] the evading player tries to transit an area, whereas

Agent Technology Center, Dept. of Cybernetics, FEE, Czech Technical University, Technická 2, 16627 Prague 6, Czech Republic, {vanek, bosansky, jakob, pechoucek}@agents.felk.cvut.cz

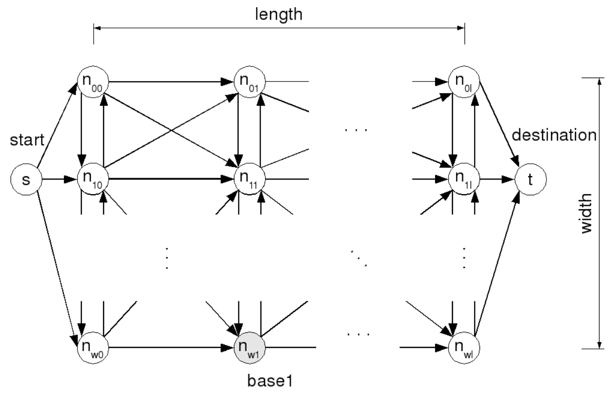


Fig. 1: Transit game graph.

the intercepting player is immobile and only selects, a priori, several static ambush points in the graph. The mobility is reversed in *search games* [7], where the evading player is immobile and the intercepting player is mobile. In *hider-seeker games* [8], both players are mobile with no explicit restrictions on the motion trajectories the players can and/or should employ.

Another differentiating factor is the players' range of sight. Whereas in all the above games the player can only see the other player if it resides at the very same location, in *pursuit-evasion games* [9] or *hunter-rabbit games* the players can also see adjacent or even more distant nodes (e.g. using the flashlight beam [10]). Longer range of sight may apply to the intercepting player, the evading player, or both.

A slightly different class of games, typically used in modeling infiltration scenarios, are *Stackelberg games* [11], where a mixed strategy employed by the guard (termed *leader* in these games) is known to the infiltrator (called *follower*). For zero-sum games, nevertheless, Stackelberg games are identical to standard normal form games.

The model closest to our approach are *infiltration games* [12] where the infiltrator tries to cross a given area between a defined entry and exit location without being caught by a guard. Our transit game differs from infiltration games in two key aspects – (1) there can be multiple patrols in the area, and (2) the patrol needs to move along a closed walk, visit its base, and eventually return to a point from where it started its movement. Existence of such patrol bases and associated movement constraints is common in real-world scenarios and the ability of our model to explicitly take this factor into consideration is an important advantage of our approach.

III. Transit Game

We now formally introduce the transit game model.

A. Game Graph

We model the environment where the players move as a discretized rectangular area[1] represented by a directed network $G = (N, A)$, where N is a set of nodes and A is a set of arcs. The nodes are assumed to be located on a homogeneous rectangular grid covering the game area (see Figure 1); we call the width w_G and length l_G of the game graph the width and length of such a grid (in the number of arcs) and we assume both to be ≥ 1. There are two nodes outside the grid – $s, t \in N, s \neq t$, are the start (origin) and the destination (target) nodes. We refer to the nodes by indexes denoting their position in the grid (i.e. node $n_{ij} \in \{N \setminus \{s,t\}\}$ denotes a node in the i-th row and the j-th column, where $i = 0, \ldots, w_G$ and $j = 0, \ldots, l_G$). Finally, we define $B_G \subseteq \{N \setminus \{s,t\}\}$ as the set of patrol bases.

Arcs are represented as ordered pairs $(n, n') \in A$ where $n, n' \in N$. The game graph contains the following arcs:

- the start node is connected to all nodes in first column of the grid: $\forall i = 0, \ldots, w_G \;\; (s, n_{i0}) \in A$
- the destination node is connected to all nodes in the last column of the grid: $\forall i = 0, \ldots, w_G \;\; (n_{il_G}, t) \in A$
- the nodes inside the grid are connected to their neighbors, $\forall i = 0, \ldots, w_G; j = 1, \ldots, (l_G - 1):$

$$\forall k \in \{\max(0, i-1), \min(w_G, i+1)\} \; (n_{ij}, n_{kj}) \in A$$
$$\forall k \in \{\max(0, i-1), \ldots, \min(w_G, i+1)\} \; (n_{ij}, n_{k(j+1)}) \in A$$

Where the game graph is clear from context, we omit the lower index G for w_G, l_G, and B_G. Finally, we refer to the nodes where the patrol bases are placed by index $b = 1, \ldots, |B_G|$.

B. Game Definition

We formalize the transit game as a zero-sum normal-form game between two players – the *Evader* and the *Patroller*. The game is defined as a tuple $\langle G, S_E, S_P, u \rangle$, where G denotes the graph of the game, S_E denotes the set of strategies of the Evader (in the form of paths for the evading unit – e.g. a person, a vessel, a car, etc.), S_P the set of strategies of the Patroller (in the form of closed walks for each of the patrolling unit). The u denotes the utility function $u : S_E \times S_P \to \mathbb{R}$ that assigns a real-valued utility based on chosen pure strategies. Let $\Delta(S_E)$ be the set of probability distributions over S_E, we denote $\sigma_E \in \Delta(S_E)$ to be the mixed strategy of the Evader; $\Delta(S_P)$, and σ_P is defined similarly for the Patroller. The game value $\mathcal{V}$ is then equal to:

$$u(\sigma_E, \sigma_P) = \sum_{s_E \in S_E} \sum_{s_P \in S_P} \sigma_E(s_E) \cdot \sigma_P(s_P) \cdot u(s_E, s_P) \quad (1)$$

Now, the Nash equilibrium is a pair of strategies $(\sigma_E^*, \sigma_P^*) \in \Delta(S_E) \times \Delta(S_P)$ such that:

$$u(\sigma_E^*, \sigma_P^*) \leq \min_{\sigma'_E \in \Delta(S_E)} u(\sigma'_E, \sigma_P^*) \quad (2)$$

$$u(\sigma_E^*, \sigma_P^*) \geq \max_{\sigma'_P \in \Delta(S_P)} u(\sigma_E^*, \sigma'_P) \quad (3)$$

C. Representation of Strategies

1) Evader: The set of strategies of the Evader S_E can be represented as the set of all possible paths from the start s to the destination t in the game graph G. We limit the

[1] This assumption can be easily dropped, as the algorithms presented in this paper can be applied to arbitrary graphs.

evading unit to use only those arcs that point towards the destination – i.e. $l = j + 1$ for each arc (n_{ij}, n_{kl}) traversed by the Evader.

With increasing the size of the graph (l_G and/or w_G), however, the number of all possible transit paths, and therefore the size of the Evader's strategy set, grows exponentially. Therefore, we employ an alternative representation of Evader's mixed strategy space, based on [1]. Instead of choosing a distribution over the set of transit paths, the Evader chooses a *network flow* over the directed arcs that can be utilized by the Evader. We define

$$P = \{p_{(n,n')} | (n, n') \in A\} \tag{4}$$

to be a network flow from origin s to destination t if

$$p_{(n,n')} \geq 0, \quad \forall (n, n') \in A \tag{5}$$

$$\sum_{i|n_{i1} \in N} p_{(s,n_{i0})} = 1 \tag{6}$$

$$\sum_{i|n_{il_G} \in N} p_{(n_{il_G},t)} = 1 \tag{7}$$

$$\sum_{i|n_{i(k-1)} \in N} p_{(n_{i(k-1)},n_{jk})} = \sum_{i|n_{i(k+1)} \in N} p_{(n_{jk},n_{i(k+1)})} \tag{8}$$

i.e. the total flow from the starting node s is equal to 1, the total flow to the destination node t is equal to 1, and for all other nodes, the node's incoming flow equals the node's outgoing flow.

The two representations are equivalent in a sense that each Evader's mixed strategy can be expressed both in the original path-based and the new flow-based representation. The difference is in the size of the set of Evader's strategies $|S_E|$. When the basic path-based representation is used the size of $|S_E|$ grows exponentially with respect to the size of the graph (w_G, l_G). When the flow-based representation is used, the size of $|S_E|$ grows only linearly. We distinguish between the two representations in the experiments, hence we further use the notation $\sigma_E^p \in \Delta(S_E^p)$ for the basic path-based representation, and $p = \sigma_E^n \in \Delta(S_E^n)$ for the network-flow-based representation.

2) Patroller: The Patroller controls one or more patrolling units. More specifically, we assume that there is exactly one patrolling unit for each patrol base, hence the Patroller controls $|B|$ patrolling units. The patrolling units can move through the network regardless of the orientation of the arcs, but their paths have to form a closed walk. Moreover, we assume that each such a closed walk consists of even number of the arcs. This assumption allows us to reduce the Patroller's strategy representation to half of the length of the closed walk – let the patrolling unit without the loss of generality start in its base and traverse through the network half of the length of the closed walk reaching some node n. Now, because the Patroller is rational and maximizes its utility, this path from the base to node n has to have maximum value. Therefore, because the unit has to return to the starting point, the second half of the closed walk has to contain the same arcs as the first one.

Note that the network-flow based representation cannot be directly used for the Patroller because the paths of patrolling units can contain loops, which are not allowed for network flows. In the rest of the paper we understand a strategy $s_P \in S_P$ to be a vector $(s_P^1, \ldots, s_P^{|B|})$ of paths (representing the half of the closed walk) for each Patroller's unit going through the unit's corresponding base.

D. Utility

Let us now define the utility function u used in the transit game. The utility for pure strategies is calculated as a sum of payoffs for each arc along the path of the evading unit. The arc has non-zero utility value, if the arc its shared by the evading unit's path and by some patrolling unit's closed walk:

$$u(s_E, s_P) = \sum_{b=1}^{|B|} \left(\sum_{(n,n') \in s_E \cap s_P^b} u((n, n'), s_P^b) \right) \tag{9}$$

where $u((n, n'), s_P^b)$ is:

$$u((n, n'), s_P^b) = \frac{1}{l(s_P^b)} \cdot d((n, n'), l(s_P^b), b) \tag{10}$$

where $l(s_P^b)$ is the length of the closed walk of the patrolling unit in the number of arcs and $d((n, n'), l(s_P^b), b)$ represents a distance of the arc from base b:

$$\begin{cases} \frac{2}{d(n,b)+d(n',b)} & \text{if } \frac{2}{d(n,b)+d(n',b)} < l(s_P^b) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $d(n, b)$ is the maximal difference between the positions of nodes $n = n_{ij}$ and $b = n_{kl}$:

$$d(n, b) = \max(|i - k|, |j - l|) \tag{12}$$

The above definition of the utility aims to capture two salient properties of the problem. Firstly, the utility depends on the length of the closed walk of the patrolling units – shorter paths are preferred by the Patroller because longer paths incur higher cost (higher fuel consumption, more supplies necessary to transport, etc.). The Evader on the other hand does not know the location of patrolling units at the beginning of the game (from the Evaders's point of view – i.e. when the evading unit enters the area). Therefore, the longer the closed walk of the patrolling unit, the lower the probability of the patrolling unit being in the "right arc" at the beginning of the game. More precisely – when the paths of the evading unit and some patrolling unit overlap (i.e. $s_E \cap s_P^b \neq \emptyset$), the fact that the two units actually meet in the game and that the patrol encounters the evading unit depends on the starting arcs of each of the units. Therefore, from the Evader's perspective, patrolling units can possibly be at one of $l(s_P^b)$ arcs at the beginning of the game. From this set of possible arcs only one would lead to a successful encounter, as the evading and the patrolling unit have to make exactly the same number of moves until they meet at the given colliding arc.

Secondly, the utility depends on the distance of the arc to the patrol base. The Patroller prefers shorter distances to

Algorithm 1 Single-Oracle Algorithm

```
F_{S_P} ← ∅, break ← false
σ_E^{n/p} ← uniform distribution over all S_E
repeat
  (σ_E^{n/p}, σ_P) ← compute NE using LP for σ_E^{n/p} and F_{S_P}
  f ← BR_P(σ_E^{n/p})
  if f ∉ F_{S_P} then
    F_{S_P} ← F_{S_P} ∪ {f}
  else
    break ← true
  end if
until not break
```

Algorithm 2 Double-Oracle Algorithm

```
break ← false
F_{S_P} ← { arbitrary strategy s_P ∈ S_P}
F_{S_E} ← { arbitrary strategy s_E ∈ S_E}
repeat
  (σ_E^{n/p}, σ_P) ← compute NE using LP for F_{S_E} and F_{S_P}
  f_P ← BR_P(σ_E^{n/p})
  f_E ← BR_E(σ_P)
  if (f_P ∈ F_{S_P})&(f_E ∈ F_{S_E}) then
    break ← true
  else
    if f_P ∉ F_{S_P} then
      F_{S_P} ← F_{S_P} ∪ {f_P}
    end if
    if f_E ∉ F_{S_E} then
      F_{S_E} ← F_{S_E} ∪ {f_E}
    end if
  end if
until not break
```

bases (in order to safely escort a captured evader to the base), while the Evader prefers the opposite.

Note that in any case the above definition of trade-offs constitutes a compromise between the accuracy of capturing the motivations of the players and staying within the framework of zero-sum games. The motivations could be modeled with higher flexibility and greater amount of detail if the zero-sum assumption were lifted. Unfortunately, general-sum games are significantly more difficult to solve and their application would render the already computationally complex problem practically intractable.

IV. Solving the Transit Game

We use three techniques for finding a Nash equilibrium of the transit game. The first technique uses network-flow-based representation of Evader's strategies and employs linear programming; we term this technique *flow-based*. The other two techniques – the *single-oracle* and the *double-oracle* algorithm – employ iterative algorithms presented in [3] which iteratively construct the game matrix and they both converge to a Nash equilibrium.

1) Linear Programming (LP): As the transit game is a zero-sum game, a Nash equilibrium can be found by solving a linear programming problem constructed from the game matrix. Due to the linear nature of network-flow constraints (Equations 5 to 8), this method can be used for our case even when the Evader's strategies are expressed using the network-flow representation.

The linear problem for the Evader can be now formulated as finding values $p = \sigma_E^n \in \Delta(S_E^n)$ minimizing the value of the game $\mathcal{V} = (\sigma_{\mathcal{E}}, \sigma_{\mathcal{P}})$ (see Equation 1) subject to the following constraints:

$$\forall a_i \in S_E \quad \mathcal{V} \geq \sum_{b=1}^{|B|} \left(\sum_{s_P^b \in S_P | a_i \in s_P^b} p_{a_i} \cdot u(a_i, s_P^b) \right) \tag{13}$$

and network flow constraints for p_{a_i} (equations 5 to 8); a_i denotes an arc in the network and $u(a_i, s_P^b)$ is calculated as defined in Equation 10.

The solution of the linear program can be found using a wide range of linear programming algorithms, e.g. the simplex method or the interior point method. The values of the primal solution p^* of the linear programming problem are the arc-transition probabilities for the Evader. The network flow constraints ensure that the support of p^* forms a set of valid paths from s to t. The problem for computing the dual solution can be formulated similarly and it results in the probabilities of the strategies σ_P^* used by the Patroller.

2) Single/Double Oracle Algorithms: The pseudo-code of the single oracle algorithm is shown in Algorithm 1, the double oracle algorithm is shown in Algorithm 2. The oracle algorithms are based on the existence of (fast) algorithms (called oracles) calculating the best response for a player to a given opponent's strategy. Best response for the Evader is a function $BR_E : \Delta(S_P) \to S_E$ such that

$$u(BR_E(\sigma_P), \sigma_P) = \min_{s'_E \in S_E} (s'_E, \sigma_P) \tag{14}$$

Best response oracle for the Patroller BR_P is defined analogously.

Both oracle algorithms construct the game matrix iteratively – the single oracle algorithm adds allowed strategies iteratively only for one player (i.e. it only adds rows or columns to the game matrix), while the double oracle adds legal strategies iteratively for both players (i.e. it adds both rows and columns to the game matrix). It has been proved (see [3]) that both algorithms converge to a Nash equilibrium, though they may need to construct a complete game matrix before it happens, thus eliminating any possible computation savings.

As shown in [13], best response can be computed rapidly for many types of games; unfortunately, for the hider-seeker games computing best responses for both players is NP-hard. Although various approximation techniques can be used, we combine the oracle algorithms with the network-flow representation of Evader's strategies. In the case of the transit game computing the best response for a player amounts to finding a minimal (maximal) allowed path in the graph using a search algorithm (e.g. branch and bound) for the Patroller or dynamic programming for the Evader.

V. Evaluation

We have performed two types of empirical evaluation. The first, technical evaluation (Section V-A) investigates the

Technique	Evader's strategy rep.	Patroller's Oracle	Evader's Oracle
FLOW-BASED	edges	no	no
PATH-SINGLE-ORACLE	paths	yes	no
FLOW-SINGLE-ORACLE	edges	yes	no
PATH-DOUBLE-ORACLE	paths	yes	yes
FLOW-DOUBLE-ORACLE	edges	yes	yes

TABLE I: Summary of the equilibrium search methods.

computational complexity of the approach by comparing the run times of different techniques for finding a Nash equilibrium on several instances of the transit game. The second evaluation (Section V-B) takes a broader perspective and evaluates the overall effectiveness of the game-theoretic approach to solving a real-world transit problem in a particular domain of maritime transport security.

A. Computational Complexity

1) Method Configuration: We evaluate five different techniques for finding a Nash equilibrium of the transit game (summarized in Table I):

1) FLOW-BASED method creates the game matrix by enumerating all arcs (i.e. the flow-based representation, σ_E^n) that can be traversed by the Evader, and all possible walks of the patrols. The sets of strategies are constant for both players, and the game is solved by a standard linear programming technique (see Section IV-.1).
2) PATH-SINGLE-ORACLE method utilizes the Patroller's oracle, the set of Evader strategies uses the path-based representation σ_E^p.
3) FLOW-SINGLE-ORACLE method utilizes Patroller's oracle, the set of Evader strategies uses the flow-based representation σ_E^n.
4) PATH-DOUBLE-ORACLE method utilizes both the Patroller's and the Evader's oracle, the set of Evader strategies uses the path-based representation σ_E^p.
5) FLOW-DOUBLE-ORACLE method utilizes both the Patroller's and the Evader's oracle, the set of Evader strategies uses the flow-based representation σ_E^n.

The approach using all possible paths of the Evader without an oracle-based algorithm was not usable due to the size of the strategy sets even on the smallest graph instances, and is thus not further described.

All evaluation was performed on a desktop PC at 2.83 GHz[2] with 2GB RAM available to the algorithms. Implementation was done in Java and OJALGO[3] Java library was used as a linear programming solver.

2) Game Parameters: We evaluated all methods on a range of scenarios differing in the following parameters:

1) *Graph width* w_G – The graph width affects solution time because the size of the strategy sets of both players is proportional to the size of the game graph.
2) *Maximum length of Patroller's walks* $l(cw)$ – Patroller's strategies have to include at least some walks reaching the opposite side of the graph from the Patroller's base (if there is no such strategy, the Evader can always select a path that is not reachable by the Patroller, thus transit the area with zero risk of being attacked). The maximum length of Patroller's walks $l(cw)$ thus has to be set at least to $l(cw) = 2 \cdot w_G + 1$ when varying w_G; this is the value to which it was set in the experiments[4].
3) *Number of patrol bases* $|B|$ – the bases are located on the sides of the graph, however in principle they can be placed anywhere in the graph.

If w_G is fixed but the length of the graph l_G is increased, the set of Evader's strategies grows proportionally, however the game value stays the same, due to the rectangular shape of the area and limit of the maximum length of Patroller's walks.

3) Results: Time Complexity: Table II summarizes solution times for each algorithm. The FLOW-BASED approach was not able to construct the linear programming matrix for graphs of width $w_G > 3$ (number of all possible Patroller walks for $w_G = 4$ and $w_G = 5$ is 13,944 and 104,278, respectively, and the linear programming problem could not be constructed due to memory limits). The PATH-DOUBLE-ORACLE algorithm outperforms all described methods because the number of iterations is in general lower and the size of the linear programming problem smaller, in spite of a slightly longer time for each iteration compared with single-oracle methods (because of the time needed for the Evader's oracle to provide the best response).

If we compare the single-oracle methods, the PATH-SINGLE-ORACLE method cannot solve larger problems because of the size of Evader's strategies (1220 and 14411 paths for width $w_G = 3$ and $w_G = 4$, respectively, in contrast to 60 and 104 edges in the case of flow-based representation for the same widths). Even for smaller graph instances, the game matrix is large enough to render the PATH-SINGLE-ORACLE method inapplicable. Overall, for the single-oracle approach, the compact flow-based strategy representation has a clear advantage over the original path-based representation.

[2]Implementation is single-threaded.

[3]http://ojalgo.org

[4]Beside the main batch of experiments, we have also quickly evaluated the dependence on $l(cw)$: increasing the maximum length of Patroller's walks $l(cw)$ while keeping other parameters constant, the size of the Patroller's strategy set $|S_P|$ grows exponentially, as does the computation time of the best response algorithm for the Patroller BR_P; the resulting strategy for the Evader remains the same, though.

w_G	$l(cw)$	$\|B\|$	Solution Time [s]					Iterations			
			FB	*PA-SO*	*FL-SO*	*PA-DO*	*FL-DO*	*PA-SO*	*FL-SO*	*PA-DO*	*FL-DO*
2	5	1	0.02	1.48	0.15	0.11	0.16	21	18	28	28
3	7	1	3.42	50.01	0.86	0.89	1.87	45	50	47	60
4	9	1	–	–	10.44	7.54	17.53	–	81	60	82
5	11	1	–	–	146.08	105.46	177.20	–	136	99	130
4	9	2	–	–	68.41	52.29	81.62	–	243	189	209
5	11	2	–	–	1097.62	838.74	1436.12	–	860	699	802

TABLE II: Comparison of the time and the number of iterations required to find a Nash equilibrium for different sizes of the transit game and different methods – FLOW-BASED (FB), PATH-SINGLE-ORACLE (PA-SO), FLOW-SINGLE-ORACLE (FL-SO), PATH-DOUBLE-ORACLE (PA-DO) and FLOW-DOUBLE-ORACLE (FL-DO) methods.

Comparing the FLOW-DOUBLE-ORACLE and the PATH-DOUBLE-ORACLE methods, the longer computation time of the FLOW-DOUBLE-ORACLE is caused by a greater number of iterations needed before reaching an equilibrium. This is because the flow-based representation generates a more complex structure of Evader's strategy subspaces, whose iterative exploration then takes longer to converge. Overall, for the double-oracle approach, the compact flow-based strategy representation has no advantage – in fact, it renders the FLOW-DOUBLE-ORACLE method slower.

In the case of multiple bases, Patroller's walks from one base does not affect walks from the others, so the best response for multiple bases comprises the combination of individual best responses for each base. The best response algorithm depends linearly on the number of bases $|B|$, however, the size of the strategy set of the Patroller $|S_P|$ grows exponentially in $|B|$. Therefore, the number of iterations needed, and thus the overall computation time, grows exponentially.

4) Results: Example Strategies: Figures 2a and 2b show example resulting equilibrium strategies (for both players and in the flow-based representation) for the game on a graph of width $w_G = 4$ with one Evader's base. Figures 2c and 2d do the same except there are two Evader's bases placed at the opposite sides of the game graph. The arcs represent possible moves of the player; the arc width is proportional to the probability of the player traversing the respective arc; the dark nodes depict the bases.

A notable feature of Evader's strategies is the increasing arc traversal probability with the increasing distance from the Patroller's base. Note, however, that with a very low probability the Evader passes the nodes adjacent to the base or even visits the base itself. This captures the rationale of staying as far from the base as possible most of the time but passing close on selected occasions in order to remain unpredictable and force the Patroller to move across the whole area.

Similar yet more complicated picture is obtained for the case of two bases. Here again the Evader tries to avoid both bases at the maximum distance most of the time yet randomly chooses not do so to confuse the Patroller. The Patroller, on the other hand, concentrates its presence into the area between the two bases as this is the most efficient way to intersect all possible Evader's paths while staying as close to one of the bases as possible. The non-symmetrical probability distribution on arcs is given by a slightly different position of each base on the horizontal axis.

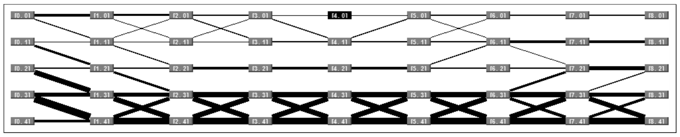
B. Effectiveness on a Real-World Use Case

In order to test the validity of the transit game model and assess its usefulness in practice, we have applied the approach to a very important example of real-world area transit problem – routing shipping traffic through areas affected by maritime piracy.

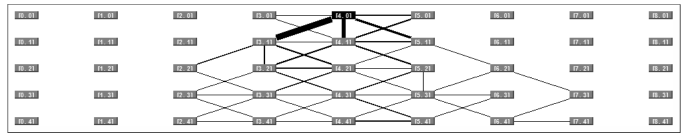
1) Transiting Pirate Waters: The recent surge in maritime piracy presents a serious threat to the international maritime transport system. The problem is currently most acute in the seas around the Horn of Africa off the Somali coast. A large fraction of attacks is targeting ships transiting through the Gulf of Aden – an area of roughly rectangular shape of size 150-by-850 nautical miles. In order to reduce the number of attacks, a narrow International Recommended Transit Corridor (further referenced as *IRTC*) was recently established in order to make it easier to monitor and protect the transit traffic. However, the pirates quickly adapted to the change and, exploiting the predictability of transit routes, continue to successfully attack transiting ships.

We are interested whether randomizing the choice of transit routes can improve the situation. Utilizing the available background knowledge [14], [15], we have mapped the problem to an instance of the transit game as follows. The Evader corresponds to a long-haul vessel transiting the gulf where it can be attacked and possibly hijacked by a pirate (modeled as the Patroller). There is one pirate base corresponding to the main pirate hub situated along the Somali coast (more specifically, to the town of Bosaso) from where the attacks are conducted. In the model, we situated the base into the node that is closest to the real location of the pirate base. The pirates cannot spend unlimited time at sea and have to return back to the base for refueling. The pirates prefer to capture long-haul vessels as close to their bases as possible because the time of the transport back to the base, and thus the probability of a successful rescue of the hijacked vessel, is lower.

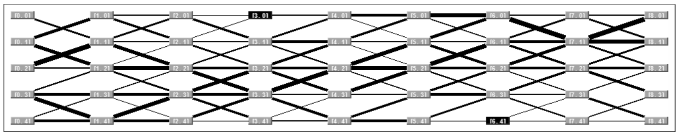
2) Pirate Behavior Model: We do not assume the pirates are employing game-theory to determine the optimum way



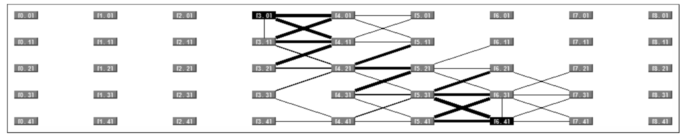
(a) Optimum Evader's strategy for the 1-base game.



(b) Optimum Patroller's strategy for the 1-base game.



(c) Optimum Evader's strategy for the 2-base game.



(d) Optimum Patroller's strategy for the 2-base game.

Fig. 2: Optimum route selection strategies in the flow-based representation. The thickness of the arcs corresponds to the probability of the respective Player traversing the respective arc. Patroller bases are represented by dark nodes.

to travel the sea[5]. We assume that the pirates go to the sea repeatedly and are able to learn from their previous experience. In the simulation, we emulate this learning capability using a simple reinforcement learning scheme based on an arm-acquiring variant of the multi-armed bandit problem [16]. The sea is not discretized – we let the pirate explore the area randomly and for each successful attack, the pirate creates an "arm" with associated probability distribution.

On the onset of a each sailing cycle, the learning algorithm chooses between the exploration of the area or the exploitation of locations where the pirate has previously encountered a transit vessel. In each round the pirate can visit more places, restricted only by the time that it can spend at sea.

3) Transit Routing Methods: We compare three methods for choosing the transit route though the Gulf of Aden:

- IRTC – the currently used method; the vessels travel through a narrow pre-defined IRTC corridor.
- UNIFORM – the transit route is chosen randomly with uniform distribution over all possible paths (leading monotonously from the start to destination – no retreats are allowed).
- GT – randomized route selection strategy corresponding to a Nash equilibrium of the corresponding transit game calculated as described in Section IV.

4) Scenario Setting: The AGENTC testbed[6], developed as part of our larger work on maritime security [4], [5], simulates the movement of both pirates and transit vessels, evaluates all encounters and records the number of attacks and the payoff accumulated.

The evaluation scenario simulates 2000 long-haul vessels (approximately one tenth of yearly traffic volume) traversing the Gulf of Aden in both directions and using routing mechanisms described in the previous section. There are 5 independent non-cooperating pirates in one base trying to attack any vessel traversing the area[7] . The simulation is run long enough for the pirates to fully adapt to the transit traffic.

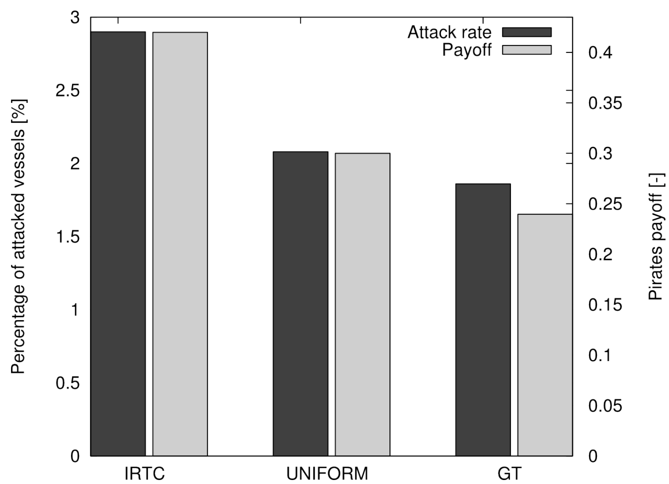


Fig. 3: Percentage of attacked vessel and the accumulated payoff of pirates when using different gulf transit routing strategies.

5) Simulation Results: Figure 3 depicts the overall percentage of attacked vessels transiting the gulf and the overall payoff accumulated by the pirates, computed as $1/dist(inc(x, y), base)$ where $inc(x, y)$ are the GPS coordinates of the attack incident, $base$ are the coordinates of the base and $dist()$ is the Euclidean distance measured in nautical miles.

In the simulation (depicted on Figure 4), the currently used IRTC corridor performs the worst (due to predictability of transiting vessels' trajectories). Randomized route selection strategies performs significantly better with the game-theoretic GT strategy being the best, in particular when the pirate payoff (which reflects the difference in the location of attacks) rather than raw attack count is considered.

6) Limitations: The importance of the simulation results should not be overstressed as several important simplifications have been made. First of all, we do not consider any armed guard vessels escorting the groups of long-haul vessels through the corridor. The pirates are also considered

[5] Even though they could without any impact on the presented results.

[6] `http://agents.felk.cvut.cz/projects/agentc/`

[7] As the pirates operate independently, increasing or decreasing the number of pirates would scale proportionally the accumulated payoff and the attack rate, except for possible saturation effects.

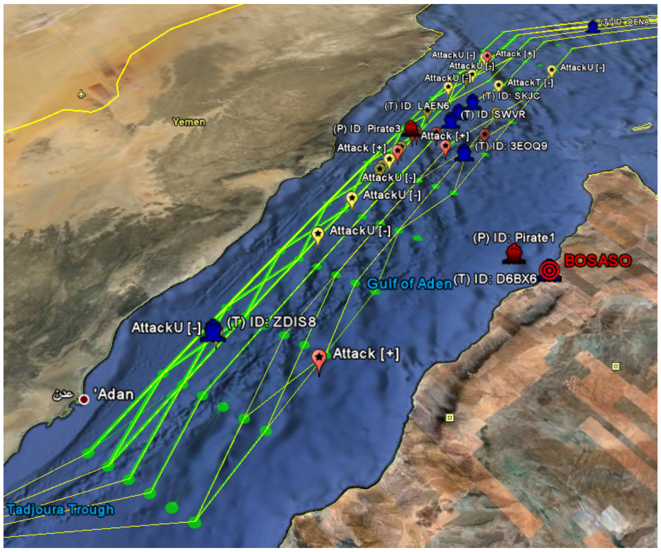


Fig. 4: Simulation of the Gulf of Aden transit in the AGENTC testbed. The route selection graph is laid over the Gulf of Aden; the Patroller's base corresponds to the well-known pirate hub Bosaso. Transiting vessels use a blue ship icon, pirate vessels use red. The red and yellow pins depict past pirate attacks.

rational in exploiting their past experience which is not always true. Nevertheless, the results indicate a promising potential of game-theoretic models in addressing real-world transit security scenarios.

VI. CONCLUSION

We addressed the problem of strategic confrontation between two mobile rational agents – an Evader trying to pass unhindered through an area where a Patroller is trying to intercept its transit. Such confrontation arises in a number of real-world scenarios, including transport through insecure regions or border protection.

Invoking the game-theoretic framework, the problem was formalized as a *transit game*, a zero-sum two-player game in the normal form on a graph; an optimum randomized route selection strategy for the players was then sought as a mixed-strategy Nash equilibrium of the game. A novel feature of our model was the explicit consideration of the limited endurance of patrolling units and the concept of patrol bases to which the units need to regularly return. Because of the game's size, two techniques for reducing computational complexity were employed, first based on an alternative network-flow-based representation of the Evader's strategy space, and the second on iterative single- and double-oracle algorithms for finding Nash equilibria.

Two types of evaluation were carried out. First, we empirically explored the computational complexity of the transit game model by comparing run times of different algorithms for computing transit game's Nash equilibrium. We combined the two above mentioned complexity-reduction approaches and evaluated the oracle algorithms both with the compact network-flow-based and the standard path-based representation of Evader's route selection strategies. While the flow-based representation significantly improves the performance of the single-oracle algorithm, it has negative impact on the double-oracle algorithm, which performs better – and best of all methods evaluated – with the standard path-based representation. This interesting finding is worth further exploration.

In addition, we tested the validity of the transit game model and the effectiveness of the produced route selection mechanism on a real-world use case of ship transit through regions affected by piracy. The evaluation on an agent-based simulation of maritime traffic proved the efficiency of the game-theoretic approach – the randomized route selection strategy performed better both in comparison to the currently used static transit corridors and to a uniform randomized route selection strategy.

VII. ACKNOWLEDGEMENTS

The work presented is supported by the Office of Naval Research project no. N00014-09-1-0537 and by the Czech Ministry of Education, Youth and Sports under Research Programme no. MSM6840770038: Decision Making and Control for Manufacturing III.

REFERENCES

[1] F. Joseph and E. Feron, "Computing the optimal mixed strategy for various ambush games," in *43th Annual Allerton conference on communication, control and computing*, 2005.
[2] M. Salani, G. Duyckaerts, and P. G. Swartz, "Ambush avoidance in vehicle routing for valuable delivery," *Journal of Transportation Security*, vol. 3, no. 1, pp. 41–55, 2010.
[3] H. B. McMahan, G. J. Gordon, and A. Blum, "Planning in the presence of cost functions controlled by an adversary," in *ICML*, 2003, pp. 536–543.
[4] M. Jakob, O. Vaněk, Š. Urban, P. Benda, and M. Pěchouček, "Employing agents to improve the security of international maritime transport," in *Proceedings of the 6th workshop on Agents in Traffic and Transportation (ATT2010)*, May 2010.
[5] ——, "AgentC: Agent-based testbed for adversarial modeling and reasoning in the maritime domain," in *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010) – demotrack*, May 2010.
[6] W. Ruckle, R. Fennell, P. T. Holmes, and C. Fennemore, "Ambushing random walks i: Finite models," *Operations Research*, vol. 24, pp. 314–324, 1976.
[7] S. Gal, *Search Games*. Academic Press, New York, 1980.
[8] M. M. Flood, "The Hide and Seek Game of Von Neumann," *MANAGEMENT SCIENCE*, vol. 18, no. 5-Part-2, pp. 107–109, 1972.
[9] T. D. Parsons, *Theory and Applications of Graphs*. Springer-Verlag, 1976, ch. Pursuit-evasion in a graph, pp. 426–441.
[10] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM J. Comput.*, vol. 21, no. 5, pp. 863–888, 1992.
[11] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1993.
[12] S. Alpern, "Infiltration games on arbitrary graphs," *Journal of Mathematical Analysis and Applications*, vol. 163, pp. 286–288, 1992.
[13] E. Halvorson, V. Conitzer, and R. Parr, "Multi-step multi-sensor hider-seeker games," in *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 159–166.
[14] B. Mpondo-Epo, "Root causes of piracy in somalia," Kuala Lumpur International Conference, On Piracy And Crimes At Sea, May 2009.
[15] R. Gilpin, "Counting the costs of somali piracy," United States Institute of Peace, Tech. Rep., 2009.
[16] R. K. Sundaram, *Studies in Choice and Welfare*. Springer Berlin Heidelberg, 2005, ch. Generalized Bandit Problems, pp. 131–162.