Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

Asynchronous Multi-Robot Patrolling Against Intrusions in Arbitrary Topologies

Nicola Basilico and **Nicola Gatti** and **Federico Villa**
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133, Milano, Italy

Abstract

Use of game theoretical models to derive randomized mobile robot patrolling strategies has recently received a growing attention. We focus on the problem of patrolling environments with arbitrary topologies using multiple robots. We address two important issues currently open in the literature. We determine the smallest number of robots needed to patrol a given environment and we compute the optimal patrolling strategies along several coordination dimensions. Finally, we experimentally evaluate the proposed techniques.

Introduction

The study of autonomous mobile robots to patrol environments under risk of intrusion has received a growing attention in the scientific community during the last few years. In particular, game theoretical approaches were demonstrated very effective in producing randomized strategies (Agmon, Kraus, and Kaminka 2008; Amigoni, Gatti, and Ippedico 2008; Basilico, Gatti, and Amigoni 2009; Paruchuri et al. 2008). The basic idea is to model the patrolling scenario as a two-player non-cooperative game (Fudenberg and Tirole 1991) between a team of patrolling robots and an intruder. The intruder is assumed to observe the strategy of the robots, derive a correct belief, and act on the basis of it by playing its best response. This makes the game a *leader-follower* game (von Stengel and Zamir 2004), where the leader is the team of robots and the follower is the intruder.

The literature provides three main game theoretical approaches for robot patrolling. In (Pita et al. 2008), the authors study the problem of placing a number of checkpoints to secure an environment without considering its specific topology. The problem is modeled as a two-player strategic form game with incomplete information and the leader-follower solution is computed. The absence of a topology makes this approach only applicable to the placement of static checkpoints, while it is unsuitable for a team of robots moving within an environment. In (Agmon, Kraus, and Kaminka 2008), the authors study the problem of patrolling a perimeter with evenly separated synchronized multiple robots (with 'synchronized' we mean that the robots are forced to move together in the same direction). The optimal patrolling strategy is computed as a max-minimization over the detection probabilities. The constraint that the environments are perimeters prevents the use of the approach in general settings. The seminal work dealing with arbitrary environments is presented in (Basilico, Gatti, and Amigoni 2009). The patrolling scenario is modeled as a two-player extensive form game with imperfect information. The environment is characterized by a set of targets connected by a given topology and a single patrolling robot is considered. This approach can be easily extended to study synchronized multi-robots in settings with arbitrary topology (relaxing the constraint that the robots must be evenly separated). Indeed, as shown in (Basilico, Gatti, and Amigoni 2009), the synchronized multi-robot problem can be reduced to a single robot problem. However, in settings with arbitrary topology, it can be easily shown that constraining the robots to be synchronized can lead to strategies that are not optimal.

We formulate the problem of multi-robot patrolling in environments with arbitrary topologies in which robots are not synchronized and we address the two following open issues.

A lower bound over the number of robots. We study the positions of robots such that the intruder cannot have success to strike a target with a probability of one. In game-theoretic related work the robot number is provided as a part of the problem, while the results developed in the pursuit-evasion field to find the smallest number of robots are not directly applicable to our problem because in them the robots ' strategies are provided as part of the problem (instead in our problem they are a part of the solution). We produce a graph-based abstraction of the patrolling setting and we show that the problem of finding the smallest number of uncoordinated robots needed to patrol the environment is equivalent to finding the smallest number of maximal cliques such that each target belongs to at least one clique.

Coordination dimensions. We identify two coordination dimensions for robots' patrolling strategies: the strategy of a robot can or cannot depend on the strategies of the other robots and the environment can or cannot be partitioned. We provide three values per dimension. The choice of the specific values to adopt depends on the problem itself (e.g., whether or not the robots can communicate) and on computational hardness (e.g., some values save computational time providing potentially non-optimal solutions). For each com-

bination of values we provide a mathematical programming formulation and we experimentally compare their effectiveness in terms of computational times and expected utility.

The Multi-Robot Patrolling Problem

A multi-robot patrolling setting is described by a direct graph $G = (V, a, T, v, d, R)$ where V is a set of n vertices to be patrolled and a is a function $a : V \times V \rightarrow \{0, 1\}$, where $a(i, j) = 1$ means that there exists an arc from i to j and $a(i, j) = 0$ otherwise. The set $T \subseteq V$ contains *targets*, i.e., vertices over which the robots and the intruder have some value. Function $v : T \rightarrow \mathbb{R}$ returns the robots' valuations over the targets. We assume that the robots and the intruder have the same preference ordering over the targets (the valuations can be different). With this assumption, the game is *strictly competitive* and can be solved without specifying the intruder's valuations. In particular, the maxmin robots' strategy is the leader-follower equilibrium of the game. Reasonably, most practical applications can be considered strictly competitive. Function $d : T \rightarrow \mathbb{N} \setminus \{0\}$ assigns each target its *penetration time*, i.e., the time needed by the intruder to successfully strike the target. R is a set of homogeneous robots. We call *configuration* a set of $|R|$ elements, each one specifying the position (i.e., a vertex) of a robot. We denote it by $c = (c_1, c_2, \ldots, c_{|R|})$, where $c_i \in V$ is the position of robot $r_i \in R$. We denote by C the configuration space, whose size is $n^{|R|}$. We report in Fig. 1 an example of environment represented as a grid map.

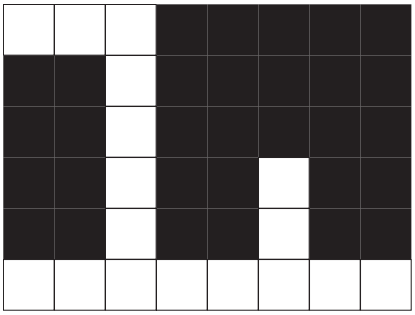

Figure 1: Grid map of a patrolling setting example.

The intruder is assumed to observe the patrollers' strategy before acting. The game develops in turns. At each turn the actions available to the robots are *move*(c, c') where c is the current configuration and $c' \in C$ is such that $a(c_i, c'_i) = 1$ for every $i \in R$. We assume that the robots' strategy is Markovian, depending only on the current configuration. We denote it as a set of probabilities $\{\alpha_{c,c'}\}$, where c is the current configuration and c' is the next one. The intruder's actions can be represented in compact way as *enter-when*(t, c), i.e., wait until the robots are not in the configuration c and then attack target t. The possible outcomes of the game are: *intruder-capture* when the intruder attempts to enter a target t at turn k and at least one robot patrols t by $k + d(t) - 1$, and *penetration-t* when the intruder enters a target t at turn k and no robot patrols t by $k + d(t) - 1$. The robots' utility over the outcomes, denoted by $U_{\mathbf{p}}$, is the cumulative value of the preserved targets, formally, $U_{\mathbf{p}}(\textit{intruder-capture}) = \sum_{t' \in T} v(t')$, while $U_{\mathbf{p}}(\textit{penetration-t}) = \sum_{t' \in T/\{t\}} v(t')$.

Lower Bound over the Number of Robots

We study the problem to find the smallest number of uncoordinated robots such that outcome *intruder-capture* occurs with a strictly positive probability. With a smaller number of robots the intruder would successfully strike a target with a probability of one. We introduce the following definitions. Given a configuration c, we say that target t is *exposed* if the shortest distance between the position of each robot and t is longer than $d(t)$ (i.e., no robot can reach t by $d(t)$ turns). When the robots are in a configuration c and target t is exposed, the intruder can make *enter-when*(t, c) and the probability of *intruder-capture* is zero. We say that a configuration causing a target to be exposed is *infeasible*. Given a number of robots, determining the set of feasible configurations can be easily accomplished by exploiting Dijkstra's algorithm. Basically, we need to find the smallest number of robots such that they can patrol all the targets without ever being in any infeasible configuration. In order to solve this problem, we introduce the following abstraction of G.

We define a undirected multigraph $Q = (T, E, l)$ where T is the set of targets of G, E is a set of edges, and l is a labeling function over E. Given two different targets, say t_i and t_j, we call $p^k_{t_i,t_j} \subseteq V$ the k-th (in general there can be more than one) path between t_i and t_j. Called $e = (t_i, t_j, k)$ the k-th edge connecting targets t_i and t_j, set E is defined as $E = \{(t_i, t_j, k)\}$ such that $|p^k_{t_i,t_j}| \leq \min\{d(t_i), d(t_j)\}$. In other words, $e = (t_i, t_j, k)$ appears in E if a robot, when moving along $p^k_{t_i,t_j}$, can always reach t_i and t_j by their respective penetration times (and then t_i and t_j are never exposed). Function $l : E \rightarrow \wp(T)$ (where $\wp$ is the power set) is defined as follows: given an edge $e = (t_i, t_j, k)$, $l(e) = \{t \in T\}$ such that for all $v \in p^k_{t_i,t_j}$ there exists a path $p_{v,t}$ with $|p_{v,t}| \leq d(t)$. In other words, target t appears in $l(e)$ if a robot, wherever it is along $p^k_{t_i,t_j}$, can reach t by its penetration time (and then t is never exposed). Notice that, by definition, if $e = (t_i, t_j, k)$ then $t_i, t_j \in l(e)$. Finally, we remove from E edges that are Pareto dominated in terms of length and labels, i.e., edges $e = (t_i, t_j, k)$ such that there exists an edge $e' = (t_i, t_j, k')$ with $l(e) \subseteq l(e')$ and $|p^k_{t_i,t_j}| \geq |p^{k'}_{t_i,t_j}|$. Essentially, when a robot moves from t_i to t_j trying to keep a set of targets not exposed, it will prefer, among all the paths preventing exposure, the shortest one. Given G, abstraction Q can be built in polynomial time in the number of vertices V by employing Dijkstra's algorithm. We report in Fig. 2 the abstraction $Q = (T, E, l)$ related to the patrolling setting reported in Fig. 1.

We resort to the graph theoretic concepts of *clique* and *maximal clique*. A *clique* of a graph H is a subgraph $H' \subseteq H$ such that every pair of vertices in H' is connected by an edge. In the case a clique is not a strict subgraph of any other clique, it is said *maximal*. In our problem, we provide a stronger concept of clique, called *labeled clique* and defined as a subgraph $W = (T', E', l)$ of Q such that:

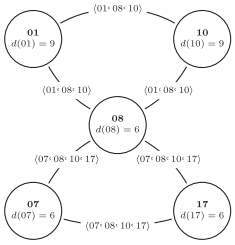

Figure 2: Abstraction related to the setting in Fig. 1.

$$\forall t_i, t_j \in T', t_i \neq t_j, \text{ there exists } k \text{ such that } (t_i, t_j, k) \in E' \quad (1)$$

$$T' \subseteq \bigcap_{e \in E'} \{l(e)\} \quad (2)$$

A labeled clique is maximal when it is not a strict subgraph of any labeled clique. A labeled clique $W = (T', E', l)$ corresponds to a sub-region of G where the targets are T' and the vertices connecting the targets are the paths associated with edges E' (a pair of targets can be connected by only one edge, W being a graph). Given a clique $W = (T', E', l)$, a single robot can patrol all the targets T' without making any target in T' to be exposed (independently of the position of the other robots). Consider Fig. 2, the subgraph with vertices $07, 17$ constitutes a (non-maximal) labeled clique, instead the subgraph with vertices $07, 08, 17$ constitutes a maximal labeled clique. According to graph theory, we define a *clique cover* of a graph H as a set of cliques such that each vertex of H appears in at least one clique. Consider Fig. 2, a (labeled) clique cover is composed of the clique constituted by vertices $01, 10$ and of the clique constituted by vertices $07, 08, 17$. We consider the multi-robot setting and we state the following theorem.

Theorem 1. *Given a patrolling setting $G = (V, a, T, v, d, R)$ and its abstraction $Q = (T, E, l)$, if there exists a labeled clique cover composed by $|R|$ or less labeled cliques, then it is possible to find a patrolling strategy that, when executed, makes no target to be exposed.*

The proof is trivial: given a labeled clique cover of $|R|$ cliques, it is always possible to produce at least a strategy in which at each turn, for every clique W of the cover, at least a robot moves along the paths associated with W. When the robots execute such a strategy, they will be exclusively in feasible configurations. The *vice versa* (i.e., if there does not exist any clique cover composed by $|R|$ or less labeled cliques, then there does not exist any patrolling strategy that makes no target exposed) holds when robots are not coordinated. With some form of coordination, a smaller number of robots can be sufficient. Moreover, it can be easily demonstrated that if Q happens not to be connected, then each one of its connected components represents a separate problem that can be independently addressed. The problem of finding the smallest number of robots needed to patrol a given environment can be formulated as the problem of finding the smallest labeled clique cover. We solve this problem, at first, by finding all the maximal labeled cliques, and, then, by finding the smallest clique cover composed of maximal labeled cliques.

In (Bron and Kerbosch 1973), the authors provide an efficient algorithm to compute all the cliques of a graph. This problem was proven to be NP-complete. The same holds for our problem, the original problem being easily reducible to ours. We extend the work in (Bron and Kerbosch 1973) to compute the labeled cliques of Q. With respect to the original algorithm, we strengthen the constraints over the cliques due to the labels and we deal with the existence of multiple edges connecting the same pair of vertices. The algorithm, based on branch-and-bound, follows.

A solution is constructively generated through a number of steps, where at each step the partial solution obtained so far (called CLI) is extended by adding an admissible *candidate*. We define CLI as a set of candidates and we define a candidate as a pair $\langle t, E_t \rangle$ where $t \in T$ and $E_t \subseteq E$ is a subset of edges connecting t (we cannot define a candidate simply as a vertex, as it is in the original algorithm, because Q is a multigraph and two vertices can be connected by multiple edges; this pushes us to consider edges in the candidates). Given a current partial solution CLI, the admissible candidates are defined as follows. If $CLI = \varnothing$, then all the candidates $\langle t, \varnothing \rangle$s are admissible. Otherwise, we define $\mathcal{T}(CLI)$ and $\mathcal{E}(CLI)$ as the set of vertices and the set of edges respectively belonging to the candidates in CLI, and the admissible candidates are $\langle t, E_t \rangle$s such that: $t \notin \mathcal{T}(CLI)$, E_t contains exactly one edge per pair (t, t') for each $t' \in \mathcal{T}(CLI)$, and, called $CLI' = CLI \cup \{\langle t, E_t \rangle\}$, the graph whose vertices are $\mathcal{T}(CLI')$ and whose edges are $\mathcal{E}(CLI')$ is a labeled clique. Notice that, by construction, each partial solution CLI contains one clique. Given a CLI, we call D the set of admissible candidates. In order to avoid the generation of duplicate cliques and to distinguish between maximal and non-maximal cliques, we use set N containing admissible candidates for the current partial solution CLI whose contribution has been already exploited. Algorithms 1 and 2 summarize our method: function CANDIDATES returns the set of candidates as discussed above, while, with a slight abuse of notation, we denote by $\mathcal{T}(D')$ the set of targets belonging to the candidates in D'. S contains the cliques found so far.

Algorithm 1: FIND_SOLUTION(T, E, l)

```
1 D ← {}
2 CLI ← {}
3 N ← {}
4 for all t in T do
5     D ← D ∪ {⟨t, ∅⟩}
6 S ← CLIQUES(CLI, D, N)
```

Once all the labeled cliques have been found, the problem of finding the smallest labeled clique cover can be easily formulated as a linear programming problem. We call $S_{\max} \subseteq S$ the set of maximal labeled cliques. We limit our search to maximal cliques because their number, even if in the worst case is $3^{\frac{|T|}{3}}$, is usually negligible w.r.t. the total number of the cliques. Given the set $S_{\max}$ of all the maximal labeled cliques returned by Algorithm 1, a clique-target covering

Algorithm 2: CLIQUES(CLI, D, N)

```
if D is empty then
    if N is empty then
        CLI is maximal clique
    else
        CLI is non maximal clique
else
    for all t ∉ N belonging to at least a candidate in D do
        for all ⟨t', E_t'⟩ in D with t' = t do
            CLI' = CLI ∪ {⟨t', E_t'⟩}
            D' = CANDIDATES(CLI')
            S ← S ∪ CLIQUES(CLI', D', N ∩ T(D'))
        N = N ∪ {t}
return S
```

matrix $\{q_{i,j}\}$ (where $q_{i,j} = 1$ if and only if target t_j appears in clique W_i and $q_{i,j} = 0$ otherwise), and called x_i the binary decision variable such that $x_i = 1$ if clique W_i is selected and $x_i = 0$ otherwise, we have $\min \sum_{i:W_i \in S_{\max}} x_i$ subject to $\sum_{i:W_i \in S_{\max}} (x_i q_{i,j}) \geq 1 \ \forall t_j \in T$ and $x_i \in \{0,1\} \ \forall i : W_i \in S$. The value returned by the above optimization problem is the smallest number of robots needed to patrol the environment. From here on, we use 'clique' as 'labeled clique' and we assume that $|R|$ is equal to the lower bound found above. The results that we shall discuss below can be easily extended to the situations in which $|R|$ is larger.

Identifying Coordination Dimensions

With multiple patrolling robots, an issue of paramount importance is the coordination of the robots themselves. We identify two main coordination dimensions with three possible values each. The first dimension refers to the degree of coordination in calculating the patrolling strategy $\{\alpha_{c,c'}\}$. The three values (in decreasing coordination strength) are:

Joint strategy: the strategy of each robot depends on its position and on the position of all the other robots. The solution is $\{\alpha_{c,c'}\}$ where $c, c' \in C$ and contains $O(n^{|R|})$ variables of probability. The strategies are computed centrally, as well as the realization of the strategy at each turn by a centralized controller.

Disjointed strategies: the strategy of each robot depends only on its position. The solution is a collection of $\{\alpha^i_{j,k}\}$, one for each r_i, where $j, k \in V$ and contains $O(|R| \cdot n^2)$ variables of probability. The randomization probabilities over the configurations $\alpha_{c,c'}$s are easily defined as $\alpha_{c,c'} = \prod_{i \in R} \alpha^i_{c_i,c'_i}$. The strategy of each robot is computed centrally keeping into account the strategy of all the other robots. Disjointed strategies provide performances potentially worse than the joint strategy's ones, because the coordination is limited. On the other hand, it requires a lighter computational effort because the number of variables of the solution is drastically smaller.

Separated strategies: as in the previous case, but the computation of $\{\alpha^i_{j,k}\}$ is accomplished by each single robot without considering the strategies of the others. The randomization probabilities $\alpha_{c,c'}$s are defined as in the previous case. This case provides performances potentially worse than the previous two, but assures the minimal computational effort. This is because the multi-robot patrolling is reduced to a set of $|R|$ single-robot independent patrolling problems whose number of variables is $O(n^2)$.

The second dimension refers to the degree of coordination in partitioning the environment. In this case each robot is assigned to a portion of G and is constrained to move within such portion. The three values (in decreasing coordination strength) are:

Full assignment: each robot can potentially move in every vertex of G. We can safely constrain the robots to move along the paths corresponding to the edges of Q such that at each turn the robots are in a feasible configuration. All the robots can potentially patrol all the targets.

Maximal clique assignment: each robot is assigned a maximal clique (i.e., the robot moves along the paths associated with the clique's edges). In this case, some targets (shared by more than one cliques) can be patrolled by multiple robots, while others are patrolled by a single robot. Consider Fig. 2, the maximal clique assignment prescribes that one robot is assigned to the clique constituted of vertices $01, 08, 10$ and the other to the clique constituted of vertices $07, 08, 17$. This case poses some constraints over the robots' strategies reducing the number of possible configurations and thus reducing the number of variables $\alpha_{c,c'}$s of the solution. These constraints make maximal clique assignment performances potentially worse than the full assignment's ones.

Separated assignment: each robot is assigned a clique that has no overlaps with the cliques assigned to the other robots (the cliques constitute a (non-maximal) clique cover). In this case, each target is patrolled by only one robot. In this paper, we do not study the problem of choosing the optimal separated assignment given all the non-maximal clique covers. In our experimental evaluation we enumerate all of them and we select the one such that the patrolling strategy is the best. Consider Fig. 2, with two robots there is only one separated assignment where the two cliques are constituted by the vertices $01, 08, 10$ and $07, 17$ respectively. The two cliques $01, 10$ and $07, 08, 17$ do not constitute a separated assignment because a robot moving between 01 and 10 necessarily patrols also 08 and then this assignment is of maximal clique. The separated assignment reduces the number of possible configurations w.r.t. the previous case, further reducing the number of $\alpha_{c,c'}$s. This makes separated assignment performances potentially worse than the performance of the previous two assignment methods.

Not all the possible combinations of values for the two above dimensions are reasonable. We denote by $\mathcal{D}_i$ the significant ones in Tab. 1. In particular, joint strategy and disjointed strategy with separated assignment can be safely reduced to separated strategy with separated assignment. This is because, when the robots patrol separated portions of G, they do not need to coordinate in choosing the next configuration to move to. Disjointed strategy with full assignment can be safely reduced to disjointed strategy with maximal clique assignment. This is because the optimal patrolling strategy with full assignment when robots' strategies are disjointed prescribes that each robot moves along a single clique (and therefore the assignment is actually of maximal

clique). Otherwise, the robots' strategies being independent on the position of the others, it would be possible that two or more robots are moving on the same clique leaving exposed targets of other cliques. The same holds for separated strategies with full assignment. In the following we discuss how a patrolling setting G can be solved for each $\mathcal{D}_i$.

	full assignment	max. clique assignment	separated assignment
joint strategy	$\mathcal{D}_1$	$\mathcal{D}_2$	–
disjointed strategy	–	$\mathcal{D}_3$	–
separated strategy	–	$\mathcal{D}_4$	$\mathcal{D}_5$

Table 1: Combinations of coordination dimensions.

Joint strategy – full assignment ($\mathcal{D}_1$) In this case, the problem can be formulated as a single-robot problem where the robot moves over the space of configurations instead of the space of vertices. The mathematical programming formulation is an extension of that described in (Basilico, Gatti, and Amigoni 2009). The variable $\gamma_{c,c'}^{h,w}$ gives the probability that patrollers reach in h turns configuration c' starting from configuration c without passing through target w. We denote by C_{-w} the set of all the configurations where w is not occupied by any robot. For brevity, we call $X = U_{\mathbf{r}}(\textit{intruder-capture})$ and $Y_i = U_{\mathbf{r}}(\textit{penetration-}i)$. The mathematical programming formulation (non linear) is:

$$\max u$$

s.t.

$$\alpha_{c,c'} \geq 0 \qquad \forall c, c' \in C \tag{3}$$

$$\sum_{c' \in C} \alpha_{c,c'} = 1 \qquad \forall c \in C \tag{4}$$

$$\alpha_{c,c'} \leq \prod_{r \in R} a(c_i, c_i') \qquad \forall c, c' \in C \tag{5}$$

$$\gamma_{c,c'}^{1,w} = \alpha_{c,c'} \qquad \forall w \in T, c, \in V, c' \in C_{-w} \tag{6}$$

$$\gamma_{c,c'}^{h,w} = \sum_{x \in C_{-w}} \left(\gamma_{c,x}^{h-1,w} \alpha_{x,c'}\right) \qquad \begin{array}{r} \forall h \in \{2, \ldots, d(w)\}, \\ \forall w \in T, c \in C, c' \in C_{-w} \end{array} \tag{7}$$

$$P(c,w) = 1 - \sum_{c' \in C_{-w}} \gamma_{c,c'}^{d(w),w} \qquad \forall w \in T, c \in C \tag{8}$$

$$u \leq P(c,w)(X - Y_w) + Y_w \qquad \forall w \in T, c \in C \tag{9}$$

Constraints (3) and (4) assure that the probabilities $\alpha_{c,c'}$ are positive and are well defined for every configuration c. Constraints (5) assure that $\alpha_{c,c'}$ is zero if the vertex of at least a robot in c' is not adjacent to its vertex in c. Constraints (6) and (7) impose the robots' strategy to be Markovian. Constraints (8) assign $P(c,w)$ the intruder's capture probability when it makes *enter-when*(w,c). The right hand of constraints (9) is the robots' expected utility when the intruder makes *enter-when*(w,c), while u is the actual robots' expected utility. Essentially, we must maximize u.

Joint strategy – maximal clique assignment ($\mathcal{D}_2$) In this case we assign each patroller r a maximal clique Q_r and we introduce the constraint that at every turn of the game r's position should be in a vertex associated with Q_r. The mathematical programming formulation is the same we used in the previous case after discarding from C those configurations that violate the clique assignment constraint.

Disjointed strategy – maximal clique assignment ($\mathcal{D}_3$) This case requires a slightly different mathematical formulation. Given a clique assignment, a robot r patrols a subset of targets, i.e., in its patrol activity it will cover only certain targets. If w is a target, we denote with $R_w \subseteq R$ the set of robots which, according to the given clique assignment, patrol target w. (Obviously, $R_w \geq 1$ for every w). The mathematical programming formulation (non linear):

$$\max u$$

s.t.

$$\alpha_{i,j}^{r} \geq 0 \qquad \forall r \in R, i, j \in V \tag{10}$$

$$\sum_{j \in V} \alpha_{i,j}^{r} = a_r(i,i) \qquad \forall r \in R, i \in V \tag{11}$$

$$\alpha_{i,j}^{r} \leq a_r(i,j) \qquad \forall r \in R, i, j \in V \tag{12}$$

$$\gamma_{r,i,j}^{1,w} = \alpha_{i,j}^{r} \qquad \forall r \in R_w, w \in T, i, j \in V, j \neq w \tag{13}$$

$$\gamma_{r,i,j}^{h,w} = \sum_{x \in V \setminus w} \left(\gamma_{r,i,x}^{h-1,w} \alpha_{x,j}^{r}\right) \qquad \begin{array}{r} \forall h \in \{2, \ldots, d(w)\}, \\ \forall r \in R_w, w \in T, i, j \in V, j \neq w \end{array} \tag{14}$$

$$P(c,w) = 1 - \prod_{r \in R_w} \sum_{j \in V \setminus \{w\}} \gamma_{r,c_r,j}^{d(w),w} \qquad \forall w \in T, c \in C \tag{15}$$

constraints (9)

Constraints (10)-(14) are the analogous to constraints (3)-(9). Constraints (15) assign $P(c,w)$ the intruder's capture probability when it makes *enter-when*(w,c) by considering the strategies of all the robots patrolling target w.

Separated strategies ($\mathcal{D}_4$, $\mathcal{D}_5$) Here the multi-robot problem is reduced to $|R|$ single-robot patrolling problems and each single problem can be solved as discussed in (Basilico, Gatti, and Amigoni 2009). Given an assignment, we solve all the associated single-robot problems. The optimal robots' expected utility related to this assignment is the smallest utility it receives among all the single-robot problems. Since there can be multiple assignments, we need to solve all of them. The optimal robots' expected utility is the largest among all the possible assignments. $\mathcal{D}_5$ presents an anomaly: once we have solved all the single-robot problems, in order to compute the robots' expected utility, we need to find the intruder's best response and, on the basis of this, we can compute the robots' expected utility. This is because the single-robot problems neglect that other robots patrol common targets.

Experimental Evaluation

We implemented our algorithms in C. We used AMPL (Fourer, Gay, and Kernighan 1990) with CPLEX (ILOG Inc. 2010) to solve the linear mathematical programming problems and with SNOPT (Stanford Business Software Inc. 2010) to solve the nonlinear problems. Initially, we experimentally evaluated our algorithm for computing the robot number lower bound. We developed a generator of random connected multigraphs $Q = (T, E, l)$ with $|T| \in [3, 15]$, $|E| \in [|T| - 1, |T|^2]$, and, for every $e = (t_i, t_j, k)$, $l(e) \in \wp(T) \cup \{t_i, t_j\}$. (The computation of $\{\alpha_{c,c'}\}$ with more than 15 settings is hard even with a single robot.) We generated 100 instances of Q and we computed the robot number lower bound. We used a UNIX computer with dual quad-core 2.33GHz CPU and 4GB RAM. For all the instances the computational time was shorter than 2 s.

	vertices	targets	range of $d()$	# max. clique ass.	#sep. ass.
setting1	8	3	$[2, 3]$	1	2
setting2	11	4	$[4, 5]$	1	4
setting3	13	3	$[4, 6]$	1	4
setting4	17	4	$[6, 9]$	1	4
setting5	23	8	$[6, 9]$	1	4

Table 2: Experimental settings.

We produced five patrolling settings with different sizes in terms of the number vertices and targets. We report their characteristics in Tab. 2. We generated 10 instances per setting where the penetration times are uniformly drawn from the corresponding ranges (constraining the robot number lower bound to be two) and the robots' values over the targets are uniformly drawn from $[0, \frac{1}{|T|}]$ in such a way the robots' utilities are normalized, i.e., $U_{\mathbf{r}}(\textit{intruder-capture}) = 1$ and $U_{\mathbf{r}}(\textit{penetration-}i) \in [0, 1]$. We solved these instances with two robots and we report the average computational times in Tab. 3 and the average robots' utilities in Tab. 4. For $\mathcal{D}_5$ we enumerated all the possible assignments and, in Tab. 3, we report the cumulative computational time and, in the parentheses, the longest one; in Tab. 4, we report the utilities with the best assignment. The percentages are calculated w.r.t. $\mathcal{D}_5$, being with lowest coordination degree.

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$
setting1	18.80 s	0.38 s	0.01 s	< 0.01 s	< 0.01 s
setting2	1 h 1 m	29 m	2.95 s	0.03 s	0.06 s (0.01 s)
setting3	1 h 20 m	2 m 48.71 s	0.28 s	0.12 s	0.30 s (0.04 s)
setting4	memory over.	memory over.	8.54 s	1.50 s	4.21 s (0.66 s)
setting5	memory over.	memory over.	284.60 s	12.42 s	36.64 s (8.12 s)
average	1 h (+10^4%)	22 m (+10^3%)	59.01 (+600%)	2.81 s (-65%)	8.25 s (8.12 s)

Table 3: Average computational times.

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	$\mathcal{D}_4$	$\mathcal{D}_5$
setting1	1.00 (+51%)	1.00 (+51%)	0.75 (+13%)	0.53 (-20%)	0.66
setting2	1.00 (+127%)	0.65 (+52%)	0.49 (+9%)	0.40 (-10%)	0.44
setting3	1.00 (+121%)	1.00 (+121%)	0.52 (+21%)	0.35 (-19%)	0.43
setting4	memory over.	memory over.	0.41 (+36%)	0.21 (-30%)	0.30
setting5	memory over.	memory over.	0.35 (+10%)	0.15 (-53%)	0.32
average	1.00 (+117%)	0.88 (+92%)	0.55 (+18%)	0.26 (-19%)	0.46

Table 4: Average robots' expected utilities.

The experimental results confirm that, as the degree of coordination decreases (i.e., as i in $\mathcal{D}_i$ increases), the computational time and the robots' utilities decrease (except for $\mathcal{D}_4$, discussed below). With instances of SETTING4 and SETTING5, $\mathcal{D}_1$ and $\mathcal{D}_2$ cannot be solved because they require more than 4GB RAM. This shows that $\mathcal{D}_1$ and $\mathcal{D}_2$ do not scale with large settings and therefore they are employable only with very small settings. $\mathcal{D}_3$ performs better in terms of expected utilities than $\mathcal{D}_5$, but it requires computational times longer than $\mathcal{D}_5$. However, this result is limited to settings where the number of possible separated assignments is small, otherwise the resolution of all the possible assignments can take a time drastically longer than the $\mathcal{D}_3$'s one. We have modified SETTING1-3 such that their robot lower bounds are 3 and 4 and we solved the related instances with $\mathcal{D}_3$ and $\mathcal{D}_5$. The average computational times with $\mathcal{D}_3$ and $\mathcal{D}_5$ are shorter than those with two robots. This is because the portion of G assigned to each robot is strictly smaller than that with two robots. That is, $\mathcal{D}_3$ and $\mathcal{D}_5$ scale with the number of robots, while the computational time strongly depends on size of the subgraphs assigned to the robots . As a result, $\mathcal{D}_3$ and $\mathcal{D}_5$ are the most appropriate for medium-large settings: the choice between them depends on the setting structure. When both $\mathcal{D}_3$ and $\mathcal{D}_5$ are intractable, $\mathcal{D}_5$ should be supported by an heuristic algorithm to choose the assignment without solving all the possible ones. In all our evaluations the worst $\mathcal{D}_5$ assignment returned an expected utility larger than the one returned by $\mathcal{D}_4$. This shows that patrolling overlapping areas without any strategy coordination negatively affects the expected utility even w.r.t. $\mathcal{D}_5$.

Conclusions and Future Research

We studied the multi-robot patrolling problem with adversaries in environments with arbitrary topologies. Extending the most general model for single-robot patrolling, we provided a game model when there are multiple robots. We studied the problem of computing the smallest number of robots needed to patrol an environment avoiding that an intruder will successfully attack a target with a probability of one, and we identified two coordination dimensions for the patrolling strategies. Our experimental evaluation shows that each dimension presents a different tradeoff in terms of computational times and expected utility making each dimension suitable for particular kinds of patrolling settings.

In the future, we shall study $\mathcal{D}_3$ and $\mathcal{D}_5$ in large settings, providing an heuristic for enumerating $\mathcal{D}_5$, and we shall develop techniques to reduce the space of search based on iterated dominance.

References

Agmon, N.; Kraus, S.; and Kaminka, G. 2008. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, 2339–2345.

Amigoni, F.; Gatti, N.; and Ippedico, A. 2008. A game-theoretic approach to determining efficient patrolling strategies for mobile robots. In *IAT*, 500–503.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, 57–64.

Bron, C., and Kerbosch, J. 1973. Algorithm 457: finding all cliques of an undirected graph. *COMMUN ACM* 16(9):575–577.

Fourer, R.; Gay, D.; and Kernighan, B. 1990. A modeling language for mathematical programming. *Management Science* 36(5):519–554.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. The MIT Press.

ILOG Inc. 2010. http://ilog.com.sg/products/cplex.

Paruchuri, P.; Pearce, J.; Marecki, J.; Tambe, M.; Ordonez, F.; and Kraus, S. 2008. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *AAMAS*, 895–902.

Pita, J.; Jain, M.; Marecki, J.; Ordonez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *AAMAS*, 125–132.

Stanford Business Software Inc. 2010. http://www.sbsi-sol-optimize.com/.

von Stengel, B., and Zamir, S. 2004. Leadership with commitment to mixed strategies. CDAM Research Report LSE-CDAM-2004-01, London School of Economics.