The Multi-Agent Patrolling Problem Theoretical Results about Cyclic Strategies

Fabrice Lauri, Jean-Charles Créput, and Abderrafiaa Koukam

IRTES-SeT
Rue Thiery-Mieg, Belfort, FRANCE

Abstract. Patrolling an environment consists in visiting as frequently as possible its most relevant areas in order to supervise, control or protect it. This task is commonly performed by a team of agents that seek to optimize a performance criterion generally based on the notion of node idleness, that is the period during which a node remains unvisited. For some patrolling strategies, the performance criterion may be unbounded or the classical iterative evaluation algorithm may be ineffective to rapidly provide this performance criterion. The contribution of this paper is fourfold. Firstly we extend the formulation of the classical multi-agent patrolling problem. Secondly we define a large class of multi-agent patrolling strategies, the consistent cyclic patrolling strategies, where every agent may visit some nodes once before ultimately visiting the same set of nodes infinitely often. Idleness-based performance criteria considered in this paper to evaluate such strategies are always bounded. Thirdly we provide theoretical results about the computation time required for evaluating efficiently and accurately any consistent cyclic strategy. Fourthly we propose an efficient and accurate evaluation algorithm of polynomial complexity based on these theoretical results.

Keywords: Multi-agent patrolling, cyclic strategies, theoretical results.

1 Introduction

A patrol is a mission involving a team of several individuals whose goal consists in *continuously* visiting the relevant areas of an environment, in order to efficiently supervise, control or protect it. A group of drones searching for wildfires in order to contribute in the forest conservation, a team of vaccum cleaning robots searching for dirt, postmen on their daily rounds, or a squad of marines securing an area are all examples of patrols. Performing such a task implies that all of the involved members coordinate their actions efficiently.

In this paper, we focus on the multi-agent patrolling problem of a known environment represented by a graph. Techniques solving this problem can be used in numerous applications, including the rescue by robots of people in danger after a disaster [13, 6] or the protection of a territory to face enemy threats [5, 13, 2, 9]. The multi-agent patrolling problem in known environments has been formulated recently [10]. This problem consists in determining a patrolling strategy that minimizes a given performance criterion. A patrolling strategy is made up of several individual patrolling strategies, one for each involved agent. An individual strategy indicates which graph nodes an agent has to visit. It can be defined prior to the patrol or while the agents are patrolling. The performance criterion which evaluates a patrolling strategy is generally based on the notion of node idleness [10], which represents the duration a node remains unvisited. The idleness of a node is zero when an agent is on the node and it increases as soon as the agent leaves the node. In [9, 1, 14, 3, 4, 7, 8, 11, 13], many patrolling strategies have been devised and experimentally validated using an evaluation criteria based on idleness. For example, one of this performance criterion, the worst idleness, consists in determining the largest period a node remains unvisited when agents follow a given patrolling strategy. This criterion is particularly adapted when some geographically distributed information has to be collected very frequently. In this paper, we focus on the framework using the worst idleness performance criterion. As all the state-of-the-art algorithms generating patrolling strategies only yield approximate solutions to this complex problem, they all require an efficient algorithm to accurately evaluate a given strategy. We provide thereafter the theoretical proofs for designing such an evaluation algorithm.

The contribution of this paper is fourfold. Firstly we extend the formulation of the classical multi-agent patrolling problem (see Section 2). Secondly we define a large class of multi-agent patrolling strategies, the cyclic patrolling strategies (see Section 4), where every agent may visit some nodes once before ultimately visiting the same set of nodes infinitely often. One of the main advantages of these cyclic strategies stems from the fact that they can be evaluated in a finite number of iterations. Another advantage is to be represented by a data structure whose size is finite. Thirdly we provide theoretical results about the computation time required for the evaluation of a cyclic strategy to converge (see Section 5). These results can be extended to the evaluation of the strategies studied by Chevaleyre [3], as cyclic strategies are generalizations of single-cycle strategies, partition-based strategies and mixed strategies. Fourthly we propose an efficient and accurate evaluation algorithm, of polynomial complexity, based on these theoretical results (see Section 6). In the remainder of this paper, Section 3 adresses the related works about the multi-agent patrolling problem, and concluding remarks and future research directions are given in Section 7.

2 Problem Formulation

The environment that has to be patrolled consists of a directed connected graph $G = (\mathcal{V}, \mathcal{E}, c)$. $\mathcal{V}$ represents the strategically relevant areas and $\mathcal{E} \subset \mathcal{V}^2$ the means of transport between them. A cost $c(x, y) \in \mathbb{R}$ is associated with any edge $(x, y) \in \mathcal{E}$. It may measure the distance (in meters for example) required to reach node y from node x. The cost function $c : \mathcal{E} \to \mathbb{R}$ satisfies the following properties: $c(x, y) \geq 0$ for any $(x, y) \in \mathcal{E}$ and $c(x, y) = 0$ iff $x = y$.

Let $r < |\mathcal{V}|$ denote the number of agents patrolling graph G. Each agent i is assumed to be located at node $sn_i \in \mathcal{V}$ prior to the patrolling and to possess a movement speed $s_i > 0$ (in m/s for instance). Node sn_i represents the deployment site of agent i. Agent i reaches node y from node x after $\frac{c(x,y)}{s_i}$ units of time (seconds for instance).

With any node x is associated an *instantaneous node idleness*, which represents the time period this node remains unvisited, and a *discount factor* $\gamma_x \in \mathbb{R}_{+*}$ [1], which influences the increase in the node idleness. When any node receives the visit of an agent, its idleness drops to zero. If node x has been left unvisited for a period Δt, its idleness equals $\gamma_x \Delta t$.

Let $\mathcal{I} = (G, r, \overrightarrow{sn}, \overrightarrow{s}, \overrightarrow{\gamma})$ be an instance of the multi-agent patrolling problem, where G is the patrolling graph, r the number of patrolling agents, $\overrightarrow{sn} \in \mathcal{V}^r$ the agent deployment sites, $\overrightarrow{s} \in \mathbb{R}_{+*}^r$ the agent speeds and $\overrightarrow{\gamma} \in \mathbb{R}_{+*}^r$ the discount factors of the nodes. Solving the multi-agent patrolling problem on $\mathcal{I}$ consists in elaborating a coverage strategy $\pi^{\mathcal{I}}$ of graph G by r agents such that any node of G is visited infinitely often. Such a patrolling strategy must optimize a given quality criterion. For the sake of clarity, a multi-agent patrolling strategy will be from now on noted π whenever there is no ambiguity on the instance $\mathcal{I}$.

Let Π be the set of all the multi-agent patrolling strategies $\pi = (\pi_1, \pi_2, \cdots, \pi_r)$ where any individual strategy $\pi_i : \mathbb{N}_* \to \mathcal{V}$ maps a discrete time space into the node set, with $\pi_i(1) = sn_i$. $\pi_i(j)$ denotes the j-th node that agent i has to visit, with $\pi_i(j+1) = x$ only if $(\pi_i(j), x) \in \mathcal{E}$.

We are concerned with determining patrolling strategies that minimize the idleness of any node $x \in \mathcal{V}$. Several criteria have been devised in [10] in order to evaluate the quality of a multi-agent patrolling strategy on a graph. For the sake of theoretical analysis, only the criterion based on the *worst idleness* will be used in this paper. The interested reader can consult Machado *et al.* [10] for other evaluation criteria. Knowing that the chosen criterion, that is the worst idleness of the graph, upper bounds the others ([3]), minimizing it implies minimizing the others.

All of the evaluation criteria can be formulated from the notion of instantaneous node idleness (INI). Assuming the agents follow strategy π on graph G, the INI $I_t^\pi(x) \in \mathbb{R}_{+*}$ of node x at time t is the elapsed *discounted* duration since this node has received the visit of an agent. If node x has been visited at time t by an agent and if Δt is the elapsed time since the last visit at node x, then the instantaneous idleness of node x at time $t + \Delta t$ is given by: $I_{t+\Delta t}^\pi(x) = \gamma_x \Delta t$

Discount factors can be used to set *visit priorities* on nodes. The higher the discount factor, the faster the idleness of the corresponding node grows. By convention, at initial time, $I_0^\pi(i) = 0$, for any strategy π and for any node $i = 1, 2, \cdots, |\mathcal{V}|$.

Evaluating the multi-agent patrolling strategy π using the worst idleness criterion consists in using the following equation:

$$WI^\pi = \limsup_{t \to +\infty} WI_t^\pi \tag{1}$$

[1] $\mathbb{R}_{+*} = \{x \in \mathbb{R} | x > 0\}$

where WI_t^π denotes the *instantaneous worst graph idleness* which is the highest instantaneous node idleness over the set $\mathcal{V}$ of nodes of G at time t, that is: $WI_t^\pi = \max_{x \in \mathcal{V}} \ I_t^\pi(x)$.

Solving the multi-agent patrolling problem thus consists in determining a strategy $\pi^* \in \operatorname{argmin}_{\pi \in \Pi} WI^\pi$ such that for any strategy π, $WI^{\pi^*} \leq WI^\pi$.

3 Related Works

In [10, 9], several multi-agent architectures and multi-agent patrolling strategy evaluation criteria were addressed. [1] improved the best architectures proposed by [9]. They have devised agents able to exchange messages freely and conduct negotiations about the nodes they have to visit. Chevaleyre [3] has formulated the patrolling problem in terms of a combinatorial optimization problem. He first proved that a patrolling strategy involving one agent could be obtained using an algorithm that solves the *Graphical Traveling Salesman Problem*. In this variant of the *Traveling Salesman Problem*, graphs are not necessarily complete. He then studied several possible classes of multi-agent patrolling strategies and showed that they all were able to reach close to optimal performance. In [14], the agents are able to learn to patrol using the Reinforcement Learning (RL) framework. All of the previously described approaches were evaluated in [1] and were compared in several configurations. Lauri *et al.* [7, 8] proposed several Ant Colony Optimization techniques, assuming all of the agents are deployed from the same initial node. Marier *et al.* [11] define the multi-agent patrolling problem as a Generalized Semi-Markov Decision Process (GSMDP). This mathematical model can handle continuous time and uncertainties in the execution of a patrol. Finally, Poulet *et al.* [13] formulate another version of the multi-agent patrolling problem, by introducing priorities on the nodes, metric performance criteria and an agent population whose size is dynamic.

In [12], the authors show that the existing multi-agent patrolling strategy search techniques have several limitations. The lack of study about the flexibility of the proposed approaches or about the efficacity of the computation resources, along with the deterministic aspect of many existing centralized approaches are part of the emphasized limitations. From a theoretical point of view, we believe that other strong limitations of some of the techniques presented above consist in using classes of patrolling strategies whose performance criteria are not well defined or using an inaccurate evaluation algorithm. Indeed, on the one hand, there exist some multi-agent patrolling strategies that have a unbounded worst idleness, for example. Trivially, these can be obtained when a node is not visited infinitely often by at least one agent, or when visits to some nodes become more and more rare. On the other hand, evaluation of patrolling strategies currently relies on an iterative algorithm, called *SEPS* (*Standard Evaluation of Patrolling Strategies*) in the rest of the paper. This algorithm updates the value of the performance criterion (the worst idleness for example) by simulating the agents' movements. It ends after T iterations, but this parameter may have been specified inadequately by the user. Briefly, we will show in Section 5 that, for consistent cyclic strategies especially, there exists T^* such that equation 1 can be rewritten as $WI^\pi = \limsup_{t \to T^*} WI_t^\pi$. An inaccurate value of the worst idleness WI may be found by $SEPS$ when $T < T^*$.

4 Cyclic Multi-Agent Patrolling Strategies

Cyclic multi-agent patrolling strategies are generalizations of single-cycle strategies, partition-based strategies and mixed strategies defined by Chevaleyre [3]. They are particularly adapted to represent tasks that consist in collecting geographically distributed information very frequently and as fast as possible.

A multi-agent patrolling strategy π is cyclic iff each of its individual strategy π_i is parameterized by a tuple (μ_i, l_i) where: $\mu_i = (\mu_i(1), \mu_i(2), \ldots, \mu_i(N_i))$ is a finite sequence of N_i nodes, $l_i \in \{1, 2, \ldots, N_i\}$, $\mu_i(1) = sn_i$, $\mu_i(l_i) = \mu_i(N_i)$, and such that:

$$\pi_i(j) = \begin{cases} \mu_i(j) \text{ for } j < N_i \\ \mu_i(l_i + (j - l_i) \mod (N_i - l_i)) \text{ for } j \geq N_i \end{cases} \tag{2}$$

The individual patrolling strategies in a cyclic multi-agent patrolling strategy are characterized by the existence of a cycle and possibly of a precycle. The patrolling cycle $\mathrm{cyc}(\pi, i)$ of agent i in a cyclic multi-agent patrolling strategy π is the finite sequence of nodes of π_i visited infinitely often by agent i, that is $\mathrm{cyc}(\pi, i) = (\pi_i(l_i), \pi_i(l_i + 1), \cdots, \pi_i(N_i))$. The precycle of agent i in a cyclic multi-agent patrolling strategy π is the sequence of nodes of π_i visited only once by agent i from its deployment site sn_i to the node $\pi_i(l_i)$ beginning its patrolling cycle. Whenever $l_i = 1$, there is no precycle in π_i. A cyclic multi-agent patrolling strategy is consistent if any node of G is visited infinitely often by at least one agent in its patrolling cycle. In the sequel, Π^{cyclic} denotes the set of all the consistent cyclic multi-agent patrolling strategies for a given instance.

Let us consider the graph represented in figure 1 that has to be patrolled by 2 agents both deployed on node 1. Let $\pi = (\pi_1, \pi_2)$ be a patrolling strategy, such that: $\pi_1 = ((1, \mathbf{4,7,8,6,5,4}), 2)$, and $\pi_2 = ((\mathbf{1,2,3,2,1}), 1)$. where the patrolling cycles are written in bold. In this patrolling strategy, agent 1 visits infinitely often the nodes $4, 7, 8, 6$ and 5 : these nodes form its patrolling cycle. The precycle of agent 1 is represented by the path $(1, 4)$. Agent 2 directly performs its

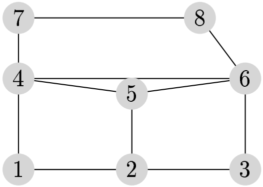

Fig. 1. Example of a patrol graph (8 nodes, 11 edges).

cycle, composed by nodes $1, 2$ and 3, without being entered previously within a pre-cycle.

5 Evaluation of cyclic strategies

Determining an optimal strategy π^* that minimizes equation 1 involves evaluating several strategies $\pi \in \Pi^{cyclic}$ before finding it. The efficiency of an algorithm capable of approximately solving an instance of the multi-agent patrolling problem then strongly depends on the computation time required to evaluate a multi-agent patrolling strategy. The more strategies evaluated in a given time period, the more likely it is that good strategies are determined by an approximate algorithm. This section provides theoretical results about the efficient and accurate evaluation of the worst idleness of cyclic patrolling strategies.

In the sequel, the following notations are used:

- $\mathbb{I}_{\{p\}}$ is the function that returns 1 if the predicate p is satisfied and 0 otherwise.
- $c(\pi_i, x) = \sum_{k=1}^{j-1} c(\pi_i(k), \pi_i(k+1))$ is the cost of the path from the deployment site $\pi_i(1)$ to node $x = \pi_i(j)$.
- $\mu = (\mu(1), \mu(2), \cdots, \mu(N_\mu))$ is a path of N_μ nodes from $\mu(1)$ to $\mu(N_\mu)$, where $\mu(j+1) = x$ only if $(\mu(j), x) \in \mathcal{E}$.
- $c(\mu) = \sum_{k=1}^{N_\mu - 1} c(\mu(k), \mu(k+1))$ is the cost of μ.
- $E(\mu) = \{\mu(k) | 1 \leq k \leq N_\mu\}$ is the set of nodes appearing in a sequence of nodes μ.
- $n^\pi(x) = \sum_{k=1}^{r} \mathbb{I}_{\{x \in E(\text{cyc}(\pi,k))\}}$ is the number of agents visiting node x in their patrolling cycle.
- $n^{\pi_i}(x) = \sum_{j=l_i}^{N_i} \mathbb{I}_{\{x = \pi_i(j)\}}$ is the number of times node x appears in the patrolling cycle of π_i.
- $WI_T^\pi(x) = \limsup_{t \to T} I_t^\pi(x)$ is the worst idleness of node x after a time period T when agents follow the patrolling strategy π.
- $WI^\pi(x)$ is the worst idleness of node x when agents follow π during a time period ensuring its convergence. In other words, $WI^\pi(x) = WI_\infty^\pi(x)$.
- $WI_T^\pi = \limsup_{t \to T} WI_t^\pi$ is the worst idleness of graph G after a time period T when agents follow the patrolling strategy π.

The problem we are faced with here can be formulated as follows:

> Identify the necessary and sufficient conditions that determine, for any $\pi \in \Pi^{cyclic}$, the time period T^π ensuring that $WI^\pi = \limsup_{t \to T^\pi} WI_t^\pi$, that is such that, $\forall T > T^\pi$, $\limsup_{t \to T} WI_t^\pi = \limsup_{t \to T^\pi} WI_t^\pi$.

Let $P_x(n)$ be the following property defined for any node x:

> $P_x(n)$: "The worst idleness $WI^\pi(x)$ of any node $x \in \mathcal{V}$ visited by n agents converges after a time period $T_x(n) = \min\{T_{x,i}\}_{1 \leq i \leq n}$, where $T_{x,i}$ corresponds to the time period agent i needs to visit node x exactly

$n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ times[2].

We are about to prove by induction in theorem 1 that $P_x(n)$ is true for all n. The demonstration of this theorem relies on two lemmas. Lemma 1 gives an upper bound on the worst idleness of any node x visited infinitely often by only one agent. This upper bound is used in lemma 2 to state that the above property $P_x(n)$ is true for $n = 1$.

Lemma 1. *The worst idleness of a node x visited infinitely often by only one agent i, that is such that $x \in E(\mathrm{cyc}(\pi, i))$ and $n^{\pi}(x) = 1$, satisfies:*

$$WI^{\pi}(x) \leq \frac{\gamma_x}{s_i} \max\{c(\mathrm{cyc}(\pi, i)), c(\pi_i, x)\} \tag{3}$$

Proof. Let t_x be the elapsed time until the first visit at node $x \in E(\mathrm{cyc}(\pi, i))$. Then, prior to the actual patrolling, the worst idleness of node x after a time period t_x is equal to:

$$WI^{\pi}_{t_x}(x) = \limsup_{t \to t_x} I^{\pi}_t(x) = \frac{\gamma_x}{s_i} c(\pi_i, x)$$

The worst idleness of x has converged after a time period $T > t_x$ that corresponds to the time span agent i needs to complete its patrolling cycle once and come back to node x. If node x appears only once in $\mathrm{cyc}(\pi, i)$ or if it is the beginning node of the patrolling cycle and it appears exactly two times in π_i, that is $n^{\pi_i}(x) = 1 + \mathbb{I}_{\{x = \pi_i(l_i)\}}$, then its worst idleness satisfies:

$$WI^{\pi}(x) = \begin{cases} WI^{\pi}_{t_x}(x) & \text{if } \frac{\gamma_x}{s_i} c(\mathrm{cyc}(\pi, i)) \leq WI^{\pi}_{t_x}(x). \\ \frac{\gamma_x}{s_i} c(\mathrm{cyc}(\pi, i)) & \text{otherwise} \end{cases}$$

If $n^{\pi_i}(x) > 1 + \mathbb{I}_{\{x = \pi_i(l_i)\}}$, then the above equality becomes a lower inequality. Hence in the general case, $WI^{\pi}(x) \leq \frac{\gamma_x}{s_i} \max\{c(\mathrm{cyc}(\pi, i)), c(\pi_i, x)\}$.

Lemma 2. *The worst idleness of a node x visited infinitely often by only one agent i is ensured to converge after a time span $T_{x,i}$ that corresponds to the time span agent i needs to visit node x $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ times exactly. In other words, $P_x(1)$ is true.*

Proof. Proof of lemma 1 reports that the worst idleness of a node x visited infinitely often by only one agent i is ensured to converge once agent i has completed its patrolling cycle once and has come back to x. If $x = \pi_i(l_i)$, then x appears at least two times in $\mathrm{cyc}(\pi, i)$. In this case, the worst idleness of x has converged once x has been visited exactly $n^{\pi_i}(x)$ times. If $x \neq \pi_i(l_i)$, then x appears at least one time in $\mathrm{cyc}(\pi, i)$. In this case, the worst idleness of x has converged once x has been visited exactly $n^{\pi_i}(x) + 1$ times.

[2] For the sake of clarity, the n agents of indices $i = 1, 2, \ldots, n$ are assumed to visit node x.

Theorem 1. *The worst idleness $WI^{\pi}(x)$ of any node $x \in \mathcal{V}$ visited by n agents converges after a time span $T_x(n) = \min\{T_{x,i}\}_{1 \leq i \leq n}$, where $T_{x,i}$ corresponds to the time period that agent i needs to visit node x exactly $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ times.*

Proof. Let us suppose that $P_x(n)$ is true and prove by induction that $P_x(n+1)$ is also true. $P_x(n+1)$ means that: "The worst idleness $WI^{\pi}(x)$ of any node $x \in \mathcal{V}$ visited by $n+1$ agents converges after a time span $T_x(n+1) = \min\{T_x(n), T_{x,n+1}\}$". If $T_x(n) \leq T_{x,n+1}$, since $P_x(n)$ is true, then $WI^{\pi}(x) = WI^{\pi}_{T_x(n)}(x)$. If $T_{x,n+1} < T_x(n)$, that is if agent $n+1$ visits node x more rapidly than the others, then $WI^{\pi}(x) = WI^{\pi}_{T_{x,n+1}}(x)$. Hence $WI^{\pi}(x) = WI^{\pi}_{T_x(n+1)}(x)$ which states that $P_x(n+1)$ is true assuming $P_x(n)$ is true. As $P_x(1)$ is true (lemma 2) and $\forall n$, $(P_x(n) \Rightarrow P_x(n+1))$, then $P_x(n)$ is true for all n.

The following corollary can be deducted from this theorem:

Corollary 1. *The worst idleness of each node x converges after a time period $T_x(n^{\pi}(x))$, that is:*

$$WI^{\pi}(x) = WI^{\pi}_{T_x(n^{\pi}(x))}(x) = \limsup_{t \to T_x(n^{\pi}(x))} I^{\pi}_t(x) \tag{4}$$

Proof. Each node x is visited infinitely often by $n^{\pi}(x)$ agents. Hence, by using theorem 1, $WI^{\pi}(x) = WI^{\pi}_{T_x(n^{\pi}(x))}(x)$.

We now demonstrate in the following theorem that the worst idleness converges once the worst idlenesses of every node have converged.

Theorem 2. *The worst idleness of graph G when agents follow π converges after a time period T^{π}, that is $WI^{\pi} = \limsup_{t \to T^{\pi}} WI^{\pi}_t$, where:*

$$T^{\pi} = \max_{x \in \mathcal{V}} T_x(n^{\pi}(x)) \tag{5}$$

T^{π} represents the time period required so that the worst idleness of every node of G has converged. T^{π} also corresponds to the time span elapsed so that each agent i visits every node x of its patrolling cycle $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ times.

Proof. Equation 1 can be reformulated as:

$$\begin{aligned} WI^{\pi} &= \max_{x \in \mathcal{V}} \limsup_{t \to +\infty} I^{\pi}_t(x) \\ &= \max_{x \in \mathcal{V}} WI^{\pi}(x) \\ &= \max_{x \in \mathcal{V}} \limsup_{t \to T_x(n^{\pi}(x))} I^{\pi}_t(x) \qquad (6) \\ &= \limsup_{t \to T^{\pi}} \max_{x \in \mathcal{V}} I^{\pi}_t(x) \qquad (7) \\ &= \limsup_{t \to T^{\pi}} WI^{\pi}_t \end{aligned}$$

Equation 6 leads to equation 7 by using equation 5.

Finally, theorem 3 below introduces the stopping criteria of the evaluation algorithm *AECPS* presented in the next section.

Theorem 3. *The following propositions are equivalent:*

- T^π *corresponds to the time span elapsed so that each agent* i *visits every node* x *of its patrolling cycle* $n^{\pi_i}(x) + \mathbb{I}_{\{x \neq \pi_i(l_i)\}}$ *times.*
- T^π *corresponds to the time span elapsed so that each agent* i *visits* M_i *nodes in its cycle, where* $M_i = \sum_{x \in E(cyc(\pi,i))} (n^{\pi_i}(x) + 1)$.

Proof. When agent i has completed its patrolling cycle the first time, each node x has been visited $n^{\pi_i}(x)$ times. The second patrolling cycle allows agent i to visit each node one more time, for a total of visited nodes equal to M_i.

6 Evaluation Algorithm

In this section, we present the algorithm *AECPS* (*Accurate Evaluation of Cyclic Patrolling Strategies*). This algorithm evaluate in a efficient and accurate ways, grounded on the theoretical results presented previously, any cyclic multi-agent patrolling strategy. An empirical comparison between *AECPS* and *SEPS* is given in Section 6.2.

6.1 Algorithm *AECPS*

```
Require: Patrol graph $G$, number of agents $r$, agents' speeds $\overrightarrow{s}$, discount factors $\overrightarrow{\gamma}$, cyclic patrolling strategy $\pi$.
Ensure: Worst idleness $WI$.

$I(x) \leftarrow 0$ for every node $x \in V$
$WI \leftarrow 0$
for every agent $i \in [1; r]$ do
    $cn(i) \leftarrow 1$
    $pn(i) \leftarrow 2$
    $d(i) \leftarrow c(\pi_i(cn(i)), \pi_i(pn(i)))$
    $n(i) \leftarrow \sum_{x \in E(cyc(\pi,i))} (n^{\pi_i}(x) + 1)$
end for
repeat
    $\Delta_t \leftarrow \min_{i \in [1;r]} \frac{d(i)}{s_i}$
    for every node $x \in V$ do
        $I(x) \leftarrow I(x) + \gamma_x \times \Delta_t$
    end for
    $WI \leftarrow \max(WI, \max_{x \in V}\{I(x)\})$
    for every agent $i \in [1; r]$ do
        $d(i) \leftarrow d(i) - \Delta_t \times s_i$
        if $d(i) = 0$ then
```

```
18:       **if** $cn(i) \geq l_i$ **and** $n(i) > 0$ **then**
19:         $n(i) \leftarrow n(i) - 1$
20:       **end if**
21:       Update indices $cn(i)$ and $pn(i)$.
22:       $d(i) \leftarrow c(\pi_i(cn(i)), \pi_i(pn(i)))$
23:       $I(\pi_i(cn(i))) \leftarrow 0$
24:     **end if**
25:   **end for**
26: **until** $n(i) = 0$ for every agent $i \in [1; r]$
```

The data structures used to compute the worst idleness WI of graph G are initialized from line 1 to line 8. These data structures represent: the instantaneous idleness $I(x)$ of each node x, the worst idleness WI of graph G, the index $cn(i)$ of the current node of each agent i, knowing that the current node of agent i is given by $\pi_i(cn(i))$, the index $pn(i)$ of the next node that each agent i must reach, the total number $n(i)$ of nodes that each agent i must visit once it has entered its cycle, and the distance $d(i)$ between the current node of each agent i and its next node. Line 10 computes the minimal period required by one of the agents to reach the next node. Lines 11 to 13 update the instantaneous idlenesses of the nodes. The update of the worst idleness WI is carried out in line 14. From line 16 to line 24, each agent i moves during a period Δ_t on the edge linking its current node to its next node according to its individual patrolling strategy π_i. If some agent i reach its next node (lines 16 and 17), the current and the next nodes (line 21) along with the distance between them (line 22) are updated for ensuring the next agent movement. Lines 18 to 20 decrease the number of nodes that remains to be visited once agent has entered its patrolling cycle. At line 23, the idleness of the current node is set to zero. The agents' movements stop when the convergence of the worst idleness criterion has been reached. This happens when every agent i has visited a total number $n(i)$ (value initialized at line 5) of nodes in its cycle (test at line 26).

6.2 Empirical comparison between *AECPS* and *SEPS*

To emphasize on the importance of having an efficient and accurate evaluation algorithm, we have conducted several experiments by using some of the graphs commonly used by the community [10, 3, 1, 7, 8] for this problem. The same patrolling strategies were evaluated successively by the algorithms *AECPS* and *SEPS*. The results of these experiments are reported in Table 1.

In this table, k is the number of iterations performed by algorithm *AECPS*, T is the number of iterations specified in algorithm *SEPS*, and WI denotes the value of the worst idleness ultimately determined. The durations shown in the table are expressed in seconds and represent the computation times of 1000 successive evaluations of a patrolling strategy. The empirical worst idlenesses that have converged to the theoretical ones are shown in bold.

One may notice that the algorithm *AECPS* determines the theoretical worst idleness in minimum computing time for most of the patrolling strategies. Be-

		AECPS			*SEPS*		*SEPS*		*SEPS*	
					$T = 50$		$T = 100$		$T = 500$	
	# agents	k	WI	Time	WI	Time	WI	Time	WI	Time
Hub 20 nodes 19 edges	5	235	**2344**	0.12	2011	0.028	2252	0.055	**2344**	0.26
	10	219	**2367**	0.158	2228	0.036	2348	0.071	**2367**	0.34
	15	46	**2141**	0.047	**2141**	0.054	**2141**	0.104	**2141**	0.53
MapA 50 nodes 104 edges	5	190	**4026**	0.077	2768	0.021	**4026**	0.042	**4026**	0.20
	10	322	**2520**	0.16	1854	0.026	2469	0.051	**2520**	0.24
	15	434	**2477**	0.25	1725	0.035	2432	0.064	**2477**	0.29
	20	543	**2348**	0.36	1569	0.041	2090	0.075	**2348**	0.35
MapB 50 nodes 69 edges	5	246	**1044**	0.196	600	0.04	981	0.087	**1044**	0.36
	10	429	**836**	0.367	419	0.05	629	0.097	**836**	0.434
	15	500	**728**	0.48	402	0.059	557	0.113	**728**	0.48
	20	634	**583**	0.64	370	0.068	481	0.128	**583**	0.56
Town 330 nodes 522 edges	5	3068	**104634**	9.13	11052	0.13	16328	0.27	56909	1.85
	10	4080	**66819**	11.72	9203	0.15	13679	0.30	34174	1.80
	15	5097	**51548**	14.80	10620	0.16	13046	0.32	26874	1.86
	20	6786	**46692**	19.56	9142	0.17	12397	0.34	22877	1.84

Table 1. Empirical comparison between the proposed algorithm *AECPS* and the standard evaluation algorithm *SEPS*. Computing time is expressed in seconds.

cause of the bound specified in lemma 1, upon which are based all the subsequent theorems, *AECPS* may use a number of iterations greater than necessary. Yet, these results perfectly illustrate the difficulty to master the trade-off between the evaluation accuracy and the computation time in the algorithm *SEPS*, especially when the number of nodes of the graph and the number of agents are high. This trade-off no longer exists when using the algorithm *AECPS*.

7 Conclusion and Future Works

The techniques solving the multi-agent patrolling problem can be used in numerous applications, ranging from the management of information networks, control of mobile and multiple patrolling robots (like in mobile wireless sensor networks) to the enhancement of non-player character behaviors in computer games, to name a few. As the multi-agent patrolling problem may be considered as NP-hard, only approximate solutions can be obtained for large instances. The pioneer work that has been conducted in this article is to deliver rigourous proofs about the computation time required for accurately and efficiently evaluating any multi-agent patrolling strategy belonging to the new introduced class of cyclic patrolling strategies. Cyclic patrolling strategies are generalizations of previously studied patrolling strategies like single-cycle strategies, partition-based strategies and mixed strategies. One research direction that might be followed consists in designing and experimentally validating algorithms that efficiently generate cyclic multi-agent patrolling strategies. Another research direction consists in providing a better bound in the proof to reduce computation time.

References

1. A. Almeida, G. Ramalho, and *al.* Recent Advances on Multi-Agent Patrolling. In *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence*, pages 474–483, 2004.
2. B. Bošanský, V. Lisý, M. Jakob, and M. Pechoucek. Computing Time-Dependent Policies for Patrolling Games with Mobile Targets. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2011.
3. Y. Chevaleyre. Theoretical Analysis of the Multi-Agent Patrolling Problem. In *International Joint Conference on Intelligent Agent Technology*, pages 302–308, 2004.
4. Y. Chevaleyre. *Combinatorial Optimization and Theoretical Computer Science*, chapter The Patrolling Problem: theoretical and experimental results. Wiley, 2007.
5. A. Jiang, Z. Yin, C. Zhang, M. Tambe, and S. Kraus. Game-theoretic Randomization for Security Patrolling with Dynamic Execution Uncertainty. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2013.
6. H. Kitano. RoboCup Rescue : A Grand Challenge for Multi-Agent Systems. In *Proceedings of the 4th International Conference on Multi Agent Systems*, pages 5–12, 2000.
7. F. Lauri and F. Charpillet. Ant Colony Optimization applied to the Multi-Agent Patrolling Problem. In *IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA*, 2006.
8. F. Lauri and A. Koukam. A Two-Step Evolutionary and ACO Approach for Solving the Multi-Agent Patrolling Problem. In *IEEE World Congress on Computational Intelligence, Honk-Kong, Chine*, 2008.
9. A. Machado, A. Almeida, and *al.* Multi-Agent Movement Coordination in Patrolling. In *Proceedings of the 3rd International Conference on Computer and Game*, 2002.
10. A. Machado, G. Ramalho, and *al.* Multi-Agent Patrolling : an Empirical Analysis of Alternatives Architectures. In *Proceedings of the 3rd International Workshop on Multi-Agent Based Simulation*, pages 155–170, 2002.
11. J.-S. Marier, C. Besse, , and B. Chaib-draa. A Markov Model for Multiagent Patrolling in Continuous Time. In *International Conference on Neural Information Processing: Part II*, pages 648–656, 2009.
12. D. Portugal and R. Rocha. A Survey on Multi-robot Patrolling Algorithms. In *Technological Innovation for Sustainability*, volume 349 of *IFIP Advances in Information and Communication Technology*, pages 139–146. 2011.
13. C. Poulet, V. Corruble, A. Seghrouchni, and G. Ramalho. The Open System Setting in Timed MultiAgent Patrolling. In *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011.
14. H. Santana, G. Ramalho, and *al.* Multi-Agent Patrolling with Reinforcement Learning. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1122–1129, 2004.