Theoretical Analysis of the Multi-agent Patrolling Problem

Yann Chevaleyre

Lamsade - university paris 9,
place du Marchal de Lattre de Tassigny
75775 PARIS CEDEX 16
yann.chevaleyre@lamsade.dauphine.fr

Abstract

A group of agents can be used to perform patrolling tasks in a variety of domains ranging from computer network administration to computer wargame simulations. The multi-agent patrolling problem has recently received growing attention from the multi-agent community, due to the wide range of potential applications. Many algorithms based on reactive and cognitive architectures have been developed, giving encouraging results. However, no theoretical analysis of this problem has been conducted. In this paper, various classes of patrolling strategies are proposed and compared. More precisely, these classes are compared to the optimal strategy by means of a standard complexity analysis.

1. Introduction

To patrol is literally the act of walking or traveling around an area, at regular intervals, in order to protect or supervise it. This task is by nature a multi-agent task and there are a wide variety of problems that may reformulate as particular patrolling task. As a concrete example, during the development of the Artificial Intelligent component of an interactive computer wargame, one may face the problem of coordinating a group of units to patrol a given rough terrain in order to detect the presence of “enemies”. The quality of the strategies used for patrolling may be evaluated using different measures. Informally, a good strategy is one that minimizes the time lag between two passages to the same place and for all places. Beyond simulators and computer games, performing this patrolling task efficiently can be useful for various application domains where distributed surveillance, inspection or control are required. An example of such a task is the identification of objects or people in dangerous situations that should be rescued by robots [3].

Recently, many different architectures of multi-agent systems have been proposed and evaluated on the patrolling problem [1], giving encouraging results. In particular, it was shown in [1] that very simple strategies implemented through reactive agents with nearly no communication ability could achieve impressive results. Also, some authors of these papers suggested that an approach based on partitioning the territory such that each agent patrols in its own region could also work well.

This paper proposes a theoretical analysis of the patrolling problem addressing the following issues : do the existing algorithms generate optimal strategies ? Are there efficient algorithms generating near-optimal strategies ? Are patrolling strategies based on partitioning the territory good ? What if the agents all follow the same circuit one behind another ? To answer these questions, strategies are proposed in this paper which are close to strategies humans could build to patrol over a territory. They need a certain amount of communication/synchronisation between agents. Which multi-agent architecture should be chosen, how to implement the algorithms will not be our purpose here, and the reader interested in these questions can refer to [7]. Our primary purpose is to understand more deeply the the patrolling problem.

To do this study, we will use the formal definition of the patrolling problem introduced in [1], presented in section 2. After a short overview of the previous work, a first class of strategies referred to as “cyclic strategies” is introduced in section 4, and it is shown that there exists an $O(n^3)$ algorithm generating such a strategy close to optimality. In section 5, strategies based on partitioning will be analyzed and compared to the previous. In section 6, the results obtained in [1, 8] will be compared to ours. Finally, section 7 draws some conclusions and indicates directions for future work.

2. The patrolling task

Many tasks (such as rescuing, tracking, detecting, etc...) can involve some sort of patrolling, which may exhibit slightly different characteristics according to the domain and circumstances. It is then necessary for our study to have a more precise definition of patrolling.

In [1], it was shown that in many applications of the patrolling problem, the territory could be represented by an undirected graph. Given such a graph, the patrolling task refers to continuously visiting all the graph nodes so as to minimize the time lag between two visits. The edges may have different associated lengths (weights) corresponding to the real distance between the nodes.

From now on, the graph representing the territory will be referred to as $G(V, E)$, where $V = \{1 \dots n\}$ is the set of nodes and $E \subseteq V^2$ the set of edges of G. To each edge (i, j) will correspond a weight c_{ij} representing the distance between nodes. Note that the graph will be assumed to be metric[1] The time taken by an agent to move across an edge (i, j) will be exactly c_{ij}. At time 0, r agents will be positioned on nodes of G. When the patrolling task starts, agents will move simultaneously around the nodes and edges of the graph according to a predetermined strategy.

Definition. The *strategy* of an agent is a function $\pi : \mathbb{N} \rightarrow V$ such that $\pi(j)$ is the j^{th} node visited by the agent. A *multi-agent strategy* $\Pi = \{\pi_1 ... \pi_r\}$is simply defined as a set of r single-agent strategies.

Knowing the strategy of agent k, we can now predict at what time a given node will be visited. For example, agent k will visit node $\pi_k(0)$ at time 0. It will also visit node $\pi_k(j)$ at time equal to the weight of the path $\pi_k(0) \dots \pi_k(j)$, which is $\sum_{i=0}^{j-1} c_{\pi_k(i)\pi_k(i+1)}$. For the sake of clarity, the weight of a path $s_0 ... s_m$ will be from now on noted $c(s_0 ... s_m)$. Also, the weight of a set of edges E' will be noted $c(E')$.

Our main goal is to find good patrolling strategies. We thus need an evaluation criterion. We will use *idleness* criteria introduced in [1].

Definition. Let r agents patrol a graph G according to a multi-agent strategy Π. The *idleness* of a node i at time t is the amount of time elapsed since that node has received the visit of an agent. The idleness of all nodes at the beginning of the patrolling task is set to 0. The *worst idleness* is the biggest value of the idleness occurred during the entire patrolling process for all nodes. It is noted $WI_\Pi(G)$ or just WI_Π when there is no ambiguity.

Figure 1 illustrates the calculation of the idleness and worst idleness for a single agent and a very simple graph. Notice that strategies are not necessarily finite over time, and that agents can patrol for ever on the graph. Of course, if strategies are tested in a simulator, the patrolling task will have to be stopped at one time. Thus, the worst idleness measured during a simulation will be an approximation of the true value.

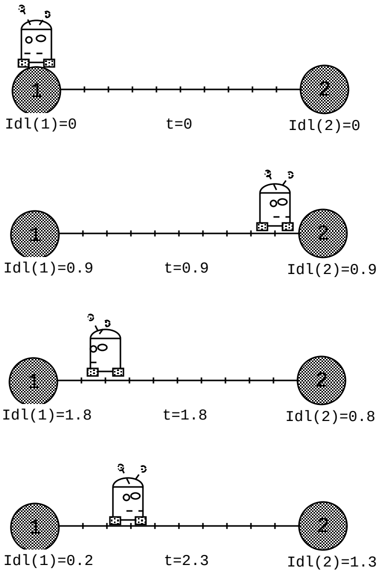

Figure 1. Agent patrolling on a graph made of two nodes and an edge of weight 1. Its strategy is $\pi = 1, 2, 1, 2, 1, \dots$. The idleness of the nodes are shown at various moments. Here, $WI_\pi = 2$.

In this work, we chose the worst idleness criterion among the various criteria defined in [1] mainly for simplicity reasons : compared to the average idleness criterion, which is the other main criterion defined in [1], the theoretical analysis will be easier than with our criterion. In addition, the former is upper bounded by the latter. Thus, minimizing worst idleness will also lead to a small average idleness.

3. Previous Work

Most work on the patrolling problem as formulated above has been done by Machado *et al.* in [1]. In their article, they proposed several multi-agent architectures varying parameters such as agent type (reactive vs. cognitive), agent communication (allowed vs. forbidden), coordination scheme (central and explicit vs. emergent), agent perception (local vs. global), decision-making (random selection vs. goal-oriented selection). For each agent, the choice of the next node to visit is basically influenced by two fac-

[1] graphs in which the triangular inequality is not violated, that is, given three nodes i, j, k connected by edges, we have $c_{ij} + c_{jk} \geq c_{ik}$.

tors: (1) node idleness, which can shared (generated by all agents) vs. individual (corresponding a single agent visits); (2) field of vision, which can be local (agent's neighborhood) or global (entire graph). The experiments they conducted showed two interesting facts: first, agents moving randomly achieved very bad results. Second, agents with absolutely no communication ability which strategies consisted in moving towards the node with the highest idleness performed nearly as well as the most complex algorithm they implemented. In the experiments section, we will the compare our approach to the two most efficient strategies of [1] : (a) The Conscientious Reactive strategy, in which the next node an agent visits is the one with the highest individual idleness from its local neighborhood; (b) The Cognitive Coordinated strategy, in which the next node an agent chooses to visit is the one with the highest idleness from the whole graph, according to the suggestions given by a central coordinator agent. To reach the chosen node, agents move through the shortest path leading to this node. The coordinator is responsible for avoiding that more than one agent chooses the same next node.More recently, a new approach based on reinforcement learning (RL) techniques was developped in [8]. In their paper, Santana et al showed that a simple Q-learning algorithm could be used to train agents to solve the patrolling problem efficiently. The results of their algorithm will also be compared to ours in the experiments section.

The strategies proposed in this paper are very different from those proposed in previous work. They have not been developped out of a specific architecture, or a communication scheme between agents. In fact, the implementation issues of our strategies will not be tackled here. However, an important work has been done concerning the implementation of the cyclic strategies in a multi-robot environment [7].

The next section proposes a new class of strategies which we called "cyclic strategies".

4. The cyclic strategies

In this section, we will first show how cycles and closed-paths can be used to create efficient single-agent patrolling strategies. Then, an extension to the multi-agent case will be proposed, and the resulting strategies will be shown to be near-optimal.

4.1. Patrolling with a single-agent

Consider a single agent patrolling over an area. The simplest strategy which comes to mind would be to find a cycle covering all the area, and then to make the agent travel around this cycle over and over. Applied to our case in which areas are represented by nodes in a graph, the notion of cycle is too restrictive. In fact, in the graph-theory terminology, a cycle is a path starting and ending on the same node and covering each edge at most once. However, for some graphs such as the one on figure 2, there does not exist a cycle covering all nodes. Instead of using cycles, we will have to use closed-paths, which are paths starting and ending on the same node and covering edges possibly more than once. A closed-path is usually represented by a list of nodes, beginning and ending with the same node. On figure 2 for example, the closed-path $s = 1, 3, 2, 1, 4, 5, 6, 4, 1$ covers all nodes, and turning over s indefinitely seems to be a good strategy for a single-patrolling problem. Single-agent strategies consisting in traveling along a closed-path indefinitely will be called *single-agent cyclic strategies*. The strategy of agent on figure 1 is a single-agent cyclic strategy based on the closed-path $1, 2, 1$. The bottom agent on the right of figure 3 is also following a cyclic strategy based on the closed-path $4, 5, 6, 5, 4$.

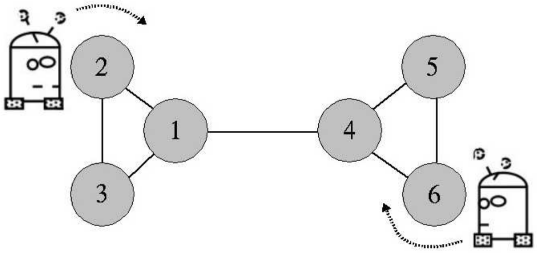

Figure 2. Example of multi-agent cyclic-based strategy. Strategy of these robots are $\pi_1 = 2, 1, 4, 5, 6, 4, 1, 3, 2, 1, 4 \ldots$ and $\pi_2 = 6, 4, 1, 3, 2, 1, 4, 5, 6, 4, \ldots$

Before extending this idea to the multi-agent case, let us concentrate on two questions: 1) which closed-path should be chosen in a single-agent patrolling problem, and 2) are strategies based on closed-paths optimal with a single-agent ? To answer the first question, let us notice that the time taken for a single agent patrolling around a closed-path to visit a node twice is at most equal to the length of this closed-path. Therefor, with a single agent patrolling around a closed-path s, the worst idleness will be equal to the length of s. Finding the smallest closed-path covering all nodes will thus result in the best possible strategy *among all single-agent cyclic strategies*. Let us show how this problem relates to the well known *Traveling Salesman Problem (TSP)*,

The traveling salesman problem is a combinatorial optimization problem which was originally formulated as follows: given a set of cities on a map, find the shortest cycle such that each city is visited *exactly once*. This problem was soon extended to metric graphs, and became: given a metric graph $G(V, E)$ with edges weighted according to c_{ij}, find the shortest closed-path such that each node is visited *at most once* [6]. Christofides [5] proposed an algorithm which generates in $O(n^3)$ a cycle (or a closed-path for the metric graphs case) whose length is less than $\frac{3}{2}$ times the shortest cycle (or closed-path). From now on, S_{TSP} will denote the closed-path being the optimal solution to the TSP, whereas S_{Chr} will denote the closed-path obtained by the algorithm of Christofides. The following holds:

Theorem 1. For a single agent, the optimal strategy in terms of worst idleness is the cyclic-based strategy based on S_{TSP}.

We already knew that the cyclic strategy based on S_{TSP} was the best possible strategy *among all single-agent cyclic strategies*. Note that this theorem states it is also the best strategy *among all possible single-agent strategies*. Due to the lack of space, the proof of this theorem which can be found in [4] will be omitted here. Note that this result will only be needed in section 5 to demonstrate theorem 2. An immediate corrolary of this theorem is that the worst idleness of a single-agent cyclic strategy based on S_{Chr} will be at most $\frac{3}{2}$ times the worst idleness of the optimal strategy. In addition, the S_{Chr} closed-path is calculable in $O(n^3)$, whereas the calculation of S_{TSP} is NP-complete. In practice, there exists many efficient algorithms which can efficiently build closed-paths very close to S_{TSP} using heuristic algorithms.

4.2. Extending to multi-agent case

One way to extend single-agent cyclic strategies to the multi-agent case is to arrange agents on the same closed-path such that when they start moving through that path all in the same direction, they keep an approximately constant gap between them. This leads to the following definition:

Definition. Let $S = s_0 ... s_m$ be a closed-path visiting all nodes of a graph G. The strategy $\Pi = \{\pi_1 ... \pi_r\}$ is a *multi-agent cyclic strategy* based on S iff there exists $d_1 ... d_r \in \mathbb{N}$ such that $\pi_i(k) = s_{(k+d_i) \mod m}$. The set of all multi-agent cyclic strategies will be referred to as $\mathbf{\Pi_{cyclic}}$.

Figure 2 illustrates this with a 2-agent cyclic strategy based on the closed-path $2, 1, 4, 5, 6, 4, 1, 3, 2$, such that $d_1 = 0$ and $d_2 = 4$.

How does the worst idleness evolve when the number of agents grows ? The following lemma shows that if r agents follow a multi-agent cyclic strategy Π and if $d_1 ... d_r$ are well chosen, then the worst idleness will be approximatively r times lower than the worst idleness obtained by a single agent patrolling around the same closed-path.

Lemma 1. *Let $S = s_0 \dots s_m$ be a closed-path covering each node of G such that there exists a node $x \in V$ covered exactly once by S. Let $l = c(S)$ be the length of the closed-path. There exists a multi-agent cyclic strategy $\Pi = \{\pi_1 \dots \pi_r\}$ based on this closed-path such that $\frac{l}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq WI_\Pi \leq \frac{l}{r} + \max\{c_{ij} \mid (i, j) \in E_S\}$. Here, E_S refers to the set of edges present in S. Note that l is also the worst idleness of the single-agent cyclic strategy based on S.*

Proof. Let $S = s_0 ... s_m$, the closed-path covering G. Consider two agents moving around S such that at time 0, agent 1 is positioned on node s_0 and agent 2 is on node s_d. Let $l(i, j) = \sum_{k=i}^{j-1} c_{s_k s_{k+1}}$. Thus, $l(0, d)$ is the time taken by agent 1 starting at s_0 to reach node s_d. Note that at any time t, the node visited by agent 2 will also be visited by agent 1 at time $t + l(0, d)$. Of course, a node visited at time t by agent 1 will be visited by agent 2 at time $t + l(d, m)$. Therefor, we have $WI_{\{\pi_1, \pi_2\}} \leq \max\{l(0, d), l(d, m)\}$. In addition, if agent 2 visits node x at time t_x, because node x is present only once in S, the next visit of the an agent to node x will occur exactly at $t_x + l(0, d)$. Thus $WI_{\{\pi_1, \pi_2\}} = \max\{l(0, d), l(d, m)\}$.

Let us generalize this to r agents. We get $WI_\Pi = \max\{l(0, d_1), l(d_1, d_2), \dots, l(d_{r-1}, m)\}$. We will now have to choose the values of d_k such that the worst idleness is as low as possible. By setting each d_k to the greatest integer verifying $l(0, d_k) \leq k \times \frac{l(0,m)}{r}$, we get $d_0 = 0$ and $d_r = m$. We can now write $WI_\Pi = \max_{k=0..r-1} l(d_k, d_{k+1})$. Let us now calculate upper and lower bounds to $l(d_k, d_{k+1}) = l(0, d_{k+1}) - l(0, d_k)$ by showing that $k \times \frac{l(0,m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(0, d_k) \leq k \times \frac{l(0,m)}{r}$, thus $(k + 1) \times \frac{l(0,m)}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(0, d_{k+1}) \leq (k + 1) \times \frac{l(0,m)}{r}$. Combining these two equations, we get for all k : $\frac{l}{r} - \max\{c_{ij} \mid (i, j) \in E_S\} \leq l(d_k, d_{k+1}) \leq \frac{l}{r} + \max\{c_{ij} \mid (i, j) \in E_S\}$. Because $WI_\Pi = \max_{k=0..m-1} l(d_k, d_{k+1})$, the lemma follows.

4.3. Optimality of cyclic strategies

We have shown previously that cyclic strategies based on S_{TSP} were optimal for single agents. We will now show a similar result for the multi-agent case. From now on, *opt* will refer to the worst idleness of the optimal strategy.

Theorem 2. *Let G=(V,E) a connected metric graph and let r agents patrolling on it. Let Π_{Chr} be the multi-agent cyclic strategy based on S_{Chr}. We have $WI_{\Pi_{Chr}} \leq 3 \times$ opt $+ 4 \times \max_{ij}\{c_{ij}\}$.*

Note that if all edges of G have the same length, then *opt* $\geq \max_{ij}\{c_{ij}\}$, and therefor, $WI_{\Pi_{Chr}} \leq 7 \times$ *opt*. To prove this theorem, we first need to demonstrate the following lemma.

Lemma 2. *For any multi-agent strategy Π, there exists multi-agent strategy $\Pi' = \{\pi'_1, ..., \pi'_r\}$ such that for each $k \in 1..r$, each strategy π'_k consists in moving through a* *path $s_1^k \dots s_{m_k}^k$ of m_k nodes forwards and backwards indefinitely, and such that $WI_{\Pi'} \leq 2\max_k c(s_0^k \dots s_{m_k}^k) \leq 2 \times WI_\Pi$.*

Proof. Consider the multi-agent strategy $\Pi = \{\pi_1 ... \pi_k\}$. Let $s_1^k \dots s_{m_k}^k$ be the list of nodes visited by agent k according to strategy π_k between time $t = 0$ and time $t = WI_\Pi$. Clearly, $\cup_{k=1..r}\{s_1^k, \dots, s_{m_k}^k\} = V$. Let $\pi_k' = s_1^k \dots s_{m_k}^k, s_{m_k-1}^k, \dots, s_1^k, \dots$ be the strategy consisting in moving forwards and backwards through $s_1^k \dots s_{m_k}^k$. Time taken by agent k using strategy π_k' to visit a node twice is at most $2 \times c(s_1^k \dots s_{m_k}^k)$. Thus, $WI_{\Pi'}(G) \leq 2\max_k c(s_1^k \dots s_{m_k}^k) \leq 2 \times WI_\Pi(G)$.

Now the theorem can be proven. The preceding lemma will be used on the optimal strategy to build a set of paths, which will be combined to form a closed-path leading to an near optimal strategy.

Proof of the theorem. Consider an optimal strategy Π. We can tell with lemma 2 that there exists a strategy $\Pi' = \{\pi_1', ..., \pi_r'\}$ in which each strategy π_k' consists in moving through a path $s_1^k \dots s_{m_k}^k$ and such that $\frac{WI_{\Pi'}}{2} \leq \max_k\{c(s_0^k ... s_{m_k}^k)\} \leq opt$. Let us first show that there exists a closed path S covering all nodes and such that $\frac{c(S)}{2r} \leq \max_k\{c(s_0^k ... s_{m_k}^k)\} + \max_{ij}\{c_{ij}\}$.

Let U be the set of edges present in these paths; because these paths cover all nodes of G, by adding at most $r - 1$ edges to U, we obtain a set U' of edges such that the graph (V, U') is connected. Clearly, there exists a tree $T \subseteq U$ covering all nodes. In addition, $c(T) \leq c(U') \leq \sum_{k=1}^r c(s_0^k ... s_{m_k}^k) + (r-1) \times \max_{ij}\{c_{ij}\}$. There exists a closed-path S covering all nodes obtained by exploring the tree T in a depth-first manner (thus twice each edge), such that $\frac{c(S)}{2r} = \frac{c(T)}{r} \leq \max_k\{c(s_0^k ... s_{m_k}^k)\} + \max_{ij}\{c_{ij}\}$. Remember that the closed-path S_{Chr} obtained by Christofides algorithm is at most $\frac{3}{2}$ times longer than the shortest closed-path covering G. Thus, we have: $c(S_{Chr}) \leq \frac{3}{2}c(S) \leq 3r \times \max_k\{c(s_0^k \dots c_{m_k}^k)\} + 3r \times \max_{ij}\{c_{ij}\} \leq 3r \times opt + 3r \times \max_{ij}\{c_{ij}\}$.

With lemma 1, we can generate from S_{Chr} a multi-agent cyclic-based strategy Π_{Chr} such that $WI_{\Pi_{Chr}} \leq \frac{c(S_{Chr})}{r} + \max_{ij}\{c_{ij}\} \leq 3.opt + 4.\max_{ij}\{c_{ij}\}$.
It is clear that because of the $\max_{ij}\{c_{ij}\}$ in the theorem, cyclic strategies will not be suited for graphs containing long edges. For this reason, let us study another kind of strategies, which we will call *partition-based strategies.*

5. Partition-based strategies

Another very intuitive way to make r agents patrol over a territory would be make a partition of this territory into r disjoint regions, and to have each agent patrolling inside a single region.

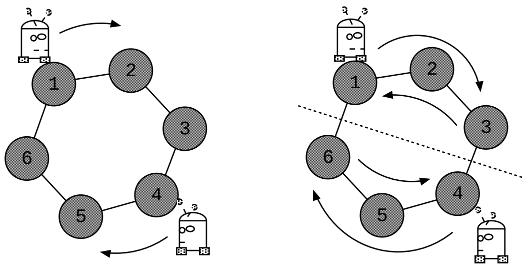

Figure 3. On the left: cyclic strategy $\Pi_{cyc} = \{\pi_{cyc1}, \pi_{cyc2}\}$ and to the right: partition based strategy $\Pi_P = \{\pi_{P_1}, \pi_{P_2}\}$ with two agents. We have $\pi_{cyc1} = 1,2,3,4,5,6,1,2,3\dots$ and $\pi_{cyc2} = 4,5,6,1,2,3,4,5,6,1\dots$. Also, $\pi_{P_1} = 1,2,3,2,1,2,\dots$ and $\pi_{P_2} = 4,5,6,5,4,5,\dots$. Thus, $WI_{\Pi_{cyc}} = 3$ and $WI_{\Pi_P} = 4$.

From now on $P = \{P_1 ... P_r\}$ will denote a partition of V. Thus, $P_1 \cup \dots \cup P_r = V$ and $P_i \cap P_j = \emptyset$. Also, $\{G_1 \dots G_r\}$ will refer to the subgraphs induced by the partition. Thus, $G_i = (V \cap P_i, E \cap (P_i \times P_i))$.

Definition. A multi-agent strategy $\Pi = \{\pi_1 \dots \pi_r\}$ is said to be based on a partition P iff each agent k following strategy π_k visits the nodes of a single region of P. The class of all strategies based on partition P is referred to as $\mathbf{\Pi_P}$.

The previous definition does not specify what moves agents should make in their own region. Given a partition P, how should agents behave inside their region ? The following lemma brings an answer.

Lemma 4. For $k \in 1..r$, let π_k be the single-agent cyclic strategy based on the TSP of G_k. Then, $\Pi = \{\pi_1 ... \pi_r\}$ is the optimal strategy based on partition P.

Proof. For any strategy $\Pi = \{\pi_1 ... \pi_r\}$ based on partition P, each node of G will not be visited by more than one agent. Thus, $WI_\Pi(G) = \max_k\{WI_{\pi_k}(G_k)\}$. Therefor, a set of r optimal single-agent strategies in $P_1 ... P_r$ is an optimal multi-agent partition-based strategy. Theorem 1 stated that single-agent cyclic strategies based on TSP were optimal. Thus, by combining such single-agent strategies, we will obtain an optimal partition-based strategy.

5.1. Comparing cyclic and partition-based strategies

Figure 3 illustrates how both strategies perform on a circular graph with two agents. On this example, the cyclic strategy wins. Consider now figure 2. On this figure, if the distance between node 1 and 4 was huge, the cyclic strategy would be disastrous, as both agents would spend most time crossing the edge $(1,4)$. However, a strategy based on the partition $\{\{1,2,3\},\{4,5,6\}\}$ would not have this problem. Thus, it seems that on graphs having "long corridors" connecting two sub-graphs as on figure 2, partition-based strategies could be better.

Most of the proofs will be omitted in this section, due to lack of space. However, the complete proofs can be found in [4]. The following theorem compares the values of the worst idleness of the optimal cyclic strategy and the optimal partition-based strategy. Here $opt_{\Pi_{Cycle}}$ and opt_{Π_P} refer to the worst idleness of the optimal cyclic strategy and of the strategies based on partition P.

Theorem 3. $opt_{\mathbf{\Pi_{cycle}}} \leq opt_{\mathbf{\Pi_P}} + 3 \times \max_{ij}\{c_{ij}\}$

sketch of the proof. The previous lemma showed that the optimal partition-based strategy was composed of r single-agent cyclic strategies based on TSP. Let S^k_{TSP} denote the closed-path being the solution to the TSP on subgraph G_k. By joining together these closed-paths, it is possible to build a closed-path S covering all nodes such that $c(S) \leq \sum_{k=1}^{r} c(S^k_{TSP}) + 2r \times \max_{ij}\{c_{ij}\}$. Therefor, we have $c(S) \leq \sum_{k=1}^{r} c(S^k_{TSP}) + 2r \times max_{ij}\{c_{ij}\} \leq r \times max_{k\in 1..r}\{c(S^k_{TSP})\} + 2r \times max_{ij}\{c_{ij}\} \leq r \times opt_{\mathbf{\Pi_P}} + 2r \times max_{ij}\{c_{ij}\}$. From closed-path S, we can build a multi-agent cyclic strategy Π_S using lemma 1. $WI_{\Pi_S} \leq \frac{c(S)}{r} + \max_{ij}\{c_{ij}\} \leq opt_{\mathbf{\Pi_P}} + 3 \times \max_{ij}\{c_{ij}\}$. Thus, given any partition P, there exists a cyclic strategy $\Pi_S$verifying the previous equation.

To conclude this section, we can say that cyclic strategies are to be prefered when graphs do not have long edges connecting far regions. Otherwise, building a partition of the graph and making agents follow cyclic strategies based on the TSP of the regions is a good solution. The following corrolary is a "computable" version of the main theorem.

Corrolary. Let $P = \{P_1, ..., P_r\}$ a partition of V. It is possible to compute in $O(n^3)$ a cyclic strategy Π_{Chr} such that $WI_{\Pi_{Chr}} \leq \frac{3}{2} opt_{\mathbf{\Pi_P}} + 4 \times \max_{ij}\{c_{ij}\}$.

6. Experiments

Six different graphs (fig 4) were proposed in [9] as a benchmark for the patrolling problem. To evaluate the cyclic strategy on these graphs, the TSP of each graph was computed using the open-source software "Concorde" [2] which contains efficient heuristic algorithms which are in practice much faster than the $O(n^3)$ Christofides algorithm [5], and often come much closer to the optimal. Then, the simulator described in [1] was used to measure the idleness of agents patrolling around the obtained closed-paths.

The graphs on figure 5 show the performance of the strategies described in [1] and in [8] on the six graphs for 5 and 15 agents per graph. On each graph, the cyclic strategy obtains the best results. When the number of agents increase largely above 15, all strategies become equivalent.

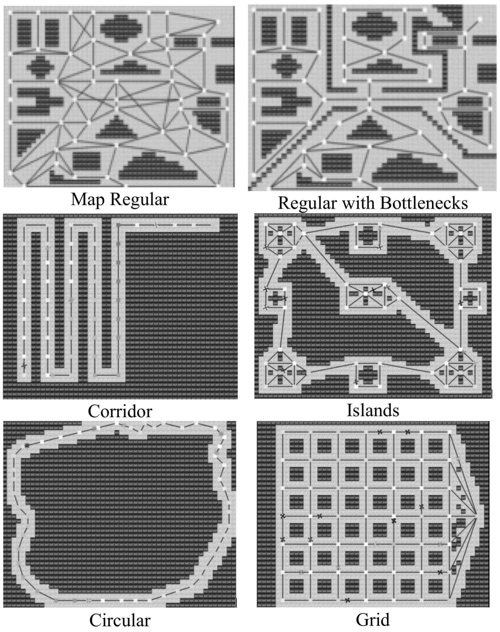
Map Regular
Regular with Bottlenecks
Corridor
Islands
Circular
Grid

Figure 4. Graphs used during experiments

7. Conclusion

We have shown various theoretical results for the patrolling problem. First, we have shown that the single-agent patrolling problem could be solved with a TSP approach. Then, we defined the class of cyclic strategies, based on an extension of this approach to more than one agent. An approximation result was obtained for this class, showing that in $O(n^3)$, a close to optimal strategy could be obtained.

The strategies based on a partitioning of the graph were also studied. A surprising result was obtained : except when $\max_{ij}\{c_{ij}\}$ is big, the cyclic strategies based on TSP were shown to be better than any partition based strategy. However, when graphs have long "tunnels" separating regions, the cyclic strategies are not well suited. Finally, some experiments were conducted to compare the state-of-the-art patrolling algorithms to the cyclic strategy based on TSP. The results are encouraging, but would need to be pursued on more graphs.

Many other interesting theoretical results have been obtained and will be published soon, in particular concerning other kind of strategies offering a good compromise between cyclic strategies and partition based strategies.

The cyclic strategy based on TSP have already been to implemented in a multi-robot patrolling problem with

[2] available at `www.math.princeton.edu/tsp/concorde.html`

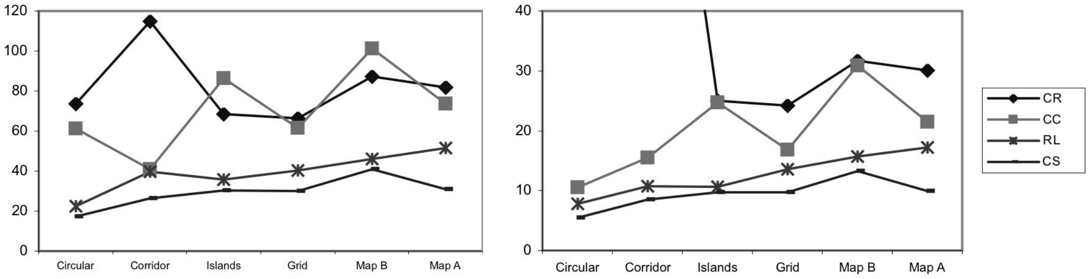

Figure 5. Idleness on different graphs with 5 agents (left) and 15 agents (right). The strategies compared here are the Conscientious Reactive (CR), the Cognitive Coordinated (CC), the reinforcement learning (RL) and the cyclic strategy (CS)

a noisy environment [7]. Many modifications had to be brought to the algorithm in order to synchronize de robots. It was shown that the cyclic strategy still performs best when the noise level is low.

References

[1] Aydano Machado, Geber Ramalho, Jean-Daniel Zucker, Alexis Drogoul (2002). Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures. Third International Workshop on Multi-Agent Based Simulation.

[2] Pottinger, Dave C.: Coordinated Unit Movement. Game Developer (January 1999) 42-51

[3] RoboCup Rescue home page: http://www.r.cs.kobe-u.ac.jp/robocup-rescue/, 2001.

[4] Chevaleyre, Y. (2003). The patrolling problem. Tech. Rep. of Univ. Paris 9. Available in french at http://l1.lamsade.dauphine.fr/˜chevaley/patrol

[5] Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. Rep. CS-93-13, Carnegie Mellon University, Graduate School of Industrial Administration.

[6] Reinelt G., (1994), ”The Traveling Salesman: Computational Solutions for TSP Applications,” Lectures Notes in Computer Science 840, Springer-Verlag.

[7] Sempe F., (2004), “Self-organizing groups of robots : application to patrolling tasks in noisy environments”, Ph.D thesis at university of Paris 6.

[8] Santana H., Corruble V., Ratitch B., (2004), “Multi-Agent Patrolling with Reinforcement Learning”, Autonomous Agents and Multi-Agent Systems. (to appear).

[9] Almeida A., Ramalho G., Santana H., Tedesco P., Menezes T., Corruble V., Chevaleyre Y. (2004), “Recent Advances on Multi-Agent Patrolling”, SBIA (to appear).