

## The Open System Setting in Timed MultiAgent Patrolling

Cyril Poulet, Vincent Corruble, Amal El Fallah Seghrouchni  
*Université Pierre et Marie Curie, LIP6, Paris, France*  
*firstname.lastname@lip6.fr*

Geber Ramalho  
*Universidade Federal do Pernambuco, CIn, Recife, Brazil*  
*glr@cin.ufpe.br*

**Abstract**—Several techniques have been proposed in the last few years to address the multiagent patrolling task. They share the assumption of a closed system setting (the set of agents present in the system is constant, no agent joins or leaves), which is a strong requirement and limits the applicability of multiagent patrolling models. In this article, we propose to revisit some of the techniques proposed in the literature to adapt them to the open society setting, and to compare their performances on a simple scenario where an agent decides to quit the patrolling task.

**Keywords**-multiagent; patrol; open system.

### I. INTRODUCTION

The patrolling problem rises in the situation when an area containing multiple points of interest needs to be supervised as often and as thoroughly as possible by a team of people or agents. *Timed patrolling* is a variation in which the task is to visit each point of interest as frequently and as regularly as possible, to optimize a metrics based on the temporal distribution of the visits.

An example of a situation that can be modeled with timed patrolling is inspired from disaster situations. A disaster happens (e.g. earthquake or tsunami), and victims must be found, then rescued. A team of agents is thus deployed on the area, searching for signs of victims in the ruins. However, with the typical patrolling problem, an agent finding a victim can only signal it to a rescue team, then proceed with patrolling. In our opinion, the agent should be able to stop patrolling and begin the rescue operation, while the rest of the patrolling agents reorganize to re-optimize the patrolling task.

With this situation in mind, we study the problem of patrolling with an open system, in which agents can join or leave the task at will. Previous work has proposed a broad range of coordination strategies for the patrolling task with a closed system, so it seems natural to adapt and compare some of them. However, we focus here mainly in the transitional phases, i.e. the phases where the MAS has to re-adapt to its new setting (fewer or more agents), as the stable phases (where no agent joins or leaves) have been already extensively studied. We propose here a framework for the patrolling problem with an open system, with new metrics to measure performances and results that can be used as references.

In this paper, we first define precisely the patrolling task

and review previous work in this section; we then present in Section 2 the adapted agents we chose to compare and the new metrics associated with the open society problem. We finally present the experimental results obtained and discuss them in Section 3.

### A. The Patrolling Task

The patrolling problem is formally defined as the tuple  $\langle G, S, M \rangle$  where  $G$  is a graph,  $S$  a society of agents and  $M$  a set of metrics. The graph  $G = \langle N, E \rangle$  is composed of a set of nodes  $N$  and a set of edges  $E$ . Each node  $n_i$  is associated with a priority  $p_i$  and an instantaneous idleness  $o_i$ , which is the time since its last visit by an agent. Each edge  $e_j$  has a length  $l_j$ , which is the distance between the two connected nodes.

The society  $S = \{a_i\}_{i \in N_S}$  is a set of size  $N_S$  of agents. Each agent is characterized by a set of perceptions and actions. Perceptions are about itself, the graph with a chosen depth, the positions of the other agents and the communications. Actions are visiting, going to a chosen node, and broadcasting with a chosen depth of perception.

$M$  is composed of metrics based on the temporal distribution of the visits that must be optimized. Patrolling is then the task given to the agents of  $S$  to repeatedly visit all the nodes of  $G$  with the goal of minimizing the metrics of  $M$ . Both  $G$  and  $S$  can be static or evolve in the course of the simulation. In this paper, we make the following assumptions:  $\forall i \leq |N|, p_i = 1$  (all nodes are of equal interest), the graph is static and simulations are cycled. There is also no restriction on the depth of communication and perception of the graph. These assumptions were made accordingly to the rescue situation presented previously, and to focus on the impact of the open system setting on the patrolling problem.

Finally, the open system implies that the number of agents can change over time. With this setting, the way the agents cope with these modifications becomes, in our opinion, a part of the patrolling task. Hence, we propose two new goals for the agents: minimizing the variations of the metrics during the transitional phases, and minimizing the durations of these phases.

### B. Related Work

The criterion of *idleness* is the most widely used (e.g. in [1]). Thus, we will present some of the proposed strategies,

then summarize the results obtained with the average graph idleness as criterion.

1) *description of the existing strategies*: Many different strategies have been proposed over the past ten years. [1] made a comparative study of agents: reactive/cognitive, with or without communication, etc. Among these strategies are the following ones, which we will use in this paper : (i) the *Random Reactive Agent (RR)*: Reactive, chooses the next node to go to randomly amongst those directly connected to the current node; (ii) the *Conscientious Reactive Agent (CR)*: Reactive, chooses the next node to go to by taking the highest instantaneous idleness amongst those directly connected to the current node; (iii) the *Cognitive Coordinated Agent (CC)*: Cognitive, communication with a global coordinator that chooses as the next node for the agent requesting a goal, the node currently with the highest idleness in the graph. They added to this comparison the *Heuristic Pathfinding Cognitive Coordinated Agent (HPCC)*, which is the same as the previous CC, but where the time to reach a node is taken into account in addition to its idleness to choose *heuristically* which node to go to, as well as the projected sum of the idlenesses of the nodes on the possible paths to choose how to reach the goal node (*pathfinding*).

[2] proposed the *Single-Cycle agent (SC)*: the shortest cycle using the nodes of the graph is calculated, then the agents are distributed evenly and revolve around it. To ensure this even distribution, one agent is started first, then each agent passed by this first agent counts an increasing amount of time before starting (an interval being the length of the cycle on the number of agent, the first agent waits 1 interval, the second 2, etc.). This strategy has been used since as a reference for comparing the efficiency of other strategies (the SC is not adaptive, but is the highest ranked strategy possible on a static problem, given complete knowledge a priori and the use of Average Graph Idleness).

Other strategies involve for exemple the use of reinforcement learning in [3]. Finally, swarm approaches have been proposed (e.g. in [4]), as well as robotic approaches (e.g. [5]).

2) *Published results*: Keeping in mind that these results are based on the Average Graph Idleness, we can summarize that cognitive agents and coordinated agents have a clear advantage on the others. The second result was that using the proposed heuristic and path-finding always improved the strategy it was used with.

Finally, the following ordering appeared on the idleness criterion: (i) Chevalerey's SC is, as expected, a clear winner; (ii) reinforcement learning, bidding techniques and HPCC are on an equal ground, with slight differences depending on the general topology of the graph (see [1] for the various graphs proposed as test environments); (iii) the first strategies proposed, though interesting as starting points, do not match the performances of the more recent strategies.

## II. GENERALYSING THE PROBLEM TO AN OPEN SYSTEM SETTING

The open system setting, in which agents can join or quit the task, is an important modification to the original multiagent patrolling problem. In this new problem, we are interested in understanding how the different strategies can be adapted to be able to cope with the changes of agents in the system, and how these adaptations effectively cope with these changes. Thus, we first propose new evaluation criteria to evaluate the various strategies, then show how we modified some of the agents described earlier to fit this new task.

### A. Evaluation Criteria

In order to study the duration of the transitional phases and their shapes (i.e. how fast the strategies return to a stable phase), we propose to use the following metrics: (i) *stabilization time*: how long (counted in cycles) does it take for the simulation to reach a stable state? For this, we propose to calculate a mean over a long stabilized phase on one of the criterion, then to compare short averages on a hundred cycle to this value, starting from the time at which the event occurred (change in the agents number) and stopping when the short average is less than 1% different of the stable value; (ii) *amplitude of variations*: how high are the variations of performances during the transition? We propose to use the ratio between the maximum value of the instantaneous idleness during the transition and its average value during the stable phase.

### B. Adapting the agents

In this work, we focused on a few of the agents proposed in earlier work, with the intention of providing some basic results that could be used as reference in further works. With this in mind, we used the following agents: the RR, CR, CC, HPCC and SC, which description can be found in I-B.

We chose these agents for the following reasons: RR is a known worst strategy, and SC a known best strategy. They will be used as references. CR is a simple reactive, yet idleness-driven agent: it is an interesting strategy to see how reactive agents behave in an open system. CC is a simple coordinated agent, and can be used as an indication of how a coordinator can cope with agents coming and going. Finally the HPCC will allow us to see if the HP techniques still provide an advantage.

Reactive agents did not need any change, as they are unaware of the other agents in the system. In the coordinated strategies, the coordinator alone had to be modified to detect that the number of agents had changed. If an agent quits the task, the coordinator removes the current goal node of this agent from the current goals, allowing it to be reassigned to another agent later. Finally, adapting the SC asked more questions: the cycle does not need to be recalculated (since

the graph did not change) but the agents need to be rearranged around it. Three situations are possible:

- an agent leaves the task: the coordinator stops all agents, then asks each agent to wait an increasing amount of time before restarting, in order to respect the new length of the interval between agents. The first agent (which does not wait) is the agent right behind the one that left. We call this the *SC\_fast* strategy
- an agent enters the task: the coordinator stops all agents, then uses the original SC algorithm. We made two versions of it: (i) *SC\_slow\_1* : the first agent waits N-1 intervals, the second N-2, etc. This way, all the agents start around the same time, which makes a shorter but rougher transitional phase; (ii) *SC\_slow\_2* : the first agent waits 1 interval, the second 2, etc. This way, the agents restart one after another, making a transitional phase longer but with smoother variations of the metrics
- an agent enters the task at the same time that an agent leaves it: the coordinator fills the hole left by the leaving agent with the new agent.

The new protagonists having been described, we will now present and discuss the experiments realized.

### III. EXPERIMENTAL RESULTS AND DISCUSSION

We chose as environment base map A described in [1]), which has 50 nodes and 106 edges. For each number of agents in the society at the beginning of the simulation, 30 simulations were run from random start positions and a length of 3000 cycles.

For N agents, N taken in 2, 5, 10, 15 and 25, we ran three scenarios: one with a closed system of N agents, one with a closed system of N-1 agents, and finally one with an open system of N agents where an agent leaves at cycle 1000.

#### A. Results

We first compared the performances of our modified agents to those of the original agents on stable phase. We found that modified agents showed the same performances as the original agents within a 5% margin on the average graph idleness criterion.

Once established that the adapted agents perform as well as the original agents on stable phases, we study the transitional phase itself. For this, we first analyze the duration of the transitional phase for the various agents, then study the shape of these transitions. Using the stabilization time presented in II-A, the Fig.1 presents the results obtained for the various sizes of system. From this Figure we can extract the following results: (i) the quickest solutions are the *SC\_fast* and the HPCC, followed by the CR, then the CC and RR, and finally the two *SC\_slow*; (ii) the *SC\_fast* and *SC\_slow\_1* slightly improve with an increasing number of agents, whereas the *SC\_slow\_2* is slower to stabilize; (iii) the SC (all), HPCC and CR variations are proportional (or inversely proportional, for the *SC\_slow\_2*) to the number of

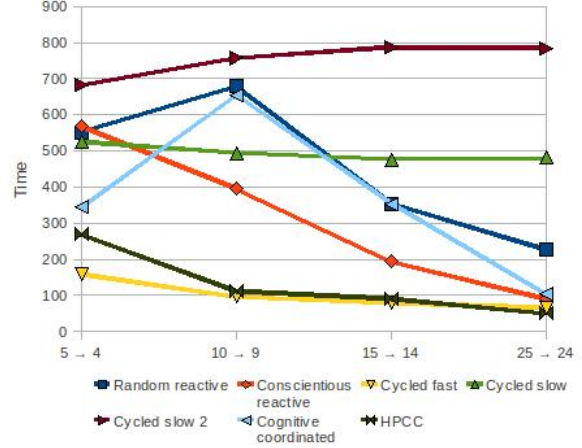


Figure 1. Stabilization time for various scenarios.

agents, whereas the RR and CC get worse until there are enough agents in the system to get them to increase.

These observations can be explained by the strategies used by the agents. RR is completely random: when an agent leaves, it takes some time for an agent to accidentally "fill the hole" left by the agent leaving. As the size of the system increases, this time will get statistically smaller. CR is a local solution: when an agent leaves, the nodes not visited quickly attracts a neighboring agent. The bigger the system is, the smaller the overall instantaneous idleness gets, and the quicker the abandoned nodes attract an agent. The difference between CC and HPCC can be explained the same way: the heuristic favors local moves, which makes HPCC a local solution. On the contrary, CC is global: when the nodes become attractive, they will be attributed one after another to all the agents of the system, until all the agents end up in the same region. This is the main reason for a bad overall performance. However, if the size of the system is high (the number of nodes by agent is small), some agents will continue to patrol the rest of the graph.

Concerning the SC solutions, it is explained by the coordination method. For the *SC\_fast*, as the size of the system increases, the interval between agents around the circular path decreases. The system thus takes less time to readapt to the new configuration. In the *SC\_slow* strategies, one agent is used as reference agent and restarted first, while the other agents wait a number of intervals corresponding to their new place after they detect the reference agent passing them. For *SC\_slow\_1*, they all restart around the same time because the first agent passed waits the most. However, as the size of the system increases, the reference agent gets quicker to the position where all agents restart. Thus, the system stabilizes quicker. For slow 2, on the contrary, the last agent passed is the one that waits the most. Since the interval between the first and the last agent grows with the number of agents, the

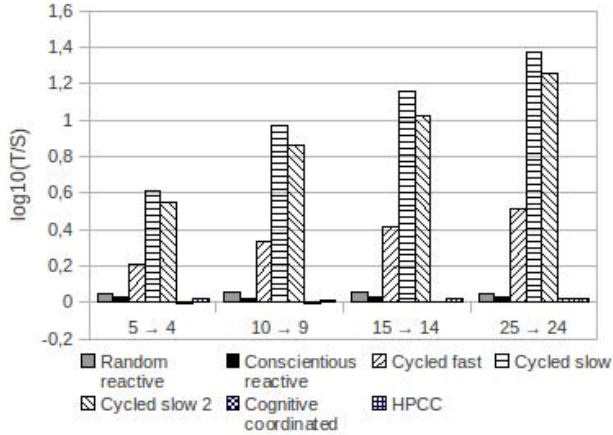


Figure 2. Max transitional instantaneous idleness (T) over stable average idleness (S) for various scenarios.

system takes more time to get to a stabilized state.

Regarding the amplitude of variations, Fig.2 shows the results associated with this criterion. It shows clearly that there is a small loss of performance for RR and CR during the transition, small to no loss for CC and HPCC, but a huge impact on performances for SC: ratios of 1.6 to 3.2 for SC\_fast, 4.1 to 23.7 for SC\_slow\_1 and 3.5 to 18.1 for SC\_slow\_2.

These results point to a clear observation: the transitional phase is very difficult for the SC\_slow strategies. They are both slow to re-adapt and present huge loss in performances during the transition. The situation is however a little brighter for SC\_fast, which compensates for its loss of performances by being the fastest strategy to re-adapt. This leads to the conclusion that the transitional phase for the  $N \rightarrow N - 1$  scenario separates the agents into three groups: RR, CR and CC which performances are not really interesting for this problem, HPCC and SC\_fast which are the two solutions that are the most adapted, and SC\_slow\_1 and 2 which are obviously not adapted to this problem.

### B. Discussion

We have shown in III-A that we can distinguish clearly between two groups of agents: one where the order of preference is clear in terms of both stabilized metrics and transitional time ( $CC > CR > RR$ ), and another where the strategies are better than those of the first group but where choosing one depends directly on which characteristic is the most important for the system.

When the key is to have a very small and smooth transition phase, at the possible cost of a slightly worse overall metrics performance, then HPCC is the choice to make. If a loss of performance is allowed during transition phase, then SC\_fast is the highest ranked choice.

All the adaptations presented here are also good for the

scenarios  $N \rightarrow N + 1$ , with the exception of the SC\_fast which works only with the loss of an agent. For this reason, studying SC\_slow\_1 and 2 gives us a comparative behavior of the SC solution for a  $N \rightarrow N + 1$  scenario. With this in mind, we can make a generalization of our results to any scenario involving the loss or gain of an agent, and probably of more than 1 agent: If the important point of the system is to remain as stable as possible, whatever the phase, then it would be a better choice to accept a small loss in performance during stable phases and choose the HPCC solution; if the system must provide the best performances during stable phases, but is allowed some time and a loss of performance for transition, then the SC\_slow is the solution (or a coupling SC\_fast/SC\_slow).

## IV. CONCLUSION

In this paper, we presented a new variation of the timed patrolling problem that we believe is a step to modeling several new interesting situations and problems: the open system setting. We proposed new metrics to compare solutions on this new problem, and adapted the five first solutions historically proposed to the timed patrolling in closed society. We showed that among these solutions, the adapted Heuristic Pathfinding Cognitive Coordinated and Single Cycle are clearly better than the others, and that choosing one or the other will depend on the characteristics one wishes to see in the system.

In future work, we plan to extend the span of our studies, in particular by adapting learning-based and bidding-based strategies, which are really decentralized. Another point of interest is to make more complicated scenarios, which would represent better the open society model we chose. Finally, we are also very interested in defining an entry and leaving cost for the open society patrol, in order to be able to interface it with a more global decision-making architecture.

## REFERENCES

- [1] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyre, "Recent advances on multi-agent patrolling," *Advances in AI-SBIA 2004*, 2004.
- [2] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. of IAT, IEEE/WIC/ACM International Conference*, 2004.
- [3] H. Santana, G. Ramalho, V. Corruble, and B. Ratitch, "Multi-agent patrolling with reinforcement learning," in *AA-MAS—Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems*, 2004.
- [4] H. Chu, A. Glad, O. Simonin, F. Sempe, A. Drogoul, and F. Charpillat, "Swarm approaches for the patrolling problem, information propagation vs. pheromone evaporation," in *IC-TAI'07*, 2007.
- [5] F. Sempe, "Auto-organisation d'une collectivité de robots: application à l'activité de patrouille en présence de perturbations," Ph.D. dissertation, Université Paris VI, 2004.