# Working as a team: using social criteria in the timed patrolling problem

Cyril Poulet, Vincent Corruble and Amal El Fallah Seghrouchni
LIP6, Université Pierre et Marie Curie, Paris, France. Email: name.surname@lip6.fr

***Abstract*—The multi-agent patrolling task constitutes a challenging issue for Artificial Intelligence and has the potential to cover a variety of domains ranging from agent-based simulations to crises management. Several techniques have been proposed in the last few years to address the multi-agent patrolling task with a closed-system setting. A few centralized strategies were also described to address the open-system setting, in which the agents can enter or leave the patrolling task at will.**
**In this article, we propose two decentralized, cooperative, auction-based strategies in which agents trade the nodes they have to visit. These strategies are inspired from the computational social choice theory and allow the agents to reason on the performances of the group rather than on their own. We show that these strategies perform at least as well as the state-of-the-art centralized performances, and better on specific criteria.**

## I. Introduction

Multi-agent patrolling is an abstract problem in which an area containing multiple points of interest need to be visited as often as possible by a team of people or agents. Two distinct variations have been studied in the past years: *adversarial patrolling*, in which the area needs to be guarded against intruders, and *timed patrolling* [1], where the task is to visit repetitively each point of interest. Performances are evaluated using metrics based on the temporal distribution of the visits. This problem is very interesting for studying coordination in Multi-Agent Systems (MAS): it is simple enough to be described and understood, and various coordination strategies can be compared experimentally by varying metrics. Indeed, the performances of the strategies are directly related to how well the agents are coordinated and share out the visits to all nodes between themselves. Yet the timed patrolling problem can model a wide variety of both real and artificial situations, such as area patrolling in war-games or repetitive maintenance in real-life factories. This is why this problem was proposed as a benchmark for MAS ([2]).

Recently, we proposed in [3] a variation for the timed patrolling problem: the open-system setting. In this new problem, agents may enter or leave the patrolling task at any time, thus requiring the system to adapt and reconfigure as the population of agents changes. This dynamicity allows for more complex applications, in which agents are multi-objectives and can participate in several concurrent tasks. These include rescue scenarios in which a rescue agent finds a victim, and must stop patrolling to begin the rescue operations. Another example is shift change in repetitive maintenance tasks.

Decentralization is a powerful way to avoid bottlenecks in computational power and communication when the population of the system rises, and also avoids a crash of the system when the coordinator fails, or when global communication is not possible. Using auctions to assign nodes to patrolling agents is an efficient, decentralized coordination strategy that has been successfully investigated in [4], using synchronous communication (allowing the auctions to be instantaneous). Lifting this assumption to use asynchronous communication leads however to an important loss in performances (see section III-A). In this context, we propose two strategies adapted from the computational social choice literature: **Minimax** and **Minisum**. They are both decentralized and asynchronous, which allows them to tackle all the previously cited problems. Computational social choice has received a lot of attention in various coordination problems. We propose an auction protocol based on [4] and show how to adapt it to cooperative auctions. We then present in details the Minimax and Minisum strategies. We finally show that these two decentralized strategies perform as well as centralized, state-of-the-art strategies on the patrolling problem, and better on specific criteria.

In this paper, section II defines formally the patrolling task, its metrics and the state-of-the-art. Section III presents an auction protocol and details the use of social criteria in this mechanism. Section IV presents our experimentation, and we discuss the results obtained in section V. Section VI concludes this paper and sketches some future work.

## II. The patrolling problem

### A. The patrolling task - formal model

The patrolling task takes place in an environment represented as a *graph*. Each node of this graph is a point of interest in the environment, and each edge is a link between two neighbouring points. Building on the formal descriptions we proposed in [3], we propose the following formal model of the multi-agent patrolling task.

The multi-agent patrolling task is formally represented as a tuple $\langle G, S, M\rangle$, with G 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 the associated set of edges $E$. Each node $n_i \in N$ has a priority $p_i$. Each edge $e_j \in E$ has a length $l_j$ representing the distance between the nodes linked by $e_j$. G may be static or evolve over time: nodes can become accessible or inaccessible, edges can become impracticable, priorities can change.

The society of agents $S = \{a_i\}_{i\leq|S|}$ is a set of size $|S|$ of agents $a_i$. Each agent is defined by the sets of perceptions and actions that it has access to. The available perceptions are the **perception of the environment**: internal time of the simulation, the agent's position on G, and the graph around the agent up to a distance of $d_G$, and the **perception of the society of agents**: the other agents' positions on G if they are under a distance ($d_{Soc}$) of the agent, and the communications. The available actions, which are not exclusive, are those **on the environment**: visiting the node the agent is situated on, and moving toward a destination, and those **on the society of agents**: sending a message by broadcast or to a single recipient, in both cases within a distance of $d_C$ of the sender. The society S can be closed - the number of agents is constant over time - or open - agents can join or leave at any time.

Finally, M is a set of evaluation criteria based on the temporal distribution of the visits. The timed multi-agent patrolling task is then the objective given to the agents of S to visit each node of G repetitively in order to optimize the set of criteria in M. In this paper, we are primarily concerned with the dynamic aspect of S (i.e. the number of agents changes over time). We will thus only consider a static graph G. We also consider each node to be equally important, thus all priorities are the same. Finally, communications are not restricted: agents can communicate as often and as far as needed ($d_C = \infty$).

### *B. Metrics*

The first metrics proposed for the patrolling problem in closed-system setting were based on the concept of *idleness* ([5]). If $t^i_{visit}$ is the time of the last visit on node $n_i$ and $t$ the current time, the instantaneous node idleness is $Id_i(t) = t - t^i_{visit}$. It can then be averaged over the graph (instantaneous graph idleness $Id_G(t)$) and over time (graph idleness $Id_G{}^{t_1}_{t_2}$). However, graph idleness is difficult to link directly to the events happening during the simulation.

For this reason, [6] proposed interval-based metrics: the average interval $I_{av}$ and the Mean Square Interval (MSI). With $N$ as the set of nodes in the graph, $\mathfrak{I}_i$ as the set of intervals between the visits of node $n_i$ during the simulation, $\mathfrak{I}$ the entire set of intervals of the simulation and $|I_{n_i}|$ the length of an interval $I_{n_i}$ of node $n_i$, the MSI is:

$$MSI = \sqrt{\frac{\sum_{\{n_i \in N\}} \sum_{\{I_{n_i} \in \mathfrak{I}_i\}} |I_{n_i}|^2}{card(\mathfrak{I})}}$$

With $|N|$ the number of nodes in the graph and $T_{max}$ the number of cycles in the simulation, $I_{av}$ is then:

$$I_{av} = \frac{\sum^{\{n_i \in N\}} \sum^{\{I_{n_i} \in \mathfrak{I}_i\}} |I_{n_i}|}{card(\mathfrak{I})} = \frac{|N| \times T_{max}}{card(\mathfrak{I})}$$

Following [6], we can prove that their criteria can be related directly to the graph idleness by the following relation:

$$Id_G{}^{T_{max}}_0 = \frac{MSI^2}{2I_{av}} - \frac{1}{2}$$

For this reason, we propose to use the average interval and the MSI, since we think that they provide complementary information: how often the visits are in average for the first criterion, and how well the visits are spread over the graph for the second. However, for instantaneous measures, the instantaneous graph idleness still provides a good measure of the state of the system.

For the patrolling problem in open-system setting, [3] proposed additional criteria to measure performance during transitional phases :

- *stabilization time*: represents the time taken by the system to return to a stable phase. For this, [3] propose to calculate a mean over the stable phase following the transition on one of the criterion, then to compare short averages on a hundred cycles 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
- *amplitude of variations*: measures the eventual loss of performances during the transitional phase. [3] propose to use the ratio between the maximum value of the instantaneous idleness during the transition and its average value during the stable phase showing the worst performances between the stable phase before the transition, and the stable phase after.

### *C. Background*

Many different strategies have been proposed for the patrolling task with a closed-system setting (e.g. [5], [7]). In this paper we will only detail those that present state-of-the-art performances that will be used as references. For the centralized approaches, two strategies stand out: the Single-Cycle agent and the Heuristic Pathfinding Cognitive Coordinated agent.

**The Single-Cycle agent (SC)** was proposed in [1]. It uses two types of agents: a single coordinator agent, and as many performer agents as needed/wanted. The coordinator agent calculates the minimal cycle of graph G with a TSP-like algorithm ($d_G$=$\infty$), and then distributes evenly the performer agents around this single chosen path according to their starting point. The performer agents simply follow the chosen path, visiting the nodes as they reach them. It is clear that since the interval between 2 visits of a single node is the exact distance between two performer agents, the more performers there are, the better this strategy performs. As we described in [3], in an open environment the agents are redistributed after each event (agents entering or leaving the patrolling task): agents are stopped various amounts of time in order to generate spaces for the new agents, or to fill the spaces left by agents leaving. This is the role of the coordinator to calculate for each performer agent the time it needs to wait before setting off again, and to transmit it.

It has been demonstrated that the SC strategy is the best possible on the $I_{max}$ criterion. However, its flaws are obvious: on the first hand it is very sensible to the size of the graph (as finding a good TSP solution is NP-hard), even more if the graph is not static, and on the second hand changing the size of the population implies to partially stop the agents, thus causing important losses in the performances (see part IV).

**The Heuristic Pathfinding Cognitive Coordinated agent (HPCC)** was proposed in [5]. Again, it uses two types of agent: a single coordinator agent, and as many performer agents as needed/wanted. The role of the coordinator is to

calculate, after each visit made by a performer on a node, the new target of the visiting performer agent using a combination of distance and expected idleness of the target. Once informed of its new target, the performer agent calculates the most interesting path to its new target, maximizing the idleness of the nodes visited along the path. This strategy is naturally adapted to the open environment. It requires $d_G$=$d_C$=$\infty$.
Though HPCC obtains the best performances currently and has no obvious flaw, it is centralized, which is not failsafe, and even without failure can become a problem as the size of the task increases. At any given time, the average number of visits, and thus the amount of messages and calculations that the coordinator has to manage, is proportional to the size of the population, and it can become a bottleneck for high-scale patrolling tasks.
Various decentralized approaches have also been proposed. We can cite the Flexible Bidder agent [4] which is discussed in section III, reinforcement learning approaches [8] and a gravitational strategy [6].

## III. Using social criteria in the auction protocol

### A. General auction protocol

In this strategy, there is only one type of agents. The nodes of G are distributed among the agents, and each agent must visit the nodes that have been assigned to him as often as possible. For this, each agent chooses its new target and the associated path according to the same heuristics as in the HPCC strategy. Coordination results from a process in which the agents can swap their nodes using auctions, with the goal of minimizing a distance function (here the length of the path they have to walk to visit all their nodes) on their set of nodes. Following [4] and the **Flexible Bidder Agent (FBA)**, the auction protocol that was selected is a private value, sealed bid auction protocol. During an auction, two roles appear: the initiator agent, and the participants. All agents can play each role, but each auction has a single initiator. An auction takes place as follows:
- The initiator agent (agent 1) identifies 1 or 2 nodes that it wants to trade, using a given value function f. With $N_i$ the set of nodes of agent i :

$$\begin{cases} n_1 = argmin_{n_{1i} \in N_1} f(N_1 \setminus \{n_{1i}\}) \\ n_2 = argmin_{n_{1i} \in N_1 \setminus \{n_{11}\}} f(N_1 \setminus \{n_{1i}, n_{11}\}) \end{cases}$$

It then informs the other agents of the beginning of an auction via an INFORM message containing the nodes put to auction.
- The participants $a_2$ to $a_n$ then choose to propose nodes in exchange (1 for 1, 1 for 2 or 2 for 2) via a PROPOSE message containing the chosen nodes, or refuse to participate in the auction (REFUSE). For example, for participant $a_k$ and the proposed node $n_{11}$, $\exists ? n_k \in N_k / f(N_k \cup \{n_{11}\} \setminus \{n_k\}) < f(N_k)$ ? If it is the case, there is an swap that is beneficial to $a_k$ and agent $a_k$ participates. Only the initiator receives the answers from the participants.
- Finally, the initiator agent reviews the propositions once it has received all the expected answers. It accepts the most interesting swap (ACCEPT) and rejects the others (REJECT). This protocol has been proposed for competitive auctions in the Flexible Bidder Agent (FBA) strategy ([4]). The valuation function chosen for the agents was to select the nodes maximizing the difference between the current path length of the agent, and the projected path length (without the chosen node for the initial choice, and after the swap for the proposition and awarding). This difference was combined with the estimated current idleness of the nodes. Although very good results were obtained, synchronous communications were used in their implementation. This is a very restrictive condition for a decentralized strategy, as it requires all agents to be instantly available when the auction is initiated. Asynchronous communication is a more realistic approach for multi-agent systems, as each agent is free to process the messages it receives whenever it is the most interesting or practical. It is also a mean to open the patrolling problem to other applications in which synchronous communication is not possible.
For this reason, we adapted this strategy for asynchronous communications: auctions can now be simultaneous, i.e. an agent can be both initiator and participant at the same time, in different auctions. As shown in [9], this causes an important loss in performances for this strategy (as can be seen in section IV). This observation leads us to the following question: how can this negative effect be balanced to achieve good performances in the patrolling task ? We show in this paper that social choice theory provides powerful answers to this question. In the remainder of this paper, FBA will refer to our asynchronous version of this strategy.

### B. Social bidding and awarding strategies

Using social choice in this auction protocol lets agents consider swaps that were not previously allowed in competitive auctions: swaps that are not mutually beneficial, but are beneficial to the whole society. This will allow a better assignment of the nodes to the agents, and thus allow better performances on the patrolling task. However, we do not guaranty optimality, due to the parallel nature of the auctions.

*1) general description:* The general protocol is the same as in the FBA strategy, with a small difference: agents will be able to propose transfers, i.e. 0 for 1 cooperative swaps in which the participant offers to unburden the initiator agent by taking one of its nodes without giving one in exchange. It is as follow :
- Initiator chooses nodes as decribed, but sends with them $i_1(N_1, n_{1i})$ where $i_1$ is a valuation function allowing other agents to evaluate if they must accept a non-beneficial transaction;
- participants check if there is a possible beneficial swap as previously described. If that is the case, they propose it. If there is none, the agent checks non beneficial but collectively acceptable transactions. With $C^1(N_k, n_1, i_1(N_1, n_{11}))$ a condition depending on the chosen well-being criterion, is $C^1$ verified? If so, $a_k$ proposes a transfer to $a_1$, i.e. to take the node for free, by sending "PROPOSE $i_k(N_k, n_{11})$". Otherwise, $a_k$ sends a "REFUSE".
- The awarding phase is in two parts. First, $a_1$ classifies

propositions in 4 categories: transfers, swaps that are beneficial to itself (i.e. which validates $f(N_1 \cup \{n_k\} \setminus \{n_{11}\}) < f(N_1)$), swaps not beneficial to itself but collectively beneficial (i.e. for which the previous equation is false but a validating condition $C^2(N_1, n_{11}, n_k, i_k(N_k, n_{11}, n_k))$ depending on the chosen well-being criterion is met), and swaps not beneficial to anyone. Then the propositions of the three first categories are classified according to the chosen well-being criterion, and the best one is accepted. The others are rejected.

As appears in this description, basic functions and valuations are dependant on the chosen well-being criterion. We will thus present now two possible criteria: Minimax and Minisum.

*2) Minimax:* The *egalitarian social welfare* criterion ([10]) aims at maximizing the utility of the poorest agent in the society. To apply this criterion to the patrolling task, we chose as utility the length of the path each agent has to walk to visit all its nodes.

$$\begin{cases} f(N_k) = pathLength(N_k) \text{ abbr. as } pL(N_k) \\ criterion\ :\ \min \max_{a \in A} pL(N_a) \end{cases}$$

Of course, we try here to minimize the longest path in the society, hence the **minimax** denomination.

To allow the agents to propose transfers, the valuation and condition used are:

$$\begin{cases} i_1(N_1, n_{11}) = pL(N_1) \\ C^1 = pL(N_1) - pL(N_k \cup \{n_{11}\}) \end{cases}$$

The participant needs to know the initial path length of the initiator agent ($i_1$), in order to be able to calculate if accepting a transfer of node $n_{11}$ will create a path longer than the initiator's current one. If $C^1$ is positive, it means that transferring $n_{11}$ to $a_k$ does not create a path that is longer than the current path. Two cases are possible: either $a_1$ has the current longest path and the transfer reduces it, or it has not and the longest path will not be changed. The transfer is collectively beneficial, as the welfare criterion is decreased or unchanged.

In order for the initiator to determine if a given, not beneficial swap is socially acceptable, we propose the following :

$$\begin{cases} i_k(N_k, n_{11}, n_k) = pL(N_k \cup \{n_{11}\} \setminus \{n_k\}) \\ C^2 = pL(N_k \cup \{n_{11}\} \setminus \{n_k\}) - pL(N_1 \cup \{n_k\} \setminus \{n_{11}\}) \end{cases}$$

The initiator uses the projected path length of participant $a_k$ to determine which projected path length is greater between its own and $a_k$'s. We know that the swap is not beneficial: $pL(N_1 \cup \{n_k\} \setminus \{n_{11}\}) > pL(N_1)$, and that it has been proposed by $a_k$ and is therefore beneficial to it: $pL(N_k) > pL(N_k \cup \{n_{11}\} \setminus \{n_k\})$. Thus if condition $C^2$ is positive, we can be sure that $a_k$'s path is currently greater that $a_1$'s, that the swap decreases $a_k$'s path length and that it does not increase $a_1$'s path length above the current path length of $a_k$. Again, there are two cases: either $a_k$'s path is the longest and it will be shortened (while ensuring that $a_1$ does not get a new max), or it is not and the longest path has not been changed. The swap is collectively beneficial.

These conditions ensure that whatever transaction is chosen, it decreases or does not change the current maximum path length. Finally, the propositions are classified as follow : 1) Consider first the swaps beneficial to $a_1$ (which are then by design mutually beneficial). Choose the one maximizing $a_1$'s gain (i.e. difference between its current path length and its projected path length). 2) if no beneficial swap was proposed, then consider swaps not beneficial to $a_1$: they have been proposed by agents whose current path is longer than $a_1$'s path. Choose the one whose $i_k$ (i.e. $a_k$'s projected path length) is maximal (since the corresponding agent has probably the current longest path). 3) finally, consider transfers: if no swap has been awarded, $a_1$ has the current longest path. Choose the transfer with the smallest $i_k$ (i.e. $a_k$'s projected path length), to unburden $a_1$ on the agent which will suffer the less as a result of the added node.

Other classifications are possible, but we chose to share the smallest information possible among the agents.

*3) Minisum:* The *utilitarian social welfare* criterion ([10]) aims at maximizing the average utility of the agents. As for the minimax, the utility that we used is the length of the path of an agent, and we aimed at minimizing the sum of all utilities:

$$\begin{cases} f(N_k) = pL(N_k) \\ criterion\ :\ \min \sum_{a \in A} pL(N_a) \end{cases}$$

For clarity, we introduce the notation $diff$, which represents the difference between agent $a_k$'s actual path length and its projected path length (by giving the set of nodes $S_2$ in exchange for the set of nodes $S_1$), i.e. how much the agent gains in the swap: $diff(_{N_k, S_1, S_2}) = pL(_{N_k}) - pL(_{(N_k \cup S_1) \setminus S_2})$. Then the valuation and condition for a participant to propose a transfer are:

$$\begin{cases} i_1(N_1, n_{11}) = diff(N_1, \emptyset, \{n_{11}\}) \\ C^1 = diff(N_1, \emptyset, \{n_{11}\}) - diff(N_k, \{n_{11}\}, \emptyset) \end{cases}$$

Participant $a_k$ uses the initiator's gain (via $i_1$) to determine if by accepting a transfer, $a_1$ gains more than $a_k$ looses. If $C^1$ is positive, it is the case, and transferring the node decreases the sum of their path length. The transfer is thus collectively interesting.

To determine if a proposed swap that is not beneficial is socially acceptable, the initiator uses the following valuation and condition:

$$\begin{cases} i_k(N_k, n_{11}, n_k) = diff(N_k, \{n_{11}\}, \{n_k\}) \\ C^2 = diff(_{N_k, \{n_{11}\}, \{n_k\}}) - diff(_{N_1, \{n_k\}, \{n_{11}\}}) \end{cases}$$

Here again, we know that the swap is beneficial to $a_k$, so $a_1$ tries to determine if it looses less in the swap than $a_k$ gains. If that is the case ($C^2$ is positive), the sum of their path lengths will decrease, which is beneficial to the society.

Finally, to choose the winning proposition, we can write that for a given auction starting at $t_1$, finishing at $t_2$ and awarded to $a_b$,

$$\sum_{a \in A} pL(N_a)(t_1) - \sum_{a \in A} pL(N_a)(t_2) =$$

$$\begin{cases} diff(N_1, \emptyset, \{n_{11}\}) - diff(N_b, \{n_{11}\}, \emptyset) \text{ (transfer)} \\ diff(N_b, \{n_{11}\}, \{n_b\}) - diff(N_1, \{n_b\}, \{n_{11}\}) \text{ (swap)} \end{cases}$$

$a_1$ thus awards the auction to the proposition (swap or transfer) maximizing the gain on the Minisum criterion.

### C. Design of a Transition mechanism

We designed and tested several mechanisms to allow the system to reorganize when agents leave or join the patrolling task with an auction-based strategy (see [9]). As it is not the focus of this paper, and due to space constraints, we only briefly present here the mechanisms that have been used for the Minimax and Minisum strategies, which were the most successful on the criteria proposed in II-B: stabilization time and amplitude of variation. Please refer to [9] for more details.
For the Minimax and the FBA strategies, the chosen mechanisms are utilitarian and give or take more nodes to the agents that are spatialy close to the agent leaving or entering (*proximity mechanism*). The Minisum strategy needed more egalitarian mechanisms : each agent gets or gives the same number of nodes. The chosen mechanisms are the *Worst Nodes mechanism* for entering, and *Worst pre-calculated group of nodes mechanism* for leaving.

## IV. Experimental evaluation

We used as experimental environment the Simpatrol simulator initiated at CIn, UFPE (Recife, Brazil) ([2]). We chose as environments the maps described in [5] and widely used in the literature: a random map that is strongly connected (50 nodes/106 edges): map A, a random map that is loosely connected (50/69): map B, a circle (50/50), a corridor (49/50), a grid (50/90) and a map of 9 islands, strongly connected inside and loosely connected between them (50/84). On each map, we ran experiments on closed settings (30 for each population size) to measure stable performances of the various strategies. Then long experiments (20000 cycles) were made with an open system whose size evolves between 2 and 13 agents. These long experiments allowed us to have a better understanding of the behaviour of the strategies on the long term. For all experiments, the original node assignment for FBA, Minimax and Minisum is uniform and randomized.
To compare the average performances of the strategies, on each map and for each population size we normalized the measures by multiplying them by the number of agent in the system. This showed as first result that for $I_{av}$ and MSI, all strategies obtain performances that are quasi-linear in the number of agents on all maps, with standard deviations under 5% for $I_{av}$ and under 20% for MSI. This allows us to consider that averaging over population size represents fairly well the performances of a given strategy. We use these averages in the remainder of this paper.
A second result is that the average interval $I_{av}$ is not a discriminating criterion. On each map, for each population size, HPCC has the best performance on $I_{av}$, followed by Minisum, but all the other strategies are less than 20% behind (above) HPCC, sometimes less than 2% for all population sizes (circle, grid). The SC strategy is on average the worse strategy on this criterion.
The MSI however shows more differences between the strategies (see Fig. 1, top). The Single Cycle has the smallest MSI on all maps, followed closely by HPCC. Minisum and Minimax have similar performances on this criterion, and FBA has the worst performance on all maps.
As the main goal of Minimax is to minimize the maximal path in the society of agents, it is natural to mention the maximum interval criterion ($I_{max}$). With the same averaging method, Fig. 1 (bottom) present the results of the various strategies on this criterion. As has been shown in [1], SC is the optimal strategy on this criterion, and thus shows the best performances on all maps. However, Minimax performs here exceedingly well by obtaining the 2nd best performance on 4 maps out of 6, and a close 3rd on the 2 other maps. As predicted in section III-A, FBA obtains very poor performances on this criterion.
In long experiments, agents may enter or leave every 100 cycles, up to 4 agents at a time. The maximum size of the population is 13 (1/4th of the number of nodes of the map), and the more (less) agents there are, the most likely it is that agents leave (enter). For each map, we ran 15 experiments with random starting positions and node assignments. These allowed to observe the behaviour of the strategies in a complex environment. A first observation is that the FBA strategy performs better during the long simulations than what could be expected from the short simulations. However, it still does not perform as well as the other strategies on the MSI and $I_{max}$ criteria. We will discuss this result in section V.
On the criteria related to transitional phases, we can make the following observations:
- Stabilization time: HPCC is the quickest to adapt to an event (with an average stabilization in 132 cycles), followed by FBA (170), Minimax (186), Minisum (173) and finally SC (205);
- Amplitude of variations: FBA is the strategy that offers the smallest loss of performance in transition with a ratio $log(\frac{Max}{Av})$ of 0.013 (i.e. 3.4%), followed by Minimax and HPCC (0.016/5%), Minisum (0.023/7%) and SC (0.24/85%).

## V. Discussion

These results offer a good picture of how the various strategies work. The SC strategy is optimized to visit every node with the same regularity, which gives it the best performances on both MSI and $I_{max}$. However, it is not optimized on $I_{av}$, and is costly to reconfigure both in time and performances.
As explained in [3], HPCC is a local strategy: each agent tends to stay in a region of the graph, and to visit often the nodes it is close to. The first ensure a good MSI and $I_{max}$ by avoiding "long-distance travels". The second assures that some nodes will be visited very often, thus allowing a very good $I_{av}$. Being centralized, the HPCC strategy reacts fast to a change in the population, but the locality of its nature is here a drawback since the agents will not go directly to their new location but multiply visits on the way. This causes the loss of performance described.
The results confirm our predictions about using asynchronous communication with the FBA strategy. The FBA shows a good $I_{av}$, a bad MSI and a worse $I_{max}$. This is due to a poor node assignment where some agents have few nodes that are visited extremely often (thus a good $I_{av}$) and with no reason to unburden the agents less fortunate (thus the

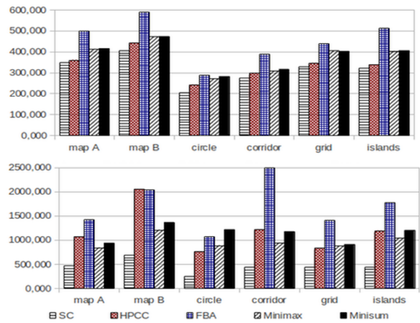


Fig. 1. Normalized performances of the various strategies, by map. Top : MSI. Bottom : Max Interval.

bad MSI and $I_{max}$). This is where the frequent changes in the population of the long simulations become helpful: it forces the agents to reorganize often, and thus improves the assignments during the entry and quitting mechanisms. Since in fact few transactions are performed after these mechanisms happen, the FBA strategy is quick to re-stabilize.

For assignment-based strategies, the loss of performance can be understood as the gain between the new assignment due to the event, and the assignment reached after the transactions. The small loss in performances can thus be explained by the fact that the poor overall performances do not allow to reach a really worse state in the system, and that the transactions do not allow a very efficient reassignment. Reallocating the nodes by proximity during transitions (see section III-C) avoid to reach worse reassignments (e.g. a node is reassigned to an agent far away, but the agent closer to the node refuses to trade it later because it is not interesting for him).

The Minimax strategy shows very good results: minimizing $I_{max}$ has the effect of ensuring a good MSI (no agent has a bad set of nodes, so each node is visited regularly), which in turn ensure a good $I_{av}$. However, a change in population triggers a longer reassignment process (with more successful transactions). This explains both the long stabilization time, and the loss of performance higher than for the FBA strategy: the reassignment is more efficient.

Finally, the Minisum strategy shows performances similar to the Minimax strategy, with a few differences. Minimizing the average path length instead of the longest allows to reach a better (smaller) $I_{av}$ at the cost of a worse $I_{max}$. Indeed, an agent can be in the situation where its path is longer than the average path, but no agent can unburden him for a cost smaller than his gain. This explains why $I_{max}$ is greater than for Minimax. However, having agents with longer paths implies having agents with smaller paths, which are able to visit their nodes very often. This explains the better $I_{av}$. Concerning the transition phases, the transition time of the Minisum strategy is very similar to those of the other auction-based strategies, FBA and Minimax. However, the amplitude of variation is greater for the Minisum strategy. This is explained by the chosen entry and leaving mechanisms: as they reassign more equally the nodes among all the agents, the new assignments after these events are less efficient than after the proximity mechanisms, thus the higher loss in performances.

In summary, both Minimax and Minisum allow an efficient assignment of nodes, with good performances on all metrics. Minimax is especially set for the $I_{max}$ criterion, on which it outperforms HPCC, whereas Minisum proposes a compromise between $I_{av}$ and MSI. They also show small losses in performances during transitions. As a final point, it is possible to bound the number of messages of the protocol independantly of the population in the system. This allows us to conclude that unlike the HPCC strategy, the proposed auction-based strategies can be used in high-scale patrolling problems without having bottlenecks appearing in the system.

## VI. Conclusion

In this paper, we presented two strategies for the multi-agent timed patrolling task with an open system setting and asynchronous communication. These strategies are auction-based, completely decentralized and inspired from the social choice theory: the Minimax and the Minisum strategies. We described how agents can bid on auctions and award them in order to optimize the egalitarian well-being of the society (Minimax) or its utilitarian well-being (Minisum). We evaluated these strategies both on closed and open system setting. Results show that the Minimax strategy performs better than the previous state-of-the-art, auction-based strategy, and is better than centralized, state-of-the-art strategies on at least one criterion. Minisum shows the same performances on closed system setting, but needs a more egalitarian transition mechanism to reach its full potential in the open-system setting. Our studies also showed that the performances achieved were quasi-linear in the number of the agent, which is very promising for larger societies.

## References

[1] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. of the Intelligent Agent Technology, IEEE/WIC/ACM Int. Conf.*, 2004.

[2] D. Moreira, G. Ramalho, and P. Tedesco, "Simpatrol - towards the establishment of multi-agent patrolling as a benchmark for multi-agent systems," in *ICAART*, 2009.

[3] C. Poulet, V. Corruble, A. Seghrouchni, and G. Ramalho, "The open system setting in timed multiagent patrolling," in *Proc. of the Intelligent Agent Technology, 2011 IEEE/WIC/ACM Int. Conf.*, vol. 2. IEEE, 2011.

[4] T. Menezes, P. Tedesco, and G. Ramalho, "Negotiator agents for the patrolling task," *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, 2006.

[5] A. Machado, G. Ramalho, J. Zucker, and A. Drogoul, "Multi-agent patrolling: An empirical analysis of alternative architectures," *Lecture notes in computer science*, 2003.

[6] P. Sampaio, G. Ramalho, and P. Tedesco, "The gravitational strategy for the timed patrolling," in *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE Int. Conf.*, vol. 1. IEEE, 2010.

[7] A. Almeida, P. Castro, T. Menezes, and G. Ramalho, "Combining idleness and distance to design heuristic agents for the patrolling task," in *II Brazilian Workshop in Games and Digital Entertainment*, 2003.

[8] H. Santana, G. Ramalho, V. Corruble, and B. Ratitch, "Multi-agent patrolling with reinforcement learning," in *AAMAS-2004-Proc.of the 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems*, 2004.

[9] C. Poulet, V. Corruble, and A. El Fallah Seghrouchni, "Auction-based strategies for the open-system patrolling task," in *PRIMA - 15th Int. Conf. on Principles and Practice of Multi-Agent Systems, Proc.*, ser. Lecture Notes in Computer Science. Springer, 2012.

[10] H. Moulin, "Axioms of Cooperative Decision Making," *Cambridge Books*, Cambridge University Press, vol. 15, 1989.