Distributed On-Line Coordination for Multi-Robot Patrolling

Alessandro Farinelli[1], Luca Iocchi[2], Daniele Nardi[2]
[1] Department of Computer Science
University of Verona, Italy
E-mail: alessandro.farinelli@univr.it
[2] Dept. of Computer Control and Management Engineering
Sapienza University of Rome, Italy
E-mail; {iocchi,nardi}@dis.uniroma1.it

Abstract

Many multi-robot applications related to surveillance and security requires the robots to continuously monitoring and patrolling an environment in order to assess the situation or look for abnormal events. Multi-Robot Patrolling has thus been studied from several different perspectives, ranging from techniques that devise optimal off-line strategies to implemented systems. However, still few approaches consider on-line decision techniques that can cope with uncertainty and non-determinism in robot behaviors.

In this article we present a Multi-Robot Patrolling system based on on-line coordination. More precisely, we cast the problem as a task assignment problem and propose a new solution technique based on sequential single-item auctions. We evaluate the performance of our system in a realistic simulation environment as well as on real robotic platforms, showing increased performance in efficiency and stability of the patrolling task.

1 Introduction

Multi-Robot Patrolling (MRP) is the task of continuous visiting a set of locations in an environment and it is a key feature for various applications related to surveillance and security. When the environment to monitor is large and many robots are involved in this task, multi-robot coordination techniques, also based on swarm methodologies, provide significant advantages. Among many proposed approaches, on-line solutions to the MRP problem [4, 5, 6, 1, 3] compute (or modify) paths while the robots are patrolling. These solutions can use most up-to-date information and hence compensate for non-modelled characteristics of the environment.

In this abstract, we address on-line coordination in MRP, by casting the MRP problem as a Dynamic Task Assignment (DTA) problem. In the proposed framework, it is possible to develop effective on-line decision techniques that can cope with uncertainty in the environment and non-determinism in robot behaviors.

We evaluate the performance of our system in a realistic simulation environment (built with ROS and Stage) as well as on real robotic platforms. In particular, in the simulated environment we compare our task assignment approach with previous off-line and on-line methods. The experimental results confirm that on-line coordination approaches improve the performance of the multi-robot patrolling system in real environments, and that coordination approaches that employ more informed coordination protocols achieve better performance with respect to on-line approaches with a weaker form of coordination.

2 Multi-Robot Patrolling as Dynamic Task Assignment

The MRP problem is characterized by a set of robots $R = \{r_1,...,r_n\}$, a patrol graph $PG =< P, E, c >$, and some MRP performance metrics. The goal of the multi-robot system is to choose a path $\pi_i = \langle p_1,...,p_t \rangle$, for each robot r_i, so to maximize such MRP performance metrics.

The DTA problem associated to MRP consists of a set of tasks $\mathscr{T} = \{\tau_1, \cdots, \tau_m\}$ a set of robots $R = \{r_1, \cdots, r_n\}$ and a reward matrix $\mathbf{V} = \{v_{ij}\}$, where v_{ij} indicates the reward the system achieves when robot r_i executes task τ_j. An allocation matrix $\mathbf{A} = \{a_{ij}\}$ defines the allocation of robots to tasks with $a_{ij} \in \{0,1\}$ and $a_{ij} = 1$ if robot r_i is allocated to task τ_j. The goal of the system is then to find the best assignment of tasks to robots with respect to the given reward, i.e.

$$a^* = \arg\max_{\mathbf{A}} \sum_{i=1}^{|R|} \sum_{j=1}^{|P|} v_{ij} a_{ij}$$

Moreover, a set of constraints $\mathscr{C}$ usually describes valid allocations of robots to tasks (e.g., one task per agent) and hence the above optimization must be performed subject to $\mathscr{C}$.

In our patrolling problem, tasks are locations to be visited, i.e., a set of patrol nodes $P = \{p_1, \cdots, p_m\}$ and rewards depend on the average idleness of a node and on the travel cost that a robot incurs to visit such node. Specifically, we have that $v_{ij} = U(r_i, p_j, t)$, where $U(r_i, p_j, t)$ is a utility function that encodes how good is for the system to allocate robot r_i to node p_j at current time t. An example of such a utility function may be

$$U(r_i, p_j, t) = \theta_1 I^{p_j}(t) + \theta_2 Tc(r_i, p_j, t)$$

where $I^{p_j}(t)$ is the idleness of p_j at time t, $Tc(r_i, p_j, t)$ is the travel cost for robot r_i to reach p_j considering the robot position at time t, and θ_1, θ_2 are parameters that balance travel cost and idleness. Moreover, we enforce the constraint that, at any time t, only one robot should be allocated to a specific location (i.e., $\forall t, j \sum_i a_{ij} \leq 1$) to maintain a similar visit frequency across the locations.

In this formalization of the DTA problem, we do not explicitly represent paths for the robots (i.e., a task is one patrol node and not a sequence of such nodes), hence at each time step only a subset of the tasks might be allocated (this depends on the solution approach as described below). However, over time robots effectively build paths that visit patrol locations based on the current value of the idleness.

The solutions of the MRP problem based on DTA that we have developed are based on a distributed coordination protocol. More specifically, two algorithms have been developed: DTAG and DTAP. DTAG is a baseline algorithm for DTA with a greedy task assignment based on a simple utility function. DTAP (summarized in the next section) is a more sophisticated coordination protocol based in sequential auctions that allocates a subset of the patrol nodes to each robot, hence considering the paths that robots will follow.

2.1 DTA based on Sequential Single Item Auctions (DTAP)

DTAP is inspired by auction based task allocation: the basic idea is that robots announce their destinations to everyone and then collect "bids" from their team-mates. Such bids encode how well each robot fits to a given destination. In more detail, each robot selects the next visit node as the one that maximizes a utility function. Then the robot broadcasts its node selection, announcing its corresponding bid, to all team mates. After collecting all bids, the robot checks whether it is the one with the best bid for the selected node. If this is the case, the robot visits the selected node, otherwise it selects the next best visit node and iterates the selection process.

Our dynamic task allocation scheme takes inspiration from Sequential Single Item auctions, where robots allocate one task at the time, and when they compute their bids, they consider previous allocated tasks. This allows to take into account important synergies between tasks (i.e., patrol nodes that are close to each other). In our approach, such bid computation considers the number of tasks a robot is responsible for and the distance of the target node to the *central node*, which is the node at minimum path distance from all other nodes (see below for a more detailed explanation). This is different with respect to standard bid computation rules employed in sequential single item auctions for task allocation (e.g., [7]), that typically consider an aggregation of the path cost to cover all allocated tasks (e.g., the sum, max or average of the path cost). The rational behind this choice is twofold: i) by considering the number of nodes, we foster a balanced workload among the robots and ii) by considering the distance to the *central node*, we aim at creating a partition of the patrol nodes that tries to minimise path crossing among the robots, hence resulting in less interferences for navigation. Moreover, we do not consider marginal costs for bid computation, as this could result in unbalanced allocations (as stated in [3]), where some robots might have significantly more visit nodes than others. This would be problematic for the MRP strategy as it would increase the standard deviation of the global idleness.

To compute the bids, each robot maintains a list of nodes that represents the locations for which it is responsible (i.e., the set of nodes for which the robot has the lowest bid), hence the path cost for a patrol node aggregates the navigation cost to all nodes the robot is currently responsible for. As it is typically the case in Sequential Single Item auctions, robots allocate one task at the time, but when they compute their bids, they consider previous allocated tasks. This allows to take into account important synergies between tasks (i.e., patrol nodes that are close to each other).

As mentioned before, our approach is based on the concept of *central node*, which is the node that has minimum travel cost from all other nodes the robot is currently responsible for. In our implementation, the travel cost is comuputed as the length (i.e., the number of edges) of the shortest path between the two nodes. The central node is updated each time the tasks associated to a robot change (i.e., when the robot acquires or loses a task). Next, we compute the bid for a destination by multiplying the number of tasks the robot is responsible for by the travel cost from the central node to the destination. This computation of the bid helps balancing the workload, as it penalizes robots that are responsible for too many tasks, and it considers synergies among tasks, by penalizing robots that have a central node which is far from the current destination.

3 Experimental Validation

Experimental validation and performance assessment of the DTAP algorithm have been performed by using a realistic MRP simulator based on ROS and Stage. The simulator and the implementation of many MRP algorithms is available as a ROS package in `http://wiki.ros.org/patrolling_sim`. Several different maps of different size and complexity are used to compare the implemented algorithms.

Standard performance measures for MRP are based on the *idleness* of the nodes [2]. The *instantaneous idleness* $I^p(t)$ for a node p at time t is the elapsed time since the last visit from any robot in the team. Let $\langle t_0, t_1, ..., t_k \rangle$ be the time frames in which any robot of the team visits p, then we can collect the idlenesses of node p as $\langle I^p(t_1), ..., I^p(t_k) \rangle$ (i.e., $I^p(t_j) = t_j - t_{j-1}$). From these values we can calculate the *average idleness* of a node I^p_{avg} and its *standard deviation* I^p_{stddev}. Finally, three global measures can be computed by determining average, standard deviation and maximum of all the values $I^p(t)$ for every time t and every $p \in P$. We refer to these measures as *global idleness average* I^G_{avg}, *global idleness standard deviation* I^G_{stddev}, and *global maximum idleness* I^G_{max}, respectively. I^G_{stddev} actually measures how balnaced are the visits to the nodes: low values for I^G_{stddev} mean that all the patrol nodes are visited approximately with the same frequency. While I^G_{max} represents a worst case analysis.

In this section, we report the results of some of the experiments. Figure 1 shows boxplots of the idleness values in six different scenarios for a set of algorithms implemented in the simulator. The boxplots show the distribution of the idleness values (in terms of a bold line for the median value, a box for delimiting 25- and 75-percentile values, an asterisk for the average, and circles for outliers) during an experiment. Best performance are obtained when these values are low and when there are no outliers with high values. In these plots it appears evident that on-line algorithms based on explicit coordination (i.e., GBS [5], SEBS [5], DTAG, and DTAP) generally outperform the others.

In Figure 2, a more detailed comparison between SEBS and DTAP is reported. In the six scenarios considered here the ratio between the number of locations to visit and the number of robots is ≥ 10. In these situations, DTAP provides better performance than SEBS, since the benefit of a stronger coordination is advantageous given the large number of coordination possibilities. However, in scenarios with a lower ratio between number of locations to visit and number of robots a strong coordination is less beneficial. In these settings, the performance of the two algorithms are similar.

The conclusion of the reported experimental activity is twofold. From one side, on-line algorithms are necessary in order to correctly compensate for uncertainty and noise in real environment (as reproduced in our realistic simulations). From the other side, when the number of choices for the robots is high, a stronger coordination is beneficial.

References

[1] Ahmadi, M., Stone, P.: A multi-robot system for continuous area sweeping tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1724–1729 (2006). URL `http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=126653`

[2] Machado, A., Ramalho, G., Zucker, J.D., Drogoul, A.: Multi-agent patrolling: an empirical analysis of alternative architectures. In: Proceedings of the 3rd international conference on Multi-agent-based simulation II. Berlin, Heidelberg (2003)

[3] Pippin, C., Christensen, H., Weiss, L.: Performance based task assignment in multi-robot patrolling. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, pp. 70–76. ACM, New York, NY, USA (2013). DOI 10.1145/2480362.2480378. URL `http://doi.acm.org/10.1145/2480362.2480378`

[4] Portugal, D., Couceiro, M.S., Rocha, R.P.: Applying bayesian learning to multi-robot patrol. In: Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, pp. 1–6 (2013). DOI 10.1109/SSRR.2013.6719325

[5] Portugal, D., Rocha, R.P.: Distributed multi-robot patrol: a scalable and fault-tolerant framework. Robotics and Autonomous Systems **61**, 1572–1587 (2013)

[6] Semp, F., Drogoul, A.: Adaptive patrol for a group of robots. In: IROS, pp. 2865–2869. IEEE (2003)

[7] Tovey, C., Lagoudakis, M., Jain, S., Koenig, S.: The generation of bidding rules for auction-based robot coordination. In: F.S. L. Parker, A.S. (editor) (eds.) Multi-Robot Systems: From Swarms to Intelligent Automata, vol. 3. Springer (2005)

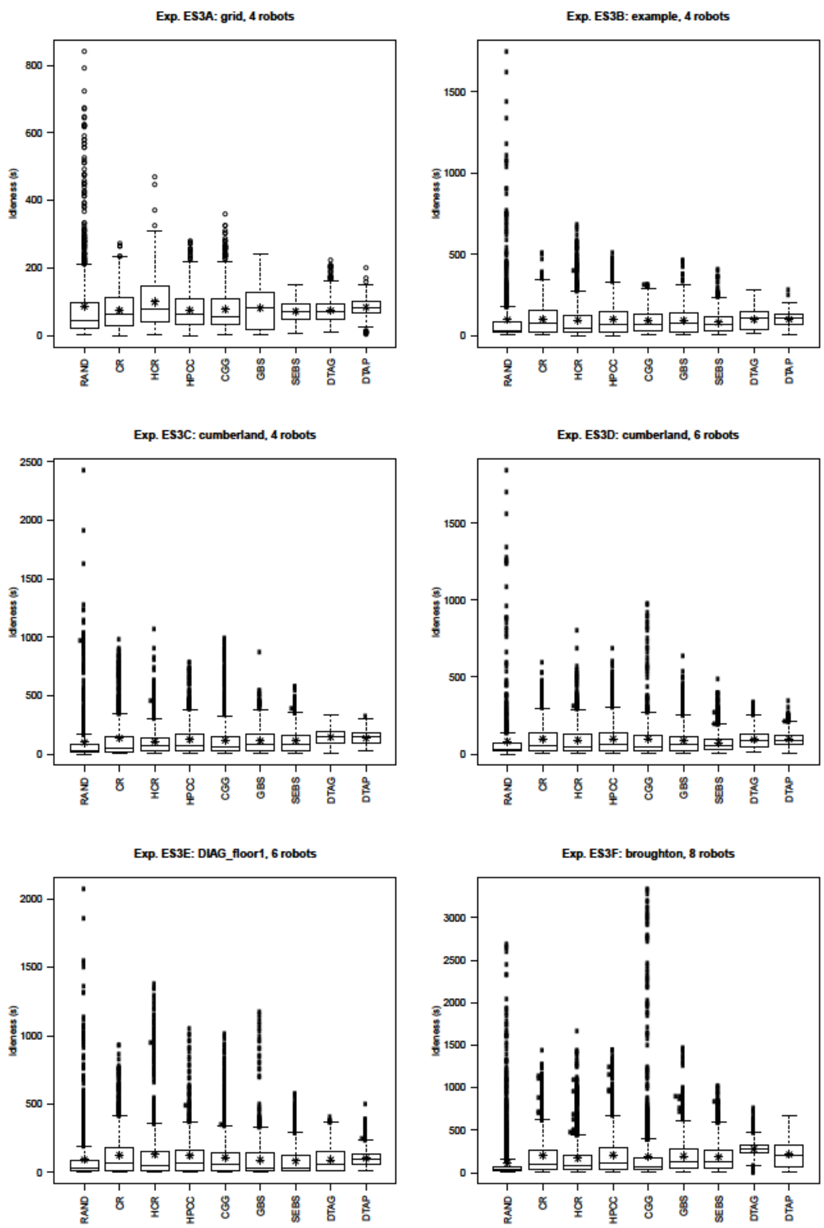

Figure 1: Comparison among all the algorithms in six scenarios.

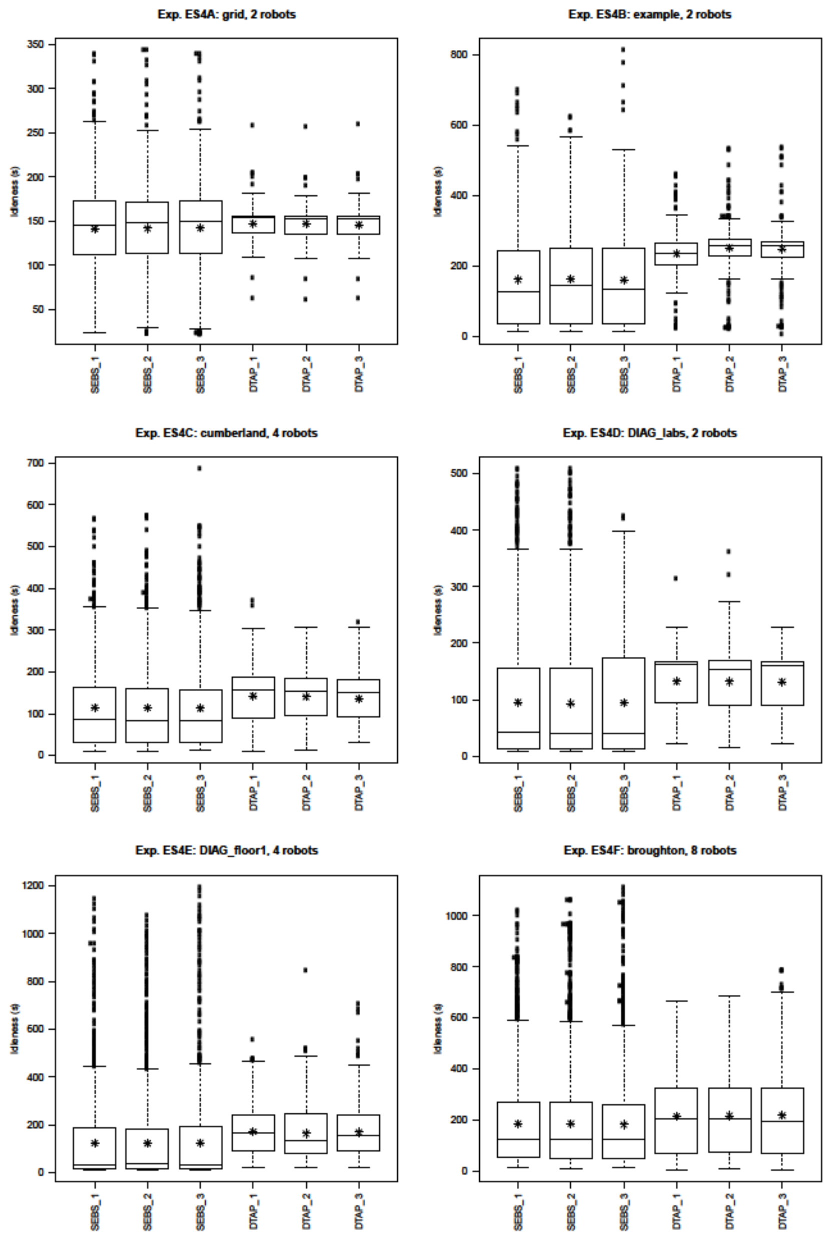

Figure 2: Comparison among all SEBS and DTAP algorithms in six scenarios with ratio size of the graph / number of robots ≥ 10.