

A Stochastic Queueing Model for Multi-robot Task Allocation

Angélica Muñoz-Meléndez^{1,2}, Pritviraj Dasgupta² and William Lenagh²

¹Computer Science Department, National Institute of Astrophysics, Optics, and Electronics,
Luis Enrique Erro 1, Sta Ma Tonantzintla, Puebla 72840, Mexico

²Computer Science Department, University of Nebraska at Omaha, Omaha NE 68182, U.S.A.

Keywords: Robot Team, Task Allocation, Demand, Stochastic Queue.

Abstract: A central problem in multi-robot systems is to solve the multi-robot task allocation problem. In this paper, a decentralized stochastic model based on stochastic queueing processes is applied for an application of collective detection of underground landmines where the robots are not told the distribution or number of landmines to be encountered in the environment. Repeat demands of inspection in the environment to ensure the accuracy of robot findings are necessary in this application. The proposed model is based on the estimation of a *stochastic queue* of pending demands that represents the alternatives of action for a robot and is used to negotiate possible conflicts with other robots. We compare and contrast this method with a decentralized greedy approach based on the distance towards the sites where inspection demands are required. Experimental results obtained using simulated robots in the Webots© environment are presented. The performance of robots is measured in terms of two metrics, completion time and distance traveled for processing a demand. Robots applying the stochastic queueing model obtained competitive results.

1 INTRODUCTION

Over the past few years, multi-robot systems have been successfully employed to solve problems in several robotic domains such as unmanned search and rescue, autonomous exploration of regions that are difficult for humans to maneuver in, automated surveillance and security, automated agriculture and domestic applications, etc. In each of these applications, the operations performed by the robots to achieve the desired objective are abstracted as tasks. A central problem in multi-robot systems is to solve the multi-robot task allocation problem (MRTA) - how to find a suitable assignment of tasks to robots so that the tasks performed by the robots can be completed in an efficient manner in terms of time and energy expended by the robots. We consider a category of MRTA problems called ST-MR-TA (single task robot, multi-robot tasks, time extended assignment) (Gerkey and Mataric, 2004), where ST stands for single-task robots, i.e., each robot is able to execute as most one task at a time, MR means multi-robot tasks, tasks that require multiple robots to be completed, and TA means time-extended assignment, problems where the information to allocate tasks to robots arrives over time. As a motivating application of such an MRTA problem, we consider a multi-robot

landmine detection scenario. A task in this scenario corresponds to a certain number of robots visiting the location of a potential landmine, not necessarily at the same time, to analyze the object using the robots' sensors. The location of potential landmines arrives dynamically and is made available to the robots. MRTA in such a scenario corresponds to the multi-city traveling salesman problem that has been shown to be NP-hard (Miller et al., 2006) (Dasgupta, 2011). Previous work in MRTA for ST-MR-TA problem considers local or market-based heuristics. In this paper, we propose to use a stochastic model called stochastic queueing to address the MRTA problem (Huang and Serfozo, 1999). Using spatial queueing is attractive for our ST-MR-TA MRTA problem as it provides a formal framework for distributed decision making by the robots so that they can respond efficiently to dynamic changes in the task distribution. We compare the performance of the spatial queueing MRTA algorithm with a greedy MRTA algorithm based on the distance to reach tasks. We have validated our algorithm on the Webots simulator using a wheeled robot called Corobot for different numbers of robots and tasks. Our experimental results done with Corobot robots on the Webots© simulator for spatially distributed task allocation scenario show that teams of 5 and 10 robots using the stochastic queueing algorithm

perform less useless movements that result in shorter traveled distances and completion time compared to the robots using the greedy algorithm.

2 RELATED WORK

The problem of MRTA has been investigated using different techniques (Gerkey and Mataric, 2004; Mataric et al., 2003), and, recently with market-based approaches (Dias et al., 2006). One of the earliest systems using for MRTA was the M+ system (Botelho and Alami, 1999). In (Gerkey and Mataric, 2004) a widely accepted taxonomy for MRTA problems is provided. The problems are classified along three dimensions: (a) single task robots (ST) vs. multi-task (MT) robots, related to the parallel task performing capabilities of robots, (b) single robot (SR) task versus multi-robot (MR) tasks, related to the number of robots required to perform a task, and, (c) instantaneous assignment (IA) versus time extended assignment (TA), related to the planning performed by robots to allocate tasks. Mataric et al. compare performance of robots teams using auction-based strategies for coordination and commitment and report that the least time is required to complete all tasks (put out all alarms) when the robots are allowed to coordinate their plans with each other as well as to dynamically change their plans (Mataric et al., 2003). The *traderbots* approach by (Dias, 2004) uses multi-round, single-item auctions for dynamic task allocation across multiple robots, while in (Jones et al., 2006) the *traderbots* approach is augmented using the Skill, Tactics, Play (STP) approach for coordinated teamwork. The MRTA problem has also been approached as an exploration problem of matching a set of robots to a set of targets using an algorithm called *PRIM-ALLOCATION* (Lagoudakis et al., 2005). Zlot and his team have also used auction-based algorithms for multi-robot task allocation (Zlot, 2006; Jones et al., 2011). The MRTA problem has also been combined with techniques from multi-agent coordination and optimization techniques such as negotiation (Viguria et al., 2007), coalition formation (Tang and Parker, 2007), reinforcement learning (Schneider et al., 2005), vector regression learning (Jones et al., 2006), Hungarian algorithm (Liu and Shell, 2011), vacancy chains (Dahl et al., 2009), and dynamic vehicle routing (Bullo et al., 2011) to improve the performance of the robots and deal with uncertainty.

3 PROBLEM FORMULATION

Our MRTA model is based on spatial queuing theory. Probability and queuing approaches provide an attractive formalism to model settings where multiple targets have to be kept under surveillance by multiple mobile units in applications such as automated surveillance, automated reconnaissance, etc. In the queuing model, the requirement of an operation by a robot on or at a target is referred to as a *demand* corresponding to that target. In our model, the demands at different targets are generated stochastically depending on the availability of target. A solution to the problem consists of each robot calculating an ordered sequence of demands based on the costs to process demands while minimizing certain metrics such as the distance traveled by the robots to process demands, or, the waiting/idle time for targets. When the spatial distribution of demands(targets) in the environment is known, a queuing approach can be enriched by applying a spatial framework. These systems generally evolve over time as Markovian processes and the robots select tasks according to a Markovian mechanism.

Let $\mathcal{E} \subset \mathbb{R}^2$ represent a bounded 2-D environment and $R = \{r_i : 1 \leq i \leq m\}$ represent a set of m mobile robots that are deployed within \mathcal{E} . $p_{r_i}(t) \in \mathcal{E}$ denotes the position of robot r_i at time t . There are n stationary targets distributed within the environment. Each target requires a subset of robots in R to operate upon it. The set of operations performed by different robots on a target is referred to as a *task*. Let $\mathcal{T} = \{\tau_i : 1 \leq i \leq n\}$ represent a set of tasks. Each task τ_i is associated with four attributes: its position in the environment $p_{\tau_i} \in \mathcal{E}$, a demand value $nd_{\tau_i} \in \mathbb{Z}$ that denotes the number of robots that need to operate on the task to complete it, a progress value $ad_{\tau_i} \in \mathbb{Z}$ that denotes the number of robots that have already serviced the task, and, a Boolean availability value $avail_{\tau_i}$ denoting whether the task is currently being serviced by a robot and is consequently unavailable. Let $\mathcal{T}_{open} = \{\tau_i \in \mathcal{T} : ad_{\tau_i} < nd_{\tau_i}\}$, $\mathcal{T}_{closed} = \mathcal{T} \setminus \mathcal{T}_{open}$, and, $\mathcal{T}_{avail} = \{\tau_i \in \mathcal{T}_{open} : avail_{\tau_i} = true\}$ represent the sets of open, closed and available tasks respectively. $d_{ij} = \|p_{\tau_i} - p_{\tau_j}\|$ is the Euclidean distance between tasks τ_i and τ_j and $\hat{d}_{ij} = \|p_{r_i}(t) - p_{\tau_j}\|$ is the Euclidean distance between robot r_i at time t and task τ_j . When a robot reaches the location of a task, one unit of the task's demand is processed and its progress increases by 1. A task is completed when its progress matches its demand, i.e., when $ad_{\tau_i} = nd_{\tau_i}$.

We represent the probability of a robot to select task τ_j after it has serviced task τ_i as an *inter-task transition matrix* M_{τ} given by:

$$M_{\tau} = \begin{pmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} \\ \dots & \dots & \dots & \dots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} \end{pmatrix}, \quad (1)$$

where $\pi_{ij} = \frac{1}{d_{ij} \sum_{j \neq i} \frac{1}{d_{ij}}}$ is the inverse of the Euclidean

distance between tasks τ_i and τ_j normalized over all tasks. Note that $\sum_i \pi_{ij} = 1$. Also, since a robot needs to service a task's demand at most once, $\pi_{ii} = 0$. Using this inter-task transition matrix, a robot has a higher likelihood of selecting available tasks that are closer than those that are further away.

The problem facing robot r_i is to select a task τ using the probabilities in the inter-task transition matrix. However, selecting the probabilities from M_{τ} does not incorporate the dynamic nature of the system manifested through robots servicing and accomplishing tasks. Therefore, each robot r_i maintains a local copy of M_{τ} denoted by M_{τ, r_i} and updates it using its own task servicing information. When a robot services a demand of task τ_j , it sets the column j in M_{τ, r_i} corresponding to task τ_j to zero, to indicate that it will not service demands from task τ_j in the future and consequently, not include τ_j in making its decisions. To select tasks, each robot r_i represents its probability of selecting a task at time t as a vector state matrix V_{r_i} , robot r_i 's initial vector state, $V_{r_i}(0)$, is given by:

$$V_{r_i}(0) = (\hat{\pi}_{i1} \hat{\pi}_{i2} \dots \hat{\pi}_{in}), \quad (2)$$

where $\hat{\pi}_{ij} = \frac{1}{\delta_{ij} \sum_{j \neq i} \frac{1}{\delta_{ij}}}$ is the inverse of the Euclidean dis-

tance between robot r_i 's initial location and the location of task τ_j , normalized over the distances of r_i to all tasks.

V_{r_i} is updated using the equation:

$$V_{r_i}(t) = V_{r_i}(t-1) \times M_{\tau, r_i}(t) \quad (3)$$

Robot r_i makes a decision about the next task to process by selecting a next task according to the highest probability of tasks in $V_{r_i}(t)$. Since the robots select tasks in a distributed manner, more than one robot end up selecting the same task. In that case, the task is allocated to the robot with the higher probability of performing the task. If more than one robot have the same probability, the task is allocated to the robot with the highest identifier.

From the probabilities in $V_{r_i}(t)$ a robot can order a *queue of open tasks* at time t , $Q_{r_i}(t) \subseteq V_{r_i}(t)$, according to the probabilities of the tasks. The values of Q_{r_i} , corresponding to the original values $\hat{\pi}_{ij}$ sorted

in descending order, represent the set of current alternatives for robot r_i at time t , as expressed in Equation 4. Note that $q_{1r_i} \in Q_{r_i}$ represents the first option for robot r_i . The stochastic queue represents, at the same time, a negotiation tool to apply in case of conflicts generated in task selection by other robots.

$$Q_{r_i}(t) = \left\{ \begin{array}{l} q_{1r_i}, q_{2r_i}, \dots, q_{n(t)r_i} : q_{jr_i} \geq q_{j+1r_i} \wedge \\ \tau_j \in \mathcal{T}_{open} \end{array} \right\} \quad (4)$$

4 STOCHASTIC QUEUEING BASED ALGORITHM FOR MRTA

The algorithm provides robots a mechanism for sorting prospective tasks according to the available knowledge. Prospective tasks are arranged in a queue of probabilities $Q_{r_i}(t)$ that represents the list of preferred tasks to be serviced by robot r_i as described in Section 3. The procedure applied for robots to allocate tasks demands is summarized in Algorithms 1 and 2.

The environment is explored applying transition matrix M_{τ} and vector states of the robot $V_{r_i}(0)$ and $V_{r_i}(t)$ calculated from the spatial distribution of tasks and location of robot, as expressed in Eqs. (1) - (2), see lines 3-16 of Algorithm 1.

Figure 1 illustrates the graph based on the transition probabilities calculated using an environment comprising 6 tasks.

5 EXPERIMENTAL RESULTS

The stochastic queueing based algorithm was compared with a greedy algorithm under identical scenarios of tasks service. In our greedy approach incoming requests of service are released by a call center to the robot team, and tasks are allocated to the nearest available robot, on a first-come first-served basis. The greedy algorithm allocates tasks to robots based on the distance between robots and demands as the first criterion, and as a second criterion in the numerical identifier of robots to solve eventual conflicts that cannot be solved by the former. In the stochastic approach robots receive a copy of the list of tasks and estimate locally the transition probabilities, vector state and queue of lists to service.

Since this research focuses on task allocation we do not deal with localization or path planning issues. We assume a robot is able to reach the point in the environment where a demand of service is active.

We conducted a set of experiments using simulated robots in the Webots© environment. We used three robot teams with 5 and 10 robots, with a varying num-

Algorithm 1: Stochastic Queueing based Task Allocation.

Input: set of tasks, \mathcal{T} ; current position of the robot, p_{r_i} .
Local variables: list of *open tasks*, \mathcal{T}_{open} ; list of available tasks \mathcal{T}_{avail} ; list of non inspected tasks for robot r_i , $\mathcal{T}_{non-insp}$; transition probability matrix, M_{τ} ; vector state of the robot, $V_{r_i}(t)$; subsequent vector state of the robot, $V_{r_i}(t+1)$; list of open available tasks selected by robot r_i , $Q_{t_{r_i}}(t)$; list of probabilities of tasks in $Q_{t_{r_i}}$, qp_{r_i} ; boolean result of an allocation, $talloc$.

- 1: **begin**
- 2: Initialize $\tau_k.ad \leftarrow 0 \forall \tau_k \in \mathcal{T}$.
- 3: Calculate $\mathcal{T}_{open} \leftarrow \{\tau_k\} : \tau_k.ad < \tau_k.nd \forall \tau_k \in \mathcal{T}$
- 4: Calculate $\mathcal{T}_{avail} \leftarrow \{\tau_k\} : p_{\tau_k} \neq p_{r_j} \forall \tau_k \in \mathcal{T}_{open} \wedge \forall r_j \in R$
- 5: Calculate $\mathcal{T}_{non-insp} \leftarrow \{\tau_k\} : \tau_k.ad = 0 \forall \tau_k \in \mathcal{T}_{open}$
- 6: Calculate the transition matrix M_{τ} from \mathcal{T} , applying Eq. (1).
- 7: Calculate the initial state vector of robot r_i , $V_{r_i}(0)$, using p_{r_i} , coordinates of tasks in \mathcal{T} , and the next state vector $V_{r_i}(1)$, applying Eqs. (2) and (3).
- 8: Build a queue of preselected tasks $Q_{r_i}(t)$ with probabilities qp_{r_i} from $V_{r_i}(1)$, applying Eq. (4).
- 9: **while** $\mathcal{T}_{non-insp} \neq \emptyset$ **do**
- 10: $talloc \leftarrow \text{SelectOneTask}(\mathcal{T}_{open}, qt_{r_i}, qp_{r_i})$.
- 11: **if** $talloc = true$ **then**
- 12: Remove the serviced tasks from the local copy of \mathcal{T} and recalculate T_{τ} .
- 13: Update the set of current open tasks \mathcal{T}_{open} .
- 14: Update $V_{r_i}(t)$ and $V_{r_i}(t+1)$ considering the current position of the robot and applying Eqs. (2) and (3).
- 15: Build a queue of preselected tasks $Q_{r_i}(t)$ with probabilities qp_{r_i} from $V_{r_i}(t+1)$, applying Eq. (4).
- 16: **end if**
- 17: **end while**
- 18: **end**

ber of tasks from 6, 12, 18 and 24. The robot model used in these experiments is based on the Coroware CoroBot robot, illustrated in Figure 2. It is equipped with a four-wheel drive base and four infra-red sensors, two located in the front and two on the sides of the robot, for avoiding collisions, and a GPS for localization. The maximum speed at which a robot can travel is 0.8m/s. The robots are able to communicate wirelessly within the environment that measures 20m \times 20m. We assume reliable communication.

The simulations were run on Webots 6.3.0 on Windows 7. The location of tasks and robots were generated randomly applying a uniform distribution. The number of demands for each task is a random number between 3 and 5. Since each task represents an inspection site for ground landmines, one task cannot be serviced twice by the same robot. The problem is considered solved if all the tasks have been inspected the associated number of demands.

We compare the performance of robot teams relying on both methods, greedy and stochastic in terms of two metrics: completion time and distance. The men-

Algorithm 2: SelectOneTask ($\mathcal{T}, \mathcal{T}_{open}, qt_{r_i}, qp_{r_i}$).

Input: set of tasks; \mathcal{T} ; set of open tasks, \mathcal{T}_{open} ; list of open tasks selected by robot r_i , qt_{r_i} ; list of probabilities of tasks in qt_{r_i} , qp_{r_i} .
Output: boolean value, $talloc$.
Global variables: set of robots R .
Local variables: float: max_qp_1 ; list of robots: Ω .
Functions: $max(x_1, \dots, x_n)$ returns the maximum value of the set $\{x_1, \dots, x_n\}$; $id(r_i)$ returns i .

- 1: **begin**
- 2: Select the first open task from qt_{r_i} , qt_{1r_i}
- 3: Broadcast qt_{1r_i} and qp_{1r_i} .
- 4: **if** $qt_{1r_i} \neq qt_{1r_j}, \forall r_j \in R : r_j \neq r_i$ **then**
- 5: qt_{1r_i} is allocated to robot r_i .
- 6: $talloc \leftarrow true$ (there are no conflicts)
- 7: **else**
- 8: **if** $qp_{1r_i} > qp_{1r_j} \forall r_j \in R : r_j \neq r_i \wedge qt_{1r_j} = qt_{1r_i}$ **then**
- 9: qt_{1r_i} is allocated to robot r_i .
- 10: $talloc \leftarrow true$ (r_i has the highest probability to reach the task).
- 11: **else**
- 12: $max_qp_1 \leftarrow qp_{1r_k} : qp_{1r_k} > qp_{1r_j} \wedge qt_{1r_k} = qt_{1r_j} = qt_{1r_i} \forall r_k, r_j \in R$.
- 13: **if** $qp_{1r_i} = max_p_1$ **then**
- 14: $\Omega \leftarrow \{r_j\} \forall r_j \in R : r_j \neq r_i \wedge qt_{1r_j} = qt_{1r_i} \wedge qp_{1r_j} = max_qp_1$.
- 15: **if** $\Omega \neq \emptyset$ **then**
- 16: **if** $id(r_i) = \max(id(r_1), \dots, id(r_x)) : r_x \in \Omega$ **then**
- 17: qt_{1r_i} is allocated to robot r_i , the robot with the highest id .
- 18: **else**
- 19: $talloc \leftarrow false$ (qt_{1r_i} is allocated to a robot in Ω with an id higher than $id(r_i)$).
- 20: **end if**
- 21: **else**
- 22: $talloc \leftarrow false$ ($r_i \notin \Omega$, qt_{1r_i} is allocated to a robot with better probability to reach it).
- 23: **end if**
- 24: **else**
- 25: $talloc \leftarrow false$ (qt_{1r_i} is allocated to a robot with better probability to reach it).
- 26: **end if**
- 27: **end if**
- 28: **end if**
- 29: **if** $talloc = true$ **then**
- 30: Update state of allocated task qt_{1r_i} in \mathcal{T} and broadcast information.
- 31: Service task qt_{1r_i} and update $qt_{1r_i}.ad$ in \mathcal{T} .
- 32: Remove qt_{1r_i} and qp_{1r_i} from qt_{r_i} and qp_{r_i} .
- 33: Update state of qt_{1r_i} , \mathcal{T} , and content of \mathcal{T}_{open} and broadcast information.
- 34: **else**
- 35: Remove qt_{1r_i} and qp_{1r_i} from qt_{r_i} and qp_{r_i} .
- 36: **end if**
- 37: **return** $talloc$.
- 38: **end**

tioned metrics were averaged across 5 runs of each set of settings. Environments are selected from a set

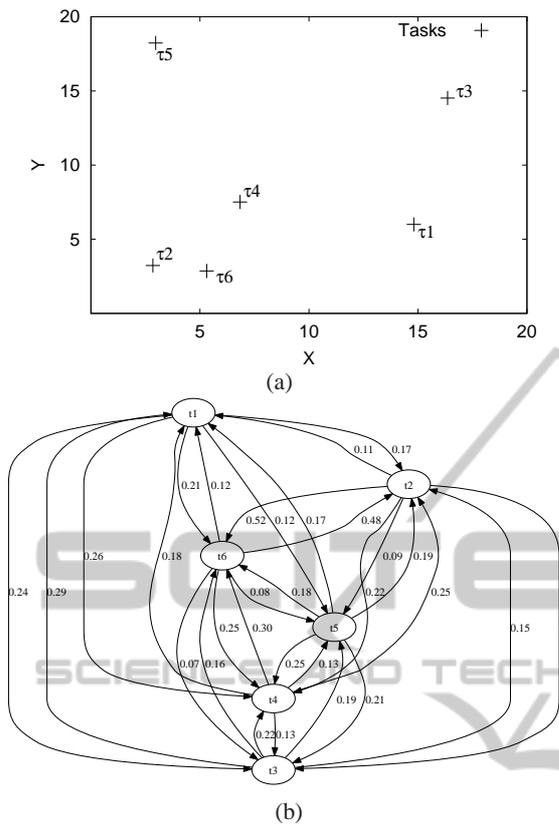


Figure 1: Example of environment containing 6 tasks, the values on the axes represent meters (a), and corresponding transition matrix (b) based on the extrapolation of inverse distances between pairs of tasks.

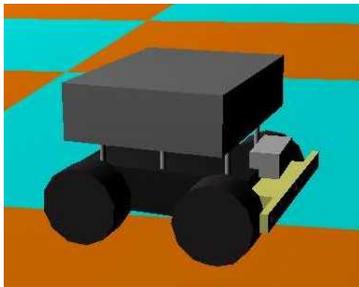


Figure 2: Corobot robot simulated in the environment Webots.

of 30 environments previously generated. Algorithms are tested under identical conditions.

Completion time is the time in the simulation when all the tasks are completed. This metric, plotted in Figure 3, indicates how long it takes to the robot team inspecting the environment. Note that, in general, robots using the stochastic algorithm invest less time to complete demands of inspection than their partners using the greedy algorithm.

The distance traveled by robots to service tasks is

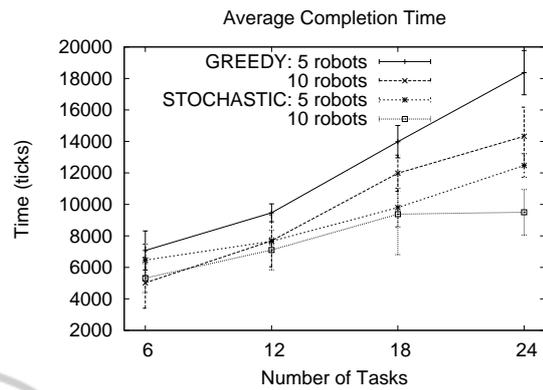


Figure 3: Average Completion Time.

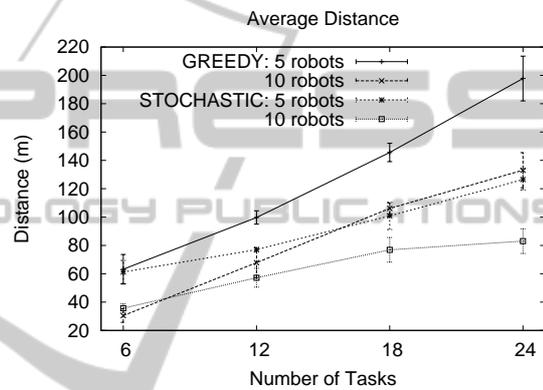


Figure 4: Average Distance.

also considered in this comparison, and it is shown in Figure 4. This plot shows how the time invested by robots when selecting tasks has a positive effect in their decisions, and that the greedy approach can face drawbacks when dealing with complex environments in terms of number of robots and tasks. Regarding the distance the robots using the stochastic algorithms travel up to 50% less than their partners using the greedy algorithm, and that happens again in the scenarios using 18 and 24 tasks.

6 CONCLUSIONS AND PERSPECTIVES

One stochastic queueing based algorithm for task allocation for robot teams is presented and compared with a simple greedy algorithm in terms of completion time and distance. These algorithms have been tested in a scenario of landmine detection where the inspection points are modeled as tasks that are serviced by different robots a number of demands.

In the near future, we plan to conduct experiments in dynamic scenarios where information of

some tasks is available *a priori*, but also information of new tasks arrives on-line, and is shared and integrated by robots in their decision making process. The stochastic queueing algorithm already recalculates a transition matrix probability during the exploration of the environment in one sense, by shrinking the alternatives for the robot. A similar procedure will be investigated to renormalize and extend the transition probability matrix.

We also plan to extend our algorithms for scenarios in which heterogeneous robots select tasks according to their skills, that represent different sensing capabilities for identifying objects of interest. Finally, we are working on the implementation of MRTA algorithms for a team of physical robots for landmine-like object detection.

REFERENCES

- Botelho, S. and Alami, R. (1999). M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proc. Int. Conf. on Robotics and Automation*, pages 1234–1239, Detroit, MI.
- Bullo, F., Frazzoli, E., Pavone, M., Savla, K., and Smith, S. (2011). Dynamic vehicle routing for robotic systems. In *Proc. of the IEEE*, volume 99, pages 1482–1504.
- Dahl, T., Matarić, M., and Sukhatme, G. (2009). Multi-robot task allocation algorithm based on vacancy chain scheduling. *Robotics and Autonomous Systems*, 57(6-7):674–687.
- Dasgupta, P. (2011). Multi-robot task allocation for performing cooperative foraging tasks in an initially unknown environment. In Jain, L., editor, *Innovations in Defense Support Systems - 2*, pages 5–20. Springer-Verlag.
- Dias, M. (2004). *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamics Environments*. PhD thesis, The Robotics Institute, Carnegie Mellon University.
- Dias, M. B., Zlot, R., Kalra, N., and Stentz, A. (2006). Market-based multirobot coordination: a survey and analysis. In *Proceedings of the IEEE, Special Issue on Multirobot Systems*, volume 94, pages 1257–1270.
- Gerkey, B. and Matarić, M. (2004). A formal analysis and taxonomy of task-allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954.
- Huang, X. and Serfozo, R. (1999). Spatial queueing processes. *Mathematics of Operations Research*, 24(4):865–886.
- Jones, E., Browning, B., Dias, M., Argall, B., Veloso, M., and Stentz, A. (2006). Dynamically formed heterogeneous robot teams performing tightly coordinated tasks. *Proceedings of the International Conference on Robotics and Automation*.
- Jones, E., Dias, M., and Stentz, A. (2011). Time-extended multi-robot coordination for domains with intra-path constraints. *Autonomous robots*, 30(1):41–56.
- Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Koenig, S., Kleywegt, A., and Tovey, C. (2005). Auction-based multi-robot routing. In *Proc. of the International Conference on Robotics: Science and Systems*, pages 343–350, Boston, MA.
- Liu, L. and Shell, D. (2011). Assessing optimal assignment under uncertainty: An interval-based algorithm. *The International Journal of Robotics Research*, 30(7):936–953.
- Mataric, M., Sukhatme, G., and Ostergaard, E. (2003). Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14:255–263.
- Miller, D., Dasgupta, P., and Judkins, T. (2006). Distributed task selection in multi-agent based swarms using heuristic strategies. In *SAB'06 Proceedings of the 2nd international conference on Swarm robotics*, pages 158–172. Springer-Verlag.
- Schneider, J., Apfelbaum, D., Bagnell, D., and Simmons, R. (2005). Learning opportunity costs in multi-robot market based planners. In *Proceedings of the International Conference on Robotics and Automation*, pages 1151–1156, Barcelona, Spain.
- Tang, F. and Parker, L. (2007). A complete methodology for generating multi-robot task solutions using asymptotic and market-based task allocation. In *Proceedings of the International Conference on Robotics and Automation*, pages 3351–3358, Rome, Italy.
- Viguria, A., Maza, I., and Ollero, A. (2007). Set: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets. In *Proceedings of the International Conference on Robotics and Automation*, pages 3339–334, Rome, Italy.
- Zlot, R. (2006). *Complex Task Allocation for Multi-robot Teams*. PhD thesis, Carnegie Mellon University.