

Summary: Distributed Task Assignment and Path Planning with Limited Communication for Robot Teams

Extended Abstract

Dario Albani*

La Sapienza University of Rome
Rome, Italy

Wolfgang Hönlig*

University of Southern California
Los Angeles, California

Nora Ayanian

University of Southern California
Los Angeles, California

Daniele Nardi

La Sapienza University of Rome
Rome, Italy

Vito Trianni

ISTC-CNR
Rome, Italy

ACM Reference Format:

Dario Albani*, Wolfgang Hönlig*, Nora Ayanian, Daniele Nardi, and Vito Trianni. 2019. Summary: Distributed Task Assignment and Path Planning with Limited Communication for Robot Teams. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)*, Montreal, Canada, May 13–17, 2019, IFAAMAS, 3 pages.

We consider multi-robot service scenarios, where tasks appear at any time and in any location of the working area. A solution to such a service task problem requires finding a suitable task assignment and a collision-free trajectory for each robot of a multi-robot team. In cluttered environments, such as indoor spaces with hallways, those two problems are tightly coupled. We propose a decentralized algorithm for simultaneously solving both problems, called Hierarchical Task Assignment and Path Finding (HTAPF). HTAPF extends a previous bio-inspired Multi-Robot Task Allocation (MRTA) framework [1]. In this work, task allocation is performed on an arbitrarily deep hierarchy of work areas and is tightly coupled with a fully distributed version of the priority-based planning paradigm [12], using only broadcast communication. Specifically, priorities are assigned implicitly by the order in which data is received from nearby robots. No token passing procedure or specific schedule is in place ensuring robust execution also in the presence of limited probabilistic communication and robot failures.

1 PROBLEM DESCRIPTION

We assume the following simplified world model. The work area is partitioned in a 4-connected grid which can be configured to represent spaces of different complexity. A grid cell $c_i \in \mathcal{C}$ might represent free space, where a robot can move and a task might appear, or an obstacle. All robots are identical and a robot $r_i \in \mathcal{R}$ is identified by its unique id i and its position in the environment.

* These authors contributed equally to the work.

DA and DN are with the Department of Computer, Control, and Management Engineering at Sapienza University of Rome, Rome, Italy. {albani,nardi}@diag.uniroma1.it. DA and VT are with the Institute of Cognitive Sciences and Technologies (ISTC), CNR, Rome, Italy. {dario.albani,vito.trianni}@istc.cnr.it. WH and NA are with the Department of Computer Science, University of Southern California, Los Angeles, CA, USA. {whoenig,ayanian}@usc.edu.

Our research was supported by NSF under grant IIS-1724392. VT acknowledges support from the project CODE, funded by ONRG under grant N62909-18-1-2093.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

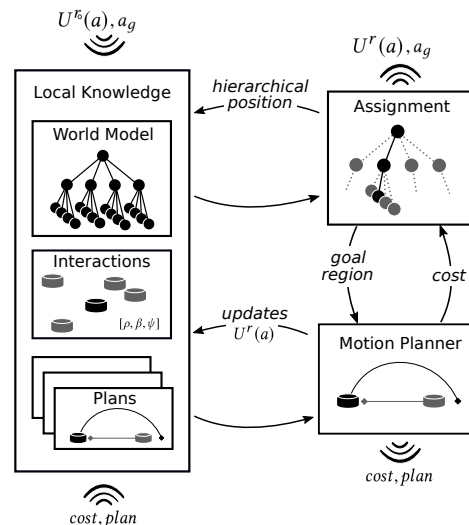


Figure 1: Interaction scheme for the proposed approach. An arrow between two blocks indicates local information passing. A sequence of \sim incoming/outgoing communication.

Robots can communicate with each other by using broadcast communication (using a probabilistic WiFi model [2, 4, 13]) without any re-broadcasting. An unfinished task $T_i \in \mathcal{T}^u$ is associated with an area a_i and requires any robot to move and work at its location (e.g., for cleaning, delivery, or maintenance). The task distribution is not known in advance, and tasks may appear at any time. Robots have limited perception of the available tasks, using the same probabilistic model that is used for communication. In this study, the primary objective is to serve the highest number of tasks.

2 APPROACH

Our approach uses three major components, see Fig. 1: the *local knowledge* available to each robot, the task *assignment* process leveraging interaction with other robots, and the *motion planner* exploiting the knowledge of other robots' plans.

The *local knowledge* is available to each robot and consists of a world model, interaction values, and motion plans of nearby robots. The world model consists of a hierarchical description of the work area represented by a quad-tree. Each leaf node relates to a cell c_i

and the root node corresponds to a region representing the whole environment. Every node in the tree is associated with an area $a_i \in \mathcal{A}$ that corresponds to a group of cells. Thus, each area of any size can contain a set of tasks that need to be served. When a task is added or a robot moves into an area, the corresponding node in the world model of robot r is updated by re-computing a local utility function, obtained as the expected number of tasks that robot r can execute in a_i :

$$U^r(a_i) = \sum_{T_i \in \mathcal{T}_{a_i}^u} \frac{1 - C^r(T_i)}{\sum_{r_o \neq r} 1 - h(r_o, T_i)}, \quad (1)$$

The term $h(\cdot, \cdot) \rightarrow [0, 1]$ is a heuristic function that estimates the normalized lower bound cost for any other robot r_o to reach task T_i or its associated area, computed using the Floyd-Warshall algorithm. The term $C^r(T_i) \rightarrow [0, 1]$ is the normalized cost for robot r to reach task T_i , as estimated by the motion planner. To this end, knowledge of the planned paths by nearby robots is used, exploiting the locally-shared plans received through communication. Finally, the local knowledge contains also the “interaction values” received from nearby robots encoding the utility of working in other areas where the other robots reside, which are used for task assignment.

The *decentralized assignment* extends a work inspired by the collective decision making abilities of honeybees [8–10], used in different contexts [3, 5, 6] and extended to MRTA [1]. We expand over previous work by considering a hierarchical representation of the world, collision-free robot motion, limited communication, and robot failures. At any time, a robot faces the choice between descending the quad-tree towards leaf nodes, exploiting current area information to reach a task to be executed, or moving up the hierarchy towards the root node, hence exploring other areas. When descending, a robot is considered “uncommitted” and can choose between the underlying four areas according to the utility of executing tasks therein. When ascending, the robot is considered “committed” to its current area and can abandon it, moving toward the parent, accessing the knowledge from neighboring areas. Ascending and descending is performed according to a probabilistic threshold-based choice, which can be tuned to balance the exploration-exploitation trade-off. Robots assigned to leaf nodes are considered allocated to task execution. Robots assigned to intermediate nodes are free to explore the area using a random walk. The assignment process is as follows. Consider a robot r to be assigned to a non-leaf node. The robot can change its assignment to other nodes according to five concurrent processes: i) spontaneous commitment, ii) recruitment, iii) spontaneous abandonment, iv) cross-inhibition, and v) self-inhibition. i) and ii) are “descending transitions” and iii) – v) are “ascending transitions”. We refer to i) and iii) as spontaneous processes, as they are determined by individual knowledge, and the others as interactive processes that take place upon interaction between pairs of robots. Specifically, spontaneous commitment (i) represents the decision of a robot to move to one of the four child nodes, assigning higher probabilities to areas with higher utility: $\gamma_i^r = k \cdot U^r(a_i)$. Conversely, spontaneous abandonment (iii) represents the probability of being assigned to the parent node: $\alpha_i^r = k(1 - U^r(a_i))$. For interactive processes, a robot r_o is randomly chosen among the known robots in a_i and a transition probability is computed according to $U^{r_o}(a_i)$. The recruitment process (ii) allows

a robot r to get recruited by other robots from the child nodes, and is defined as $\rho_i^r = h \cdot U^{r_o}(a_i)$. Self-inhibition (v) is used to cope with overcrowding which impairs robots’ motions. It is defined as: $\psi_i^r = h \cdot U^{r_o}(a_i) \cdot \mathcal{H}(|\mathcal{R}_{a_i}| - \theta_\psi)$, where $|\mathcal{R}_{a_i}|$ represents the percentage of the population of the area a_i and \mathcal{H} is the Heaviside step function, which enables the process only if the population in the i^{th} area exceeds a specific capacity. Cross-inhibition (iv) forces a robot i to abandon a region with low utility in favor of a region j with higher utility. It is used to focus the assignment to areas of high utility, allowing the robots to abandon their current area to explore one of the sibling areas. The purpose of this process is to balance poor commitments that might arise from outdated/limited knowledge, and is defined as: $\beta_{i,j}^r = h \cdot U^{r_o}(a_i) \cdot \mathcal{H}(\theta_\beta - |\mathcal{R}_{a_i}|)$, $i \neq j$, where the Heaviside function enable the process only if the population does not exceed a given capacity. Interactive transitions provide a means to tune the assignment process to the density of robots in a given area, as the probability of selecting a robot assigned to a certain area is proportional to the density of that area. Processes are weighted by two free parameters, k and h , used to define the ratio between interactive and spontaneous processes [10]. Changing this ratio allows switching between a utility-proportional deployment, to a greedy-like allocation, to a reactive deployment in which robots promptly adapt to changes in the area utility [1].

Motion planning uses a priority-based planning scheme assuming that robots plan sequentially, treating the plans of previous robots as known dynamic obstacles. We implement this approach in a distributed way, similar to previous work [12]. At the beginning, we pre-compute the all-pair shortest paths in a pre-processing step using the Floyd-Warshall algorithm. We use the all-pair shortest paths in two ways: i) to normalize the all-pair shortest paths and use the resulting values for our heuristic function $h(\cdot, \cdot)$ (see Eq. (1)); ii) to seed an admissible heuristic that speeds up the actual path planning. We use Safe Interval Path Planning (SIPP) [7], a variant of A^* by making two adjustments: i) we plan up to a user-specified planning horizon and ii) we consider any cell in our goal region a_g as goal. This allows the assignment process to reason about the expected number of tasks rather than having to consider low-level congestion information of the path planner. Our approach is fully distributed and asynchronous with no explicit synchronization.

3 RESULTS

We compare our approach against three baseline algorithms: two Greedy variants and a Contract Net Protocol [11] approach. We perform tests on environments of different size, with robot failures, and different communication settings. Results show consistently good results in all cases with respect to the number of finished tasks and task response time. In contrast, the performance of our baseline algorithms varies much more. To summarize, HTAPF is observed to be scalable and robust to communication and robot failures. Scalability stems from our bio-inspired task-assignment approach mediated from swarm robotics research, which is tightly coupled to a search-based path planning approach. Thanks to this coupling, HTAPF is able to dynamically adjust the robots’ density within the work area, granting consistently good performance across different problem instances.

REFERENCES

- [1] Dario Albani, Tiziano Manoni, Daniele Nardi, and Vito Trianni. 2018. Dynamic UAV Swarm Deployment for Non-Uniform Coverage. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 523–531. <http://dl.acm.org/citation.cfm?id=3237462>
- [2] Razvan Beuran, Junya Nakata, Takashi Okada, Lan Tien Nguyen, Yasuo Tan, and Yoichi Shinoda. 2008. A Multi-Purpose Wireless Network Emulator: QOMET. In *International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 223–228. <https://doi.org/10.1109/WAINA.2008.111>
- [3] M Caleffi, V. Trianni, and A S Cacciapuoti. 2018. Self-Organizing Strategy Design for Heterogeneous Coexistence in the Sub-6 GHz. *IEEE Transactions on Wireless Communications* 17, 11 (2018), 7128–7143. <https://doi.org/10.1109/TWC.2018.2864734>
- [4] Daniel B. Faria. 2005. *Modeling Signal Attenuation in IEEE 802.11 Wireless LANs - Vol. 1*. Technical Report TR-KP06-0118. Stanford University. <http://www-cs-students.stanford.edu/~dbfaria/files/faria-TR-KP06-0118.pdf>
- [5] Chloe Fleming and Julie A. Adams. 2018. Recruitment-Based Robotic Colony Allocation. In *Int. Symp. on Distributed Autonomous Robotic Systems (DARS)*. To Appear.
- [6] Roman Miletitch, Marco Dorigo, and Vito Trianni. 2018. Balancing exploitation of renewable resources by a robot swarm. *Swarm Intelligence* 86, 11 (2018), 3078–20. <https://doi.org/10.1007/s11721-018-0159-8>
- [7] Mike Phillips and Maxim Likhachev. 2011. SIPP: Safe interval path planning for dynamic environments. In *IEEE Intl Conf on Robotics and Automation (ICRA)*. 5628–5635. <https://doi.org/10.1109/ICRA.2011.5980306>
- [8] Andreagiovanni Reina, Thomas Bose, Vito Trianni, and James A R Marshall. 2018. Psychophysical Laws and the Superorganism. *Scientific Reports* 8, 1 (2018), 4387–8. <https://doi.org/10.1038/s41598-018-22616-y>
- [9] A Reina, JAR Marshall, V Trianni, and T Bose. 2017. Model of the best-of-N nest-site selection process in honeybees. *Physical Review E* 95, 5 (2017), 052411–15.
- [10] A Reina, G Valentini, C Fernández-Oto, M Dorigo, and V Trianni. 2015. A Design Pattern for Decentralised Decision Making. *PLoS ONE* 10, 10 (2015), e0140950–18. <https://doi.org/10.1371/journal.pone.0140950>
- [11] Reid G. Smith. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Trans. Computers* 29, 12 (1980), 1104–1113. <https://doi.org/10.1109/TC.1980.1675516>
- [12] Prasanna Velagapudi, Katia P. Sycara, and Paul Scerri. 2010. Decentralized prioritized planning in large multirobot teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4603–4609. <https://doi.org/10.1109/IROS.2010.5649438>
- [13] Shangxing Wang, Bhaskar Krishnamachari, and Nora Ayanian. 2015. The optimism principle: A unified framework for optimal robotic network deployment in an unknown obstructed environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2578–2584. <https://doi.org/10.1109/IROS.2015.7353728>