Distributed Task Allocation Optimisation Techniques

Doctoral Consortium

Joanna Turner
Loughborough University, UK
joanna.z.turner@gmail.com

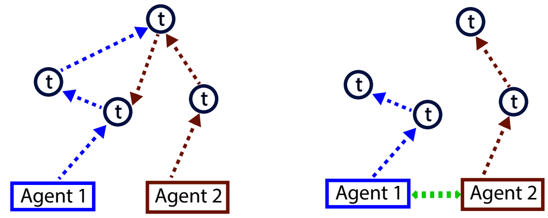
ABSTRACT

Multi-agent task allocation is a complex problem compounded by various constraints such as deadlines, agent capabilities, and communication delays. In high-stake real-time environments, such as rescue missions, it is difficult to predict in advance what the requirements of the mission will be, and what resources will be available. Yet, a fast response and speedy execution are critical to the outcome. This thesis proposes distributed optimisation techniques to tackle the following questions: how to maximise the number of assigned tasks in time restricted environments with limited resources; how to reach consensus within a reasonable time-frame; and how to allow for changeability of factors such as the availability of agents.

KEYWORDS

Distributed Task Allocation; Scheduling; Multi-Agent Systems

ACM Reference Format:
Joanna Turner. 2018. Distributed Task Allocation Optimisation Techniques. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 2 pages.



(a) Autonomous scheduling (b) Consensus

Figure 1: Tasks 't' are located in a 2D space. Dashed arrows represent the path that an agent has chosen. The dashed link between agents represents networked communication. (a) Agents independently select which tasks to service and in which order. (b) Agents communicate task assignment bids among each other and resolve conflicts based on an auction mechanism. These two phases repeat until all agents reach consensus.

1 INTRODUCTION

In multi-agent task allocation problems, the aim is to find an assignment of tasks to agents that respects all constraints while optimising one or more objectives. As the numbers of tasks and agents grow, in real-time environments it becomes computationally intractable to find the optimal solution to a task allocation problem. Therefore, suboptimal approximate solutions are found using heuristic approaches. In complexity theory, the problem is said to be NP-hard [3].

Multi-agent task allocation architectures can be classified under two main umbrellas: centralised and decentralised. Centralised architectures have the advantage of computing a global plan based on all available information, but have the main disadvantage of being a single point of failure. Distributed task allocation avoids this pitfall, there is no centralised controller and instead all agents contribute to the task allocation process. This approach also has the advantage of scalability and robustness.

Numerous frameworks have been developed to tackle the multi-agent task allocation problem. This research builds upon market-based distributed consensus algorithms. The consensus-based bundle algorithm (CBBA) [1] is a state-of-the-art distributed market-based task allocation algorithm that generates provably good approximate solutions over networks of heterogeneous agents. The algorithm iterates over two main phases. In the first phase, agents autonomously decide which tasks to add to their schedules using a scoring function. In the second phase, agents place bids on their selected tasks, share the bids with networked agents, and the agent with the highest bid (or equivalently lowest cost) wins the task. A simple example of this process is illustrated in Figure 1.

The task allocation problem considered in this research requires that agents perform one task at a time, and each agent can be assigned multiple tasks that they execute based on a schedule. Agents travel to and between tasks. The predicted cost of an agent performing a task depends on other tasks in that agent's schedule. Using the taxonomies suggested by [2], [3] and [5], this is known as the single-task (ST), single robot (SR), time-extended assignment (TA) with time windows (TW) problem with in-schedule dependencies (ID), i.e. the ID[ST-SR-TA:TW] class of task allocation problem. The task allocations are deterministic and the constraints are hard. Therefore, the scheduling of tasks cannot violate any temporal constraints. Furthermore, a task can not be assigned to more than one agent, this is referred to as a 'conflict'.

The proposed optimisation techniques were tested in MATLAB with a simulated rescue scenario situated in a large open 3D space. The number of agents ranged between 5 to 15, and the number of tasks ranged up to 250.

2 EXPLOITING THE BIDDING MECHANISM

In scenarios with time constraints and a greater number of tasks than can be assigned, a key challenge is to find an allocation of tasks to agents that allocates the highest possible number of tasks.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

Due to the high complexity of the problem, distributed task allocation algorithms use heuristic methods that generate sub-optimal solutions.

Consider a scenario in which an agent A is the only agent capable of servicing a task t_1, due to time constraints. However, A's schedule is occupied by another task t_2. Consider another agent B that would be capable of servicing t_2, but not t_1. In this situation, existing algorithms lacked the ability to have the agents reassign t_2 to agent B so that agent A may service t_1. As a consequence, only one task is allocated when two could be allocated. As the problem size increases, so does the number of potentially assignable tasks left unassigned.

One solution I devised to improve on this problem was published in [8]. The principal idea is to ensure that an agent loses the bid for a task if a feasible slot could instead be created for an unassigned task, while also ensuring that the agent keeps the task if such a slot cannot be created. Agents first check whether a task, if removed from their schedule, would create a feasible slot for an unassigned task. If so, agents then place relatively low bids on those tasks according to the proposed bidding policy. The particular bidding policy introduced allows for task reassignment chains that can involve a predefined maximum number of agents. Starting from an initial allocation, simulation results demonstrated that the proposed approach enabled agents to reassign tasks among each other to create feasible schedule space for unassigned tasks, and thereby increased the overall number of assigned tasks.

3 REDUCING THE TIME TO CONVERGENCE

In highly dynamic and time critical environments, a fast convergence time is an essential property of a distributed algorithm. With consensus-based task allocation algorithms, the time it takes for all agents to converge is determined by the number of times the two phases of the algorithm repeat (as in Figure 1) until all agents reach consensus. This in turn is largely dependent on the number of conflicting task allocations i.e. when multiple agents bid for the same tasks.

In the research published in [7], we proposed an approach to reduce convergence time while maintaining the same or a higher number of task allocations. Traditionally, agents' bids on task assignments indicate the optimality of an assignment with respect to an optimisation objective. When conflicts occur, the agent that can perform the task most optimally keeps the assignment. For example, if the objective is to minimise the latency between a task becoming available and the time at which the task is serviced, bids may be based on agents' predicted arrival time at the task location. Changes in an agent's schedule are likely to affect its arrival times and therefore the bids placed on tasks. The proposed approach resolves conflicting task allocations based exclusively on agents' relative rankings in a hierarchy. Compared with using variable bids, the proposed rank-based approach stabilises the convergence process which has the effect of speeding up the rate of convergence.

Simulation results confirmed that the proposed rank-based conflict resolution approach was able to converge faster than a benchmark method using variable bids. The findings suggest that the proposed approach is most effective and can significantly reduce the time to convergence when agents' ranks are determined by the network topology. Future work may extend this approach to assign agents' ranks based on the network topology.

4 OPTIMAL STRATEGY ADAPTATION

In consensus-based task allocation algorithms, a common approach in the autonomous scheduling phase is to have the agents determine which tasks to assign and in which order using a heuristic score function. The effectiveness of a given heuristic is dependent on various factors such as the problem constraints and the objective being optimised. Two well-known heuristics that perform well in time constrained scenarios are earliest-deadline-first (EDF) and nearest-task-first (NTF) [4]. The research we presented in [6] proposes the idea that, given a choice of heuristics, agents can locally predict and select the best task inclusion heuristic, based on the limited information shared among networked agents. The proposed method augments the task allocation algorithm with a learned prediction function in combination with a strategy switching behaviour. The method is effectively a prediction mechanism that uses past experience to select which task allocation strategy (i.e. heuristic) yields the optimal global task allocation. This method enables agents to independently adapt task allocation strategies in line with changing environmental factors, and thereby boost performance.

To test the proposed method, the prediction function was trained to predict which heuristic, between EDF and NTF, yields the highest number of task allocations. Simulation results showed that for most scenarios tested, the agents were able to predict and select the optimal heuristic using locally communicated task assignment information. The method boosted performance in terms of the overall number of allocated tasks without a significant change in number of iterations until convergence.

Future work may investigate in greater depth the variable factors such as the machine learning tool, the inputs, and the time at which agents make a prediction, with regards to how these impact the accuracy of the predictions. Furthermore, the proposed approach could be extended to adapt to changing optimisation objectives.

REFERENCES

[1] Han-Lim Choi, Luc Brunet, and Jonathan P. How. 2009. Consensus-based decentralized auctions for robust task allocation. *IEEE Transactions on Robotics* 25, 4 (2009), 912–926.

[2] Brian P Gerkey and Maja J Matarić. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research* 23, 9 (2004), 939–954.

[3] G. Ayorkor Korsah, Anthony Stentz, and M. Bernardine Dias. 2013. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research* 32, 12 (2013), 1495–1512.

[4] Hakim Mitiche, Dalila Boughaci, and Maria Gini. 2015. Efficient Heuristics for a Time-Extended Multi-Robot Task Allocation Problem. In *First International Conference on New Technologies of Information and Communication (NTIC).* IEEE.

[5] Ernesto Nunes, Marie Manner, Hakim Mitiche, and Maria Gini. 2017. A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems* 90 (2017), 55–70.

[6] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. 2018. Distributed Strategy Adaptation with a Prediction Function in Multi-Agent Task Allocation. In *17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018).* IFAAMAS.

[7] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. 2018. Fast Consensus for Fully Distributed Multi-Agent Task Allocation. In *The 33rd Symposium On Applied Computing (SAC '18).* ACM/SIGAPP.

[8] Joanna Turner, Qinggang Meng, Gerald Schaefer, Amanda Whitbrook, and Andrea Soltoggio. 2017. Distributed Task Rescheduling With Time Constraints for the Optimization of Total Task Allocations in a Multirobot System. *IEEE Transactions on Cybernetics* 99 (2017), 1–15.