Multi-robot task allocation problem: current trends and new ideas

Mattia D'Emidio[1], Imran Khan[1]

Gran Sasso Science Institute (GSSI)
Via F. Crispi, 7, I–67100, L'Aquila (Italy)
{mattia.demidio,imran.khan}@gssi.it

Abstract. The use of robotic fleets, a.k.a. *multi-robot computing systems*, is a de–facto standard in a variety of real-world applications. Efficiently solving the *multi-robot task allocation* (MRTA) problem is perhaps one of the most fundamental feature of such systems. In this paper, we attempt at overviewing recent research on the matter. In particular, we first briefly survey some of the most interesting results of the last decade. Then, we focus on a prominent variant, namely the *single-task single-robot* task allocation problem with *hard temporal constraints* (ST-SR-HC, in short), and on *auction-based* algorithms, which have been shown to be rather effective solutions in the ST-SR-HC setting. To this regard, we discuss the state-of-the-art and highlight some ideas we are currently exploring to improve the best algorithm of the category, i.e. TESSI. We provide preliminary experimental evidences of how such ideas seem to lead to better balancing number of *performed tasks, distance* and *makespan* optimization objectives.

1 Introduction

The use of robotic fleets is becoming increasingly compelling for many reasons. Chiefly, there has been a significant decrease in the cost of robotic devices (e.g. drones or ground robots), while at the same time hardware continues to improve. This makes so–called *multi-robot computing systems* (MRS) a pertinent option for many applications of critical practical importance as, e.g., exploration of hazardous zones or rescue of people in emergency contexts. In this context, many different problems/scenarios have been studied. Robots might be asked to form desired geometric patterns [8], to explore unknown zones [2]. They can be constrained to move on an Euclidean plane [1], or on a graph [4, 6], which can be either known in advance or not [3]. They can be equipped with different kinds of hardware [7], and be able to act synchronously or not [5].

One of the major thread of research in the mentioned area, due to its high industrial and academic relevance, is that regarding the *multi-robot task allocation* (MRTA) problem, i.e. the problem of partitioning tasks (to be performed) across robots such that some objective of interest is optimized [11]. Unfortunately, even the basic version of the problem, where, given a set R of robots, and a set T of tasks, one wants to find an *assignment* of the tasks in T to the robots in R

such that the total time for completing all the tasks (i.e. the *makespan*) is minimized, is NP–Hard [11] For this reason, a plethora of alternative approaches have been proposed over the years, ranging from polynomial time approximation algorithms to techniques that chase good solutions via heuristics with no provable guarantees. The situation is even more complicated, algorithmically speaking, when it comes to incorporate further practically relevant aspects (e.g. precedence constraints). Nonetheless, a countless number of variants of the general MRTA problem have been defined and investigated in last decade.

In this paper, we attempt at overviewing the results of such investigation. In particular, we first briefly survey some of the most interesting works on specializations of the MRTA problem (Section 2). Then, we focus on a prominent variant, namely the *single-task single-robot* task allocation problem with *hard temporal constraints* (ST-SR-HC, in short), and on *auction-based* algorithms, which have been shown to be rather effective solutions w.r.t. the ST-SR-HC setting [11] (Section 3). To this regard, we discuss the state-of-the-art and highlight some ideas we are currently exploring to improve the best algorithm of the category, i.e. TESSI. We provide preliminary experimental evidences of how such ideas seem to lead to better balancing number of *performed tasks, distance* and *makespan* optimization objectives (Section 4).

2 Variants of the MRTA problem

In the robotic field, the MRTA problem is always considered in the realistic setting where both robots and tasks are spread within a physical environment [11]. In this case, time for moving robots where the tasks have to be performed needs to be considered and, possibly, optimized. Other main flavors of the problem are mostly due to the variance in: i) abilities of robots; ii) nature of tasks; iii) system goals; iv) computational paradigm. In what follows, we briefly summarize the most commonly considered distinctions w.r.t. these four directions.

Regarding i), the broadest classification concerns the amount of tasks that can be executed by a robot at a time. In particular, a robot can either perform at most one task at a time (*single-task robot* setting, or ST) or more than one task at a time (*multi-task robot* setting, or MT) [11]. Moreover, sometimes robots can have an upper bound on the maximum allowed operational time [9, 12].

Concerning ii), symmetrically w.r.t. i), the main distinction is usually due to the number of robots that are required to execute a task, i.e. a task can either be accomplished by a single robot (a.k.a. *single-robot task* setting, or SR) or require more than one robot (*multi-robot task* setting, or MR) [11]. Secondary classifications are induced by domain-related constraints on *when* or *how* tasks must be completed, i.e. *temporal* and *ordering* constraints [11]. On the one hand, temporal constraints require a task to be executed within a specific *time window*, i.e. a time interval that implicitly imposes a *latest starting time* and a *deadline* within which the task must be completed. In more details, deadlines can be *hard* (HC setting) or *soft* (SC setting) where in the HC (SC, resp.) case the execution of the task outside the corresponding time window is not allowed (allowed by a predefined fixed amount, resp.) [11]. On the other hand, ordering constraints restrict the execution of some tasks to be dependent on the completion of some other tasks (a.k.a, *precedence constraints*) [11].

Regarding iii), the main difference is induced by the number of objective functions to be optimized (single vs multi-objective) [11]. Commonly adopted optimization goals in the area are temporal objectives (e.g., *makespan* or *tardiness* [9]), spatial objectives (e.g. distance traveled [10]), cardinality of the required robots (e.g. minimizing number of used robots [11]).

Finally, concerning iv), as in many other optimization problems, heavy differences in the problem are induced by the possibility of having a central coordinating entity (*centralized* setting) or not (*decentralized* setting). In the former case, there is a robot or a ground station responsible for allocating tasks, while in the latter the computation is distributed to the robotic fleet itself. Centralized computation offers the typical benefits of better optimizing efforts, resources, cost and time, since the central entity has complete environment awareness.However, it suffers from both the well-known drawbacks, e.g. *single point of failure*. Distributed approaches are often preferred since, despite they require more complex coordination strategies, they achieve higher reliability and scalability [9].

3 The ST-SR-HC problem

In this section, we focus on the ST-SR-HC problem, which can be defined as follows. We are given a set of n ST robots and a set of m SR tasks whose time window is of type HC, with $n < m$. Each robot has an associated *budget* in terms of operational time and both tasks and robots are embedded in a 2D plane, i.e. they have an associated set of coordinates. The objective is to assign the largest amount of tasks to robots such that: i) temporal constraints are satisfied; ii) makespan and distance covered by the robots are minimized.

Of the wide variety of approaches addressing the different variants of MRTA problem, only a handful of them deal with the ST-SR-HC setting. Among these, *auction-based* algorithms enjoyed a wide range of popularity [11]. Auction-based algorithms are market inspired and their behavior can be summarized as follows. The computation takes place in rounds and each round consists of two phases: an *announcement* phase and a *selection* phase. During the announcement, a set of *available tasks* T is broadcasted to be ready for execution to all robots, by a so–called *auctioneer* which is assumed to become aware of T by monitoring the environment. Then, each robot computes a so–called *bid* for each available task which is transmitted to the auctioneer. The bid is basically a number, computed by a robot by considering both constraints and system goals, i.e. the impact that allocating a task to the robot has on the considered objectives. Bids are then collected by the auctioneer and processed to select the so-called *winning bidder* (robots are often referred to as *bidders* here), according to some optimization criteria, e.g. by choosing the one who has sent the bid of minimum value. Finally, the auctioneer sends a message to the winning robot to confirm the allocation of the corresponding task. The robot accordingly adds the task to its *schedule*, i.e. a data structure storing assigned tasks along with their temporal information.

Some of the most relevant algorithms of the area are of *single sequential item auction* (SSI) type (they allow bidding on a single task at the time [9]). They are usually preferred over other approaches (e.g. *combinatorial auction* solutions [12]) since they are more parsimonious w.r.t. communication and computational resources, therefore suited to be used in MRS where robots are less powerful in this sense. The state-of-the-art SSI algorithm is TESSI, which has been shown to outperform previous methods in producing quality solutions w.r.t. number of allocated tasks, makespan and distance objectives [10].

4 Beyond TESSI

TESSI requires a robot to check the internal *consistency* of the current schedule whenever a bid needs to be placed, where a schedule is said to be consistent if all tasks can be performed without violating any temporal constraint [10]. To this aim, each schedule is stored in a suited data structure, called STN, whose space complexity is linear in the number k of tasks currently associated to the schedule, and each of the above consistency checks take $O(k^3)$ time, which is clearly not practical whenever k is large [10]. To overcome this limit, we are investigating the possibility of devising an enhanced version of the STN that, by storing some additional data, can allow faster consistency tests. In particular, by exploiting some properties of the ST-SR settings (e.g. the STN, stored locally on each ST robot, cannot have two or more events, i.e. tasks, occurring in parallel to each other), so far we are able to show that consistency checking can be done in $O(k)$ time at the price of an extra $O(k)$ of space (which asymptotically does not affect the space requirements of the original STN). Furthermore, we are also studying how to better optimize the distance objective by modifying the criterion used by TESSI for placing bids. The intuition here is to weight "more" the distance traveled by a robot in such criterion. More details on the above modifications will be provided in an extended version of this paper.

To assess the effectiveness of this enhanced version (called ETESSI) against TESSI, we carried out a preliminary experimental evaluation. We considered both real-world and synthetic inputs and various settings. Our results (Fig. 1) suggest ETESSI to be computationally more parsimonious than conventional TESSI (as theoretically expected), while at the same time being slightly better w.r.t number

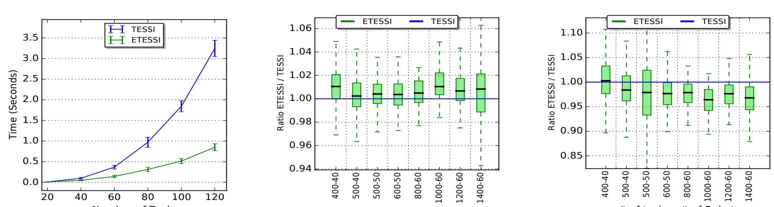

Fig. 1: Results of the experimental evaluation of ETESSI against TESSI: (left) CPU Time, (middle) number of tasks scheduled by ETESSI over number of tasks scheduled by TESSI, (right) maximum distance travelled by robots using ETESSI over maximum distance travelled by robots using TESSI.

of scheduled tasks and distance objectives, with a similar makespan (results w.r.t. latter omitted due to space limitations). To extend our work, we are planning to: i) expand the theoretical foundations of ETESSI and improve its performance; ii) consider other settings, e.g. SC or dynamic settings where rescheduling tasks is allowed; iii) enlarge our experimental evaluation.

Bibliography

[1] S. Das, P. Flocchini, G. Prencipe, N. Santoro, and M. Yamashita. The power of lights: Synchronizing asynchronous robots using visible bits. In *Proc. of 32nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 506–515, 2012.

[2] M. D'Emidio, D. Frigioni, and A. Navarra. Exploring and making safe dangerous networks using mobile entities. In *Proc. of 12th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW)*, volume 7960 of *Lecture Notes in Computer Science*, pages 136–147. Springer, 2013.

[3] M. D'Emidio, D. Frigioni, and A. Navarra. Explore and repair graphs with black holes using mobile entities. *Theor. Comput. Sci.*, 605:129–145, 2015.

[4] M. D'Emidio, D. Frigioni, and A. Navarra. Characterizing the computational power of anonymous mobile robots. In *Proc. of 36th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 293–302, 2016.

[5] M. D'Emidio, D. Frigioni, and A. Navarra. Synchronous robots vs asynchronous lights-enhanced robots on graphs. In *Proc. of 17th Italian Conference on Theoretical Computer Science (ICTCS)*, volume 322 of *Electronic Notes in Theoretical Computer Science*, pages 169–180. Elsevier, 2016.

[6] M. D'Emidio, G. D. Stefano, D. Frigioni, and A. Navarra. Improved protocols for luminous asynchronous robots. In *ICTCS*, volume 1720 of *CEUR Workshop Proceedings*, pages 136–148. CEUR-WS.org, 2016.

[7] P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Computing without communicating: Ring exploration by asynchronous oblivious robots. *Algorithmica*, 65(3):562–583, 2013.

[8] N. Fujinaga, Y. Yamauchi, H. Ono, S. Kijima, and M. Yamashita. Pattern formation by oblivious asynchronous mobile robots. *SIAM Journal on Computing*, 44(3):740–785, 2015.

[9] L. Luo, N. Chakraborty, and K. Sycara. Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, 12(3):876–888, 2015.

[10] E. Nunes and M. L. Gini. Multi-robot auctions for allocation of tasks with temporal constraints. In *Proc. of 29th AAAI conference on aritifical intelligence*, pages 2110–2116, 2015.

[11] E. Nunes, M. Manner, H. Mitiche, and M. Gini. A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems*, 90:55–70, 2017.

[12] W. Zhao, Q. Meng, and P. W. Chung. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE transactions on cybernetics*, 46(4):902–915, 2016.