

RESEARCH ARTICLE

Multi-Robot Coalitions Formation with Deadlines: Complexity Analysis and Solutions

Jose Guerrero^{1*}, Gabriel Oliver¹, Oscar Valero¹

University of the Balearic Islands, Mathematical and Computer Science Department, Palma de Mallorca, Illes Balears, Spain

¹ These authors contributed equally to this work.

* jose.guerrero@uib.es



Abstract

Multi-robot task allocation is one of the main problems to address in order to design a multi-robot system, very especially when robots form coalitions that must carry out tasks before a deadline. A lot of factors affect the performance of these systems and among them, this paper is focused on the physical interference effect, produced when two or more robots want to access the same point simultaneously. To our best knowledge, this paper presents the first formal description of multi-robot task allocation that includes a model of interference. Thanks to this description, the complexity of the allocation problem is analyzed. Moreover, the main contribution of this paper is to provide the conditions under which the optimal solution of the aforementioned allocation problem can be obtained solving an integer linear problem. The optimal results are compared to previous allocation algorithms already proposed by the first two authors of this paper and with a new method proposed in this paper. The results obtained show how the new task allocation algorithms reach up more than an 80% of the median of the optimal solution, outperforming previous auction algorithms with a huge reduction of the execution time.

OPEN ACCESS

Citation: Guerrero J, Oliver G, Valero O (2017) Multi-Robot Coalitions Formation with Deadlines: Complexity Analysis and Solutions. PLoS ONE 12 (1): e0170659. doi:10.1371/journal.pone.0170659

Editor: Long Wang, Peking University, CHINA

Received: May 19, 2016

Accepted: January 9, 2017

Published: January 24, 2017

Copyright: © 2017 Guerrero et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All the source codes used in this paper and all the obtained results are available from <https://github.com/joseGuerreroUIB/PLOSONEMulti> and from <https://github.com/joseGuerreroUIB/RoboCoT>.

Funding: This work has been partially supported by project DPI2014-57746-C03-2-R and FEDER funds.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Systems with two or more mobile robots (multi-robot-systems) can perform tasks that with one robot would be impossible to carry out or would take much more time. Moreover, such systems are more robust, scalable and flexible than those with only one robot. However, a lot of new challenges and problems must be solved before taking advantage of the potential benefits of multi-robot systems. Among all possible issues that arise in a natural way in multi-robot systems, this paper focuses on the problem commonly referred to as “Multi-robot Task Allocation” (MRTA) which consists in selecting the best robots to execute each one of the tasks that must be performed. MRTA is still an open problem, specially when several robots (a coalition) have to execute concurrently a task before a deadline. Furthermore, in most cases, this problem is either NP or needs a very long time to provide an optimal solution. To address this problem, auction-like algorithms are widely used, where the robots negotiate between them to achieve a solution.

This paper is focused on the formation of multi-robot coalitions under an auction-based framework, where the tasks must be executed before deadlines. In this case the knowledge of how long the coalition will take to finish a task is essential in order to determine the proper alliances. The time needed to finish a task is frequently unknown because it depends on many complex and dynamic factors, among them this paper will center its attention on the physical interference between robots. The interference appears when some individuals of a community compete for a shared resource. In multi-robot systems, a frequent situation occurs when two or more robots need to access the same point simultaneously. It has been demonstrated in different studies [1, 2], that the physical interference has an important impact on the system performance. This effect can be dramatic when the system has to address tasks with deadlines because the interference increases the time needed by the coalition to finish the task and, therefore, diminishes the coalition utility.

This paper provides the first mathematical formalization of a MRTA problem with deadlines that takes into account the interference effect. Moreover, it analyzes the complexity of the resulting optimization problem by means of the coalition utility and interference. As will be demonstrated, depending on the interference function and on the task characteristics, the allocation can be faced up as an integer linear optimization problem. The interference function values have been obtained from the execution of a large number of experiments with physical simulators in the authors' previous works [3]. Recently, other authors (see [4]) gave a mathematical model for MRTA with interference but without deadlines. Another example of the growing interest for this field is the Nunes' et al. work [5] where a formalization of the task allocation with deadlines is given only from a theoretical point of view, without neither implementing nor testing their models. Our paper extends this work and carries out an extensive number of experiments to compare the first two authors' previous works, given in [3], regarding to the optimum solution. Thus, the experiments show that the algorithms given in [3] can reach up to more than an 80% of the optimal solution with a very low execution time. In the light of the formalization and, as a secondary contribution of this paper, a new auction-like MRTA algorithm, called Multi Objectives Double Round Auction (MDRA), is proposed and tested. The experimental results show that this new method outperforms the previous ones.

The interference can cause a high impact when implemented in real robots and, as Rubenstein et al. points out in [6], is crucial to implement motion strategies for avoiding failures. Despite this need, this paper is only focused on MRTA problems ignoring the lower motion issues. As future work, we will address how to join both our MRTA algorithms and motion control algorithms in an unique strategy.

The rest of the paper is organized as follows: first, Section MRTA Problem Fundamentals reviews the basic formal definitions of the MRTA problem that can be found in literature; Section Previous work on coalition formation is focused on the algorithms formerly proposed to deal with multi-robot coalitions proposed in literature; Section Coalition formation complexity analysis provides a new formalization for MRTA problem that take into account interference; Section Task Allocation: Double Round Auction for tasks with deadline reviews the authors' previous work based on auctions and, in addition, introduces the new MDRA algorithm; the experimental results are shown and analyzed in Section Experimental results and finally, conclusions and future work are presented in Section Conclusions and future work.

MRTA Problem Fundamentals

In this section we review the fundamental aspects of multi-robot task allocation proposed in the literature. Specially, we focus on the problem of the complexity and on the relationships with other knowledge fields such as multi-agent or task-scheduling. As will be seen, most

MRTA problems for the coalition formation are NP-hard, thus, making necessary the use of heuristic functions for obtaining a solution close enough to the optimum value in a moderate time. The MRTA methods can be classified in three groups: centralized approaches, swarm-based methods and auction algorithms. The last two approaches are the most used nowadays, but as will be seen, these methods present serious limitations when the robots must form coalitions to face up tasks with deadlines.

Some taxonomies for MRTA systems have been proposed in the literature, but probably one of the most broadly used is authored by Gerkey and Mataric [7]. Gerkey's taxonomy breaks down the MRTA problem in three orthogonal axes:

- Single-Task robots (ST) vs Multi-Task robots (MT): in a ST problem every robot can only execute a task at a time. While in contrast, in a MT problem the same robot can execute several tasks simultaneously.
- Single-Robot tasks (SR) vs Multi-Robot tasks (MR): if two or more robots can collaborate to carry out the same task, the problem is MR. The group of robots assigned to a task is called coalition or working group. Otherwise, if only one robot can be assigned to each task at the same time, the problem is called SR.
- Instantaneous assignment (IA) vs. Time-Extended assignment (TE): in an IA problem, the task allocation is made without taking into account future allocations, that is, without planning the future. Conversely, in a TE problem, a planner considers the future assignments.

This taxonomy will be used to classify the different MRTA methods exposed from now on. Regardless of Gerkey's category, the algorithms can be preemptive or non preemptive. A preemptive algorithm is a method where the robots can change their current assigned tasks before ending them.

A single robot per task: the SR problem

This section reviews some definitions used to address the SR problem. As proved by Gerkey in [8], the ST-SR-IA algorithms can be described in terms of the well-known Optimal Assignment Problem (OAP). The OAP is defined as: given n agents (robots) and m tasks to carry out, each agent can only be assigned to one task and each task requires only one agent. For each couple (agent-task), a value is defined that forecasts the agents performance for that task, that is, this value models the agents utility regarding that task. The goal is to assign an agent to each task to maximize the total utility U . This goal function is given as:

$$U = \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq m} \alpha_{ij} U_{ij} w_j, \tag{1}$$

where $\alpha_{ij} = 1$ if the task i is assigned to agent j , and $\alpha_{ij} = 0$ otherwise; U_{ij} is the utility gained by the system when agent j is assigned to task i ; and w_j is the weight or importance of the task j . Thus, w_j represents the priority of the task. The Hungarian's method allows to get the optimal solution for this kind of problems in a time $O(nm^2)$ through a dynamic programming centralized method [9]. We have to note that, if the utilities are interrelated, that is, if the agent's utility for a task depends on the tasks previously assigned to that agent, then the assignment is a NP-hard problem of type ST-SR-TE.

Multiple robots per task: MR problem

In this section we introduce the definitions for MR problems, where the robots can form coalitions for carrying out a task. In this case, the definition for the Set Partitioning Problem (SPP)

can be given in the following way according to [10]: given a finite set E , a family F of all feasible subsets of E and a utility function $U : F \rightarrow \mathbb{R}^+$, the goal is to find a family X of F that has the maximum utility and which is a partition of E . The SPP problem is NP-hard, as it was proved in [11]. The Set Covering Problem (SCP) is similar to SPP but it is enough that X is a covering of E , being this a SCP NP-complete problem [12]. Both problems can be viewed as general cases of the multi-robot coalition formation problem, where E is the set of robots, F is the set of feasible robot coalitions formed (each one with a task assigned), and U is the utility function assigned to each coalition [13]. If time constraints are needed, the task deadlines should be considered just as one of the conditions to validate a family F of subsets of E . On the one hand, the SCP definition allows to allocate more than one task to the same robot and, therefore, it can address MT problems. On the other hand, the SPP definition is more restrictive because each robot can only belong to a single coalition at the same time and, therefore, these problems are ST. In [14], it is proved that, using the SPP and SCP definition, the multi-robot coalition formation problem is not only NP-hard but, in general, it cannot be approximated by a factor $O(m^{1-\epsilon})$, for any $\epsilon > 0$, where m is the number of tasks. This means that, unless $P = NP$, the solution obtained by means of an algorithm with a complexity bounded by a polynomial function will always be $O(m^{1-\epsilon})$, for any $\epsilon > 0$ worst than the optimal solution. As can be observed, the complexity of the coalition formation algorithms is very high. This fact justifies the great number of *ad-hoc* solutions that can be found in the literature.

In [15], a new coalition formation algorithm is proposed, based on the SCP and SPP definitions, where the maximum number of agents per coalition (k) is fixed beforehand. The algorithm complexity is $O(n^k m)$, where n is the number of agents and m the number of tasks. In [13], it is described a new algorithm with a complexity $O(n^k m)$ that extends to multi-robot systems the previous algorithms proposed only for multi-agents in [15]. In [14], Service and Adams improve the above cited algorithm given in [13], with a $O(n^2 m)$ complexity. In any case, this solution is, in general, not optimal yet. Only, if the robots can be grouped in j homogeneous groups, in such a way that each robot only belongs to one of these groups, Service and Adams' algorithm provides an optimal solution, with a complexity equal to $O(n^{2j} m)$. In this algorithm, the task must be assigned to a certain number of robots to start its execution and only when this number is reached the task i is executed with a utility u_i . In next sections, a new method, based on the SPP's definition, will be explained where the number of robots k do not need to be known beforehand, the robots are not homogeneous and the utility u_i depends, among other factors, on the coalition size. Therefore, Service and Adams' algorithm is not suitable for these kind of environments. It is also remarkable the Agarwal et al. contribution [16], where genetic algorithms optimize multiple objectives simultaneously for the coalition formation. These objectives can be the distance traveled by the robots, the number of finished tasks, or any other system parameter. Due to its complexity, the evolutive algorithms cannot deal with neither dynamic tasks nor tasks with time restrictions. Nevertheless, the Agarwal's evolutive methods are very good candidates to improve somehow the real time system's performance.

Scheduling: ST-MR-TE Problems

The above definitions, based on SPP or SCP, do not take into account any temporal condition for the tasks, that is, they cannot deal with any scheduling consideration. On account of [17], the coalition formation problem can be defined in terms of the Job Shop Scheduling (JSS) problem as follows: given a set of m jobs $J = \{J_1, J_2, \dots, J_m\}$ and a set of n machines, each job J_i has m_i associated operations $O_i = \{O_{i1}, O_{i2}, \dots, O_{im_i}\}$ each one of them with an execution time for each machine T_{ij} . The system's goal is to decide (schedule) in which order the tasks must

be carried out on the machines to minimize the execution time. Moreover, this formulation allows defining a precedence order among operations and, even more, set a maximum execution time bound for the tasks. Balas et al. in [18] propose a method that take into account all these constraints in a multi-agent scenario. It was demonstrated that, in general, the JSS problem is NP-hard. The JSS problem can be adapted to a multi-robot problem in such a way that the system should decide the execution order of the tasks assigned to each robot to minimize, among other factors, the total execution time, the traveled path or the robots' energy consumption. Thus, these problems are classified as TE, following the Gerkey's taxonomy. For example, Jones et al. in [19] use genetic algorithms to solve a variation of the JSS problem for multi-robot systems.

The Vehicle Routing Problem with Time Windows (VRPTW) can also be considered as a scheduling problem, where a set of vehicles must transport goods to several "clients". Each vehicle has a load capacity and each client demands a quantity of goods that have to be delivered within a time window. The system goal is to find a route, as short as possible, that satisfies the clients constraints with the fewest number of vehicles [20]. As the JSS definition, the VRPTW problem is NP-hard too. A variation of the VRPTW is the Tasks with Overlapping Time Window Problem (TOTWP) where the time window can be overlapped. Nunes et al. in [21] try to solve the TOTWP but with non realistic environments and with a single robot per task (SR problem).

Finally, it should be noticed that MT problems, where a robot is able to execute several tasks at the same time, are not feasible in most of the multi-robot missions because, in general, the robots must be located in an specific place to carry out a task.

Previous work on coalition formation

This section focuses on the coalition formation methods that can deal with task deadlines. Depending on how the task allocation process is distributed among the robots, the coalition formation methods can be classified into the following categories: centralized systems, self-organized approaches (swarm intelligence) and auction methods. The Gerkey's taxonomy will be applied into each one of these groups.

Centralized systems

In the centralised systems all the information about the robots or the environment characteristics is sent to a central agent that takes all the decisions. Typically, these systems use techniques such as, linear programming or heuristic searches on graphs to find the optimal solution or the closest to the optimal one. Koes et al. [22] utilize linear programming to address simultaneously task allocation, path planning and task scheduling. Yu and Cai, in [23], propose a centralized mechanism called Heterogeneous Interactive Cultural Hybrid Algorithm (HICHA) where, among other techniques, use genetic algorithms. Another centralized system has been introduced by Smith and Bullo in [24], where different task allocation methods are analyzed as a function of the arrival frequency of the tasks.

Swarm Intelligence

Swarm methods are inspired by insect colonies behavior, such as bees or ants, where a global action emerges from the interaction between very simple entities. These methods belong to a more general category of systems called self-organized systems which are defined by Cao et al. [25] as follows: "[...] systems of non-intelligent robots exhibiting collectively intelligent behavior". That is, they consist on system without any central node, where the cooperative behavior emerges from the interaction of very simple behaviors running on each robot.

The main problem of swarm systems is its lack of global knowledge about the decision made by other robots. Due to this fact, the solutions provided by the swarm methods are far away from the optimal one, moreover, they can only address SR problems of limited complexity such as foraging or cleaning with homogeneous robots. This restriction is superseded in some already explained swarm systems by the authors [26, 27]. Due to its communication complexity, Low's et al. algorithm [27] does not show some advantages of swarm methods. Furthermore, in Low's system the robots must be homogeneous, and it cannot handle task time restrictions.

One important issue is how swarm system manage the physical interference effect between robots. This aspect, that will be addressed in Section Physical interference between robots, has already been broadly faced up in several papers, such as [28, 29]. Khaluf's swarm methods [30] can manage deadlines but without taking into account the interference effect.

Auction methods

Auction methods use communication protocols between robots to implement an explicit negotiation about which task must execute each robot or coalition of robots. These protocols are based on the Contract Net Protocol (CNP) proposed by Smith in 1980 [31]. Smith's work defines a mechanism to assign tasks to communication nodes (machines) connected between them, that must solve some cooperative problem in a distributed way. When a node, called manager or auctioneer, finds or generates a new task, sends a message to announce it before starting the execution. The other nodes, called bidders, send to the auctioneer a value, called bid, that indicates how suitable are they for executing the new task. Once the manager node has received all the bids, it selects the best node (node with the highest bid) for the task. In multi-robot systems, the nodes are the robots and the manager is a central agent or another robot.

Most current auction methods only admit a robot per task, therefore, they are SR methods. In contrast, MR problems, which require coalition formation mechanisms, have hardly been addressed by auction approaches. Vig and Adam's RACHNA method [32] allows MR problems but without deadlines. Paquet in [33] proposes a method where the working group size is unknown beforehand and considers task deadlines. This mechanism has only been tested on multi-agent systems, such as Robocup competition, but not on real or simulated physical robots. Furthermore, due to the high computational cost of its algorithms, the decision about how many agents are needed to finish a task cannot be made on-line. In [34], Tang and Parker introduce another auction approach, called ASyMTRe, that allows high tight tasks and coalition formation. More auction methods that admit coalition formation can be found in [35]. None of these methods consider the physical interference or the task deadlines.

Among the auction methods that allow time restrictions, one of the most remarkable work is that proposed by Jones [36]. This approach makes use of a learning algorithm to find the most suitable value for the bids in MR problems. Jones' method uses a learning Support Vector Regression (SVR) model to obtain the bid value, but does not estimate the execution time. Lemaire's auction method [37] allows soft deadlines but with a single robot per task. In [38], the proposed method can solve the VRPTW without overlapping windows. In [39], Campbell introduces a method that, that is partially inspired by auctions, where the robots can not decide to bid for a task despite it was the best option at that moment. The Campbell's experimental results show a great improvement regarding the aforementioned methods.

Heretofore, all the methods cited does not try to forecast the expected execution time of the task. Sellner's and Simmons' SR method [40] predicts this time making use of kernels, but no physical interference is taken into account. In contrast, Sellner's approach requires very heavy

computational algorithms to carry out the tasks. As has been mentioned in previous sections, Nunes et al. [21] allow overlapped time windows with auctions to address SR problems. Finally, other auction-like mechanisms [41] also face up tasks with deadlines but with only one robot per task.

As has been exposed, the auction mechanisms allow the distribution of the decision process among the robots through communication protocols. Thus, the typical problems produced in centralized systems, such as single point of failure or overloading of the central agent, are avoided. Furthermore, these algorithms are, in general, much more simple than the centralized approaches. In contrast, to the aforesaid advantages, the solutions provided by the auction methods require more time and increase the energy consumption compared to the centralized systems. In addition, the auction methods are more complex than the swarm ones because they require communication protocols between robots. Despite of that, the auction systems outperform the swarm-based ones increasing the number of finished tasks and reducing the energy consumption, among other factors. Furthermore, due to the inter-robot communication protocols, the auction methods can deal with much more complex and tight tasks.

Physical interference between robots

The physical interference between robots is one of the most fundamental problems to address by the task allocation mechanisms. In [2] Rosenfeld et al. clearly state the huge impact of the physical interference on the multi-robot systems performance. There are a lot of methods that try to minimize the interference effect in multi-robot systems without deadline (see, for instance, [42]).

In [43], Dahl et al. propose a mechanism called Vacancy Chains to carry out foraging tasks for MR problems. The system assigns robots to a task (transport an object that can be divided in small parts) until it detects that the amount of transported object weight decreases. This reduction can only be due to the interference between robots. Dahl's method neither deals with tasks with deadlines nor can estimate tasks execution time. Finally, Lerman's et al. studies [1] and Nam and Shell [44] present methods to model the interference but none of them deal with time restrictions.

Other authors describe swarm-like strategies to deal with the physical interference in systems with a huge number of robots (see [6, 45, 46]). For example, Rubenstein et al. in [45] use self-assembly with thousand of robots to build complex structures.

Coalition formation complexity analysis

The MR problem with deadlines has not been properly addressed by previous task allocation approaches. Moreover, no formal analysis regarding the complexity of this problem has been proposed. This section presents a formal definition of the MRTA-ST-MR with deadlines problem from which we will propose a classification, according to the computational complexity under different situations.

Generic problem

The MRTA-ST-MR with deadlines problem (from now on simply MRTA problem) can be seen as a particular case of the SPP. From now on, \mathbb{N} will denote the set of positive integer numbers. Moreover, in the following, R will denote the set of robots with $R = \{r_1, \dots, r_N\}$ and T will denote the set of tasks to carry out with $T = \{t_1, \dots, t_N\}$, where $N_r, N_t \in \mathbb{N}$. Furthermore, we will assume that both sets, R and T , are kept constant over the time. Next, denote by G the the subset of 2^R formed by all working groups or coalitions of robots allowed over R , where 2^R

stands for the power set of R . The criteria to decide if a working group is allowed depends on specific task characteristics.

Now, taking into account the preceding notation, we introduce the main notions in order to formalize the MRTA problem with deadlines. To this end, let us denote by \mathbb{R}^+ the non negative real numbers.

Definition 1 Let $t_j \in T$ and $n(t_j) \in \mathbb{N}$ with $1 \leq j \leq N_t$. The sets of coalitions assigned to t_j over the time, Sg_{t_j} , is defined by

$$Sg_{t_j} = \{(g_1, \tau_1), \dots, (g_n(t_j), \tau_n(t_j))\},$$

where each pair $(g_k, \tau_k) \in (G \times \mathbb{R}^+)^2$ stands for the coalition of robots g_k allocated at task t_j at time τ_k for all $k = 1, \dots, n(t_j)$ and, in addition, satisfies that $\tau_k < \tau_m$ whenever $k < m$ for all $k, m = 1 \dots n(t_j)$.

It must be pointed out that each coalition g_k allocated at a task t_j is kept constant from time instant τ_k until τ_{k+1} for all $k = 1, \dots, n(t_j)$, where $\tau_{n(t_j)+1}$ denotes the finishing time for that task provided that the set of coalitions Sg_{t_j} is able to carry out the task.

Notice that if the allocation is non preemptive, that is, a robot cannot leave its current working group after allocating it in a task, then the cardinality of the set of coalitions must be lower or equal to 1, i.e., $|Sg_{t_j}| \leq 1$ for all $j = 1 \dots N_t$ and, thus, $Sg_{t_j} \in G \times \mathbb{R}^+$ for all $j = 1, \dots, N_t$.

Definition 2 Let $j \in \{1, \dots, N_t\}$. Then the execution time of the task $t_j \in T$ is defined as the function $\beta_j : Sg_{t_j} \rightarrow \mathbb{R}^+ \cup \{\infty\}$ such that

$$\beta_j(Sg_{t_j}) = \begin{cases} \tau_{n(t_j)+1} & \text{if } Sg_{t_j} \text{ is able to carry out the task } t_j \\ \infty & \text{otherwise} \end{cases}.$$

Note that $\beta_j(Sg_{t_j})$ provides the time needed by the set of coalitions to finish the task t_j whenever the aforementioned set is able to carry out t_j .

As has been pointed out in Section Previous work on coalition formation, giving the exact value of β function for a certain coalition is often impossible or quite difficult because of, among other factors, the interference effect. Therefore, in most cases, the function β must be predicted through different methods.

Definition 3 Let $j \in \{1, \dots, N_t\}$. Then the utility of a task t_j is defined as a function $U_j : \mathbb{R}^+ \cup \{\infty\} \rightarrow \mathbb{R}^+$ such that $U_j(\infty) = 0$.

Note that the utility function of a task will depend on task's characteristics such as its deadline which will be denoted by DL_j . Fig 1(a) shows an example of an utility function U_j called hard-deadline utility function, where $U_j(\tau)$ is constant and equal to U_{max_j} if $\tau < DL_j$ and abruptly drops to zero if $\tau \geq DL_j$ (the task cannot be executed on time). Another kind of utility function are so-called soft-deadline utility functions. An example of these type of functions can be seen in Fig 1(b), where the value of $U_j(\tau)$ progressively falls to zero from $\tau = DL_j$.

In the light of Definition 3 it is clear that the value $U_j(\beta_j(Sg_{t_j}))$ yields the utility after assigning a coalition set Sg_{t_j} to a task $t_j \in T$. Depending on the characteristics of the composition function $U_j \circ \beta_j$, as for instance linearity or continuity, the allocation problem must be addressed by a different kind of strategy, as will be seen later.

As we have noticed before, $Sg_{t_j} \in G \times \mathbb{R}^+$ whenever the allocation is not preemptive. Hence in this case the global utility associated to the MRTA problem matches up with the function

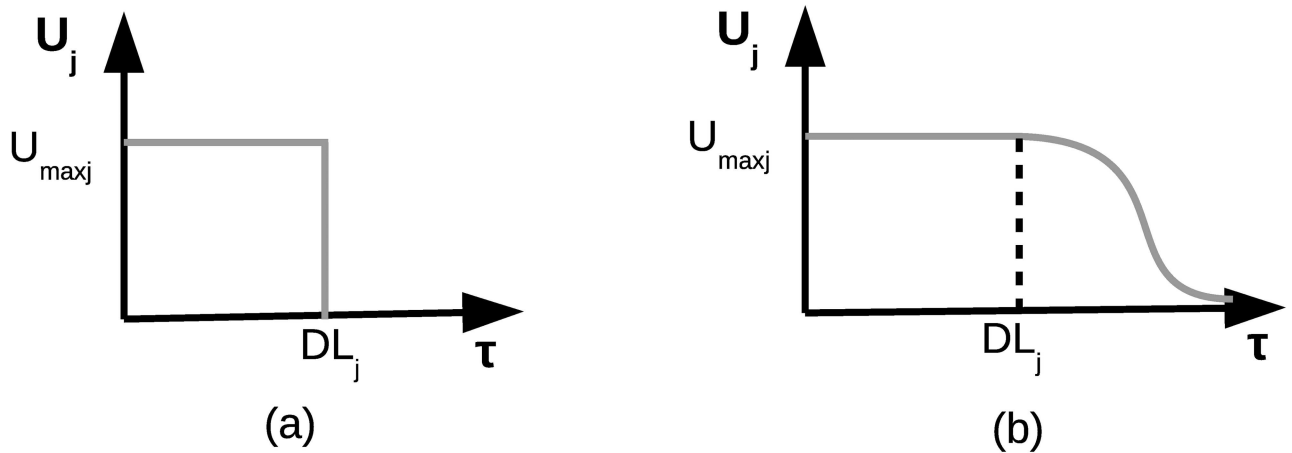


Fig 1. Utility function (U_j) examples. (a) Hard-deadline utility function, (b) Soft-deadline utility function, where τ stands for the time needed by the coalition to finish the task.

doi:10.1371/journal.pone.0170659.g001

$U : (G \times \mathbb{R}^+)^{N_t} \rightarrow \mathbb{R}^+$ given as follows:

$$U(Sg_{t_1}, \dots, Sg_{N_t}) = \sum_{j=1}^{N_t} U_j(\beta_j(Sg_{t_j})). \tag{2}$$

It is clear that an element $(Sg_{t_1}, \dots, Sg_{N_t}) \in (G \times \mathbb{R}^+)^{N_t}$ can be identified with a function $TA : T \rightarrow (G \times \mathbb{R}^+)^{N_t}$ such that

$$TA(T) = (TA_1(t_1), \dots, TA_{N_t}(t_{N_t})),$$

where $TA_j(t_j) = Sg_{t_j}$ for all $j = 1, \dots, N_t$.

Under the preceding assumptions, the goal of the MRTA problem is to maximize the global utility function, that is, to find a function TA^* which maximizes the function U . Thus, the aforesaid goal matches up with obtaining the solution to the following optimization problem:

$$\begin{aligned} &\underset{TA}{\text{maximize}} \sum_{j=1}^{N_t} U_j(\beta_j(TA_j(T))) \\ &\text{subject to} \end{aligned} \tag{3}$$

$$TA_i(t_i) \cap TA_j(t_j) = \emptyset \quad \text{for } i, j = 1, \dots, N_t$$

Observe that we need to consider the restriction of [Problem 3](#), since the MRTA problem is an SPP problem.

To validate the model explained above we propose a simple but generic example of a foraging-like task. To this end, we assume, in addition to the non preemptive allocation assumption, an IA strategy and that the coalitions for the tasks are formed simultaneously, i.e., τ_1 is the same for all t_j with $j = \dots, N_t$.

Example: Foraging with tasks' deadline. Foraging is one of the canonical tasks utilized by a lot of authors to state their algorithms. Moreover, it can easily be extended to other kind of missions, such as area coverage or repeated coverage.

In a foraging task there is a set of objects or tasks ($T = \{t_1, \dots, t_{N_t}\}$) to process. Each task $t_j \in T$ has a workload (L_j) that, for example, is the weight of the object; and a utility function

(U_j) with a deadline (DL_j) (see the paragraph about utility functions with deadlines below Definition 3). Each robot $r_i \in R = \{r_1, \dots, r_{N_r}\}$ has a work capacity, C_{ij} , associated to each task t_j that represents the amount of work (object's workload) that the robot can process per time unit. For example, in a foraging task, C_{ij} would be the object's weight that a robot can carry per time unit. Each coalition of robots $g \in G$, in charge of a task t_j , has a work capacity GC_{gj} that gives the amount of work that the group of robots can perform per time unit. Thus, the expected time to finish a task performed by a group of robots g (function β_j) is denoted by:

$$\beta_j(g) = \frac{L_j}{GC_{gj}} \tag{4}$$

The L_j is known before executing the task, for example, the weight of an object to be transported or the area of a region to be covered by the robots' group. However, it is hard to determine precisely the GC_{gj} because it usually depends on dynamic and a priori unknown factors. It is widely accepted that the factor with the strongest effect is the physical interference, as pointed out in Section Physical interference between robots. In this work the deviation from the ideal capacity of a robot's team is summarized in a single term called the interference factor defined by a function $I : G \times T \rightarrow \mathbb{R}$ (\mathbb{R} stands for the real numbers set), where $I(g, t_j) = I_{gj}$ denotes the interference effect of executing task t_j by a coalition of robots g . I_{gj} must be viewed as a noise of the ideal model (*idealCapacity_{gj}*) that is able to include the physical interference or any other kind of perturbations, such as odometry errors (see [3]). Thus, the work capacity of a group to perform a task t_j is considered to be:

$$GC_{gj} = \text{idealCapacity}_{gj} - I_{gj}, \tag{5}$$

where *idealCapacity_{gj}* is the ideal group capacity. When the individual characteristics of the robots are known a priori, the *idealCapacity_{gj}* can be easily estimated as follows:

$$\text{idealCapacity}_{gj} = \sum_{1 \leq i \leq n_g} C_{ij}, \tag{6}$$

where n_g is the number of robots of the group g . This is a simple but very effective model, as has been proved by the first two authors in a previous work in [3].

From all these definitions the objective function to maximize can be defined in two different ways as follows:

If there is no interference, that is $I_{gj} = 0$ for all $g \in G$ and for all $j = 1, \dots, N_t$, then the utility function, provided by Eq (2), is given now by

$$\sum_{j=1}^{N_t} U_j \left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} C_{ij}} \right). \tag{7}$$

Hence the Problem (3) becomes the following optimization problem:

$$\begin{aligned} & \underset{\alpha_{ij}}{\text{maximize}} \sum_{j=1}^{N_t} U_j \left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} C_{ij}} \right) \\ & \text{subject to} \quad , \\ & \sum_{j=1}^{N_t} \alpha_{ij} \leq 1 \quad \text{for } i = 1, \dots, N_r \end{aligned} \tag{8}$$

where

$$\alpha_{ij} = \begin{cases} 1 & \text{if robot } r_i \text{ is assigned to task } t_j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Notice that the restriction in [Problem \(3\)](#), $TA_i(t_i) \cap TA_j(t_j) = \emptyset$ for all $i, j = 1, \dots, N_b$ have been rewritten in terms of α_{ij} in order to develop our numerical schemes later on.

If there is interference then the utility function, provided by [Eq \(2\)](#), is given now by

$$\sum_{j=1}^{N_t} U_j \left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} C_{ij} + I(TA_j(t_j), t_j)} \right) \tag{10}$$

So the [Problem \(3\)](#) becomes the following optimization problem:

$$\begin{aligned} & \underset{\alpha_{ij}}{\text{maximize}} \sum_{j=1}^{N_t} U_j \left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} C_{ij} + I(TA_j(t_j), t_j)} \right) \\ & \text{subject to} \\ & \sum_{j=1}^{N_t} \alpha_{ij} \leq 1 \quad \text{for } i = 1, \dots, N_r \end{aligned} \tag{11}$$

The method under consideration to face up the [Problems \(8\)](#) and [\(11\)](#) depends on the characteristics of the composed function $(U_j \circ \beta_j)$. Next we summarize, on the one hand, the optimization methods that we will use to solve the aforesaid problems and, on the other hand, the range of considered characteristics of the utilities functions.

Case studies

Case 1 (Linear and soft-deadline utility function). If U_j is a soft-deadline linear, continuous and decreasing function for all $j = 1, \dots, N_b$, then, in general, [Problems \(8\)](#) and [\(11\)](#) becomes binary integer nonlinear optimization problems. This is due to the fact that the functions β_j , given by [Eq \(4\)](#), are not linear. For example, if the utilities functions are defined by $U_j(\tau) = -\tau + U_{max_j}$ for all $\tau \in \mathbb{R}^+$ (U_{max_j} is a constant for each $j = 1, \dots, N_t$), then [Problem \(11\)](#) can be expressed as follows:

$$\begin{aligned} & \underset{\alpha_{ij}}{\text{maximize}} \sum_{j=1}^{N_t} \left(-\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} C_{ij} + I(TA_j(t_j), t_j)} + U_{max_j} \right) \\ & \text{subject to} \\ & \sum_{j=1}^{N_t} \alpha_{ij} \leq 1 \quad \text{for all } i = 1, \dots, N_r \end{aligned} \tag{12}$$

Case 2 (Soft-deadline utility function and $U_j \circ \beta_j$ linear). If the functions U_j are given, for all $j = 1, \dots, N_b$, as follows

$$U_j(\tau) = \frac{U_{max_j}}{\tau} \tag{13}$$

(again U_{max_j} is a constant for each $j = 1, \dots, N_t$) and I_g is linear, then it is not hard to see that

Problem (11) is an integer linear optimization problem, as will be demonstrated in following paragraphs. This utility function form fits a great number of realistic tasks such as foraging toxic wastes where as the time passes, the damage of the environment increases and therefore the utility of the function also decreases.

In general the interference function is not linear with respect to the number of robots, as proved in [2]. However, in some cases this assumption is pretty realistic, as the experimental results of Section Experimental results demonstrate. Of course whenever the interference function is linear it has the following form:

$$I(g, t_j) = a_j n_g + b_j, \tag{14}$$

where we recall that n_g is the number of robots of the coalition g and, in addition, a_j, b_j are constants that depend on the tasks characteristics. Section Experiments with physical robots: Interference function shows how a_j and b_j can be obtained. Taking into account the preceding information, **Problem (11)** can be rewritten as follows:

$$\begin{aligned} & \underset{\alpha_{ij}}{\text{maximize}} \sum_{j=1}^{N_t} \sum_{i=1}^{N_r} \left(U_{\max_j} \alpha_{ij} \left(\frac{C_{ij} - a_j}{L_j} \right) \right) - U_{\max_j} \frac{b_j}{L_j} \\ & \text{subject to} \\ & \sum_{j=1}^{N_t} \alpha_{ij} \leq 1 \quad \text{for all } i = 1, \dots, N_r \end{aligned} \tag{15}$$

Therefore, we conclude that in this case the optimization problem can be addressed, under the considered assumptions, as an integer linear programming problem.

Case 3 (Hard-deadline utility function). If the utility functions U_j , for all $j = 1, \dots, N_t$, are two-step hard deadline discontinuous function (see Fig 1(a)) then we show that the problem can be addressed as an integer linear programming problem. To this end, let us recall that according to [47] a renowned classical optimization problem consists in minimizing a function f defined by

$$f = \sum_{j=1}^{N_t} H \left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) - b_j} - DL_j \right), \tag{16}$$

where each α_{ij} is given as in Eq (9) and H is the heaviside function defined by

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{17}$$

Following [47], the preceding optimization problem can be converted into an integer program problem adding for each task t_j a new binary auxiliary variables $\eta_j \in \{0, 1\}$ and the constraint $\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) - b_j} - DL_j \leq M \eta_j$, where M is a value greater than all the values for the argument of H . Thus the optimization problem of minimizing the function f can be reformulated as the next

optimization problem where the variable are now η_j and α_{ij} :

$$\begin{aligned} & \underset{\eta_j, \alpha_{ij}}{\text{minimize}} \sum_{j=1}^{N_t} \eta_j \\ & \text{subject to} \end{aligned} \tag{18}$$

$$\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) - b_j} - DL_j \leq M\eta_j \text{ for all } 1 \leq j \leq N_t$$

In the light of the exposed considerations, it is clear that the multi-robot task allocation problem under consideration can be seen as an optimization problem which fits the characteristics of the Problem (18). Indeed, the fact that any η_j takes the value 0 can be interpreted as the coalition of robots allocated in task t_j can carry out the task before its deadline (of course $\eta_j = 1$ can be interpreted dually). Moreover, each utility function U_j can be expressed in terms of the H functions as follows:

$$U_j(\tau) = U_{\max_j} H(\tau - DL_j), \tag{19}$$

where τ is the amount of time required to finish the task t_j . This expression can be rewritten to add robots' capabilities, task's characteristics and interference, as follows:

$$U_j\left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) - b_j}\right) = U_{\max_j} H\left(\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) - b_j} - DL_j\right). \tag{20}$$

After taking into account the information provided by the utility functions U_j and the interference effect, Problem (18) becomes the following optimization problem:

$$\begin{aligned} & \underset{\eta_j, \alpha_{ij}}{\text{minimize}} \sum_{j=1}^{N_t} U_{\max_j} \eta_j \\ & \text{subject to} \end{aligned}$$

$$\eta_j m - \sum_{i=1}^{N_r} \alpha_{ij} (C_{ij} - a_j) \leq -\left(\frac{L_j}{DL_j} + b_j\right) \text{ for all } j = 1, \dots, N_t, \tag{21}$$

$$\sum_{j=1}^{N_t} \alpha_{ij} \leq 1 \text{ for all } i = 1, \dots, N_r$$

where $m = \max_j \left\{ -\frac{L_j}{\sum_{i=1}^{N_r} \alpha_{ij} (a_j - 1) + b_j} \right\}$ for all $j = 1, \dots, N_t$. Consequently, when the utility functions meet the characteristics of the Case 3, the optimization problem matches up with a binary linear programming problem.

Table 1 summarizes the aforementioned cases, assuming that the interference function I_g is linear. In spite of most MRTA cases can be addressed as linear (binary-integer) optimization problems they still require too much time to be solved in dynamic environments, especially when there are a large number of tasks or robots. As pointed in section about previous work, many approaches have tried to give a good enough solution with a reasonable amount of time: swarm intelligence, auction approaches, and so on, but only few of them take into account task with deadlines. Inspired by the preceding fact and by the exposed example, a new auction-like algorithm to address MRTA-IA problems is introduced in the next section.

Table 1. Summary of the complexity regarding the utility function, where BINLP is binary integer non-linear programming problem; BILP denotes integer linear programming problem; LP is linear programming problem.

Case 1	BINLP
Case 2	BILP
Case 3	BILP

doi:10.1371/journal.pone.0170659.t001

Task Allocation: Double Round Auction for tasks with deadline

In this section we first overview an auction method, called from now on Simple Double Round Auction (SDRA), that was previously proposed by the first two authors in [3]. The experimental results showed that the aforementioned method outperforms the current auction approaches. Furthermore, to our best knowledge, this is the only method that addresses coalition formation for tasks with deadlines. Then, in the light of the experimental results and motivated by the problem formalization explained in the last section, a new auction method, called Multiple Objectives Double Round Auction (MDRA), is proposed for the first time.

Simple Double Round Auction (SDRA)

The SDRA algorithm splits the classical auction process into two steps: *auction for a task* and *auction for a robot*. In each task there is a robot, called the *task leader*, that will manage an auction process. Thus, several auction processes can be executed in parallel; one for each task. Each task must have a single leader which is the robot that handles the coalition or work group, that is, it will be the auctioneer. In the initial stage, each robot is looking for a task. When a robot finds a new task, it tries to lead it. As there can only be one leader per task, the candidate will execute the leadership request process to ensure at most one leader per group.

If a robot is promoted to the leadership of the task t_j , it creates a coalition of robots, if necessary. In that case, the leader must decide the optimum group size and which robots are part of it. To take this decision, the leader uses the first step of the SDRA, called *auction for a task*. In this process the leader is the auctioneer and the other robots bid using their work capacities, C_{ij} . After a given time, the leader selects the robots with the highest work capacity using a greedy algorithm until it detects that, even if the next best robot is added to the group, the task utility function U_j is not improved. Following the new formulation, if $U_j \circ \beta_j$ function meets the constraints of the Case 2 detailed in Section Case studies, the task leader selects the best robots until this condition is verified:

$$U_j(\beta_j(g_B)) \geq U_j(\beta_j(g'_B)), \tag{22}$$

where g_B stands for the group of best robots already selected and $g'_B = g_B \cup \{r_{nmax}\}$ where r_{nmax} is the next best robot. If U_j is a hard-deadline function (see Case 3), the task leader stops selecting new robots when it detects that the group is able to reach the task deadline, that is until $\beta_j(g_B) \leq DL_j$.

Furthermore, the leader does not include any selected robot in its group until it receives the confirmation from these candidates. This confirmation is generated during the *auction for a robot* round. In this round, to get the selected robots, the leader bids for them sending the expected utility of the task. Therefore, now the leader becomes the bidder and the selected robots are the auctioneers. When a robot receives a bid from a leader, it waits a certain time for more bids from other leaders. After this time, the robot selects the coalitions with the greatest bid and it becomes a member sending a confirmation message to the corresponding leader.

The interference factor I_{g^j} , required to calculate the expected execution time, (see Eqs (4) and (5)), was estimated using a Support Vector Regression method (SVR). In the experimental results section, we will explain in a more detailed way how the interference parameters have been estimated.

Multiple Objectives Double Round Auction (MDRA)

When applying the SDRA method, the *auction for a robot* round only used the utility sent by the leader and did not take into account the individual capacity of the robots. The new method, called Multiple Objectives Double Round Auction (MDRA), uses both the individual and the leader's utility during the *auction for a robot* round in order to improve the system performance. The other two steps of the auction process: leader's selection and *auction for a task*, are not modified with respect to SDRA.

In the MDRA method, as in SDRA, the robots selected by the leaders receive the expected utility of the tasks, denoted by $B = \{(t_1, b_1), \dots, (t_n, b_n)\}$, where $b_i \in \mathbb{R}^+$ represents the utility of the group that attempts to execute the task $t_i \in T$. Let (t_{best}, b_{best}) be the tuple in B with highest utility; and denote by $B_{bests} = \{(t_i, b_i) \in B: b_i \geq \lambda_B \cdot b_{best}\}$ where $\lambda_B \in [0, 1]$. The robot will select the task from B_{bests} for which it has the highest capacity. Algorithm 1 specifies the steps executed by a non-leader robot to implement the *auction for a robot* stage, where *TIME_BID_ACCEPTED* stands for the amount of time that the robot waits for leaders bids (*AWARD* messages). After this time, the robot selects the coalition with the greatest bid and it becomes a member sending a confirmation message (*ROBOT_ALIVE*) to the corresponding leader. If the leader doesn't receive any answer from a robot or receives a reject message (*REFUSE*), it removes that robot from its list of selected robots. On the other hand, a robot is definitely added to the group, when the leader receives a confirmation message from it. The λ_B value is inspired by the selection factor used in [13], but it is now applied in an auction process with tasks' deadlines.

Algorithm 1 Auction for a robot round executed in each no leader robot

```

Require:  $\lambda_B \in [0, 1]$ 
Require: time current time
1: if AWARDmessage,  $(t_0, b_0)$ , from a leader then
2:    $B \leftarrow (t_0, b_0)$ 
3:    $time_{init} \leftarrow time$ 
4:   while  $time - time_{init} \leq TIME\_BID\_ACCEPTED$  do
5:     if AWARDmessage,  $(t_j, b_j)$ , from a leader then
6:        $B \leftarrow B \cup (t_j, b_j)$ 
7:        $j \leftarrow j + 1$ 
8:     end if
9:   end while
10:   $(t_{best}, b_{best})$  tuple in  $B$  with the maximum bid value
11:   $B_{bests} = \{(t_i, b_i) \in B: b_i \geq \lambda_B \cdot b_{best}\}$ 
12:   $t_{sel} \leftarrow$  task from  $B_{bests}$  with the highest robot's capacity
13:  Send a ROBOT_ALIVEmessage to the leader of  $t_{sel}$ 
14:  for all  $t_j$  in  $B$  and  $t_i \neq t_{sel}$  do
15:    Send REFUSEmessage to the leader of  $t_j$ 
16:  end for
17: end if

```

Experimental results

In this section we detail the experiments executed in order to analyze the behaviour of the auction algorithms, comparing them to the optimal solution given by the optimization problems of Section Coalition formation complexity analysis. Firstly, we will explain the experiments carried out with a realistic (physical) simulator called RoboCoT (see [3]) in order to obtain the I function parameters. From the function I , new experiments have been carried out using Matlab (Version R2014a and Optimization Toolbox version 7). The solutions for the optimization, Problems 15 and 21, have been obtained executing the optimal linear programming-based branch-and-bound algorithm together with a simplex method [48].

Besides the approaches already explained, another simple method, called greedy selection, has been implemented. In this method each robot selects the best task from its point of view, in a selfish mode. Thus, a robot r_i selects the task t_{sel} such that, $t_{sel} = \max_j(C_{ij})$. The robots do not execute any auction algorithm nor make any prediction about the interference.

Experiments with physical robots: Interference function

First, we describe how the parameters of the interference function (Eq 14) have been estimated. In a previous work [49], the first two authors exposed how, for foraging-like tasks (see Section Example: Foraging with tasks' deadline), the interference can be modelled as a set of polynomial functions that only depends on the number of robots and on the distance between the delivery point and the object. To get the parameters of the polynomial functions the following experiment in the realistic simulator RoboCoT was carried out: several robots must transport a unique object, formed by transportable bits, and the total weight transported by the robots after 40,000 time units is calculated. Seven different distances between object and delivery point were tested and the parameters of I were fitted for each one of them (see Table 2). Fig 2 shows a snapshot of one of these executions, where the little square represents the object, the big circle the delivery point and the little circles are the robots. These results were also validated with other well-known simulator called Player/Stage [50]. Although all the interference values in this paper relate to physical simulations, the algorithms were also implemented on four real Pioneer-3DX. Videos with some executions can be watched in [51] and a more detailed description of these experiments can be found in [49].

From the results of the aforesaid simulations, it can be seen that the maximum utility of the group (the maximum of Eq (5)) is reached when the number of robots is equal to or lower than 30. Thanks to the condition given by Eq (22), the maximum number of robots in a coalition created by the auction algorithms is 30. Although in the original paper [49] the polynomials have a degree equals to two, if $n_g < 30$ the I_g can be fitted by a linear polynomial function

Table 2. Parameter a_j of the linear interference function.

Distance	$a_j \cdot 10^3$
D1 = 138	10.3428
D2 = 184	6.1682
D3 = 247	3.4728
D4 = 282	2.8746
D5 = 329	1.9278
D6 = 359	1.7190
D7 = 400	1.3657

doi:10.1371/journal.pone.0170659.t002

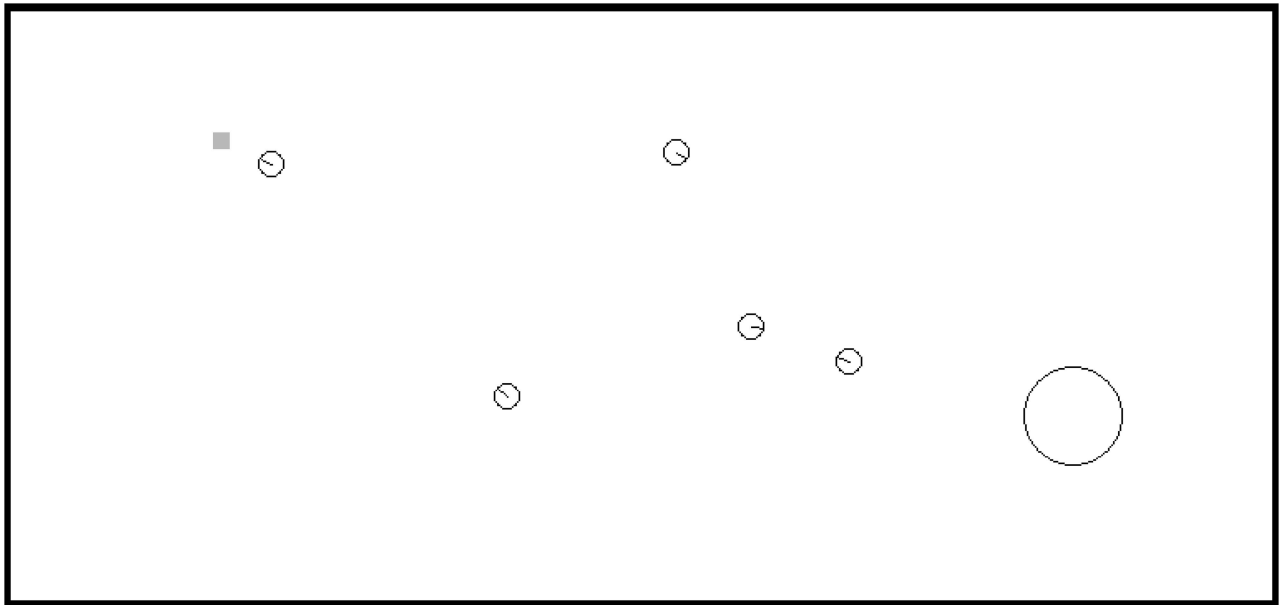


Fig 2. Execution example of a RoboCoT experiment to calculate the parameters of the interference function I .

doi:10.1371/journal.pone.0170659.g002

with a very low error, meeting all the conditions of the cases exposed in section about the formalization (Cases 2 and 3). The parameter b_j of Eq (14) was set to 0 because it is assumed that a coalition with a single robot does not have physical interference. Table 2 shows the interference linear function parameter a_j that minimizes the square error for each distance tested.

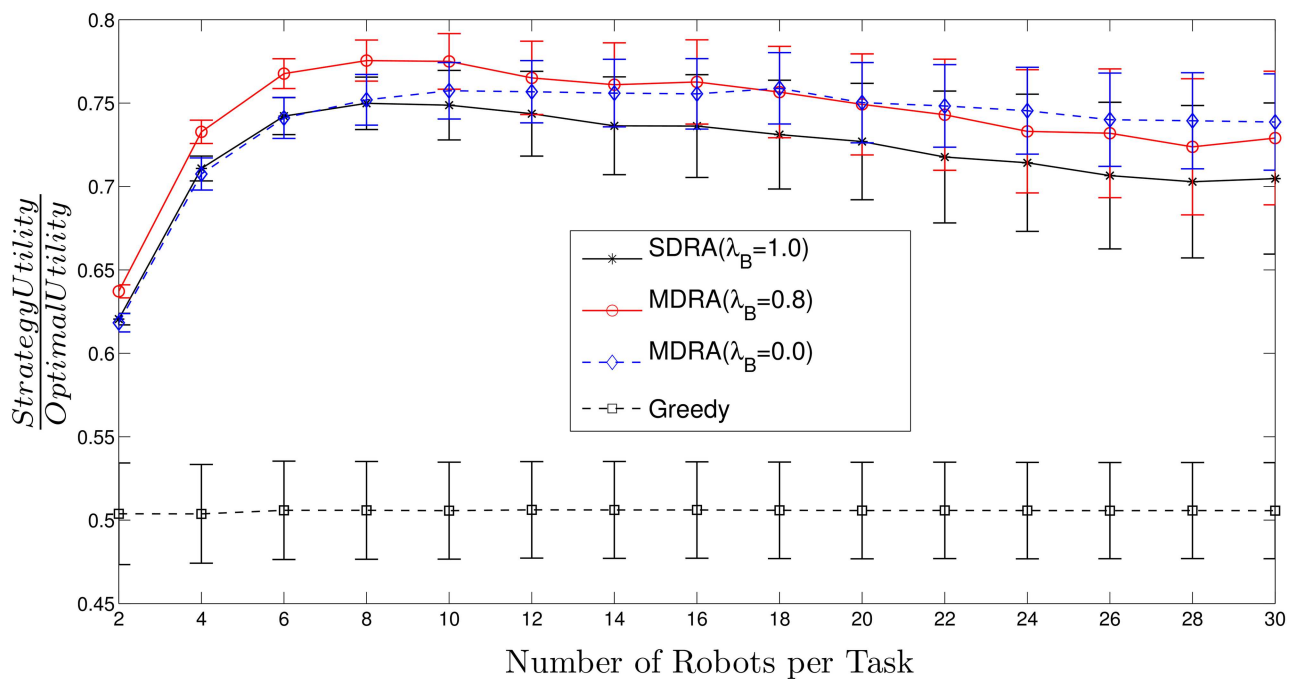


Fig 3. Ratio between utility obtained with MDRA and the utility of the optimal strategy with $N_t = 15$. The bars stand for the standard deviation (σ^2).

doi:10.1371/journal.pone.0170659.g003

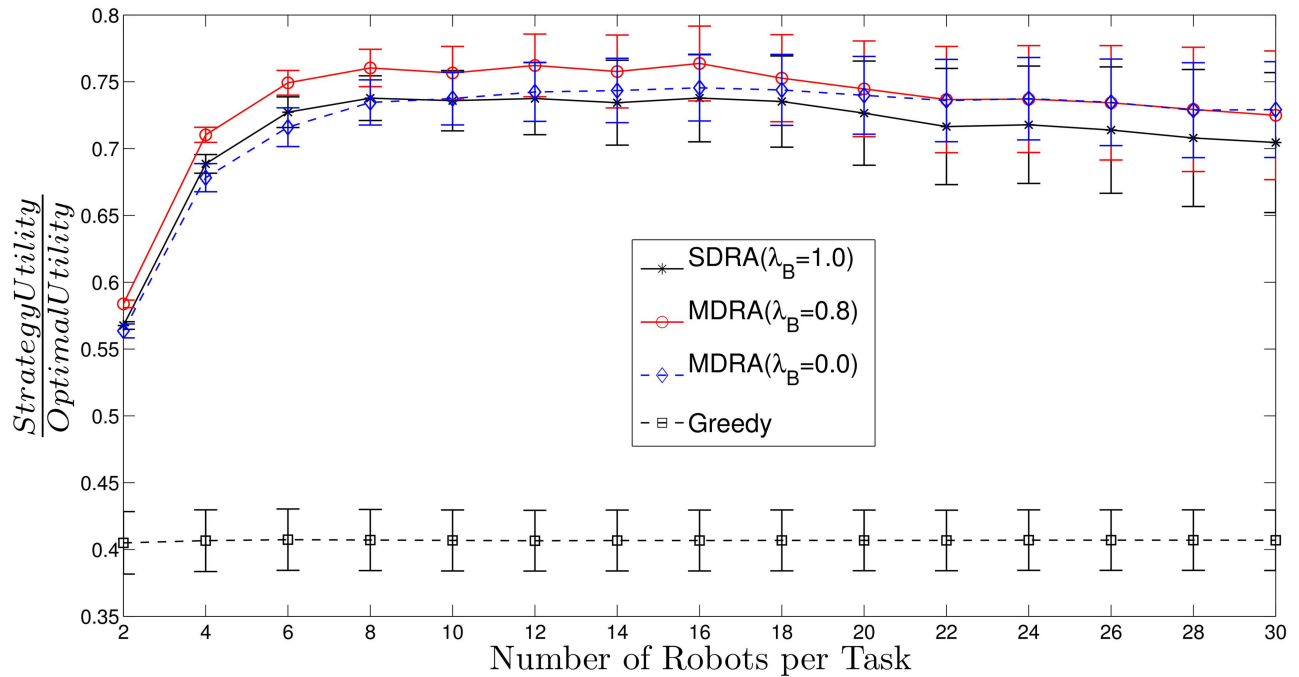


Fig 4. Ratio between utility obtained with MDRA and the utility of the optimal strategy with $N_t = 30$. The bars stand for the standard deviation (σ^2).

doi:10.1371/journal.pone.0170659.g004

Experiments with soft-deadline utility functions

The first set of experiments assume that each $U_j \circ \beta_j$ is linear and it meets the condition of Case 2. Therefore the problem to solve is Problem (15). The characteristics of the experiments are the following: The number of robots (N_r) varies from 2 to 30 times the number of objects (N_t). There are 5 different types of objects and each robot has a different load capacity depending on the object type. In order to apply the parameters of Table 2, the distances always fall into the bounds described in above section. In addition, robots' work capacity C_{ij} depends on the type of task and on its velocity, as is pointed out by the authors in [3]. The main goal of these experiments is to test the new MDRA algorithm with a very large number of robots. It is worth to point out that when $N_t = 15$ there are up to 450 robots, and if $N_t = 30$ then this number is increased up to 900 robots. Each experiment has been performed 500 times using a uniform random distribution to generate the position of the objects, their weights, the load capacity of the robots and their velocities. S1 Table shows the parameters of the uniform random variables for each one of these characteristics. In order to keep the robots as simple as possible, the robots use their load capacities for selecting the best task in the Greedy strategy.

Fig 3 ($N_t = 15$) and Fig 4 ($N_t = 30$) show the utility per robot obtained by different strategies (SDRA, MDRA and greedy) divided by the utility per robot obtained with the optimal method (see S2 and S3 Tables). For all these results, utility per robot stands for the mean of the total obtained utility divided by the number of robots. Thus, higher values stand for results closer to the optimal. The X-axis shows the ratio between the number of robots and the total number of tasks and the bars stand for the standard deviation (σ^2). As can be seen, the total utility obtained with MDRA utilities is close to the optimal and the better results (higher values) are obtained when the number of robots per task is 8. With more than 8 robots per task, the results stay constant or show a slight decrease. For some cases, MDRA utility reaches more than the 75% of the optimal utility and from around 14 robots per task this value keeps almost constant.

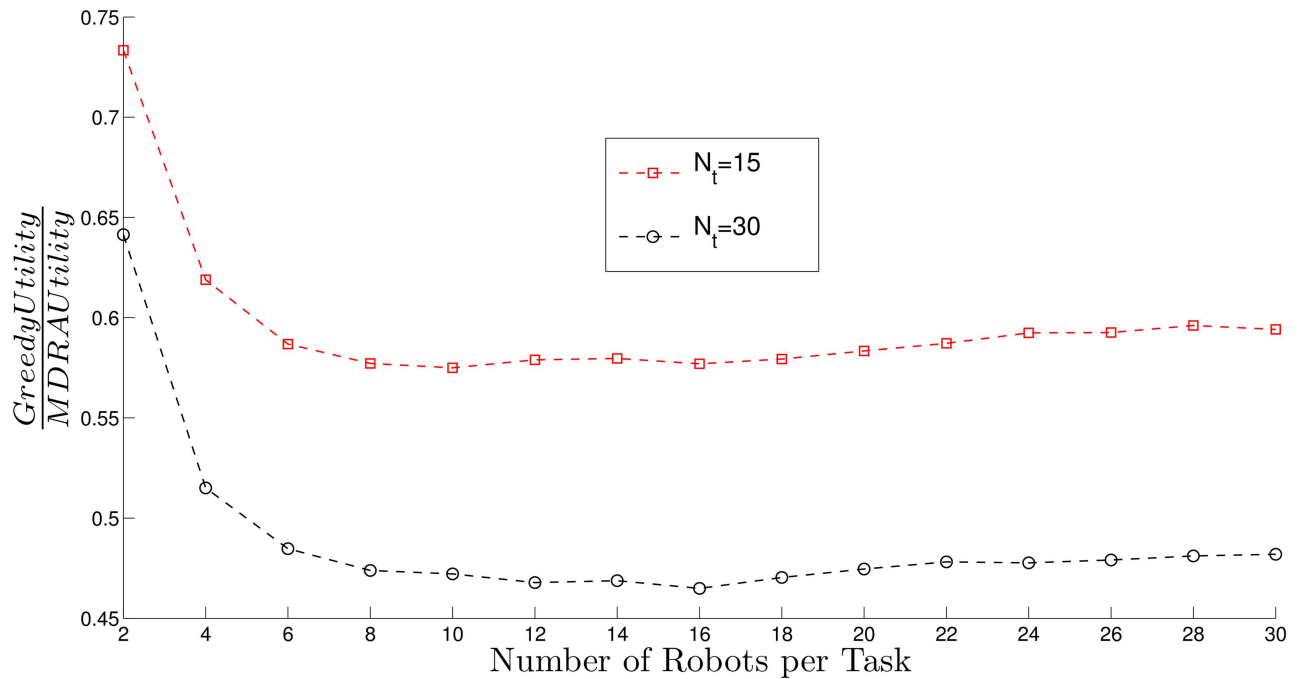


Fig 5. Ratio between greedy utility and MDRA ($\lambda_B = 0.8$).

doi:10.1371/journal.pone.0170659.g005

If, instead of the mean, it is calculated the median, the MDRA utility reaches more than an 80% of the optimal (see S3 Table). The Greedy strategy provides the worst results, always lower than a 55% of the optimal utility. In most cases, the new MDRA algorithm gives the best results when $\lambda_B = 0.8$ and outperforms the SDRA. It should be recalled that the tested environments are very heterogeneous and, therefore, this issue increases the σ^2 values of the experimental results.

Fig 5 shows the ratio between utility per robot using greedy algorithm and MDRA with $\lambda_B = 0.8$, similar results are obtained for other values of λ_B . The experiments demonstrate that the higher number of robots the lower greedy performance with respect to MDRA and hence, the new auction algorithm improves the results compared to a fully distributed approach. The standard deviation of these experiments can be obtained from the information of the Figs 3 and 4.

As can be seen in Fig 6 (see also S4 Table), the optimal algorithm needs much more execution time to give a solution than MDRA with $\lambda_B = 0.8$ or greedy algorithm and it clearly grows with the number of robots. Moreover, the MDRA algorithm increases very slowly its execution time with respect to the number of robots. Therefore, it can be concluded that the results provided by the new MDRA method are pretty close to the optimal results with much less execution time and it outperforms the SDRA approach.

Experiments with hard-deadline utility functions

This section describes the experiments carried out with hard-deadline utility functions and linear interference, that is, Case 3 described in Section Case studies. Therefore, the problem to solve is the same as Problem (21). In order solve it numerically, we must find coalitions with a performance (amount of work carried out per time unit) greater or equal than the performance needed to finish the task before the deadline, instead of the execution time. Notice that

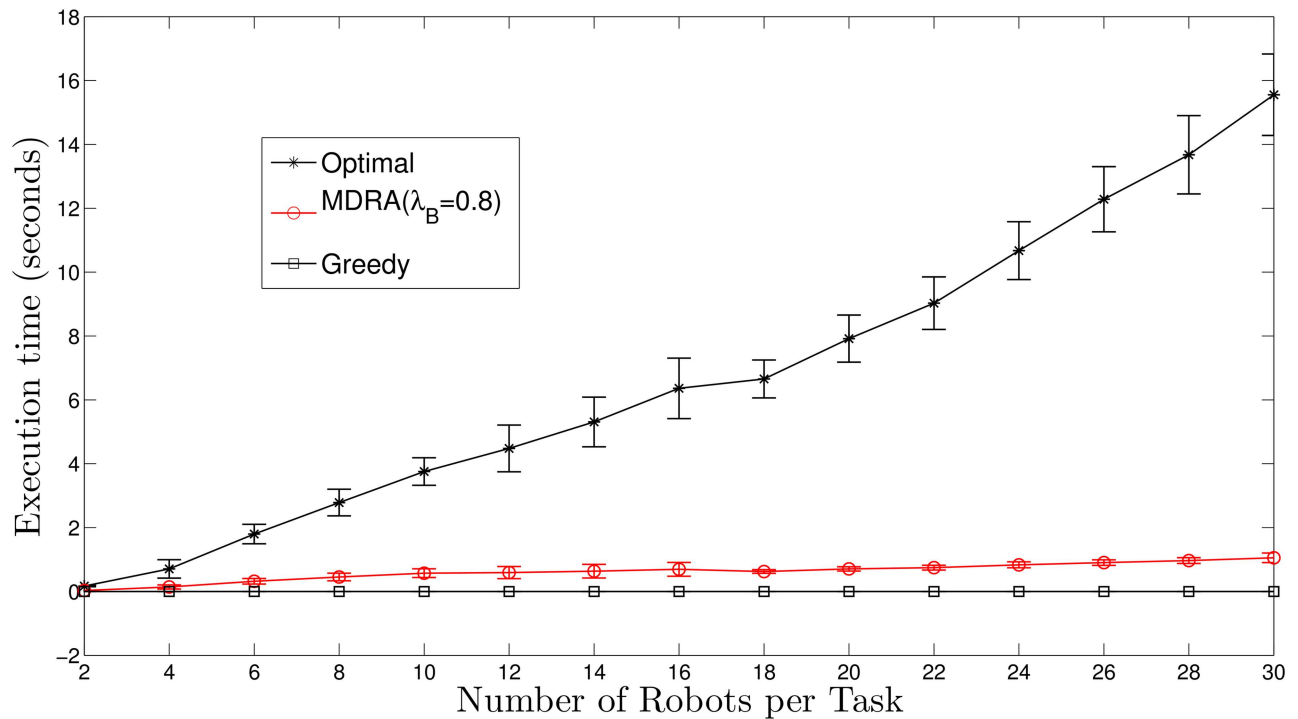


Fig 6. Execution time for soft deadline experiments and $N_t = 30$. The bars stand for the standard deviation (σ^2).

doi:10.1371/journal.pone.0170659.g006

both problems are equivalent and, therefore, the resulting solutions are exactly the same.

Moreover the new problem has the same form as the [Problem 21](#) but with

$m = \min_j \left(-\frac{L_j}{DL_j} - b_j \right)$, for all $j = 1, \dots, N_t$. The new m value is simpler than the original one and avoids division by zero problems.

As for soft-deadline experiments, each experiment has been performed 500 times using a uniform random distribution to generate the position of each object, its weight and deadline, the load capacity of the robots and their velocities. As the scalability of the system has already been demonstrated in the previous section, all the simulations have been performed with 4 tasks ($N_t = 4$) and the robots always use their expected capacities for selecting the best robot (Greedy) and for bidding (SDRA or MDRA).

[Fig 7](#) (see also [S5 Table](#)) shows the utility per robot for different strategies (SDRA, MDRA and greedy) divided by the utility per robot obtained with the optimal method, where the bars stand for the standard deviation σ^2 . As can be seen, the higher number of robots the higher value of this ratio for MDRA, which means that the MDRA results are getting closer to the optimal. Despite of the utility per robot provided by MDRA decreases compared to the soft-deadline case, it reaches values near the 70% of the optimal (ratio = 0.7). This must be considered a very good result, specially taking into account that the complexity of the problem is dramatically increased, which implies that the MATLAB solver takes order of minutes to give a results while MDRA keeps bounded on the order of seconds. In all the cases, the MDRA outperforms the SDRA when $\lambda_B = 0.8$, as happened for soft-deadline utility function experiments. So the optimal value of λ_B seems to be very stable for any environment or utility function. Furthermore, the greedy algorithm keeps very far from the optimal and even from the auction approaches. Finally, the standard deviation decreases regarding the

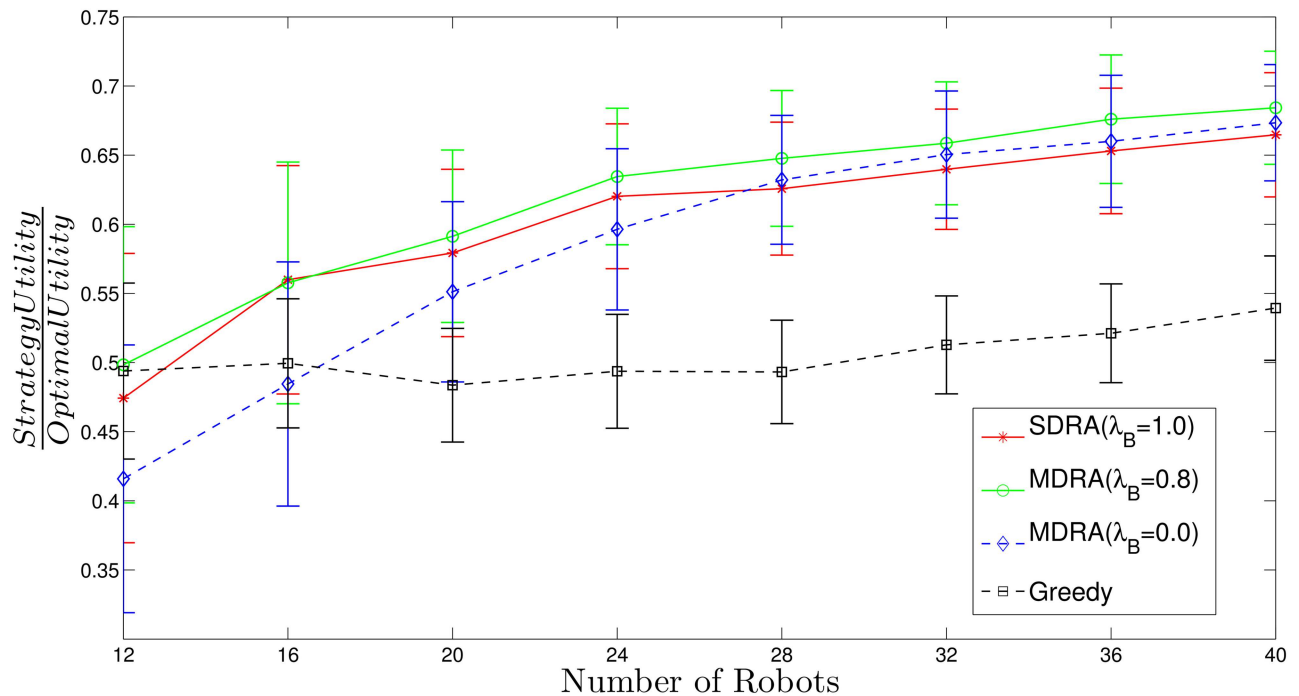


Fig 7. Ratio between auction or greedy utility and optimal utility with hard-deadline. The bars stand for the standard deviation (σ^2).

doi:10.1371/journal.pone.0170659.g007

number of robots. The overlap in the standard deviation bars is due to the great heterogeneity of the carried out experiments. In order to cover a broad spectrum of situations, the standard deviations of the uniform random variables used to generate each experiment are very high, as can be seen in [S1 Table](#). If we focus on a specific set of experiments, the aforementioned overlap is reduced. For example, [Fig 8](#) (see also [S6 Table](#)) shows the ratio between auction or greedy utility and optimal utility with hard-deadline over the 50 experiments with the highest tasks' deadline values. For the sake of simplicity this figure only shows the MDRA results with $\lambda_B = 0.8$. As can be seen, the overlap between the standard deviation of SDRA and MDRA is reduced regarding [Fig 7](#). In order to address these problems, future work will analyze the impact of the environment configuration on the system performance.

As in previous section, [Fig 9](#) shows the ratio between utility per robot using greedy algorithm and MDRA with $\lambda_B = 0.8$, similar results are obtained for other values of λ_B . The results demonstrate how the performance of the greedy algorithm decreases regarding the number of robots. The standard deviation of these experiments can be obtained from the information of [Fig 7](#).

Conclusions and future work

This paper has presented the first formal study about the interference impact on the complexity of the multi-robot coalition formation problem to allocate tasks with deadlines. The formalization of the MRTA problem shows that, on the one hand the interference can be modelled by a linear function and, on the other hand, the optimal solution can be obtained solving an integer linear problem which depends on the utility function characteristics.

As a secondary contribution of this paper, a new auction-like method, called Multiple Objectives Double Round Auction (MDRA), is also presented. A large number of experiments, with both hard and soft-deadline utility functions, has been carried out in order to compare

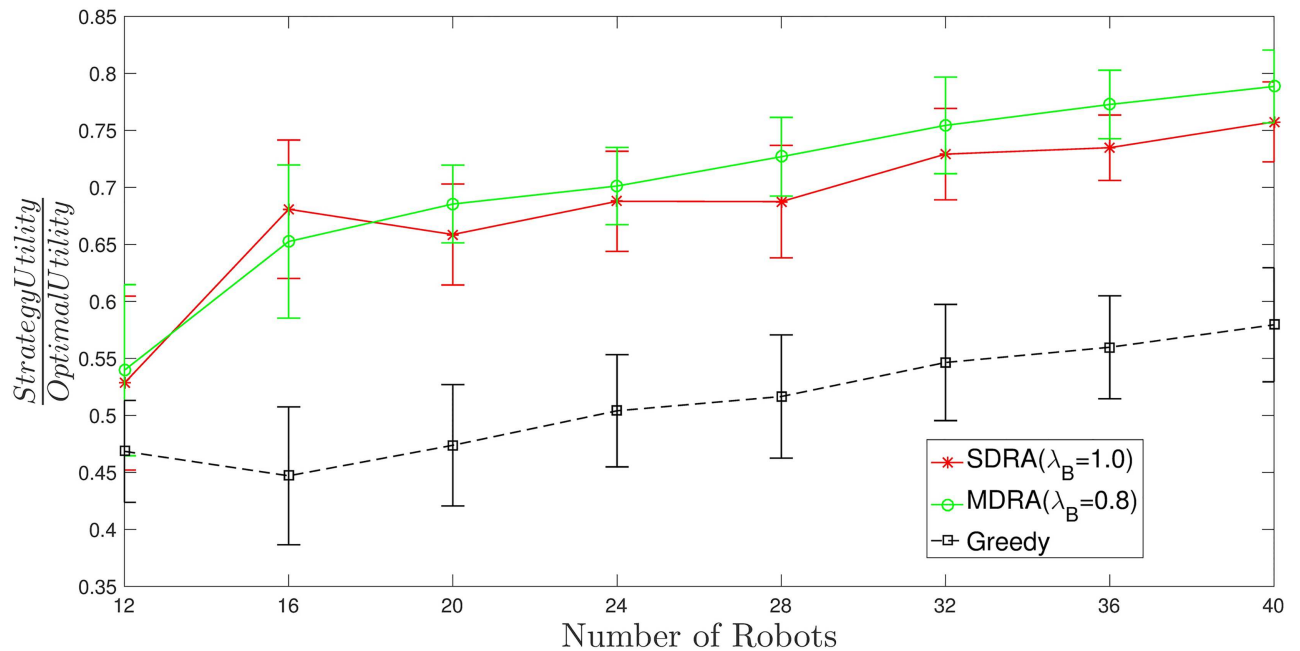


Fig 8. Ratio between auction or greedy utility and optimal utility with hard-deadline using the 50 experiments with the highest tasks' deadline values. The bars stand for the standard deviation (σ^2).

doi:10.1371/journal.pone.0170659.g008

this new approach to the optimal solutions. The results show that MDRA outperforms previous auction methods in the literature and can reach between a 70% (for the hard-deadline utility functions) and more than an 80% (for the soft-deadline utility functions) of the optimal with a very low computational time.

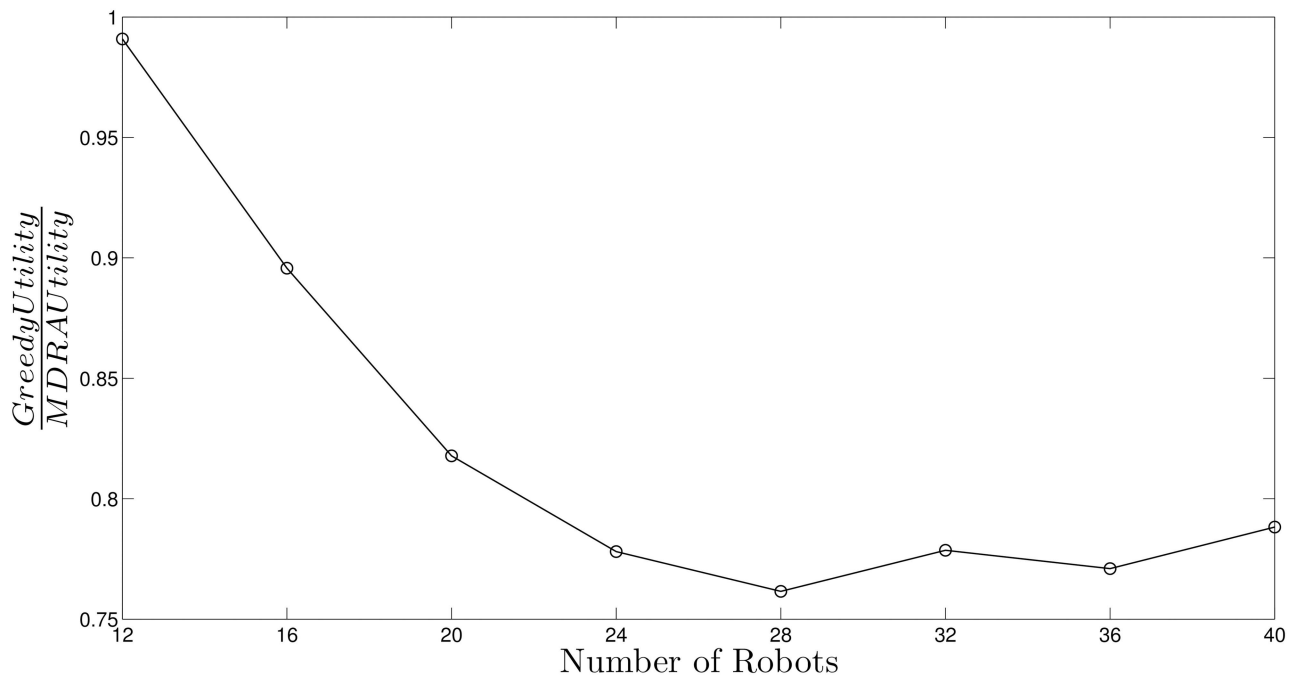


Fig 9. Ratio between greedy utility and MDRA ($\lambda_B = 0.8$) for hard-deadline tasks.

doi:10.1371/journal.pone.0170659.g009

Regarding the aspects to improve this work in the near future, our research is now focused on defining new models for the interference and for measurable errors like: robot's kinematic errors, localization errors, and so on. As pointed out in this paper, other authors propose swarm-like strategies to face up the physical interaction between thousands of robots. Despite our paper does not address swarm issues, the proposed MRTA strategies can be seen as a previous stage before applying lower level swarm-like methodologies. Thus, from the results of these authors (see [45]), we will study how the physical interaction impacts on our MRTA algorithms. Moreover, non-linear interference models, that would need to be addressed with genetic algorithms, and methods for optimizing several goals simultaneously are under consideration. A preemptive auction method, that is, a method that allows the exchange of robots between working groups, is also under study. Finally, we will study how the environments characteristics impact on the system performance.

Supporting Information

S1 Table. Parameters of the uniform random variables ($U(a, b)$) used to generate the experiments.

(PDF)

S2 Table. Mean, standard deviation and median for experiments with soft deadline and $N_t = 15$. These results show the ratio between the utility obtained with MDRA and the utility of the optimal strategy.

(PDF)

S3 Table. Mean, standard deviation and median for experiments with soft deadline and $N_t = 30$. These results show the ratio between the utility obtained with MDRA and the utility of the optimal strategy.

(PDF)

S4 Table. Mean, standard deviation and median of the execution time with soft deadline and $N_t = 30$.

(PDF)

S5 Table. Mean, standard deviation and median of the execution with hard deadline. These results show the ratio between the utility obtained with MDRA and the utility of the optimal strategy.

(PDF)

S6 Table. Mean, standard deviation and median of the execution with hard deadline using the 50 environments with the highest deadline values. These results show the ratio between the utility obtained with MDRA and the utility of the optimal strategy.

(PDF)

Acknowledgments

This work has been partially supported by project DPI2014-57746-C03-2-R and FEDER funds.

Author Contributions

Conceptualization: JG GO OV.

Data curation: JG GO OV.

Formal analysis: JG GO OV.

Funding acquisition: GO.

Investigation: JG GO OV.

Methodology: JG GO OV.

Resources: JG GO OV.

Software: JG GO OV.

Validation: JG GO OV.

Visualization: JG GO OV.

Writing – original draft: JG GO OV.

Writing – review & editing: JG GO OV.

References

1. Lerman K, Galstyan A. Mathematical Model of Foraging in a Group of Robots: Effect of Interference. *Autonomous Robots*. 2002; 13(2):127–141. doi: [10.1023/A:1019633424543](https://doi.org/10.1023/A:1019633424543)
2. Rosenfeld A, Kaminka G, Kraus S. A Study of Scalability Properties in Robotic Teams. In: *Coordination of Large Scale Multiagent Systems*. Springer-Verlag; 2006. p. 27–51.
3. Guerrero J, Oliver G. Multi-robot coalition formation in real-time scenarios. *Robotics and Autonomous Systems*. 2012; 60:1295–1307. doi: [10.1016/j.robot.2012.06.004](https://doi.org/10.1016/j.robot.2012.06.004)
4. Nam C, Shell D. Assignment Algorithms for Modeling Resource Contention and Interference in Multi-Robot Task-Allocation. In: *2014 IEEE International Conference on Robotics & Automation (ICRA)*. Hong Kong: IEEE; 2014. p. 2158–2163.
5. Nunes E, Manner M, Mitiche H, Gini M. A Taxonomy for Task Allocation Problems with Temporal and Ordering Constraints. Technical Report TR 16-007. University of Minnesota.; 2016.
6. Rubenstein M, Cabrera A, Werfel J, Habibi G, McLurkin J, Nagpal R. Collective Transport of Complex Objects by Simple Robots: Theory and Experiments. In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*. St. Paul, MN, USA; 2013. p. 47–54.
7. Gerkey BP, Mataric M. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*. 2004; 23 (9):939–954. doi: [10.1177/0278364904045564](https://doi.org/10.1177/0278364904045564)
8. Gerkey BP. *On Multi-Robot Task Allocation*. Center of Robotics and Embedded Systems, University of Southern California. Los Angeles, USA; 2003.
9. Kuhn HW. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*. 1955; 2(1):83–97. doi: [10.1002/nav.3800020109](https://doi.org/10.1002/nav.3800020109)
10. Balas E, Padberg MW. Set partitioning: A survey. *Siam Review*. 1976; 18(4):710–760. doi: [10.1137/1018115](https://doi.org/10.1137/1018115)
11. Korte B, Vygen J. *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag, Berlin; 2000.
12. Garey MR, Johnson DS. ‘Strong’ NP-Completeness Results: Motivation, Examples, and Implications. *Journal of the ACM (JACM)*. 1978; 25(3):499–508. doi: [10.1145/322077.322090](https://doi.org/10.1145/322077.322090)
13. Vig L. *Multi-Robot Coalition Formation*. Graduate School of Vanderbilt University. Nashville, USA; 2006.
14. Service T, Adams J. Coalition formation for task allocation: theory and algorithms. *Autonomous Agents and Multi-Agent Systems*. 2010;.
15. Shehory O, Kraus S. Methods for task allocation via agent coalition formation. *Artificial Intelligence*. 1998;(1–2):165–200. doi: [10.1016/S0004-3702\(98\)00045-9](https://doi.org/10.1016/S0004-3702(98)00045-9)
16. Agarwal M, Kumar N, Vig L. Non-additive multi-objective robot coalition formation. *Expert Systems with Applications*. 2013; 41:3736–3747. doi: [10.1016/j.eswa.2013.11.044](https://doi.org/10.1016/j.eswa.2013.11.044)
17. Thamilselvan R, Balasubramanie P. Integrating Genetic Algorithm, Tabu Search Approach for Job Shop Scheduling. *International Journal of Computer Science and Information Security*. 2009; 2(1):134–139.
18. Balas E, Simonetti N, Vazacopoulos A. Job shop scheduling with setup times, deadlines and precedence constraints. *Journal of Scheduling*. 2008; 11(4):253–262. doi: [10.1007/s10951-008-0067-7](https://doi.org/10.1007/s10951-008-0067-7)

19. Jones EG, Dias MB, Stentz A. Time-extended Multi-robot Coordination for Domains with Intra-path constraints. In: *Robotics: Science and Systems (RSS)*. Seattle, USA; 2009.
20. Gendreau M, Tarantilis CD. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Montreal, Canada: Interuniversity Research Center on Enterprise Networks, Logistic and Transportation; 2010. CIRRELT 2010 04.
21. Nunes E, Nanjanath M, Gini M. Auctioning robotic tasks with overlapping time windows. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*. vol. 3; 2012. p. 1211–1212.
22. Koes M, Nourbakhsh I, Sycara K. Heterogeneous Multirobot Coordination with Spatial and Temporal Constraints. In: *20th National Conference on Artificial Intelligence (AAAI)*. Boston, USA; 2005. p. 1292–1297.
23. Yu L, Cai Z. Robot Exploration Mission Planning Based on Heterogeneous Interactive Cultural Hybrid Algorithm. In: *5th International Conference on Natural Computation*. Tianjin, China; 2009. p. 583–587.
24. Smith SL, Bullo F. The dynamic team forming problem: Throughput and delay for unbiased policies. *Systems and Control Letters*. 2009; 58:709–715. doi: [10.1016/j.sysconle.2009.07.001](https://doi.org/10.1016/j.sysconle.2009.07.001)
25. Cao YU, Fukunaga AS, Kahng AB. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*. 1997; 4(1):7–23. doi: [10.1023/A:1008855018923](https://doi.org/10.1023/A:1008855018923)
26. Ducatelle F, Förster A, Caro GD, Gambardella L. Task allocation in robotic swarms: new methods and comparisons. Lugano, Suiza: IDSIA—Dalle Molle Institute for Artificial Intelligence; 2009. IDSIA-01-09.
27. Low KH, Leow WK, Ang MH. Task Allocation via Self-Organizing Swarm Coalitions in Distributed Mobile Sensor Network. In: *19th National Conference on Artificial Intelligence*. San Jose, USA; 2004. p. 28–33.
28. Dawson S, Wellman BL, Anderson M. Identification of Issues in Predicting Multi-Robot Performance through Model-Based Simulations. *Intelligent Control and Automation*. 2011; 2:133–143. doi: [10.4236/ica.2011.22016](https://doi.org/10.4236/ica.2011.22016)
29. Pini G, Brutschy A, Birattari M, Dorigo M. Interference Reduction through Task Partitioning in a Robotic Swarm, Or: Don't you Step on My Blue Suede Shoes. In: *6th International Conference on Informatics in Control, Automation and Robotics*. Milan, Italy; 2009. p. 52–59.
30. Khaluf Y, Birattari M, Rammig F. Analysis of long-term swarm performance based on short-term experiments. *Soft Computing*. 2015; 20(1):37–48. doi: [10.1007/s00500-015-1958-0](https://doi.org/10.1007/s00500-015-1958-0)
31. Smith RG. The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*. 1980; 29(12):1104–1113. doi: [10.1109/TC.1980.1675516](https://doi.org/10.1109/TC.1980.1675516)
32. Vig L, Adams JA. Market-based Multi-Robot Coalition Formation. In: *8th International Symposium on Distributed Autonomous Robotic Systems*. Minneapolis, USA; 2006. p. 227–236.
33. Paquet S. Distributed Decision-Making and Task Coordination in Dynamic, Uncertain and Real-Time Multiagent Environments. *Faculté de Sciences et de Génie, Université Laval Québec*. Quebec, Canada; 2006.
34. Tang F, Parker LE. Distributed multi-robot coalitions through ASyMTRe-D. In: *International Conference on Intelligent Robots and Systems (IROS)*. Edmonton, Canada; 2005. p. 2606–2613.
35. Bayram H, Bozma HI. Coalition Formation Games for Dynamic Multirobot Tasks. In: Akin HL, Amato NM, Isler V, van der Stappen AF, editors. *Algorithmic Foundations of Robotics XI*. vol. 107 of Springer Tracts in Advanced Robotics. Springer International Publishing; 2015. p. 37–54.
36. Jones EG, Dias MB, Stentz A. Learning-enhanced Market-based Task Allocation for Oversubscribed Domains. In: *International Conference on Intelligent Robots and Systems (IROS)*. San Diego, USA; 2007. p. 2308–2313.
37. Lemaire T, Alami R, Lacroix S. A distributed Tasks Allocation Scheme in Multi-UAV Context. In: *International Conference on Robotics and Automation (ICRA)*. vol. 4. New Orleans, USA; 2004. p. 3622–3627.
38. Melvin J, Keskinocak P, Koenig S, Tovey C, Ozkaya BY. Multi-Robot Routing with Rewards and Disjoint Time Windows. In: *International Conference on Intelligent Robots and Systems (IROS)*. San Diego, EUA; 2007. p. 2332–2337.
39. Campbell A, Wu A, Shumaker R. Multi-Agent Task Allocation: Learning When to Say No. In: *10th annual conference on Genetic and evolutionary computation*. Atlanta, USA; 2008. p. 201–208.
40. Sellner B, Simmons R. Duration Prediction for Proactive Replanning. In: *International Conference on Robotics and Automation (ICRA)*. Pasadena, USA; 2008. p. 1365–1371.
41. Luo L, Chakraborty N, Sycara K. Distributed algorithm design for multi-robot task assignment with deadlines for tasks. *IEEE Transactions on Automation Science and Engineering*. 2015; 12(3):876–888. doi: [10.1109/TASE.2015.2438032](https://doi.org/10.1109/TASE.2015.2438032)

42. Lein A, Vaughan RT. Adaptive multi-robot bucket brigade foraging. In: 11st International Conference on the Simulation and Synthesis of Living Systems. Winchester, UK; 2008. p. 337–342.
43. Dahl TS, Mataric M, Sukhatme GS. Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*. 2009; 57:674–687. doi: [10.1016/j.robot.2008.12.001](https://doi.org/10.1016/j.robot.2008.12.001)
44. Nam C, Shell DA. Assignment algorithms for modeling resource contention and interference in multi-robot task-allocation. In: *IEEE Int. Conf. on Robotics and Automation*; 2014.
45. Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. *Science*. 2014; 345(6198):795–799. doi: [10.1126/science.1254295](https://doi.org/10.1126/science.1254295) PMID: [25124435](https://pubmed.ncbi.nlm.nih.gov/25124435/)
46. Winfield AF. Foraging Robots. In: Meyers RA, editor. *Encyclopedia of Complexity and Systems Science*. New York, NY: Springer New York; 2009. p. 3682–3700. Available from: http://dx.doi.org/10.1007/978-0-387-30440-3_217.
47. Chinneck JW. *Practical Optimization: A Gentle Introduction*. Available Online (May 2016); 2015. Available from: <http://www.sce.carleton.ca/faculty/chinneck/po.html>.
48. Chen DS, Batson RG, Dang Y. *Applied Integer Programming: Modeling and Solution*. Wiley; 2010.
49. Guerrero J, Oliver G. Physical interference impact in multi-robot task allocation auction methods. In: *IEEE Workshop on Distributed Intelligent Systems*. Praga, Czech Republic; 2006. p. 19–24.
50. Gerkey B, Vaughan RT, Howard A. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems'. In: *In Proceedings of the 11th International Conference on Advanced Robotics*; 2003.
51. Guerrero J. *Robotics Youtube Channel*; 2016. Available from: <https://www.youtube.com/user/jguerreroSRV>.