A Market Mechanism for QoS–aware Multi–Robot Task Allocation

Francesco Barile*, Alessandra Rossi†, Maricarla Staffa‡, Claudia Di Napoli§ and Silvia Rossi‡
* Dipartimento di Matematica e Applicazioni, Università degli Studi di Napoli "Federico II"
francesco.barile@unina.it
† Dipartimento di Fisica, Università degli Studi di Napoli "Federico II"
alessandra.rossi@unina.it
‡ Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli "Federico II"
{ mariacarla.staffa, silvia.rossi}@unina.it
§ Istituto di Calcolo e Reti ad Alte Prestazioni, CNR
claudia.dinapoli@cnr.it

***Abstract*—Market mechanisms, such as auctions and negotiations, are often used for efficient task allocation in multi–robot domains where tasks are characterized by quality parameters that are related to the way the task is executed depending on the specific robot capabilities. In these cases, usually robots negotiate their respective assignments in order to optimize task distribution according to their own utility function. In this work, a market–based negotiation mechanism is proposed to allocate tasks to a set of robots by taking into account end–to–end requirements that the complete allocation should meet in terms of the considered quality parameters. Negotiation takes place on these parameters that are considered goods to be traded by the individual robots, depending on their strategies, so that they can successfully negotiate to obtain the task allocation.**

I. INTRODUCTION

Networked robotic devices, such as mobile robots, drones, unmanned vehicles and position sensors, are starting to be concretely employed in many real contexts, especially in emergency and rescue activities, where navigation and search operations take place both in indoor and outdoor environments with the human supervision. In this context, the use and the arrangement of multiple robots has been proven to help in achieving the task, both in terms of execution speed, increase of robustness, reliability, quality and performance of solutions in general. However the displacement of multiple robots comes with the cost of coordination to allocate resources and tasks among them in a way that enables them to accomplish their mission efficiently and reliably.

Researchers have recently applied the principles of market economies to multi–robot coordination [1]. Market mechanisms, such as auctions and negotiations, are often used for efficient task allocation in multi–robot domains [2]. More specifically, they are used to determine the optimal allocation (distribution) of tasks to robots by considering it as an optimization problem with some cost functions, such as, for example, distance to travel, battery consumption or battery autonomy. These functions depend on the specific application, and they are usually expressed as a minimization of the robots individual costs, or the total cost of the mission, although others are possible. In any case, the optimal task allocation problem is known to be NP–hard.

In this work, our assumption is that, in a more broad context, ubiquitous robots can be represented as heterogeneous self–interested agents that can provide different services (e.g., to execute a specific task) depending on their own capabilities, which are a priori defined. According to their capabilities, robots may provide services with different performance levels, which can be evaluated depending on dynamic information and that can be traded, so allowing robots to dynamically adjust the performance they execute a task with, in order to be assigned the task. Of course, it is not possible to obtain an optimal allocation, but any allocation of tasks that meets global constraints on the complete allocation is considered an acceptable solution.

II. MULTI-ROBOT TASK ALLOCATION

Mobile robot teams can fulfill a goal more efficiently than a single robot by sharing the workload and by optimizing the use of available resources. In fact, a given system–level (or global) task can be divided into m sub–components $\{T_1, \ldots, T_m\}$ which can be assigned to individual n robots $\{R_1, \ldots, R_n\}$ that can execute them. How a global task is decomposed is a crucial problem addressed by the planning methods, while the distribution of subtasks to robots is known as task allocation.

A task decomposition algorithm has to consider both the tasks nature, and restrictions in order to accommodate them among the agents for the execution. Tasks can be long-term (e.g. monitoring an environment) or transient (e.g. looking for objects), they can have different complexity and specificity, and they can be performed by a single robot or multiple robots. Tasks have a well–defined set of constraints depending on the problem domain, the use of limited technologies, the scarcity of resources, and environment conditions. The most common ones are:

- Partial ordering, i.e. a task has to be completed before or after one or more others;
- Coupling, i.e. two or more tasks have be executed concurrently;
- Incompatibility, i.e. two or more tasks can not be concurrently executed;

- Time windows, i.e. a task has a duration condition (i.e. a deadline) to be met;
- Mobility interferences, i.e. robot's ability to perform a task is limited (e.g., it can not move in a narrow space due to its dimensions).

Such constraints model the functional relationships among tasks that can be expressed using a particular data structure. There are several planning approaches to generate such constrained decompositions that can be expressed by task trees. In [3], Doherty et al. presented a task specification language and an abstract distributed data structure, called Task Specification Tree (TST). Each node of a TST represents a task. A TST has a constraint network formed by a constraint model for each node and tree constraints, expressing the relations between the nodes. Hierarchical Task Networks (HTN) are used to represent various levels of task semantics, that have been extensively used for planning in AI domains, and have been imported to the robotic domain in abundant researches and applications.

In the literature [4], tasks may be also characterized by different parameters that are related to the way the task is executed, i.e. to non–functional attributes whose values are determined by the specific robot able to execute it. The most common parameters are:

- Cost, i.e., a measure of the effort made by the robot to execute a task, in terms of time to reach a goal, energy and resources consumed, and so on [5];
- Accuracy, i.e., a characterization of goodness of the task execution [6] performed by a robot (e.g. the map accuracy produced using a laser range-finder);
- Reward, i.e., a measure of the profit gained by the robot for performing a task [7];
- Priority, i.e., the task urgency required for its execution (higher priority tasks have to be executed before lower priority ones [8]).

When a complex task has to be executed by a team of robots with specific values of these non–functional parameters, the allocation of subtasks to the suitable robots, is an NP–hard decision problem. Market–based approaches for task allocation have received significant attention within the robotics research community [1], [9], in order to efficiently produce sub–optimal allocations [2].

Generally, in reply to a task request, robots may submit bids based on their abilities to perform the tasks and the highest (lowest) bid wins the assignment. Moreover, when more than one task can be assigned to each robot (and the evaluation of the robot non–functional parameters depends on its complete assignment), the robots can negotiate their respective assignments in order to optimize the task distribution. Typical allocation approaches adopt optimization algorithms based on some utility function. Indeed, the measure of utility is considered as a combination of these non–functional parameters, and it is used for evaluating the impact of the chosen tasks execution on the system performance. Utility functions can be based on different parameters, such as sensors-based metrics [10] or sophisticated planner–based measures [11] (multi–attribute utility theory). Utility represents a trade–off between accuracy and costs able to obtain an approximated result [10].

III. QoS Market–based Negotiation for Task Allocation

The problem of allocating subtasks composing a complex task with specific non–functional constraints to a team of robots is similar to select services for delivering a composition of services with Quality of Service (QoS) constraints. So, service–based computing technologies can be effectively integrated and utilized into robotic applications [12], [13].

In this work, we propose the use of automated negotiation as a mechanism to allocate subtasks to a set of robots, by allowing the individual robots to negotiate on the values of the non–functional parameters characterizing the complex task. These values may depend on dynamic conditions of each robot, such as its computational and physical resources at the time the allocation process starts, the number of tasks it was already allocated, the reward it gets, and so on. It is crucial to adopt allocation mechanisms that can take into account of such a variability. In addition, robots may decide to change the values of these non–functional parameters to accommodate some global requirements specified for the complex task in order to have the subtask assigned.

Here, it is assumed that a complex task is represented as a task tree, we refer to as an *Abstract Task Tree* (ATT), whose leaves are the subtasks composing it, we refer to as *Abstract Task*s (Ts). The complex task is required to be executed with specific QoS end–to–end requirements, referring to non-functional attributes of the complex task. The request is managed by an agent, we refer to as the *Task Allocator Agent* (TAA), responsible for finding an allocation of these subtasks to a team of robots providing them with values of the considered attributes that, once aggregated, meet the end–to–end requirements. Robots are modeled as task providers, i.e. "market vendors" of tasks characterized by QoS values accounting for these non–functional attributes. They negotiate these values with the TAA, so that a successful negotiation determines an allocation of subtasks to the robots able to execute them with suitable values of the non–functional attributes. Robots, referred to as *Task Provider Agent*s (TPAs), aim to win the negotiation so to obtain the allocation of the task, and they may change the QoS values they provide during negotiation according to their own negotiation strategies. In fact, while some values may depend only on the robot capabilities, others can be modified proactively by the robot. For example, it can dynamically modify the execution speed of a task or the accuracy of a provided sampling.

More specifically, each robot is modeled as composed of:

- a *deliberative* layer responsible for negotiating with the TAA the allocation of tasks;
- a *dispatching* layer responsible for scheduling the tasks allocated to the robot;
- an *execution* layer responsible for the actual execution of the allocated tasks.

In the following only the deliberative layer is addressed, while the other two layers are outside the scope of the work.

The negotiation mechanism adopted in this work is based on the one proposed in [14], a one–to–many iterative protocol, used to select services when a composition of services is required with specific end–to–end requirements. It allows a composer agent, acting on behalf of the consumer, to negotiate QoS attribute values of the functionality requested in the composition, both with different providers of the same functionality, and with the providers of different functionalities in a coordinated way, since it is not always possible to decompose these requirements in individual requirements for each functionality of the composition.

Here the same negotiation protocol is adopted, while new negotiation strategies are provided, specifically designed for the multi–robot task allocation domain.

The negotiation protocol consists of a number of *negotiation rounds* proceeding until either the negotiation is successful (i.e., all subtasks are allocated), or a *deadline* (i.e., a maximum number of allowed rounds) is reached. The deadline can be set according to the nature of the required tasks, or other criteria depending on the considered scenario. At each negotiation round, the TAA sends n *call for proposal*s (cfps), one for each available TPA, specifying the subtasks in the ATT to be allocated ($\{T_1, \ldots, T_m\}$), with $n \geq m$, and it waits for replies for a given time, known as the *expiration time* of a negotiation round. Each TPA_j provides an offer o_j containing the k QoS values $q_{i,k}$ for each of the T_i it is able to perform, and $m - k$ $NULL$ values for the T_i it is unable to perform. Such QoS values represent measures of non–functional parameters characterizing the way the robot can execute the task T_i (such as for example battery consumption, the speed, the accuracy, and so on). Hence, the TAA receives a set of offers $\{o_1, ..., o_n\}$, with n the number of TPAs.

At the first negotiation iteration, the TAA checks if there are offers for each required subtasks specified in the ATT. If there are no offers for all the required subtasks, it declares a failure since it is not possible to find a set of TPAs for all required T_i. Otherwise, it evaluates the received offers and the iteration number, and, according to the result of such evaluation, it performs one of the following actions:

1) if the aggregated QoS values of the received offers do not meet the global QoS requirements and the deadline is not reached (not final iteration), it asks for new offers by sending again n cfps, so starting another negotiation round;
2) if the aggregated QoS value of the received offers meets the global QoS requirements (final iteration), it selects the best set of offers, in terms of its own utility, and it accepts such offers and rejects the others, so ending the negotiation successfully.
3) if the deadline is reached without a success, it declares a failure to all robots that took part in the negotiation (final iteration).

A. Negotiation strategies

In order to decide whether to accept a set of offers, the TAA evaluates first if there is a combination of offers satisfying the global end–to–end requirements, intended as upper bounds for the aggregated offers values. The evaluation function used by the TAA is a solver of an Integer Linear Programming problem formulated as follows:

$$\sum_{j=1}^{n} x_{i,j} = 1, \forall i = 1, \ldots, m \tag{1}$$

$$aggr_i(\sum_{j=1}^{n} x_{i,j} \cdot q_{i,j,k}) \leq Q_k, \forall k = 1, \ldots, r \tag{2}$$

$$\sum_{j=1}^{n} x_{i,j} \cdot q_{i,j,k} \neq NULL, \forall k = 1, \ldots, r \tag{3}$$

Hence, there are $n{\cdot}m$ decision variables $x_{i,j}$, where i identifies one of the m Ts, and j identifies one of the n TPAs that replied to the cfp. Such variables assume value 1 if the $j - th$ TPA is selected for the $i - th$ T, 0 otherwise. Equation 3 verifies that agent j is able to execute the task corresponding to the $i - th$ T. Equation 1 verifies that exactly one TPA has to be selected for each T (such value can be changed in case task replication is required). Equation 2 evaluates that the global constraint for each considered non–functional parameter is met. Typically, additive (e.g., price and execution time) and multiplicative (e.g., reliability and availability) parameters are considered [15], so $aggr$ is either a sum or a multiplication over the number of Ts.

Once that combinations of offers that satisfy the ILP are found, the TAA selects the one that maximizes its own utility, that is evaluated as follows:

$$U_{TAA} = \frac{1}{r} \sum_{k=1}^{r} \frac{Q_{max'(k)} - aggr_i(\sum_{j=1}^{n} x_{i,j} \cdot \overline{q}_{i,j,k})}{Q_{max'(k)} - Q_{min'(k)}} \tag{4}$$

where, $Q_{max'(k)} = aggr_k(max(q_{i,j,k}))$ aggregates the local maxima of the offers received for the $i - th$ task, $Q_{min'(k)} = aggr_k(min(q_{i,j,k}))$ aggregates the corresponding local minima, and $\overline{q}_{i,j,k}$ are the values of one offer of the found combinations.

The adopted negotiation mechanism is one–sided, since the TAA cannot make counterproposals, but it only evaluates offers in an aggregated manner, while TPAs can send new offers. The TAA could formulate counterproposals relying on heuristics methods to decompose the end–to–end requirements into individual requirements. But this approach is not followed in the present work, since it increases the constraints to be satisfied by individual offers. In fact, when an aggregated value is required, it is possible that individual offers are acceptable when aggregated with the others, while they are not acceptable according to the chosen decomposition criteria.

TPAs are provided with strategies and tactics to generate offers to send at each negotiation round to the TAA for the subtasks they are able to execute. Several strategies and tactics have been proposed in the literature for different types of negotiation [16]. Here, a stochastic monotonic concession strategy is adopted for the TPAs, that models a cooperative behavior coming from the robot objective to obtain the allocation of the task.

IV. A Practical Example

In our reference scenario, we assume that robots may execute different type of tasks, and that they have the goal of maximizing the number of tasks to be executed according to the resources they have. For simplicity, we consider as a working example a global task ATT that consists in covering a specific total distance, and it is requested to be executed with a global completion time TCT_{req}, that represents the only constraint to be met. The task is decomposed in 3 subtasks $\{T_1, T_2, T_3\}$, each one characterized by a pair $< x_i, dir_i >$, where x_i is a distance in a specific direction dir_i. The tasks have to be executed sequentially. At each negotiation, the 3 subtasks will be assigned to 3 different robots if the negotiation is successful, i.e. if the sum of the execution times tct_i proposed by each triple of robots does not exceed the upper bound TCT_{req}:

$$\sum_{i=1}^{3}(\sum_{j=1}^{n} x_{i,j} * tct_{i,j}) \leq TCT_{req} \tag{5}$$

where n is the number of available robots. Hence, the negotiation is single–issue and the issue is additive.

It is assumed that each robot is able to cover a distance only in a specific direction, so it is able to execute only one of the 3 tasks, i.e. each subtask has to be allocated to a different robot.

The TAA initiates the negotiation by issuing the n call for proposals, one for each robot (or TPA), containing the reference to the 3 subtasks to be executed. Each robot R_j will reply providing an offer characterized by a triple of values, with a $NULL$ value for the subtasks it is unable to execute, and a task completion times $tct_{i,j}$ the robot R_j offers for the subtask T_i it is able to execute. The offered time $tct_{i,j}$ depends on the velocity $v_{i,j}$ the robot j decides to cover the distance of subtask T_i with.

Each robot R_j is represented by a state s affecting the offers it provides to the TAA for a specific task T_i. In particular, it is assumed that the state s of each robot is a n–tuple composed of the current battery level $b_s \in [bmin, bmax]$, depending on the current task allocation, a function f determining the battery consumption according to the velocity the robot offers to cover the distance of a task T_i with, the tasks $\{T_k, \ldots, T_h\}$ that it committed to execute and that are in its agenda, and a minimum allowed velocity $vmin$ that is the minimum functioning robot velocity (i.e., it depends on the robot and not on the task).

The initial state is defined as $s_0 =< b_{s0}, f, \{\}, vmin >$, and it is updated at the end of each negotiation. The maximum level of battery b_{s0} can be different for each robot, and it determines the maximum velocity the robot can execute a task with, when it is fully charged (a maximum initial value when the robot is not committed to execute any task), i.e. $b_{s0} = \alpha \cdot vmax_{i,j} \cdot x_i$. Hence, if the robot executes the task at the maximum velocity, it will use all its battery.

In order to reply to a cfp, the $j-th$ robot evaluates its current state in terms of the current battery level b_s, to determine the maximum velocity it can execute a task with ($vmax_{i,j}$), where

$$vmax_{i,j} = b_s/T_i. \tag{6}$$

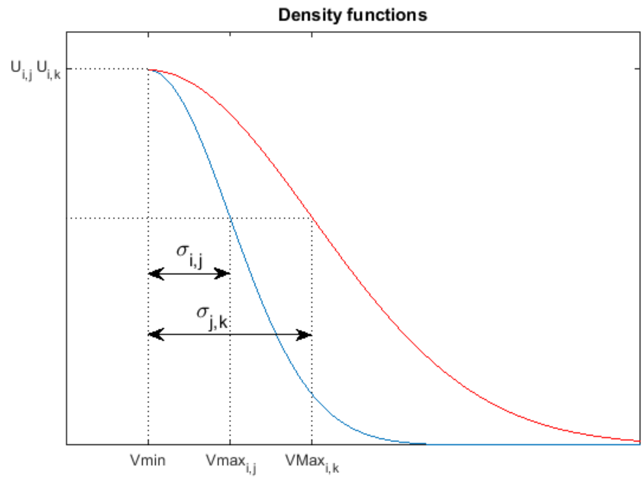

Fig. 1. An example of Gaussian functions for two robots j and k.

To model a stochastic behavior of robots taking part in the negotiation, the offers generation strategy is given by a Gaussian function, representing the probability distribution of the offers in terms of the robot utility, as proposed in [17]. In particular, the Gaussian function depends on the specific task T_i, and it accounts for the robot utility obtained when covering the distance of the task T_i at velocity $v_{i,j}$ ($tct_{i,j} = x_i/v_{i,j}$). As shown in Figure 1, the mean value of the Gaussian $U_{i,j}(vmin)$ represents the best offer the R_j may propose in terms of its own utility with the highest probability to be selected, and it corresponds to the maximum $tct_{i,j}$ it may provide the T_i with, i.e. the one obtained by executing the task at the minimum velocity possible. The rationale of this choice is that the robot prefers to use the less battery possible when providing the task, so that it can try to maximize the number of tasks it may be assigned, so its most convenient offer is the one that minimizes the battery discharge. The standard deviation σ represents the attitude of the R_j to concede during negotiation, and it is given by $\sigma_{i,j} = vmax_{i,j} - vmin$, if $vmax_{i,j} > vmin$, 0 otherwise. It takes into account the computational load of the robot in terms of the number of tasks it was assigned to execute. In fact, at each new negotiation the σ is decreased according to the battery consumption due to an assigned task since the $vmax_{i,j}$ depends on the remaining battery level, so generating a new Gaussian function to be used in the new negotiation. The value $vmax_{i,j}$ represents the reservation value for the robot, since it is the maximum velocity it can execute the task with its current level of battery. Furthermore, the robot can make an offer only if the remaining battery level allows it to cover the distance of the task T_i at the minimum velocity. The parameter σ varies from robot to robot providing the same task T_i, in such a way that the lower its computational load (in terms of available battery) is, the more it is available to concede in utility, and the lower its reservation value is.

The battery consumption is assumed to be linearly dependent on the velocity the robot can cover the distance x_i of the task T_i, according to the following equation:

$$f = \alpha \cdot v_{i,j} \cdot x_i \quad (7)$$

Whenever the robot R_j is selected for executing task T_i at the end of a negotiation, it adds it to its agenda and evaluates the remaining battery as follows:

$$b_{s'} = b_s - \alpha \cdot v_{i,j} \cdot x_i \quad (8)$$

The new state will be $s' =< b_{s'}, f, \{T_i\}, vmin >$.

In Figure 1, the functions associated to two different robots for the same T_i are reported. The best offer is the same for both robots (i.e., $U_{i,1}(vmin) = U_{i,2}(vmin)$), since it is assumed that the robots have the same minimum velocity, while their concession strategies are different according to their workload when the negotiation takes place. In fact, $\sigma_{i,1}$ is greater than $\sigma_{i,2}$ meaning that R_1 has a lower computational load than R_2, so it concedes more in utility than R_2.

At each negotiation round, each robot generates, following its Gaussian distribution, a new utility value corresponding to a new offer. If this value is lower than the one offered in the previous round and within the negotiation set, then the robot proposes the new value. The negotiation set is $[U_{i,j}(vmin); U_{i,j}(vmin+\sigma_{i,j}]$. If the new utility value generated is higher than that offered in the previous round, or it is outside the negotiation set, the robot proposes the same value offered in the previous round. This strategy allows to simulate different and plausible behaviors of robots that prefer not having a consistent loss in utility, even though by increasing the number of negotiation rounds the probability for the robot to move towards its reservation value increases.

A. Results

In our testing scenario, we evaluate the behavior of a set of negotiations carried out with 9 TPAs to allocate the complex task ATT composed of 3 subtasks T_1, T_2, T_3. There are 3 robots able to perform one of the three tasks.

Multiple negotiations are carried out since the TAA aims to allocate as many complex tasks as possible with the available robots. In this preliminary analysis it is assumed that each $x_i = 3m$, $dir_i = north, south, west$, and each robot can perform the task with a velocity v_j in a range between $vmin_j = 0.05m/s$ and $vmax_j = 0.3m/s$. The initial battery for each robot is $b_{s0} = vmax_j * x_i$, and the battery consumption is computed as defined in Equation 8 with $\alpha = 1$.

In such scenario, if a robot performs a single task with its minimum velocity ($vmin_j = 0.05m/s$), each task is completed in $tct_{i,j} = 60$ seconds, while the complex task requires $TCTreq = 180$ seconds. Hence, at the minimum velocity, each robot could execute 6 subtasks before consuming all its battery. Without considering the global constraint, 18 complex tasks (ATT) could be allocated to the robots. So, we set the maximum number of negotiations at 18. If we assume that robots perform the tasks at maximum velocity, each robot can execute only 1 task, so 3 complex tasks could be allocated to the available robots.

We simulate the set negotiations with 3 different global time constraints for the complex task ($TCT_{req1} = 100s$, $TCT_{req2} = 130s$ and $TCT_{req1} = 160s$), and, for each configuration, we perform 100 runs. The deadline of each negotiation is 100 rounds. Table I reports for each configuration the following information: average values, standard deviation, maximum and minimum values of the number of allocated complex tasks, the number of tasks allocated to each robot, the remaining battery at the end of the all negotiations, the time to complete each complex task, and the number of rounds for successful negotiations (i.e., the ones terminated with a complete allocation).

When a global constraint is required, the possibility to offer different values of the constraint parameter, allows to allocate a greater number of complex tasks with respect to a static allocation at a fixed velocity for each robot, in fact, at a fixed maximum or medium velocity, only 3 complex tasks could be allocated. As expected, when increasing the required TCT for the task execution, the number of complex tasks allocated increases, with a consequent increase in battery consumption. Moreover, let us note that the probabilistic distributions of offers lead to a global time value that is lower than the set constraint. Hence, different concession strategies may further improve these results on the constraints satisfaction. Finally, the number of rounds necessary to reach a complete allocation is smaller in the first negotiations, while it drastically increases in the final ones since it is more difficult to find a complete allocation once the battery level of the robots decreases. This is also shown by the high values of standard deviation for the *Rounds* number reported.

More specifically, this behavior is shown in Figures 2 and 3, reporting respectively, for one run of a complete set of negotiations with the $TCT_{req} = 130s$, the battery levels for each robot and the number of rounds for each negotiation. As shown in Figure 3, in the first four negotiations, robots easily find an agreement in 1 round; while, from the forth to the seventh negotiation the number of rounds in order to reach an agreement increases considerably leading to a lack of agreements from the seventh negotiation onward. This behavior is explained in Figure 2 where the trends of the battery level for each robot show that after the seventh negotiation the battery levels decreased for each robot. In addition the stochastic behavior of the offers generation leads to different final battery levels for different robots.

V. Conclusion

In this paper, we investigated the possibility to adopt heuristic approaches to find allocations of tasks to teams of robots when tasks are characterized by non–functional attributes, and the complete allocation has to meet global constraints expressed as end–to–end requirements of these non–functional attributes. A market–based negotiation mechanism where robots are modeled as task providers with negotiable non–functional parameters is proposed as an heuristic method to address the NP–hard task distribution problem.

This preliminary study showed that the adopted negotiation mechanism is a promising approach when trying to optimizing the number of complete allocations given a fixed number of available robots in the team.

The approach relies on the possibility for the robots to change the values of the parameters characterizing the tasks

Limit	**100s**				**130s**				**160s**			
	AVG	**SD**	**MIN**	**MAX**	**AVG**	**SD**	**MIN**	**MAX**	**AVG**	**SD**	**MIN**	**MAX**
Allocated Tasks	5.02	0.43	4	6	6.27	0.55	5	7	8.06	0.51	7	9
Tasks for each robot	1.67	0.48	1	3	2.08	0.55	1	3	2.68	0.58	1	4
Remaining Battery	37%	9%	2%	64%	30%	7%	19%	50%	21%	6%	7%	39%
Time	83	15	70	93	99	27	85	109	116	38	106	126
Rounds	9	16	1	97	5	10	1	97	4	7	1	100

TABLE I. NEGOTIATION RESULTS WITH TCT_{req} =100S, 130S AND 160S.

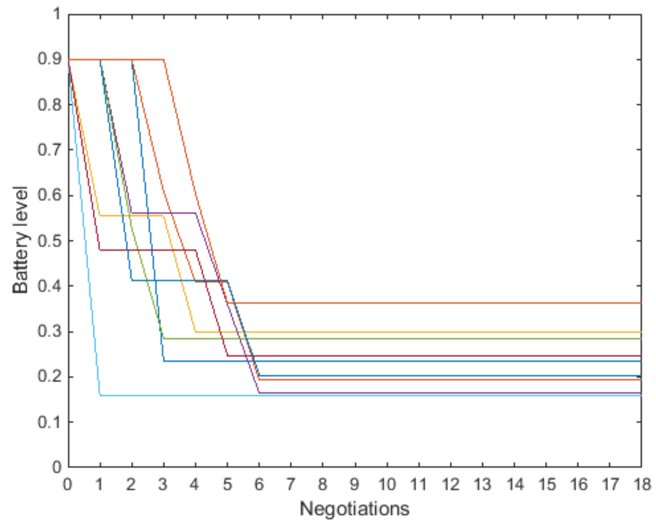

Fig. 2. Robot battery consumption trends for one set of negotiations.

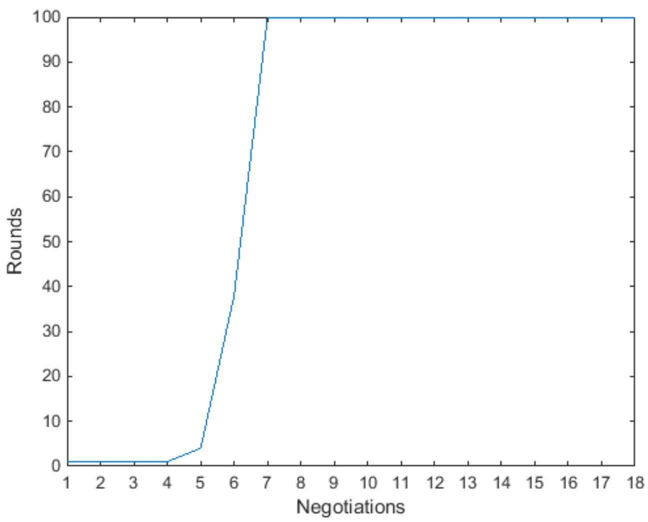

Fig. 3. Negotiation rounds for one set of negotiations.

they can execute dynamically, so modeling an innovative behavior of robots that can make decisions on how to perform a task.

Further investigation is necessary to optimize the battery level consumption of each involved robot that allows to still meet the constraints so leading to an increased number of complete allocations.

REFERENCES

[1] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, July 2006.

[2] R. Cui, J. Guo, and B. Gao, "Game theory-based negotiation for multiple robots task allocation," *Robotica*, vol. 31, pp. 923–934, 9 2013.

[3] P. Doherty, F. Heintz, and J. Kvarnström, "High-level mission specification and planning for collaborative unmanned aircraft systems using delegation," *Unmanned Systems*, vol. 1, no. 1, pp. 75–119, January 2013.

[4] L. M. Alejandro R. Mosteo, "A survey of multi-robot task allocation," Aragon Institute of Engineering Research, Tech. Rep., 2010.

[5] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing," *Robotics: Science and Systems*, 2005.

[6] P. Stone and M. M. Veloso, "Task decomposition and dynamic role assignment for real–time strategic teamwork," in *Agent Theories, Architectures, and Languages*, 1998, pp. 293–308.

[7] A. Stentz and M. B. Dias, "A free market architecture for coordinating multiple robots," Tech. Rep., 1999.

[8] F. Antonelli, G.; Arrichiello and S. Chiaverini, "The null–space-based behavioral control for mobile robots," in *IEEE Computational Intelligence in Robotics and Automation (CIRA'05)*, 2005, pp. 15–20.

[9] R. M. Zlot and A. T. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics*, vol. 25, no. 1, pp. 73–101, 2006.

[10] B. P. Gerkey, M. J. Matari, and M. robot Systems, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. of Robotics Research*, 2004.

[11] S. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1999, pp. 1234–1239 vol.2.

[12] S. Mokarizadeh, A. Grosso, M. Matskin, P. Kungas, and A. Haseeb, "Applying semantic web service composition for action planning in multi-robot systems," in *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on*, May 2009, pp. 370–376.

[13] Y.-G. Ha, J.-C. Sohn, and Y.-J. Cho, "Service-oriented integration of networked robots with ubiquitous sensors and devices using the semantic web services technology," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Aug 2005, pp. 3947–3952.

[14] C. Di Napoli, P. Pisa, and S. Rossi, "Towards a dynamic negotiation mechanism for qos-aware service markets," in *Trends in Practical Applications of Agents and Multiagent Systems*. Springer, 2013, vol. 221, pp. 9–16.

[15] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, may 2004.

[16] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation Decision Functions for Autonomous Agents," *Robotics and Autonomous Systems*, vol. 24, pp. 3–4, 1998.

[17] S. Rossi, D. Di Nocera, and C. Di Napoli, "Normal distributions and multi-issue negotiation for service composition," *Advances in Soft Computing*, vol. 293, pp. 1–8, 2014.