

SET: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets

Antidio Viguria
University of Seville
41092 Seville, Spain
and Georgia Institute of Technology
30332 Atlanta, USA
antidio@gatech.edu

Ivan Maza and Anibal Ollero
Robotics, Vision and Control Group
University of Seville
41092 Seville, Spain
{imaza,aollero}@cartuja.us.es

Abstract—The *multi-robot task allocation* (MRTA) problem has become a key research topic in the field of distributed multirobot coordination in recent years. In this paper, two algorithms for the distributed solution of the MRTA problem are presented. In our market-based approach, robots consider their local plans when bidding and multiple tasks can be allocated to a single robot during the negotiation process. The second algorithm described in the paper is based on the negotiation of subset of tasks and can be considered as a generalization of the first one, which only negotiates single tasks. Both algorithms have been tested in a multirobot simulator with multiple missions consisting in visiting waypoints with promising results.

I. INTRODUCTION

An important issue in distributed multirobot coordination is the *multi-robot task allocation* (MRTA) problem that has recently become a key research topic. It deals with the way to distribute tasks among the robots and requires to define some metrics to assess the relevance of assigning given tasks to such or such robot. This paper is focused on the distributed solution of the MRTA problem, but centralized ([3], [4]) and hybrid ([5], [14]) approaches have been also addressed in the literature. On the other hand, only independent loosely-coupled tasks will be considered, i.e., tasks that only need one robot to be executed (task allocation problems where tasks need tight coordination of more than one robot can be found in [9], [12] and [13]).

Several multirobot architectures considering the MRTA problem in a distributed manner have been validated on either physical or simulated robots. *ALLIANCE* [16], one of the earliest demonstrated approaches and *Broadcast of Local Eligibility* (BLE) [21], a distributed behavior-based architecture, are examples of systems based on behaviors with high fault tolerance and adaptability to noisy environments. On the other hand, in the last years a very popular approach to the MRTA problem considers the application of market-based negotiation rules. This negotiation is usually implemented by using some variant of the *Contract Net Protocol* (CNP) ([18], [19]).

One of the first distributed market-based systems was M+ [1], defined within a general architecture for the cooperation among multiple robots [2]. In this system, when a robot computes the cost of a task, it considers the next one in its local plan in order to increase the efficiency of the solution. In MURDOCH ([8], [10]), a robot which is executing a task, does not take part in the different negotiation processes. Therefore, the mechanism of task allocation is based on a purely greedy method that assigns each new task to the most suitable available robot in the system. TraderBots [6] considers dynamic environments [7] and total/partial failures of the robots and communication links. Unlike the previous mentioned works, robots have a local plan and more than one task can be allocated to each robot during the negotiation. In general, this approach leads to solutions closer to the global optimum. In this work, as our goal is to find solutions close to the optimum, robots consider their local plans when bidding and multiple tasks can be allocated to a single robot during the negotiation process. Therefore, the work in this paper can be posed as an instance of the ST-SR-TA case [11]: single-task robots, single-robot tasks and time-extended assignment.

The paper is organized as follows. The next section is devoted to describe a first algorithm called SIT-MASR which is based on the ideas presented in [6]. Then, a simple mission is presented to point out some limitations of this algorithm. In order to reduce those limitations, a new algorithm called SET-MASR which considers subsets of tasks in the negotiation process was developed. This algorithm is presented in Section III. Then, synchronization issues related with both algorithms are described in Section IV. Finally, their performance is compared in Section V with missions consisting in visiting waypoints. Conclusions and future work are discussed in Section VI.

II. DYNAMIC SINGLE TASK NEGOTIATIONS WITH MULTIPLE ALLOCATIONS TO A SINGLE ROBOT (SIT-MASR)

A. Description

As our goal was to find solutions close to the global optimum (minimize the sum of all the individual costs

assuming independent tasks), the approach presented in [5] was taken as a starting point as it has been mentioned above. In the same manner, robots with a local plan and multiple tasks allocated to a single robot during the negotiation were considered. Following those ideas, a first algorithm called SIT-MASR (Single Tasks negotiation with Multiple Allocations to a Single Robot) was developed. Regarding the implementation of this algorithm, several differences with the work in [5] can be pointed out: revenues are not used, a different synchronization method is applied (see Section IV) and there is an agent acting as an entry point for the tasks.

In the negotiation process, each robot bids for a task with the cost of inserting this task in its local plan (marginal cost). Therefore, this marginal cost associated with a task T_i for the robot j is the difference between the local plan including the new task and the current local plan:

$$MC_{ij} = C(P_j, T_i) - C(P_j), \quad (1)$$

where $C(P_j)$ is the total cost of the current local plan for robot j and $C(P_j, T_i)$ is the cost of a new local plan including task T_i . Each robot should compute the optimal insertion point of a new task in its current plan. Taking into account the local plan of each robot in the negotiation leads to better solutions, as it will be shown in Section V.

The SIT-MASR algorithm is based on two different roles played dynamically by the robots: auctioneer and bidders. In each auction there is only one auctioneer which is the robot that has the token (see Section IV). The auction is opened for a period of time and all the bids received during it are considered. When the auction is finished and the task is allocated, the auctioneer considers to pass the token to another robot. If so, the auctioneer changes its role to a bidder role and the robot with the token becomes the new auctioneer. These basic steps of the algorithms executed by the auctioneer and the bidders are given in Algorithms 1 and 2. In Algorithm 1, the best bid collected by the auctioneer is increased by a given percentage (usually 1%) to avoid transactions that will not significantly improve the solution.

Algorithm 1 SIT-MASR auctioneer algorithm

```

if there is any task to announce then
  announce task
  while timer is running do
    receive bids
  end while
  calculate best bid
  if best bid is bigger than the auctioneer bid then
    send task to best bidder
  end if
  delete task from announcement list
end if

```

Algorithm 2 SIT-MASR bidder algorithm

```

a new message is received
if new message is a task announcement then
  calculate the optimal position of the task in the local
  plan
  calculate bid (marginal cost)
  send bid to the auctioneer
else if new message is a task award then
  insert task in the local plan in the position calculated
  before
  introduce task in announcement list
end if

```

The main difference with the basic CNP protocol is that the bid of each robot depends on its current plan and every time the local plan changes, the negotiation continues until no bids improve the current global allocation. When the initial negotiation is over, the mission execution starts, but new tasks can be generated at any moment. Therefore, the negotiation is dynamic in the sense that new tasks are handled also during the mission execution. All the robots take part in the negotiation of those new tasks with the only restriction that the current tasks in execution are not re-negotiated.

The SIT-MASR algorithm has been tested in multi-robot missions consisting in visiting waypoints and returning to their home positions (see Section V. In this case, the local plan cost for a robot j visiting a set of waypoints P_i with $i = 1, 2, \dots, M$ can be expressed as:

$$C(P_j) = D(R_j, P_1) + \sum_{i=2}^M D(P_{i-1}, P_i) + D(P_M, H_j), \quad (2)$$

where R_j and H_j are the current and initial positions of the j -th robot and $D(X, Y)$ is the euclidean distance between points X and Y . In this particular missions, each robot should build its own local plan visiting the waypoints in an order that minimizes the total distance travelled. This problem is equivalent to the TSP problem which is a well known *NP-hard* problem. In our implementation, a greedy approach has been applied to solve it, inserting the new task in the position which minimizes its insertion cost. In Section V, several simulation results are compared with the global optimal solution.

III. DYNAMIC TASK SUBSETS NEGOTIATION WITH MULTIPLE ALLOCATIONS TO A SINGLE ROBOT (SET-MASR)

A. Motivation

Although the SIT-MASR algorithm leads to good solutions in general, simple missions can be found where it does not find the global optimum. For example, let us consider the mission in Figure 1, consisting in visiting waypoints **wp1** and **wp2**. If task **wp2** is announced before task **wp1**, the SIT-MASR algorithm will not find the global optimal solution (represented in Figure 1,a):

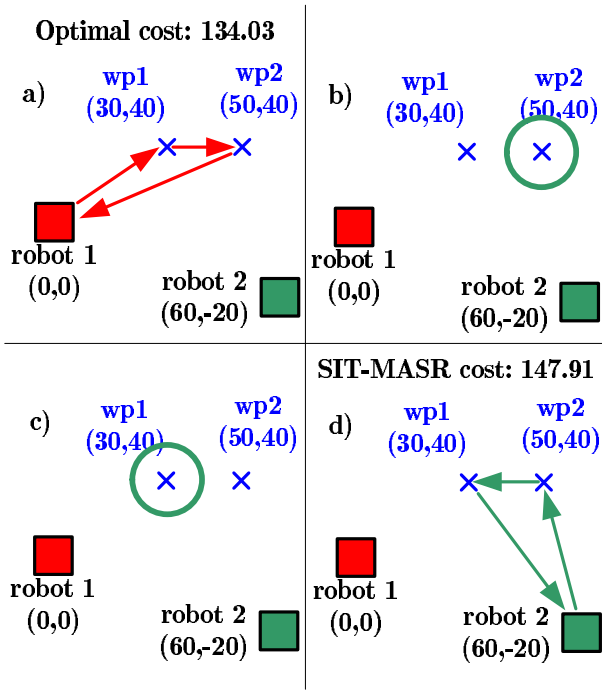


Fig. 1. A particular mission that shows some limitations of the SIT-MASR algorithm.

- 1) Task wp2 will be allocated to robot 2 (Figure 1,b) which is the nearest one.
- 2) As the marginal cost of task wp1 is lower for robot 2 than for robot 1, this task is also allocated to robot 2 (Figure 1,c).
- 3) Robot 2 announces both tasks, but it has the lower marginal costs for them and keep both tasks (Figure 1,d).
- 4) After a given timeout expires, robot 2 starts executing their tasks.

In this particular mission, the optimal solution would have been found if robot 2 had announced a subset of tasks composed by wp1 and wp2. This idea has been used to develop the algorithm described in the next subsection.

B. Description

In order to find better solutions for the mission explained above, the negotiation of subset of tasks was considered in the design of a new algorithm called SET-MASR (dynamic task subSETs negotiation with Multiple Allocations to a Single Robot). The basic idea behind this algorithm is that negotiating subsets of tasks provides more information to the robots than negotiating tasks one by one. It should be also noted that the SET-MASR can be considered as a generalization of the SIT-MASR algorithm, which tries to improve the quality of the solutions.

Given a subset of tasks $\Gamma_i = \{T_1, T_2, \dots, T_N\}$ with cardinality $|\Gamma_i| = N$, the marginal cost associated with this subset for the robot j is given by:

$$MC_{ij} = C(P_j, \Gamma_i) - C(P_j), \quad (3)$$

where $C(P_j)$ is the total cost of the current local plan for robot j and $C(P_j, \Gamma_i)$ is the cost of a new local plan including the subset of tasks Γ_i . In our implementation, a greedy approach has been applied to find where to insert the tasks of the subset in the current local plan in order to minimize the insertion cost.

On the other hand, a policy for building the subsets of tasks to be auctioned during the negotiation process is required. As a brute force algorithm to try all the possible combinations is not feasible, in our algorithm each robot computes the subset of tasks with the highest cost in its local plan. The computational cost to find this subset is not significant for the number of tasks usually managed by a single robot (less than 50).

As in the *SIT-MASR* algorithm, there are two roles: auctioneer and bidders. The basic steps of the algorithm for the auctioneer is given in Algorithm 3 whereas Algorithm 4 is used by the bidders. When the cardinality of the subset of tasks to be announced is one, the algorithm behaves exactly as the *SIT-MASR*. Once all the tasks have been allocated and there are no changes in the local plans of the robots during a given period, the subset cardinality is increased by some robot. This robot will start the next phase of auctions with subsets of two tasks. Finally, the algorithm will stop when there is no interchange of tasks during a given phase or when the subset cardinality is greater than the number of tasks to be announced in every robot. Once the algorithm has finished, the robots will start executing their local plans.

Algorithm 3 SET-MASR auctioneer algorithm

```

if subset cardinality is  $\leq$  dim of announcement list
then
  calculate the subset of tasks to be announced
  announce the subset of tasks
  while timer is running do
    receive bids
  end while
  calculate the best bid
  if best bid is bigger than the auctioneer bid then
    send subset of tasks to the winner of the auction
  end if
  delete subset of tasks from announcement list
end if

```

IV. SYNCHRONIZATION DURING THE NEGOTIATION

The synchronization during the negotiation process has a relevant impact on the solutions. For example, in [6] it is shown that having only one auction process running at a given time leads to better solutions. When the auctions run in parallel, robots can be bidding with invalid marginal costs if tasks in their local plans are not

Algorithm 4 SET-MASR bidder algorithm

```

new message received
if new message is task announcement then
  set the cardinality of the subset equal to the number
  of tasks announced;
  calculate the optimal position of the subset of tasks
  in the local plan
  calculate bid (marginal cost)
  send bid
else if new message is a task award then
  insert subset of tasks in the local plan in the posi-
  tions calculated before
  introduce the subset of tasks in the announcement
  list
end if

```

finally allocated to them. In our approach, a single token with a modified round robin algorithm has been used to guarantee only one auction running at a given time.

The token is created by the first robot which starts an auction. Each robot with tasks to announce, requests the token periodically. Once the current auction is over, its owner passes the token to the robot with more tasks to be announced. In order to implement properly this basic idea, communication channel characteristics, such as delays and errors in the messages, should be considered. For example, the robots assume that the last robot which has announced a task is the owner of the token. But due to communication delays, the owner can change after some requests for the token have been sent to the old owner. To solve this problem, this old owner answers with a rejection message containing the identification of the new owner.

On the other hand, robot failures are also considered. For example, if the token is requested and no answer is received in a given period of time, the token is assumed lost (due to a robot communication system failure for example) and a new token is generated.

V. SIMULATION RESULTS

A multi-robot simulator has been programmed in C++ [20] using a three layer architecture [15] for each robot. An important objective in the design of the simulator was to have C++ code directly reusable in the real robots. Therefore, each simulated robot runs the same software which is on board the real robot and uses a layer that emulates the hardware and its interfaces. It should be also pointed out that the software allows simulating heterogeneous robots (i.e. ground and aerial robots) in a common virtual space.

On the other hand, inter-processes communication has been implemented by using the BBCS (BlackBoard Communication System) recently developed by the Technical University of Berlin [17]. It is a robust communication system implemented via a distributed shared memory, the blackboard (BB), in which each network node has a

local copy of the BB portion it is accessing. This communication system allows to run a multi-robot simulation in a single or multiple machines, and is used also to communicate the real robots. However, communication range constraints have not been considered in the simulations.

Multirobot missions consisting in visiting waypoints and returning to home positions have been used to test the algorithms in the simulator. Hundreds of simulations with different number of robots and tasks have been performed to compare the algorithms presented in the previous sections. Moreover, a third algorithm has been implemented in order to evaluate the relevance of the local plans in the quality of the solutions. This algorithm will be called *NoP* (No Local Plan) and uses a basic CNP protocol where robots only participates in the auction when they are idle. Furthermore, a brute force algorithm has been used to compute the global optimal solutions when the sum of robots and tasks is below a certain value.

In particular, for each given number of robots and waypoints, one hundred missions have been run in a virtual world of 1000x1000 meters using random positions for the robots, with a size of 1x1 meter, and the waypoints. Each mission has been simulated with the three algorithms implemented and Table I shows the different solutions compared with the global optimum. In each cell the first number is the arithmetic mean of the global cost for the 100 random missions, the value between brackets is its standard deviation (in meters) and the third number is the difference in percentage with the optimal solution. The global cost is given by the sum of the individual costs of the robots.

From the results, it should be noted that using a local plan during the auction process improves the solutions significantly. On the other hand, the algorithms presented in this paper achieve very good results, with a maximum difference of 4.7% w.r.t. the optimal solution. Moreover, the SET-MASR algorithm computes better solutions in mean than the SIT-MASR, but the difference is small. In fact, it has been found that the SET-MASR algorithm improvement is very sensitive to the particular initial positions of the robots and waypoints. Using the mean global cost of one hundred random missions “smooths” this improvement and the different is not so significant. Also, it is important to point out that the standard deviation values are high because of the random nature of the missions (i.e., the global cost for those random missions can have quite different values).

Tables II, III and IV show the results from testing the algorithms with more waypoints for three, five and seven robots. In those cases, it was not possible to compute the optimal solution with our brute force algorithm due to the NP-hard nature of the problem. Therefore, in these tables the percentage value corresponds to the difference with the solutions found with the SET-MASR algorithm. The rest of values have the same meaning and units explained for Table I.

With three and five robots the results are quite similar:

Robots	Tasks	NoP	SIT	SET	Optimum
3	3	2371, 22 (742, 4) 65, 18%	1453, 44 (369, 63) 1, 25%	1444, 9 (364, 23) 0, 66%	1435, 4 (362, 36)
3	5	4144, 7 (923, 42) 101, 23%	2097, 83 (414, 52) 1, 74%	2088, 1 (405, 38) 1, 27%	2061, 8 (396, 3)
3	7	5073, 2 (788, 75) 114, 73%	2473, 65 (385, 46) 4, 7%	2457, 5 (381, 27) 4, 01%	2362, 6 (335, 81)
3	9	6070, 8 (850, 46) 129, 13%	2816, 59 (398, 16) 6, 3%	2773, 3 (392, 7) 4, 6%	2649, 5 (332, 58)
5	3	1764, 36 (627, 87) 39, 49%	1274, 88 (365, 74) 0, 79%	1268, 21 (359, 53) 0, 27%	1264, 78 (356, 04)
5	5	3808, 55 (921, 45) 112, 37%	1842, 96 (363, 62) 2, 76%	1816, 77 (348, 94) 1, 3%	1793, 35 (337, 56)
5	7	5407, 59 (1238, 38) 150, 15%	2225, 67 (384, 74) 2, 96%	2209, 71 (381, 82) 2, 22%	2161, 68 (365, 82)

TABLE I

SOLUTIONS COMPUTED WITH THREE DIFFERENT DISTRIBUTED TASK ALLOCATION ALGORITHMS AND THE OPTIMAL RESULT. THE FIRST NUMBER IS THE ARITHMETIC MEAN OF THE GLOBAL COST, THE VALUE BETWEEN BRACKETS IS ITS STANDARD DEVIATION (BOTH VALUES IN METERS) AND THE THIRD NUMBER IS THE DIFFERENCE IN % W.R.T THE OPTIMAL SOLUTION.

Tasks	NoP	SIT	SET
9	6070, 8m. (850, 46m.) 118,9%	2816, 59m. (398, 16m.) 1,5%	2773, 3m. (392, 7m.)
15	9146, 9m. (1104, 5m.) 152, 90%	3655, 61m. (401, 85m.) 1, 07%	3616, 78m. (372, 64m.)
20	12324, 99m. (1085, 64m.) 198, 9%	4157, 12m. (379, 07m.) 0, 83%	4122, 79m. (364, 75m.)
30	16780, 74m. (1490, 19m.) 236, 97%	5035, 43m. (414, 09m.) 1, 11%	4979, 8m. (476, 56m.)
40	22045, 37m. (1909, 91m.) 294, 9%	5634, 98m. (332, 9m.) 0, 94%	5582, 42m. (353, 1m.)

TABLE II

SOLUTIONS COMPUTED FOR MISSIONS WITH THREE ROBOTS AND DIFFERENT NUMBER OF WAYPOINTS.

a significant difference between the NoP algorithm and the others (at least 118,9%) and very similar results for the SIT-MASR and the SET-MASR algorithms, being the largest difference of 3,01%. But with seven robots the solutions of the NoP algorithm are better as expected: with few robots, a single robot has a higher probability to execute a task with high cost if the others are not idle.

Finally, Figure 2 compares the mean of messages transmitted by each robot using the different algorithms in one hundred missions with five robots. As expected, the number of messages increases with the number of tasks. The SET-MASR algorithm needs more messages than others due to its more complex negotiation protocol,

Tasks	NoP	SIT	SET
9	6558, 51m. (1282, 5m.) 142, 31%	2744, 81m. (382, 15m.) 1, 41%	2706, 59m. (378, 55m.)
15	9779, 13m. (11434, 31m.) 182, 64%	3488, 06m. (381, 93m.) 0, 81%	3459, 86m. (383, 12m.)
20	12277, 43m. (1389, 83m.) 205, 70%	4058, 52m. (398, 18m.) 1, 05%	4016, 01m. (389, 47m.)
30	17312, 18m. (1561, 04m.) 253, 74%	4969, 31m. (361, 43m.) 1, 53%	4894, 02m. (391, 54m.)
40	22353, 78m. (1721, 68m.) 306, 06%	5727, 55m. (342, 5m.) 3, 01%	5559, 75m. (756, 3m.)

TABLE III

SOLUTIONS COMPUTED FOR MISSIONS WITH FIVE ROBOTS AND DIFFERENT NUMBER OF WAYPOINTS.

Tasks	NoP	SIT	SET
9	3214, 16m. (653, 28m.) 30, 37%	2472, 31m. (424, 77m.) 0, 28%	2465, 29m. (422, 53m.)
15	4788, 04m. (716, 12m.) 43, 46%	3360, 78m. (414, 97m.) 0, 7%	3337, 36m. (400, 78m.)
20	6093, 22m. (806, 31m.) 54, 08%	3975, 39m. (381, 4m.) 0, 52%	3954, 57m. (364, 07m.)
30	8592, 23m. (959, 65m.) 79, 1%	4818, 46m. (508, 73m.) 0, 50%	4794, 23m. (430, 94m.)
40	10932, 07m. (1152, 79m.) 95, 37%	5619, 95m. (521, 65m.) 0, 44%	5598, 34m. (470, 45m.)

TABLE IV

SOLUTIONS COMPUTED FOR MISSIONS WITH SEVEN ROBOTS AND DIFFERENT NUMBER OF WAYPOINTS.

but the number of messages also scales linearly with the number of tasks. On the other hand, the best ratio between the improvement of the solutions and the number of messages required is achieved with the SIT-MASR algorithm. Finally, the NoP algorithm can be used if the communication among robots should be minimized.

VI. CONCLUSIONS AND FUTURE WORK

The *multi-robot task allocation* (MRTA) problem has recently become a key research topic in the field of distributed multirobot coordination. In this paper, two algorithms for the distributed solution of the MRTA problem have been presented. In both algorithms, robots have a local plan and multiple tasks can be allocated to a single robot during the negotiation. Moreover the second algorithm, based on the negotiation of subsets of tasks, can be considered as a generalization of the first one (which only negotiates single tasks) designed to improve the solutions. Some implementation details related to the problems derived from the asynchronous nature of the negotiation process have been also pointed out.

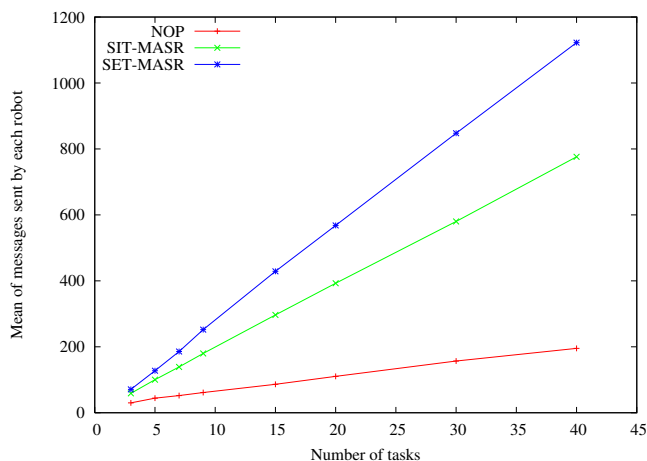


Fig. 2. Mean of the messages sent per robot in one hundred missions with five robots and different number of waypoints.

From the simulation results, it is derived that using a local plan during the auction process improves the solutions significantly. Furthermore, the algorithms presented in the paper lead to good results when comparing with the global optimal solutions in missions consisting in visiting waypoints. Moreover, the SET-MASR algorithm computes better solutions in mean than the SIT-MASR, being this improvement sensitive to the particular initial positions of the robots and waypoints. On the other hand, both algorithms performance scale well when the number of robots and tasks increases.

Future work includes evaluating the impact of partial or total communication and robot failures in the performance of both algorithms. Furthermore, the autonomous generation of new tasks from the distributed perception of the environment is also being implemented. Finally, it would be interesting to have some kind of formalism to represent the distributed algorithms in such a way that researchers could understand other approaches easier. Moreover, a common set of missions should be also available with the solutions computed by different algorithms to allow an easier comparison. In this sense, the missions used in Section V have been uploaded in <http://grvc.us.es/mrta> with the solutions computed by the algorithms presented in this paper.

VII. ACKNOWLEDGMENTS

This work was partially funded by the European Union AWARE Project (IST-2006-33579) and the AEROSENS Project of the Spanish Research and Development Program (DPI2005-02293).

REFERENCES

[1] S. C. Botelho and R. Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, 1999.

[2] S.C. Botelho and R. Alami. Multi-robot cooperation through the common use of “mechanisms”. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 375–380, Maui, USA, 2001.

[3] B Brumitt and A. Stenz. GRAMMPS: A generalized mission planner for multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998.

[4] P. Caloud, W. Choi, J. Latombe, C. Le Pape, and M. Yim. Indoor automation with many mobile robots. In *Proceedings of the IEEE International Workshop on Intelligent Robotics and Systems (IROS)*, 1990.

[5] M. B. Dias and A. Stenz. Opportunistic optimization for market-based multirobot control. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2714–2720, Lausanne, Switzerland, 2002.

[6] M.B. Dias. *TraderBots: A New Paradigm for Robust and Efficient MultiRobot Coordination in Dynamic Environments*. PhD thesis, Carnegie Mellon University, 2004.

[7] M.B. Dias, Marc B. Zinck, R. M. Zlot, and A. Stenz. Robust multirobot coordination in dynamic environments. April 2004.

[8] B.P. Gerkey and M.J. Mataric. Murdoch: Publish/subscribe task allocation for heterogeneous agents. In *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 203–204, Barcelona, Spain, 2000.

[9] B.P. Gerkey and M.J. Mataric. Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, 2002.

[10] B.P. Gerkey and M.J. Mataric. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[11] B.P. Gerkey and M.J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[12] E. G. Jones, B. Browning, M.B. Dias, B. Argall, M. Veloso, and A. Stenz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, 2006.

[13] Nidhi Kalra, Anthony (Tony) Stenz, and David Ferguson. Hoplitest: A market framework for complex tight coordination in multi-agent teams. Technical Report CMU-RI-TR-04-41, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2004.

[14] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical decision-theoretic approach to multi-robot mapping and exploration. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2714–2720, 2003.

[15] I. Maza, A. Viguria, and A. Ollero. Networked aerial-ground robot system with distributed task allocation for disaster management. In *IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.

[16] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[17] V. Remuss, M. Musial, and U.W. Brandenburg. BBCS robust communication system for distributed system. In *Proceedings of the International Workshop on Safety, Security, and Rescue Robotics*, Bonn, Germany, 2004.

[18] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, 1993.

[19] G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12), 1980.

[20] A. Viguria and I. Maza. Realsim: Simulador multirobot generico. Technical Report US-GRVC-06-09, University of Seville, Seville, Spain, July 2006.

[21] B. B. Werger and M. J. Mataric. Broadcast of local eligibility for multi- target observation. In *Distributed Autonomous Robotic Systems 4*, pages 347 – 356. Springer-Verlag, 2000.