Controlled Reconfiguration of Robotic Mobile Sensor Networks using Distributed Allocation Formalisms

Ayanna Howard and Antidio Viguria
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
ayanna.howard@ece.gatech.edu and antidio@gatech.edu

***Abstract*— In this paper, we discuss the realization of a robotic mobile sensor network that allows for controlled reconfiguration of sensor assets in a decentralized manner. The motivation is to allow the construction of a new system of science observations that requires higher spatial and temporal resolution models that are needed for understanding environmental changes. To enable controlled reconfiguration of these sensor assets, we discuss four formal algorithms that address the deployment challenges in a distributed way. We discuss these algorithms in detail and present results of their applications to a science-driven coverage task.**

I. INTRODUCTION

Mobile sensor networks have been shown to be a powerful tool for enabling a number of activities that require recording of spatial and temporal variations in environmental parameters required for such activities as monitoring of seismic activity, monitoring of civil and engineering infrastructures, and detection of toxic agents throughout a region of interest [15]. In most sensor network applications, individual sensor agents collect information about their neighboring agents using peer-to-peer communication. Unfortunately, as the size of the network increases, bandwidth limitations and the absence of feasible communication channels severely limits the possibility of conveying and using global information. As such, the utilization of decentralized techniques for forming new sensor topologies and configurations is a highly desired quality of mobile sensor networks. Establishment of these sensor configurations involves determining how to allocate sensor positions to mobile sensor agents in order to achieve a desired topology - a similar research objective that is found when focusing on the task allocation problem with teams of robots.

In the last few years, different approaches have been used to solve the task allocation problem: centralized [3], [4], hybrid [5], [10] and distributed. The distributed approach, considered ideal for large teams of robots or mobile sensors, has characteristics that fit most applications: high fault tolerance, fast response for dynamic changes in the environment and low computational complexity. Basically, two main approaches have been studied in depth in order to solve the problem for independent loosely-coupled tasks: behavior-based [14], [20] and market-based [2], [5], [6], [17]. While the first approach presents high fault tolerance and adaptability to noisy environments, the second approach obtains efficient solutions close to the optimum.

Most of the research in this area has typically focused on solving the general task allocation problem where more than one task can be allocated to a single robot. However, there are other types of problems that cannot be solved with these algorithms, for example, the Initial Formation Problem [1], [13]. This problem can be stated as follows:

Given a number of tasks, $\{T_1, T_2, ...T_N\}$, a team of robots $\{R_1, R_2, ..., R_M\}$, a function $C(T_i, R_j)$ that specifies the cost of executing task T_i by robot R_j and considering that the number of tasks must be less or equal than the number of robots, i.e., $N \leq M$. Find the assignment that allocates one task per robot and minimizes the global cost defined as $\sum_{j=1}^{M} C(T_i, R_j)$, where i is the task assigned to robot j.

This type of problem becomes really important within the field of formation control [8], [12] where using local information and control laws, the distributed algorithm is able to drive a given formation error to zero. However, as it is stated in [9], these algorithms require a first step that assigns the robots to the formation positions while taking into account their initial positions, i.e., answer the question who goes where? Usually this problem has been solved using centralized solutions such as the Hungarian method [11], since the Initial Formation Problem can be viewed as a classical job assignment problem where robots are the workers and tasks are the jobs to be executed by those workers. However, this type of solution requires that all the robots have to communicate between each other and has all the disadvantages related to centralized systems: low fault tolerant, computational complexity and slow response for dynamic changes in the environment. Furthermore, it is not possible to take advantage of all the good characteristics related to distributed algorithms if part of your problem has to be solved in a centralized way.

For that reason, it is important to come up with an algorithm that solves the Initial Formation Problem in a distributed way. Our interest is to obtain not only a feasible solution, but also an efficient one. Due to that fact, a market-based approach has been chosen which uses negotiations in order to allocate the different tasks. This negotiation is typically implemented by using some variant of the *Contract Net Protocol* [16], [18], where two roles are played dynamically by robots: auctioneer and bidders. The auctioneer is the robot in charge of announcing the tasks and selecting the best bid from all the bids received from the bidders. The best bid is considered the one with the lowest cost.

In order to use the distributed algorithm to solve the Initial Formation Problem, we must reformulate it as a task allocation problem where the tasks are waypoint tasks that coincide with the positions of the formation. For that reason, the cost used in the bids is a quantity that reflects how much it will cost the robot to go to a certain waypoint, such as the euclidean distance or the traversability index [7]. Also, it is important to point out that one robot can only be allocated one task, since the final objective is to assign one position of the next formation to each robot.

The paper is organized as follows. In the next section, a basic market-based algorithm that solves the Initial Formation Problem will be explained. This algorithm obtains good results when the initial position of the robots and the formation positions are calculated at random. However, in specific types of scenarios the results obtained have large errors. In Section III, different modifications of the basic market-based algorithm that improve its results will be addressed. These algorithms try to improve the basic algorithm in two aspects:

- Use a better cost function.
- Select the task to be bid on in a more clever way.

The first algorithm called RMA improves the results using more intelligent logic when selecting a task, while the next algorithm, TMA, uses a new cost function. The last algorithm, RTMA, combines both features. In Section IV, simulation results will be presented and discussed showing the advantages and disadvantages of each algorithm. Finally, conclusions and future work are provided in Section V.

II. BS: Basic Market-Based Approach

A market-based algorithm has been used to solved the Initial Formation Problem. As usual in algorithms based on the Contract Net Protocol, two roles are played dynamically by robots: auctioneer and bidders. The auctioneer is the agent in charge of announcing the tasks and selecting the best bid from all the received bids. In our case the best bid is the one with the lowest cost and the cost is equal to the distance from the robot to the task. The complete algorithm is explained in Algorithm 1. On the other hand, the bidder role is explained in Algorithm 2. The basic idea is that each robot must have only one task, so it will keep the task with the lowest cost. If it wins a new task that has a lower cost than the one already won, it would sell the old task to the robot with the best bid but worse than its own bid. The best bid worse than the robot's bid is selected in order to avoid infinite loops in the negotiation. This scenario could happen when two robots have the best bids for at least three tasks as shown in Figure 1.

For formation initialization, a slight modification is implemented for the market-based structure. At the beginning, tasks are introduced by a human operator, such as a scientist, using a monitoring center or a planner that generates tasks from an abstract mission. Therefore, in our system there are

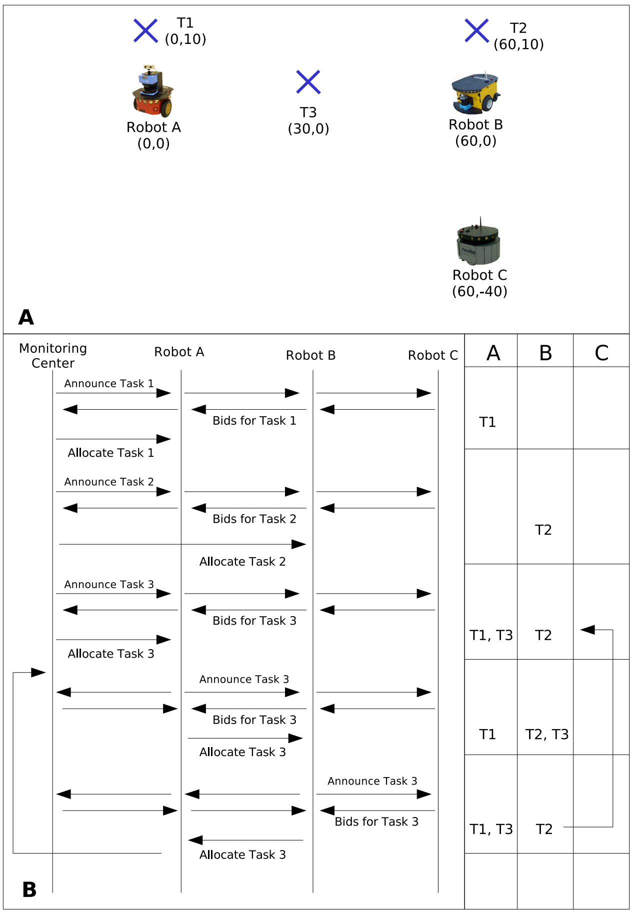

Fig. 1. Figure A presents the initial position of the robots and the tasks. Figure B presents the messages exchanged among the different agents and shows how an infinite loop appears in the negotiation protocol.

two types of agents: robots and monitoring center, and two types of roles: auctioneer and bidders. Both types of agents can play both roles. However, the monitoring center plays the auctioneer role at the beginning and after all the tasks are introduced, it switches to the bidder role with a constant bid equal to infinite for all tasks in order to assure that it will never win a task after the auction starts. It is important to point out that the monitoring center need not be unique, i.e, the same algorithm works with distributed insertion of tasks. Also, tasks can be generated dynamically by robots, and therefore, there is no firm requirement for existence of a monitoring center. It is rather just an implementation detail.

Algorithm 1 Auctioneer algorithm

if announcement-task list is not empty **then**
    announce task
    while timer is running **do**
        receive bids
    end while
    calculate best bid worse than the robot's bid
    send task to best bidder
    delete task from announcement-task list
end if

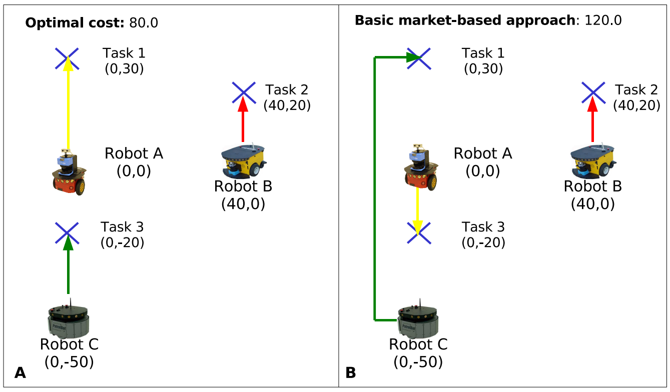

Fig. 2. Difference in cost between the optimal allocation and the one obtained with the basic market-based algorithm.

Algorithm 2 Bidder algorithm

a new message is received
if new message is a task announcement **then**
  calculate bid (distance to the task)
  send bid to the auctioneer
else if new message is a task award **then**
  if the robot has already won a task **then**
    if cost of the new task $<$ cost of the won one **then**
      introduce old task in announcement-task list and delete it from won-tasks list
      introduce won task in the won-tasks list
    else
      introduce won task in the announcement-task list
    end if
  else
    introduce won task in the won-tasks list
  end if
end if

From the results shown in Section IV, it can be stated that this algorithm obtains not bad results when the initial position of the robots and the tasks are calculated totally at random. However, there are situations when this algorithm does not obtain good results which usually happens when a robot has to take a task that is the worst one for its own interest, as can be seen in Figure 2. In this example, the global cost obtained with the market-based algorithm is 66.67% greater than the optimal allocation.

III. IMPROVED ALGORITHMS

In order to solve the initial formation problem, the task allocation algorithm has to solve two main problems:

- How do I calculate the bid for a certain task?
- If I won more than one task, how do I determine which one to keep?

In the basic market-based approach, bids are the distance between the robot position and the tasks (we are only considering waypoint tasks) and if one robot wins more than one task, it keeps the one that is closest to itself, i.e., the one with the lowest cost or best bid. Therefore, if our objective is to improve the basic market-based algorithm, one or both of these aspects must change. Moreover, the improved algorithm must keep the advantages of the market-based approach: fault tolerance, independent from the number of robots and high adaptation to changes in the environment using reallocations.

A. RMA: Robot Mean Allocation algorithm

Our first improved algorithm is focused on trying to choose in a more clever way the task that must be kept when a robot wins more than one task. This is accomplished using additional knowledge available to the system. Instead of keeping the task with the smallest distance to the robot, the task with highest difference between the distance to the robot and the mean of its distance to all the robots will be selected. In other words, suppose that there are a finite number of tasks T and robots N and robot R_k has won tasks T_i and T_j. In this case, robot R_k will keep task T_i if and only if:

$$\sum_{l=1}^{N} \frac{D(R_l, T_i)}{N} - D(R_k, T_i) > \sum_{l=1}^{N} \frac{D(R_l, T_j)}{N} - D(R_k, T_j) \tag{1}$$

where $D(R_a, T_b)$ is the distance between robot R_a and task T_b.

The question that arises now is how to calculate the mean of the distances for a certain task. During the normal operation of the algorithm, the auctioneer receives bids from all functioning robots in order to allocate the task to the best robot. At this moment, the auctioneer knows all the distances between every robot and the current task. Thus, the mean is calculated by the auctioneer and transmitted to the robot within the message that informs the robot that has won the task. The major difference with the basic market-based algorithm is that the robot must remember the mean associated with the won task. Furthermore, the robot is able to compare their means to different tasks because it remembers the mean of the task already won and the mean of the new allocated task is sent by the auctioneer as it was done previously.

This new algorithm obtains better results than the basic market-based approach as can be seen in Section IV and, in comparison with the resources used for the basic algorithm, it only requires that the winner of a task remember the mean associated with it.

B. TMA: Task Mean Allocation algorithm

In this algorithm instead of changing the way that the tasks are selected, the cost function will be changed. In the original algorithm the cost function used to calculate the bid for a certain task is the distance between the robot and the task. However, in this improved algorithm the cost function will be the difference between the distance of the robot and the task minus the mean of the distances between that robot and all the tasks, i.e.:

$$C(R_i, T_j) = D(R_i, T_j) - \sum_{k=1}^{N} \frac{D(R_i, T_k)}{N} \tag{2}$$

where $C(R_i, T_j)$ is the cost function for robot R_i and task T_j and $D(R_a, T_b)$ is the distance between robot R_a and task T_b.

The rest of the algorithm works as the basic market-based approach but using the new cost function instead of the distance. Therefore, when one robot wins two tasks instead of comparing the distances to choose the closest one, it will compare the costs using the new cost function and it will select the task with the lowest cost for itself. Thus, a robot R_k that has won two tasks, T_i and T_j will keep T_i if and only if:

$$C(R_k, T_i) < C(R_k, T_j), \tag{3}$$

or

$$D(R_k, T_i) - \sum_{l=1}^{N} \frac{D(R_k, T_l)}{N} < D(R_k, T_j) - \sum_{l=1}^{N} \frac{D(R_k, T_l)}{N} \tag{4}$$

As it can be seen in Equation 4, the sum factor is equal in both parts of the inequality. So, the algorithm will select a task by using either the distance or the new cost function.

The only drawback to this algorithm is that robots must know the different tasks at the beginning in order to calculate the mean of the distances. This has been implemented using a new kind of message that is sent at the beginning by the monitoring center to all the robots. The only purpose of this message is to transmit the different task to the robots, so they can calculate the mean of the distances necessary for the new cost function. Afterwards, the extra resources needed for the algorithm are almost the same as that for the basic market-based approach since robots only have to memorize the mean calculated at the beginning and implement one basic cost function operation.

C. RTMA: Robot and Task Mean Allocation algorithm

The last algorithm is a mix between the RMA and the TMA algorithms. Therefore, the cost function will be the one used in the TMA algorithm, while the logic used to select tasks is the one used in the RMA. As it was shown for the TMA, it has the same results if the distance or the new cost function is used in the logic that selects tasks. If the distances are used, task T_i will be the one selected if and only if:

$$\sum_{l=1}^{N} \frac{D(R_l, T_i)}{N} - D(R_k, T_i) > \sum_{l=1}^{N} \frac{D(R_l, T_j)}{N} - D(R_k, T_j) \tag{5}$$

On the other hand if the new cost function is used, task T_i will be the one selected if and only if:

$$\sum_{l=1}^{N} \frac{C(R_l, T_i)}{N} - C(R_k, T_i) > \sum_{l=1}^{N} \frac{C(R_l, T_j)}{N} - C(R_k, T_j) \tag{6}$$

where $C(R_a, T_b) = D(R_a, T_b) - \sum_{t=1}^{N} \dfrac{D(R_a, T_t)}{N}$.

Thus,

$$\begin{aligned}
&\sum_{l=1}^{N} \frac{D(R_l, T_i)}{N} - \sum_{l=1}^{N}\sum_{t=1}^{N} \frac{D(R_l, T_t)}{N^2} \\
&\quad - D(R_k, T_i) - \sum_{t=1}^{N} \frac{D(R_k, T_t)}{N} \\
&> \sum_{l=1}^{N} \frac{D(R_l, T_j)}{N} - \sum_{l=1}^{N}\sum_{t=1}^{N} \frac{D(R_l, T_t)}{N^2} \\
&\quad - D(R_k, T_j) - \sum_{t=1}^{N} \frac{D(R_k, T_t)}{N}.
\end{aligned} \tag{7}$$

Simplifying the last equation, it is obtained:

$$\sum_{l=1}^{N} \frac{D(R_l, T_i)}{N} - D(R_k, T_i) > \sum_{l=1}^{N} \frac{D(R_l, T_j)}{N} - D(R_k, T_j) \tag{8}$$

As can be seen, Equations 5 and 8 are exactly the same. However, due to practical implementation it is easier to compare the tasks using the new cost function, because the bids are calculated with it.

IV. SIMULATIONS AND DISCUSSION

A multi-robot simulator has been used to test decentralized algorithms. This simulator is based on an architecture designed for heterogeneous robots [19] and divided into three layers. The highest layer is independent from the type of robot and is the one aware of the existence of other robots. Thus, the task allocation algorithm is implemented in this layer. Moreover, the communication among robots is based on IP, so it can also be used as an interprocess communication method for simulations. The other two layers are used to execute the different tasks allocated to the robot and make easier the creation of new algorithms by using to a modular and component-based architecture.

The different algorithms have been tested using initial positions of the robots and formations calculated at random in a virtual world of 1000x1000 meters. The simulations have been accomplished using a variety of scenarios in which the number of robots and tasks ranged from 2 up to 20, and for every case one hundred simulations were run. These results are shown in Figure 3 where the mean of the global cost and the error in percentage in comparison with the optimal solution are presented. The optimal solution has been calculated using the Hungarian method [11]. It can be observed that the best algorithm is the RTMA and the worst one is the basic market-based (BS) algorithm, although all the algorithms obtains efficient results up to 8 robots and tasks where the largest error is less than 10%. For more than 8 robots only the RTMA algorithm obtained good results, with a maximum error of 5.98% in the case of 20 robots. As can be seen in Figure 3, the error with the optimal solution is bounded by a linear function for all the algorithms with the number of robots and tasks. However, the RTMA algorithm is the one with lowest slope. Furthermore, it is interesting to

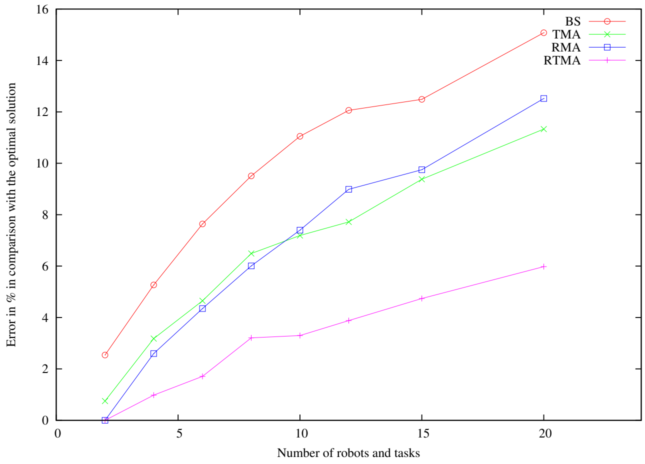

Fig. 3. Error in percentage in comparison with the optimal solution for the different types of algorithms and calculating the inital positions of the robots and the points of the formations at random over 100 simulations.

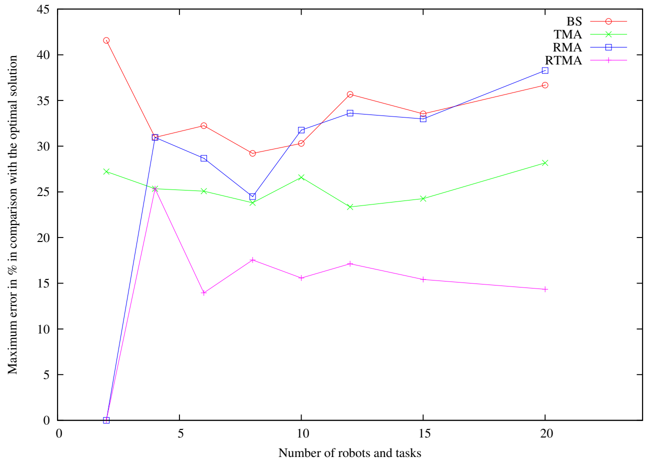

Fig. 4. Maximum errors in percentage in comparison with the optimal solution in 100 simulations for the different types of algorithms and calculating the inital positions of the robots and the points of the formations at random.

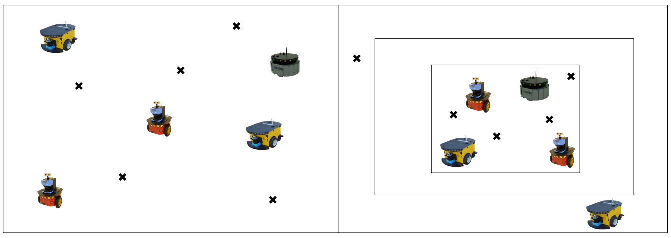

Fig. 5. Types of formations used in the simulations. Left: initial positions of the robots and the formations calculated at random. Right: most of the formation points and the initial positions of the robots calculated at random in the small box and the others calculated outside the big box and calculated also at random.

comment that with less than 10 robots the RMA algorithm obtains results slightly better than the TMA algorithm, but over 10 robots it is the TMA algorithm which obtains better results than the RMA. It is also important to point out that for 2 robots and tasks the RMA and RTMA algorithms always obtain the optimal solution.

The results of Figure 3 only show statistically how good the algorithm is, based on the mean. However, it could be the case that an algorithm could have good results on average but there are some situations where its results have large errors. Therefore, another important parameter to consider is the maximum error with the optimal solution over all the simulations. In Figure 4, the maximal errors in percentage is shown. First of all, it can be observed that the RMA algorithm obtains worse maximal errors than the TMA algorithm and, in some cases, even worse than the BS algorithm, but the mean of the global cost is lower for the RMA algorithm as can be seen in Figure 3. Therefore, the RMA algorithm has a better behavior on average but in certain circumstances the results can be worse than the TMA and BS algorithms. On the other hand, the BS algorithm is still the worst one for most of the cases, while the RTMA algorithm presents the best results. As can be seen in Figure 4, the mean of the maximum errors considering all the cases is 14.91% for the RTMA algorithm and 33.77% for the BS algorithm which is greater than the mean error commented in Figure 3. That means these algorithms do not have a constant behavior and for a specific situation results could be worse than the average.

All the results presented have been calculated using random position of the robots and random points of the formations uniformly distributed. However, the quality of the solution for some of the algorithms depends on the type of formations. In Figure 5, there are two types of formations: the one on the left is calculated totally at random and is the one used so far, the other formation on the right has a structure formed by two boxes. Most of the points and robots of the formation are in the small box, and the others outside the big box. As can be seen in Figure 6, the BS algorithm obtains worse results than the ones obtained with the other type of formation, specially for low number of robots and tasks. Another important characteristic of this type of formations is that the error in percentage in comparison with the optimal solution remains more or less constant for different number of robots and tasks. Therefore, for this type of formations the behavior of the algorithms for a specific situation are more predictable than with the totally random formations. Finally, the RTMA algorithm obtains also the best results while the BS algorithm the worst ones and unlike the first type of formations, the TMA algorithm always obtains worse results than the RMA algorithm for all the cases simulated.

V. CONCLUSIONS AND FUTURE DEVELOPMENTS

The Initial Formation Problem has been stated and four different algorithms that solve this problem in a distributed way have been explained. The first one is based on the basic market-approach, it is the simplest algorithm but obtains the worst results in most of the cases. Also, it is the algorithm that is most affected by the structure of the formation due to the fact that its results get worse more with the second

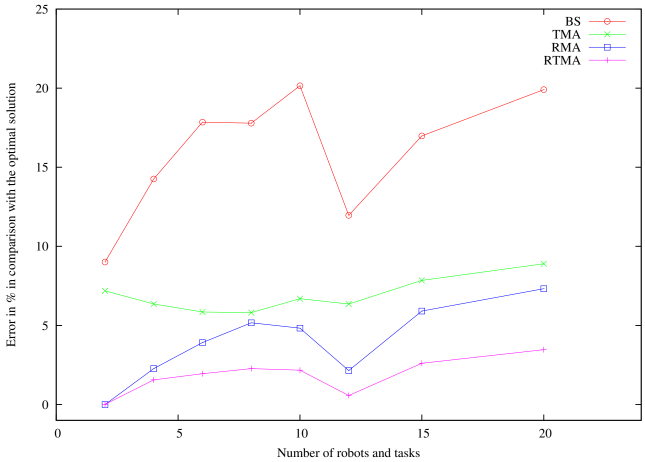

Fig. 6. Error in percentage in comparison with the optimal solution for the different types of algorithms and calculating the inital positions of the robots and the points of the formations as it is described in the right part of the Figure 5 over 100 simulations.

type of formations. The second and third algorithms use the mean of the costs (considering all the tasks associated to a robot or all the robots for a specific task) in order to increase the information about the whole system and improve the results, but always keeping the distributed computation of the algorithm. These two algorithms obtain similar results for all the cases and better than the ones obtained with the first algorithm. Finally, the RTMA, which is a combination of the RMA and TMA algorithms, obtains the best results in all the cases for both types of formations since combines the good characteristics of the RMA and TMA algorithms.

Two different types of formations have been used. In the first one, the error in comparison with the optimal solution increases in a linear way with the number of robots and tasks. In the sencod one, the error keeps slightly constant. Therefore, the behavior of the algorithms is more predictable for the second type of formations.

Future work includes the consideration of the sensor information in the task allocation algorithms. Therefore, the local environment information will be considered in the bidding process using cost functions that express in a more realistic way the effort needed to achieve a task. Also, more realistic situations will be considered that include aspects such as limited communication radius and failures in the robots. Finally, it is also planned to implement these algorithms in real robots and test their robustness and performance in different terrain environments.

VI. Acknowledgments

This work supports research in Reconfigurable Sensor Networks for the National Aeronautics and Space Administration, Earth Science Technology Office.

References

[1] A. Barrett. Autonomy architectures for a constellation of spacecraft. In *International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS)*, Noordwijk, The Netherlands, June 1999.

[2] 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.

[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, Switzwerland, 2002.

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

[7] A. Howard, H. Seraji, and B. Werger. Global and regional path planners for integrated planning and navigation. *Journal of Robotic Systems*, 22(12):767–778, December 2005.

[8] X. Hu and M. Egerstedt. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, December 2001.

[9] M. Ji and M. Egerstedt. Role-assignment in multi-agent coordination. *International Journal of Assistive Robotics and Mechatronics*, 7(1):32–40, 2006.

[10] 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.

[11] H.W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, pages 83–97, 1955.

[12] J. Lawton, R. Beard, and B. Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, December 2003.

[13] F. McQuade, R. Ward, and C.R. McInnes. The autonomous configuration of satellite formations using a generic potential function. In *5th International Symposium on Spaceflight Dynamics*, France, June 2000.

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

[15] NSF Report. National workshop on future sensing systems - living, nonliving, and energy systems. Technical report, Lake Tahoe, NV, August 2002.

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

[17] S. Sariel and T. Balch. A distributed multi-robot cooperation framework for real time task achievement. In *Proceedings of the 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2006.

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

[19] A. Viguria, I. Maza, and A. Ollero. Set: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007.

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