Market-based multi-robot coalition formation

Lovekesh Vig and Julie A. Adams

Vanderbilt University, Nashville TN 37212
{lovekesh.vig, julie.a.adams}@vanderbilt.edu

Summary. Task allocation is an issue that every multi-robot system must address. Recent task allocation solutions propose an auction based approach wherein robots bid for tasks based on cost functions for performing a task. This paper presents RACHNA, a novel architecture for multi-robot task allocation based on a modified algorithm for the winner determination problem in multi-unit combinatorial auctions. A more generic utility based framework is proposed to accommodate different types of tasks and task environments. Preliminary experiments yield promising results demonstrating the system's superiority over simple task allocation techniques.

1 Introduction and Motivation

Task allocation is a challenging problem due to the unpredictable nature of robot environments, sensor failure, robot failure, and dynamically changing task requirements. While market-based task allocation systems have traditionally found favor with the software-agent research community ([1], [2] and [3]), market-based control architectures are proving to be an effective distributed mechanism for multi-robot task allocation as well. Stentz and Dias [4] utilized a market-based scheme to coordinate multiple robots for cooperative task completion that introduced the application of market mechanisms to intra-team robot coordination. The common feature in market-based allocation mechanisms is an auction protocol to coordinate tasks between different robots [5], [6], [7] or between different components of the same robot [8], [9]. When an auction is announced, robots compute bids based on their expected profit for the tasks and the robots with the lowest cost bid are awarded contracts.

A number of elegant non market-based solutions to the task allocation problem have been proposed. The ALLIANCE [10] architecture uses motivational behaviors to monitor task progress and dynamically reallocate tasks. Recently Low et al. [11] proposed a swarm based approach for the cooperative observation of multiple moving targets (CMOMMT). Dahl et al. [12] present a task allocation scheme based on "Vacancy Chains," a social structure modeled on the creation and filling of vacancies in an organization. The Broadcast of Local Eligibility system (BLE) [13] system uses a Publish/Subscribe method to allocate tasks that are hierarchically distributed.

The common underlying factor in the above systems is the single robot-single task (SR-ST) assumption which entails that tasks are indivisible and may be performed by a single robot. As multi-robot tasks become more complex, this assumption is proving to be an oversimplification. Many task domains contain multiple tasks requiring a team of robots to work on them simultaneously, thus further complicating task allocation.

A relatively unexplored problem is the allocation of multi-robot teams to different tasks (the ST-MR problem), commonly known as the *Multi-Robot Coalition Formation (MRCF)* problem. Many coalition formation techniques within Distributed Artificial Intelligence (DAI) have been proposed for this provably hard problem [14], [15], [16]. Multi-robot coalition formation adds further complexity due to additional real world constraints [17] (fault tolerance, sensor location, communication costs, etc.). Recently a variety of market based solutions to the ST-MR task allocation problem have been proposed, [18], [19]. This paper proposes RACHNA[1], a novel market-based solution to the MRCF problem that leverages the inherent redundancy in sensor/actuator capabilities of robots to enable a more tractable, utility-based formulation of the MRCF problem.

This paper is organized as follows; Section 2 details the RACHNA architecture, the negotiation protocol, and the task environments. Section 3 provides the experimental details and results. Section 4 provides conclusions and outlines potential avenues for future work.

2 The RACHNA system

A common feature of the market based systems discussed in Section 1 is that they require the robots to bid on the tasks. The bidding process is central to determining the auction outcome. Therefore when dealing with complex tasks, the bidder should have a global view of the available resources. The RACHNA system reverses the bidding process. The auction is performed by the tasks for the individual robot services, thus allowing the bidding to be performed with a semi-global view of the resources necessary for coalition formation.

One of the most prominent differences between multi-agent and multi-robot domains is the level of redundancy in multi-robot and software-agent capabilities. Robots are manufactured on a large scale and are more likely to have greater redundancy in their sensor/actuator capabilities. RACHNA leverages this redundancy to enable a more tractable formulation of the MRCF

[1] Robot Allocation through Coalitions using Heterogeneous Non-Cooperative Agents

problem. RACHNA achieves this through the formulation of the MRCF as a multi-unit combinatorial auction. While single item auctions allow the bidders to bid on only one item, combinatorial auctions permit bidding on combinations of items.

Definition: The auctioneer has a set of items, M = *1, 2,..., m* to sell. The auctioneer has some number of each item available: $U = \{u_1, u_2, ..., u_m\}, u_i \epsilon Z+$. The buyers submit a set of bids, $B = \{B_1, B_2, ..., B_n\}$. A bid is a tuple $Bj = <(\gamma_j^1, ..., _j^m), p_j>$, where $\gamma_j^k \geq 0$ is the number of units of item k that the bid requests, and p_j is the price. The *Binary Multi-Unit Combinatorial Auction Winner Determination Problem (BMUCAWDP)* is to label the bids as winning or losing so as to maximize the auctioneers revenue under the constraint that each unit of an item can be allocated to at most one bidder:

$$max \ \ \sum p_j x_j \ s.t. \sum_{j=1}^{n} \gamma_j^i x_j \leq u_i, i = 1, 2, \ldots, m \tag{1}$$

The MRCF problem can be cast as a combinatorial auction with the bidders represented by the tasks, the items as the different types of robots, and the price as the utility that each task has to offer. Unfortunately, the BMUCAWDP problem is inapproximable [20] however some empirically strong algorithms exist [21], [20].

2.1 The Architecture

Two types of software agents are involved in the task allocation process:

1. **Service Agents** are the mediator agents through which the tasks must bid for a service. RACHNA requires that each robot have a set of services or roles it can perform. The roles are determined by the individual sensor and behavioral capabilities resident on each robot. One service agent exists for each service type that a robot can provide. A service agent may communicate with any robot that provides the particular service to which the agent corresponds. Service agents reside on any robot capable of providing the service. Thus, the global task information is acquired in a decentralized manner via service agents.
2. **Task agents** place offers on behalf of the tasks so as to acquire the necessary services. The task agents only communicate with the service agents during negotiations. Once the task is allocated, the task agent may communicate directly with the robots allocated to the task. Task agents may reside on a workstation or a robot and communicate with the necessary service agents.

An economy is proposed where the tasks are represented by task-agents that are bidding for the services of the individual robots. The economy has a

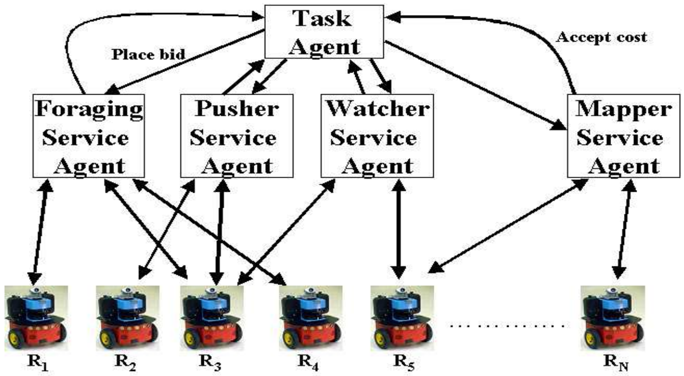


Fig. 1. An example RACHNA Implementation

set of robots $R_1, R_2, ..., R_N$ where each robot is equipped with sensor capabilities that enable it to perform various services such as pushing, watching, foraging, etc. The tasks are assumed to be decomposable into the sub-task behaviors. For example, a box-pushing task may require two pusher sub-task roles and one watcher sub-task role as shown in Figure 1. Each role is represented by a service agent that is responsible for negotiating with the robots that have the desired capability. The bids are relatively sparse compared to the overall space of coalitions and will yield a more tractable formulation of the MRCF. Unlike other heuristic based algorithms coalition formation [15], no restriction is placed on the coalition size.

2.2 The Allocation Environments

Three task types were permitted in the presented experiments:

1. **Urgent** tasks can pre-empt an ongoing standard task and generally have a higher average reward per robot. These tasks are emergency tasks that require immediate attention, such as fire extinguishing or rescue tasks.
2. **Standard** tasks are allocated only when sufficient free resources exist and when the task utility is sufficient to merit allocation. These tasks may be pre-empted by urgent tasks. Loosely coupled tasks (i.e. foraging) or tasks that may easily be resumed comprise this category.
3. **Non preemptable** tasks are allocated similar to standard tasks but cannot be pre-empted. Tightly coupled tasks fall into this category because, preemption would completely debilitate task performance.

Two different types of allocation are considered:

1. *Instantaneous Allocation*: A number of tasks are introduced into the environment and the algorithm allocates resources to the optimal set of tasks.

2. *Pre-emptive Allocation*: Involves introduction of a single urgent task that requires immediate attention. The urgent task offers higher rewards in an attempt to obtain bids from the robots.

Instantaneous Assignment

Instantaneous assignment requires multiple auctions while the system's objective is to allocate resources to tasks while maximizing overall utility. Services correspond to items, robots correspond to units of a particular item (service), and task offers correspond to bids. This work distributes this solution in order to leverage the inherent redundancy in robot capabilities, thereby obtaining a more tractable formulation of the MRCF problem.

The auction begins with each task agent sending request messages to the individual service agents. The service agents attempt to obtain the minimum possible price for the requested services. The robot's current minimum salaries are evaluated and a minimum increment is added in order to lure the robots to the new task. The service agents then forward this information to the task agents. The task agents determine if sufficient utility exists to purchase the required services. If this is the case, then the services are temporarily awarded the task. This offer-counteroffer process proceeds in a round robin fashion with the robots' salaries increasing at every step until there is a round where no service (robot) changes hands. At this point, a final stable solution is attained.

Random Assignment

Urgent tasks are randomly introduced and are allocated robot services according to a negotiation process between tasks. The negotiation begins when the new task submits a request to the required service agents for a certain number of services of that type. The service agents take into account the current robots' salaries and a bargaining process ensues with tasks increasing robot salaries until either the new task successfully purchases the resources or waits for additional free resources.

2.3 Utility vs. Cost

A difficulty with the employed market based approach is that the resulting teams are highly dependent on the initial utilities assigned to various tasks. However, this may not be an entirely undesirable property. While the system is sensitive to initial utilities, it also empowers the user to prioritize tasks by varying the task utilities. Most definitions of utility incorporate some notion of balance between quality and cost ([22], [23]). Cost quantification is relatively straightforward, however quantifying quality task execution prior to coalition formation for a new task can be difficult. Independent of the utility measure employed, what matters is that a mapping exists between coalition task pairs to scalar values permitting comparisons between coalitions for performing a task.

2.4 Multiple decompositions

Many scenarios involve more than one potential decomposition for a particular complex task and many possible decompositions may be considered when evaluating the potential coalitions. It may be possible to permit multiple decompositions via a task decomposition system [23] and introducing 'dummy' items to incorporate these, as described in [21].

3 Experiments

Preliminary experiments were conducted by simulating the RACHNA system on a single computer. The experiments recorded the variation in robot salaries and overall utility with bid numbers. A set of real world tasks were simulated in the Player/Stage environment to demonstrate task preemption.

3.1 Wage increase

The first set of experiments simulated a set of 68 robots and ten services such that each service had exactly ten possible robots capable of providing that particular service. 100 tasks were generated with each task requiring a random vector of resources. The variation in the average salary for each service type was recorded as the bids increased. Fig 2 shows the average, maximum, and minimum salary curves for all services. The results depict how the increasing competition (more tasks) increased the salaries as robots received better offers when demand increases. Initially the salaries are low (Number of tasks $\leq$ 20), the salaries rise at different rates depending on demand for a particular service ($20 \leq$ Number of tasks ≤ 40), and eventually if the demand for each service increases sufficiently, the salaries for all service agents approach high values (Number of tasks $= 100$). The robots in RACHNA that are capable of performing services that are in high demand have a high likelihood of participating in the final allocation.

3.2 Effect of diversity

RACHNA leverages the redundant sensory capabilities in a set of robots in order to group robots and make the allocation problem more tractable. RACHNA does make any assumptions about the diversity of the resulting teams, only the diversity of the entire collection of robots. Fig 3 shows RACHNA's performance deteriorates as the number of services is increased and the number of robots remains constant. The higher the number of services, the lower the redundancy, and hence the higher the execution time of the algorithm. If there was only one service agent, all robots would be identical and task allocation is the least expensive. However, if each robot was different, task allocation would be more expensive. Thus the execution time increases with the increased diversity of the set of robots.

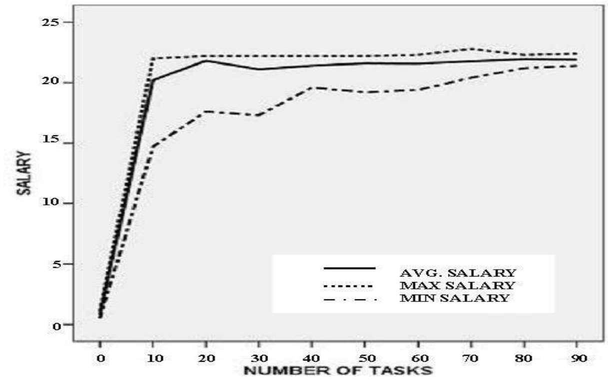


Fig. 2. Average salary across all robots vs. Tasks (bids).

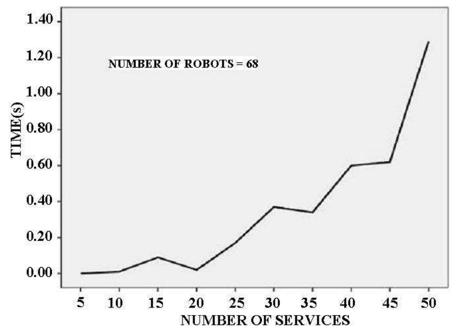


Fig. 3. Execution time vs. Number of Services.

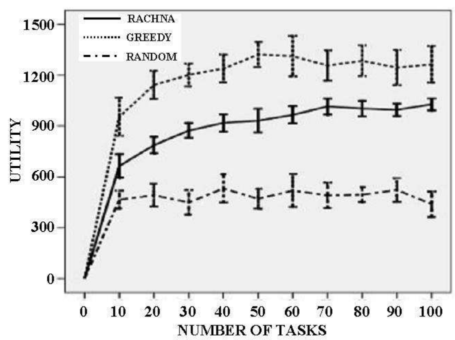


Fig. 4. Comparison of global greedy, random a allocations to RACHNA.

3.3 Utility Comparision

RACHNA's solution quality was compared to that obtained by two simple task allocation schemes. Fig 4 provides comparison for the average solution quality between solutions produced by RACHNA to those produced by the global greedy (best task first) and random allocation algorithms as the number of tasks varied. Each data point represents the mean performance from ten trials. A random task was generated for each trial and each algorithm's performance was recorded. RACHNA outperforms both the greedy and random allocation algorithms (as shown in Fig 4) because unlike greedy or random search, RACHNA refines the solution in each auction round to include better tasks (bids) and remove less profitable tasks.

3.4 Preemption Simulations

The preemption experiments involve a set of five services, ten robots, (see Table 1) and four heterogeneous tasks as described in Table 2. The first three tasks are introduced using the procedure described in Section 2. Fig 5(a) shows the initial task allocation. There are sufficient resources to satisfy all three tasks and all robots, except for Robot 7, are allocated with a minimum wage of 5 units. Introduction of Task 4 initiates a bargaining process where Task 4 attempts to acquire the idle Robot 7 for the minimum wage (5) and acquire Robots 6 and 8 from Task 3 by offering a higher salary. Since, Task 3 cannot match Task 4's best offer, Task 3 relinquishes Robots 6 and 8. At the end of this bargaining process the demand increase for robots of type 6 and 8 results in their salaries being increased to 10 and 20 respectively and Task 3 is preempted.

Table 1. Services

Services	Capabilities					Robots
	LRF	Camera	Bumper	Gripper	Sonar	
Foraging	0	1	0	1	1	R_1, R_2
Pushing	1	0	1	0	0	R_3, R_4, R_6, R_7
Object Tracking	0	1	0	0	1	R_1, R_2, R_5, R_8
Sentry-Duty	1	0	0	0	0	$R_3, R_4, R_6, R_7, R_9, R_{10}$

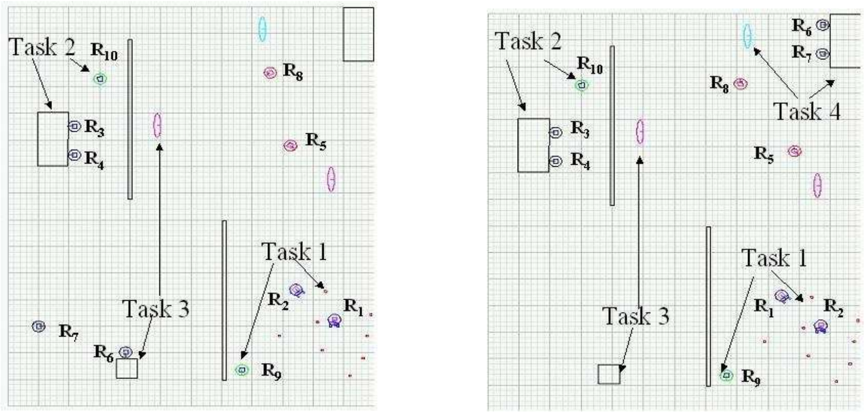
Table 2. Tasks

Tasks	Services				Priority	Utility
	Foraging	Pushing	Object-Tracking	Sentry-Duty		
1	2	0	0	1	Standard	70
2	0	2	0	1	Non-premptible	40
3	0	1	2	0	Standard	45
4	0	2	1	0	Urgent	50

The results reported in this section demonstrate the potential applicability of the system to different types of tasks and environments. RACHNA also allows for a more generic, task-independent system. It is important to note the favorable comparison of the suggested allocation to simple techniques like global greedy and random allocation. The fact that the algorithm leverages sensor redundancy makes the coalition formation tractable and the experiments demonstrate the improved performance.

4 Conclusions and future work

This paper presents RACHNA, a market-based distributed task allocation scheme based on the multi-unit combinatorial auction problem. RACHNA



(a) The initial allocation of tasks 1,2 and 3.

(b) Allocation of task 4 and pre-emption of task 3.

Fig. 5. The pre-emption of the standard task 3 by the urgent task 4

reverses the auction scheme found in other market-based coordination schemes by allowing tasks to bid on robot services rather than the other way around. RACHNA is a utility based system, allowing the user to specify the task priority. Finally the system produces higher quality solutions than simple greedy or random task allocation strategies and enables a more tractable formulation of the coalition formation problem by leveraging the redundancy in robot sensor capabilities. Future work involves real robot experiments to demonstrate allocation and dynamic task preemption.

References

1. Sandholm, T.: An implementation of the contract net protocol based on marginal cost calculations. In: Proceedings of the Eleventh National Conference on Artificial Intelligence. (1993) 256–262
2. Collins, J., Jamison, S., Mobasher, B., Gini, M.: A market architecture for multi-agent contracting. Technical Report 97-15, University of Minnesota, Dept. of Computer Science (1997)
3. Sycara, K., Zeng, D.: Coordination of multiple intelligent software agents. International Journal of Intelligent and Cooperative Information Systems **5** (1996) 181–211
4. Stentz, A., Dias, M.B.: A free market architecture for coordinating multiple robots. Technical Report CMU-RI-TR-01-26, Robotics Institute, Carnegie Mellon University (1999)
5. Gerkey, B.P., Matarić, M.J.: Murdoch: Publish/subscribe task allocation for heterogeneous agents. In: Proceedings of Autonomous Agents. (2000) 203 – 204
6. Botelho, S.C., Alami, R.: M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In: Proceedings of IEEE International Conference on Robotics and Automation. (1999) 1234 – 1238

7. Dias, M.B.: TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments. PhD thesis, Robotics Institute, Carnegie Mellon University (2004)
8. Laengle, T., Lueth, T.C., Rembold, U., Woern, H.: A distributed control architecture for autonomous mobile robots. Advanced Robotics **12** (1998) 411–431
9. Caloud, P., Choi, W., Latombe, J.C., Pape, C.L., Yim, M.: Indoor automation with many mobile robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. (1990) 67–72
10. Parker, L.E.: ALLIANCE: An architecture for fault tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation **14** (1998) 220–240
11. Low, K.H., Leow, W.K., M. H. Ang, J.: Task allocation via self-organizing swarm coalitions in distributed mobile sensor network. In: Proceedings of the American Association of Artificial Intelligence. (2004) 28–33
12. Dahl, T.S., Matarić, M.J., Sukhatme, G.S.: Multi-robot task-allocation through vacancy chains. In: Proceedings of IEEE International Conference on Robotics and Automation. (2003) 14–19
13. Werger, B., Matarić, M.J.: Broadcast of local eligibility: Behavior based control for strongly cooperative multi-robot teams. In: Proceedings of Autonomous Agents. (2000) 21–22
14. Sandholm, T.W., Larson, K., Andersson, M., Shehory, O., Tomhe, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence **111** (1999) 209–238
15. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence Journal **101** (1998) 165–200
16. Abdallah, S., Lesser, V.: Organization-Based Cooperative Coalition Formation. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Techonology. (2004) 162–168
17. Vig, L., Adams, J.A.: Issues in multi-robot coalition formation. In: Proceedings of Multi-Robot Systems. From Swarms to Intelligent Automata. Volume 3. (2005) 15–26
18. Zlot, R., Stentz, A.: Complex task allocation for multiple robots. In: Proceedings of the IEEE Conference on Robotics and Automation. (2005) 67–72
19. Lin, L., Zheng, Z.: Combinatorial bids based multi-robot task allocation. In: Proceedings of the International Conference on Robotics and Automation. (2005) 1145–1150
20. Sandholm, T.: Algorithm for optimal winner determination algorithm. Artificial Intelligence **135** (1998) 1–54
21. Leyton-Brown, K., Y. Shoham, M.T.: An algorithm for multi-unit combinatorial auctions. In: Proceedings of the 17th National Conference on Artificial Intelligence. (2000) 56–61
22. Gerkey, B., Matarić, M.J.: A framework for studying multi-robot task allocation. In: Proceedings of the Multi-Robot Systems Workshop: From Swarms to Intelligent Automata. Volume 2. (2003) 15–26
23. Tang, F., Parker, L.E.: ASyMTRe: Automated synthesis of multi-robot task solutions through software reconfiguration. In: Proceedings of the IEEE International Conference on Robotics and Automation. (2005) 1501–1508