In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*
pages 3862-3868, Taipei, Taiwan, September, 2003. (Also Technical Report CRES-02-005)

Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures

Brian P. Gerkey Maja J Matarić
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA
{bgerkey|mataric}@cs.usc.edu

***Abstract*— Important theoretical aspects of multi-robot coordination mechanisms have, to date, been largely ignored. To address part of this negligence, we focus on the problem of multi-robot task allocation. We give a formal, domain-independent, statement of the problem and show it to be an instance of another, well-studied, optimization problem. In this light, we analyze several recently proposed approaches to multi-robot task allocation, describing their fundamental characteristics in such a way that they can be objectively studied, compared, and evaluated.**

I. INTRODUCTION

Since the early 1990s, the problem of task allocation in multi-robot systems has received significant and increasing interest in the research community. As researchers design, build, and use cooperative multi-robot systems, they invariably encounter the question: "which robot should execute which task?" This question must be answered, even for relatively simple multi-robot systems, and the importance of task allocation grows with the complexity, in size and capability, of the system under study. Even in the simplest case of homogeneous robots with fixed, identical roles, intelligent allocation of tasks is required for good system performance, if only to minimize physical interference.

Of course, task allocation need not be explicit; it may instead emerge from the interactions of the robots (physical and otherwise), as is the case with coordination methods put forward by proponents of *swarm robotics* [9]. Such an emergent system, when constructed skillfully, can be extremely effective and may provide the simplest and most elegant solution to a problem. However, it is a solution to a *specific* problem, and if robots are to be generally useful, we believe that they must be capable of solving a variety of problems.

Over the years, a significant body of work has been done on *explicit* multi-robot task allocation (MRTA), generally involving task-oriented inter-robot communication. A variety of such architectures have been proposed; while they are sometimes evaluated experimentally, they are rarely subject to formal analysis. It is coordination architectures of this type on which we focus our analysis in this paper. Our aim is to address two key shortcomings of the MRTA work to date[1]:

- computation and communication requirements are generally unknown
- aside from experimental validation in specific domains, there is no characterization of the *solution quality* that can be expected

The rest of this paper is organized as follows. In the next section we give a formal statement of the general problem of multi-robot task allocation, by way of reduction to an instance of a well-known optimization problem from Operations Research. In Section III, we analyze some significant MRTA architectures that have been proposed to date by considering them to be algorithms for (approximately) solving the underlying optimization problem. By doing so we will gain a deeper understanding of how these approaches function and what cost/benefit tradeoffs they introduce. We conclude in Section IV with a summary and a brief discussion of extensions and future directions for our work.

II. PROBLEM STATEMENT

We claim that multi-robot task allocation can be reduced to an instance of the *Optimal Assignment Problem* (OAP) [11], a well-known problem from Operations Research. A recurring special case of particular interest in several fields of study, this problem can be formulated in many ways. Given our application domain, it is fitting to describe the problem in terms of jobs and workers. There are n workers, each looking for one job, and m available jobs, each requiring one worker. The jobs can be of different priorities, meaning that it is more important to fill some jobs than others. Each worker has a nonnegative skill rating estimating his/her performance for each potential job (if a worker is incapable of undertaking a job, then the worker is assigned a rating of zero for that job). The problem is to assign workers to jobs in order to maximize the overall expected performance, taking into account the priorities of the jobs and the skill ratings of the workers.

[1] In a recent important paper [20], Pynadath & Tambe perform a similar critique and analysis of strategies for multi-agent teamwork.

Our multi-robot task allocation problem can be posed as an assignment problem in the following way: given n robots, m prioritized (i.e., weighted) single-robot tasks, and estimates of how well each robot can be expected to perform each task, assign robots to tasks so as maximize overall expected performance. However, because the problem of task allocation is a dynamic decision problem that varies in time with phenomena including environmental changes, we cannot be content with this static assignment problem. Thus we complete our reduction by iteratively solving the static assignment problem over time.

Of course, the cost of running the assignment algorithm must be taken into account. At one extreme, a costless algorithm can be executed arbitrarily fast, ensuring an optimal assignment over time. At the other extreme, an expensive algorithm that can only be executed once will produce a static assignment that is only initially optimal and will degrade over time. Finally there is the question of how many tasks are considered for (re)assignment at each iteration. In order to create and maintain an optimal allocation, the assignment algorithm must consider (and potentially reassign) every task in the system. Such an inclusive approach can be computationally expensive and, indeed, some implemented approaches to MRTA use heuristics to determine a subset of tasks that will be considered in a particular iteration.

Together, the cost of the static algorithm, the frequency with which it is executed, and the manner in which tasks are considered for (re)assignment will determine the overall computational and communication overhead of the system, as well as the solution quality. Thus it is these characteristics of MRTA architectures that we examine in Section III. Before continuing with a formal statement of our problem, we undertake a necessary aside regarding *utility*.

A. Utility

Utility is a unifying, if sometimes implicit, concept in economics [10], game theory [21], and operations research [2], as well as multi-robot coordination (see Section III). The idea is that each individual can somehow internally estimate the value (or the cost) of executing an action. It is variously called fitness, valuation, and cost. Since the exact formulation varies from system to system, we now give an instructive and generic, yet practical, definition of utility for multi-robot systems.

We assume that each robot is capable of estimating its fitness for every task of which it is capable. This estimation includes two factors, both task- and robot-dependent:

- expected quality of task execution, given the method and equipment to be used (e.g., the accuracy of the map that will be produced using a laser range-finder)
- expected resource cost, given the spatio-temporal requirements of the task (e.g., the power that will be required to drive the motors and laser range-finder in order to map the building)

Given a robot R and a task T, if R is capable of executing T, then we can define, on some standardized scale, Q_{RT} and C_{RT} as the quality and cost, respectively, expected to result from the execution of T by R. We can now define a combined, nonnegative utility measure[2]:

$$U_{RT} = \begin{cases} Q_{RT} - C_{RT} & \text{if } R \text{ is capable of executing } T \text{ and } \\ & Q_{RT} > C_{RT} \\ 0 & \text{otherwise} \end{cases}$$

The robots' utility estimates will be inexact for a number of reasons, including sensor noise, general uncertainty, and environmental change. These unavoidable characteristics of the multi-robot domain will necessarily limit the efficiency with which coordination can be achieved. We treat this limit as exogenous, on the assumption that lower-level robot control has already been made as reliable, robust, and precise as possible and thus that we are incapable of improving it. When we later discuss "optimal" allocation solutions; we mean "optimal" in the sense that, given the union of all information available in the system (with the concomitant noise, uncertainty, and inaccuracy), it is impossible to construct a solution with higher overall utility.

Note that although we have not discussed planning or learning, our definition of utility permits their introduction. The addition of a predictive model, for example, will (presumably) improve the accuracy of utility estimates by considering expected future events. To achieve the best system performance, such techniques should be used whenever possible.

B. Formalism

We are now ready to state our MRTA problem as an instance of the OAP. Formally, we are given:

- the set of n robots, denoted $I_1, \ldots, I_n$
- the set of m prioritized tasks, denoted $J_1, \ldots, J_m$ and their relative weights[3] $w_1, \ldots, w_m$
- U_{ij}, the nonnegative utility of robot I_i for task J_j, $1 \leq i \leq n, 1 \leq j \leq m$

We assume:

- Each robot I_i is capable of executing at most one task at any given time.
- Each task J_j requires exactly one robot to execute it.

These assumptions, though somewhat restrictive, are necessary in order to reduce MRTA to the classical OAP, which is given in terms of single-worker jobs and single-job workers. We are currently working on a more general

[2] We thank Michael Wellman for suggesting this formulation.

[3] If the tasks are of equal priority (as is often the case), then the weights are all equal to 1.

formulation that will allow us to relax these assumptions. Regardless, in most existing MRTA work (including the architectures that we study in Section III), these same assumptions are made, though often implicitly.

The problem is to find an optimal allocation of robots to tasks. An allocation is a set of robot-task pairs:

$$(i_1, j_1) \dots (i_k, j_k),\ 1 <= k <= \min(m,n)$$

Given our assumptions, for an allocation to be *feasible* the robots $i_1 \dots i_k$ and the tasks $j_1 \dots j_k$ must be unique. The benefit (i.e., expected performance) of an allocation is the weighted utility sum:

$$U = \sum_{m=1}^{k} U_{i_m j_m} w_{j_m}$$

We can now cast our problem as an integral linear program [11]: find n^2 nonnegative integers α_{ij} that maximize

$$\sum_{i,j} \alpha_{ij} U_{ij} w_j \tag{1}$$

subject to

$$\begin{aligned} \sum_i \alpha_{ij} &= 1, \quad \forall j \\ \sum_j \alpha_{ij} &= 1, \quad \forall i \end{aligned} \tag{2}$$

The sum (1) is just the overall system utility, while (2) enforces the constraint that we are working with single-robot tasks and single-task robots (note that since α_{ij} are integers they must all be either 0 or 1). Given an optimal solution to this problem (i.e., a set of integers α_{ij} that maximizes (1) subject to (2)), we construct an optimal task allocation by assigning robot i to task j only when $\alpha_{ij} = 1$.

By creating a *linear* program, we restrict the space of task allocation problems that we can model in one way: the function to be maximized (1) must be linear. Importantly, there is no such restriction on the manner in which the components of that function are derived. That is, individual utilities can be computed in any arbitrary way, but they must be combined linearly.

C. Scheduling formulation

The OAP, and thus MRTA, can also be seen from a scheduling perspective. Phrased in Brucker's terminology [4], our problem is in the class of scheduling problems described by:

$$R \,||\, \sum w_i C_i \tag{3}$$

That is, the system is composed of heterogeneous parallel machines and overall performance is computed as a weighted sum of the utility values for the individual tasks (Brucker uses C_i as the utility estimate of the machine assigned to task i and w_i as the scalar weight for task i). The problem class (3) is a superset of the simpler case of identical parallel machines:

$$P \,||\, \sum w_i C_i \tag{4}$$

Problems in the class (4) are known to be $\mathcal{NP}$-hard [4], and thus so are problems in the class (3).

Fortunately, we can simplify our problem by making two domain-specific observations. First, we recognize that the MRTA problem is a degenerate scheduling problem. Whereas in scheduling one must assign tasks to machines over time, in MRTA we consider only a single time-slot at each iteration. Second, we can incorporate the task weights directly into the utility estimates if we make the reasonable assumption that the task weights are known to the robots and can be used in utility estimation. Given a utility estimate U_{ij} for robot i and task j and a scalar task weight w_j, we can define a new weighted utility estimate:

$$U'_{ij} = w_j U_{ij}$$

For a single time-slot, we can trivially make this change of variables for each of the mn utilities U_{ij} in $O(mn)$ time. Thus, our particular problem reduces to an unweighted scheduling problem, becoming an instance of the class:

$$R \,||\, \sum C_i \tag{5}$$

Problems in the class (5) are known to be polynomially solvable, for example by Bruno et al.'s job scheduling algorithm [5], which runs in $O(mn^3)$ time. There also exist specialized algorithms that solve the OAP even faster, such as Kuhn's Hungarian method [13], which runs in $O(mn^2)$ time.

This result is important, because it suggests that we can develop practical, efficient mechanisms for making *optimal* allocations of tasks in multi-robot systems. If instead our problem were $\mathcal{NP}$-hard then we could only realistically expect to employ heuristic, potentially sub-optimal algorithms of the sort used to date for MRTA, and which we analyze in the next section.

Parker has shown [18] that a variant of MRTA, which she calls the ALLIANCE Efficiency Problem (AEP), is $\mathcal{NP}$-hard by restriction to the the $\mathcal{NP}$-complete problem PARTITION. This result does not contradict our preceding analysis of MRTA, for the AEP is in fact harder than the instantaneous task allocation problem that we consider in this paper. In the AEP, given is a set of tasks making up a mission, and the objective is to allocate a subset of these tasks to each robot so as to minimize the maximum time taken by a robot to serially execute its allocated tasks. Thus in order to solve the AEP, one must construct a time-extended schedule of tasks for each robot. This problem is clearly an instance of the scheduling problem:

$$R \,||\, C_{max} \tag{6}$$

which is known to be $\mathscr{N}\mathscr{P}$-hard [4]. Thus we have corroborated Parker's conclusion with a scheduling analysis.

III. ANALYSIS

Having given a formal statement of the MRTA problem, we are now in a position to analyze some of the key task allocation architectures from the literature. In this section we examine six approaches to MRTA, focusing on three characteristics:

- computation requirements [7]
- communication requirements [14]
- task consideration

In part because of trends in the research community that stress the importance of experimental validation with physical robots, such theoretical aspects of multi-robot coordination mechanisms have been largely ignored. However, they are vitally important to the study, comparison, and objective evaluation of the mechanisms. The large-scale and long-term behavior of the system will be strongly determined by the fundamental characteristics of the underlying algorithm(s). Thus we endeavor to derive and explicate those characteristics here. Before we continue, however, it will be necessary to explain the methodology that we use in our analysis.

A. Methodology

As we stated earlier, the key to effective task allocation for multi-robot systems is to iterate the assignment, in order to deal with changes in the tasks, the robots, and the environment. The architectures under study achieve this iteration in different ways, along two dimensions. First, while some approaches allow assignment and reassignment of all tasks at each iteration, some never reassign tasks (or at least only reassign them because of robot failure). Second, some approaches periodically consider all tasks simultaneously, while others consider single tasks sequentially as they are offered for (re)assignment. Thus, when we discuss complexities, we state them in terms of iterations, though the details of an "iteration" may vary across architectures.

We determine computation requirements, or running time, in the usual way, as the number of times that some dominant operation is repeated. For our domain that operation is usually either a calculation or comparison of utility, and running time is stated as a function of n and m, the numbers of robots and tasks, respectively. Since modern robots have significant processing capabilities onboard and can easily work in parallel, we assume that the computational load is evenly distributed over the robots, and state the running time as it is *for each robot*. For example, if we need to find for each robot the task with the highest utility, then the running time is $O(m)$, because each robot performs m comparisons, in parallel.

We determine communication requirements as the total number of inter-robot messages sent over the network. We do not consider message sizes, on the assumption that they are generally small (e.g., single scalar utility values) and approximately the same for different algorithms. We also assume that a perfect shared broadcast communication medium is in use and that messages are always broadcast, rather than unicast. So if, for example, each robot must tell every other robot its own highest utility value then the overhead is $O(n)$, because each robot makes a single broadcast.

B. The architectures

We have chosen for study six MRTA architectures that have been validated on either physical or simulated robots. Our choices are somewhat subjective, for there are a great many more architectures in the literature. However, we believe that we have gathered a set of approaches that is fairly representative of the work to date. In the following sections, we analyze these architectures; our results are summarized in Table I.

1) ALLIANCE and BLE: We begin our analysis with the behavior-based [15] ALLIANCE architecture [19], one of the earliest and best-known approaches to MRTA. At each iteration, all tasks are considered for (re)assignment, based on the robots' utility estimates. In this case, utilities are distributed among measures of *acquiescence* and *impatience*. For example, when a robot is currently executing a task, its utility for that task is decreased over time by its own increasing acquiescence and by the increasing impatience of the other robots. Similarly, a robot's utility for a task that is being executed by another robot is increased over time by its own impatience and by the other robot's acquiescence.

By splitting the utility estimation in this way, the ALLIANCE architecture decreases communication overhead. Since each robot is effectively modeling internally the progress of the others, the robots need not broadcast their utilities for each task (as is the case with many of the approaches described below). Specifically, assuming that all available tasks are currently underway, each engaged robot broadcasts only a heartbeat message each iteration, yielding a communication overhead of $O(m)$ per iteration. A significant drawback to this approach is that a variety of parameters that govern the robots' update rules for impatience and acquiescence must be carefully tuned. This problem was addressed in ALLIANCE in part by introducing parameter learning.

With regard to computation, each robot executes a greedy task-selection algorithm: for each available task, compare its own utility to that of every other robot and select the shortest task for which it is most capable (thus tasks are implicitly prioritized by length). This algorithm can be executed in $O(mn)$ time per iteration.

Name	Computational Requirements / iteration	Communication Requirements / iteration	Task Consideration
ALLIANCE [19]	$O(mn)$	$O(m)$	simultaneous, reassignment
BLE [22]	$O(mn)$	$O(mn)$	simultaneous, reassignment
M+ [3]	$O(mn)$	$O(mn)$	simultaneous, no reassignment
MURDOCH [12]	$O(1)$ / bidder $O(n)$ / auctioneer	$O(n)$	sequential, no reassignment
First-price auctions [23]	$O(1)$ / bidder $O(n)$ / auctioneer	$O(n)$	sequential, reassignment
Dynamic role assignment [6]	$O(1)$ / bidder $O(n)$ / auctioneer	$O(n)$	sequential, reassignment

TABLE I

Summary of selected MRTA architectures. Shown here are the computational and communication requirements for six key architectures. Note that "iteration" has a different meaning depending on whether tasks are considered simultaneously or sequentially.

Broadcast of Local Eligibility (BLE) [22] is another behavior-based approach to MRTA, with fixed-priority tasks. For each task, each robot has a corresponding behavior that is capable of executing the task, as well as estimating the robot's utility for the task. Utilities are computed in a task-specific manner as a function of relevant sensor data. These utilities are periodically broadcast to the other robots, with all tasks simultaneously considered for (re)assignment.

Since each robot must broadcast its utility for each task, we have communication overhead of $O(mn)$ per iteration. Upon receipt of the other robots' utilities, each robot executes a simple greedy algorithm: find the highest-priority task for which it is most fit. This algorithm requires each robot to compare, for each task, its own utility to that of every other robot, resulting in running time of $O(mn)$ per iteration.

The BLE algorithm has also been used to coordinate the actions of the Azzurra Robot Team (ART) [1] in a soccer domain. We note that, if task priorities are incorporated into utility estimates (see Section II-C) and all utility estimates have been gathered in a central table, both the ALLIANCE and BLE task algorithms can equivalently be stated in the following way:

1) Find the robot-task pair (i, j) with the highest utility.
2) Assign robot i to task j and remove them from consideration.
3) Go to step 1.

Exactly this greedy algorithm, operating on a global blackboard, has also been used in a recent study of the impact of communication and coordination on MRTA [17].

2) Auction-based approaches: The M+ system [3] achieves task allocation by use of a variant of the well-known Contract Net Protocol (CNP) [8]. The basic idea of the CNP is that when a task is available, it is put up for auction, and candidate robots make "bids" that are their task-specific utility estimates. The highest bidder (i.e., the best-fit robot) wins a contract for the task and proceeds to execute it.

In the M+ system, each robot considers, at each iteration, all currently available tasks. For each task, each robot uses a planner to compute its utility and announces the resulting value to the other robots. With each robot broadcasting its utility for each task, we have communication overhead of $O(mn)$ per iteration.

Upon receipt of the other robots' utilities, each robot executes essentially the same greedy task-selection algorithm that is used in ALLIANCE: find those tasks for which its utility is highest among all robots and pick from that set the highest-utility task. This algorithm can be executed in $O(mn)$ time per iteration.

Similar to M+, the MURDOCH task allocation mechanism [12] also employs a variant of CNP. Tasks are allocated by first-price auction [16] sequentially as they are stochastically introduced to the system; reassignment is not allowed. Utility is computed in a task-specific manner, as a function of relevant sensor inputs.

For each task auction, each available robot broadcasts its bid (i.e., utility), yielding communication overhead of $O(n)$ per iteration. Because of the asymmetric nature of MURDOCH's auctions, the running time varies between the bidders and the auctioneer. Each bidder need only compute its utility, while the auctioneer must find the highest utility among the bidders. Thus computational overhead per iteration is $O(1)$ for bidders and $O(n)$ for the auctioneer.

Another CNP-based approach to MRTA, this one applied to multi-robot exploration, is described in [23]. Tasks are allocated by first-price auction sequentially, but reassignment is allowed. Utilities are computed in a task-specific manner, in this case as a function of the estimated time required to travel to a target location. As with MURDOCH, each task auction requires each robot to broadcast its utility, resulting in communication overhead of $O(n)$ per iteration. Similarly, the computational overhead is asymmetric: $O(1)$ for bidders and $O(n)$ for the auctioneer (per iteration).

Finally, the dynamic role assignment architecture described in [6] is another CNP-based approach to MRTA. The complexities are the same as for MURDOCH.

That such auction-based allocation methods work in practice is not surprising, for it is well known that synthetic economic systems can be used to solve a variety of optimization problems. In fact, an appropriately constructed price-based market system (which the previously described architectures approximate to varying degrees) can optimally solve assignment problems. At equilibrium, such a market optimizes costs in the so-called *dual* of the original OAP, resulting in an optimal allocation [11], [2].

C. Solution quality

We have yet to characterize the results that can be expected from the architectures that we have analyzed. Unfortunately, such a characterization is difficult, if not impossible, to make. The crux of this difficulty is that all of the architectures execute some kind of *greedy* algorithm for task allocation. The solution quality of greedy optimization algorithms can be difficult to define, because it can depend strongly on the nature of the input. In the MRTA domain, the input (as described in Section II-B), is the set of robots, the set of tasks, and the environment that governs their evolution.

Although each of the approaches discussed in the previous section may in some cases produce allocations that are close to (or even equal to) an optimal allocation, it is trivial to construct pathological inputs that elicit arbitrarily sub-optimal allocations[4]. For example, consider the following matrix, giving the utilities (already weighted by task priority) of robots A and B for tasks x and y:

	x	y
A	100	99
B	99	1

(7)

Using either ALLIANCE or BLE, task x would be assigned to robot A, leaving task y for robot B. This solution yields the maximally sub-optimal value of 101, far less than the obvious optimal solution.

Similarly, CNP-based approaches are highly susceptible to the time ordering of tasks. The problem with such sequential architectures is really a time-extended analogue of the problem with simultaneous architectures such as ALLIANCE and BLE. Still working with the same utility matrix (7), imagine that the tasks x and y are introduced, in that order, to MURDOCH. Task x would be auctioned off to the highest bidder, robot A, and task y would be left for robot B, resulting in the same poor allocation.

Thus we go no further than to classify the algorithms analyzed in Section III-B as greedy, and assert that they should be expected to produce nearly identical solutions, subject to the following observations:

- Simultaneous consideration of tasks will produce allocations that are at least as good as those produced with sequential consideration.
- Allowing reassignment of previously assigned tasks will produce allocations that are at least good as those produced without reassignment.

We are currently working to establish more informative performance bounds for these algorithms.

IV. CONCLUSION

With the goal of bringing some objective grounding to an emerging area of research that has, to date, been largely experimental, we have presented a formal study of the problem of multi-robot task allocation (MRTA). We have given a domain-independent statement of the problem and shown that it can be understood as an instance of the well-known optimal assignment problem (OAP). By this reduction, as well as a related scheduling analysis, we have shown that it is possible to *optimally* solve the original MRTA problem in polynomial time. By interpreting them as algorithms for solving the underlying OAP, we have analyzed the computation and communication requirements and (sub-)optimality of several robot task-allocation architectures from the literature.

There are many ways in which to exploit this information. For example, when building a multi-robot system, one can use the results presented in Table I in order to select an appropriate task-allocation architecture (if necessary, the analysis in Section III-B can be easily extended to include other architectures). Similarly, when designing a new method for MRTA, our definition of the problem and our exposition on previous approaches may prove useful.

For our own research, we plan to pursue the opportunities provided by the substantial body of work regarding the OAP that is available in other fields, including operations research, economics, and game theory. We are currently investigating the applicability to the robot domain of a wide variety of efficient, optimal assignment algorithms,

[4]In the parlance of algorithmic analysis, we are showing by counterexample that the MRTA problem, as it is approached by the architectures under study, lacks the *greedy-choice property*, which is a prerequisite for a greedy algorithm to produce an optimal solution [7].

both distributed and centralized. Based in part on their communication and computation requirements (which are generally well-known), we plan to adapt, implement, and experiment with some of these algorithms in the domain of multi-robot task allocation.

ACKNOWLEDGMENTS

The research reported here was conducted at the Interaction Lab, part of the Center for Robotics and Embedded Systems (CRES) at USC. The work is supported in part by the Intel Foundation, DARPA Grant DABT63-99-1-0015 (MARS), and DARPA Grant F30602-00-2-0573 (TASK). We thank Michael Wellman, Herbert Dawid, and Gaurav Sukhatme for their insightful comments.

V. REFERENCES

[1] Giovanni Adorni, Andrea Bonarini, Giorgio Clemente, Daniele Nardi, Enrico Pagello, and Maurizio Piaggio. ART'00 - Azzurra robot team for the year 2000. In Peter Stone, Tucker Balch, and Gerhard Kraetzschmar, editors, *RoboCup 2000, LNAI 2019*, pages 559–562. Springer-Verlag, Berlin, 2001.

[2] Dmitri P. Bertsekas. The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial. *Interfaces*, 20(4):133–149, July 1990.

[3] Sylvia C. Botelho and Rachid Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1234–1239, Detroit, Michigan, May 1999.

[4] Peter Brucker. *Scheduling Algorithms*. Springer-Verlag, Berlin, 2^{nd} edition, 1998.

[5] John L. Bruno, Edward G. Coffman, and Ravi Sethi. Scheduling Independent Tasks To Reduce Mean Finishing Time. *Communications of the ACM*, 17(7):382–387, July 1974.

[6] Luiz Chaimowicz, Mario F. M. Campos, and Vijay Kumar. Dynamic Role Assignment for Cooperative Robots. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 293–298, Washington, DC, May 2002.

[7] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 1997.

[8] Randall Davis and Reid G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, 20(1):63–109, 1983.

[9] Jean-Louis Deneubourg, Guy Theraulaz, and Ralph Beckers. Swarm-made architectures. In *Proc. of the European. Conf. on Artificial Life (ECAL)*, pages 123–133, Paris, France, 1991.

[10] Francis Y. Edgeworth. Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences, 1881. Translated and reprinted, Augustus M. Kelley, New York, 1967.

[11] David Gale. *The Theory of Linear Economic Models*. McGraw-Hill Book Company, Inc., New York, 1960.

[12] Brian P. Gerkey and Maja J Matarić. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, October 2002.

[13] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1):83–97, 1955.

[14] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, Cambridge, 1997.

[15] Maja J Matarić. Behavior-based control: Examples from navigation, learning, and group behavior. *J. of Experimental and Theoretical Artifical Intelligence*, 9(2–3):323–336, 1997.

[16] R. Preston McAfee and John McMillan. Auctions and Bidding. *J. of Economic Literature*, 25(2):699–738, June 1987.

[17] Esben H. Østergård, Maja J Matarić, and Gaurav S Sukhatme. Distributed multi-robot task allocation for emergency handling. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 821–826, Wailea, Hawaii, October 2001.

[18] Lynne E. Parker. L-ALLIANCE: A Mechanism for Adaptive Action Selection in Heterogeneous Multi-Robot Teams. Technical Report ORNL/TM-13000, Oak Ridge National Laboratory, October 1995.

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

[20] David V. Pynadath and Milind Tambe. Multiagent Teamwork: Analyzing the Optimality and Complexity of Key Theories and Models. In *Proc. of Intl. Conf. on Autonomous Agents and Multi Agent Systems*, pages 873–880, Bologna, Italy, July 2002.

[21] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. J. Wiley, New York, 3^{rd} edition, 1964.

[22] Barry Brian Werger and Maja J Matarić. Broadcast of Local Eligibility for Multi-Target Observation. In Lynne E. Parker, George Bekey, and Jacob Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 347–356. Springer-Verlag, 2000.

[23] Robert Zlot, Anthony Stentz, M. Bernardine Dias, and Scott Thayer. Multi-Robot Exploration Controlled by a Market Economy. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3016–3023, Washington, DC, May 2002.