**Brian P. Gerkey**

Artificial Intelligence Lab
Stanford University
Stanford, CA 94305-9010, USA
gerkey@stanford.edu

**Maja J. Matarić**

Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA

# A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems

## Abstract

*Despite more than a decade of experimental work in multi-robot systems, important theoretical aspects of multi-robot coordination mechanisms have, to date, been largely untreated. To address this issue, we focus on the problem of multi-robot task allocation (MRTA). Most work on MRTA has been ad hoc and empirical, with many coordination architectures having been proposed and validated in a proof-of-concept fashion, but infrequently analyzed. With the goal of bringing objective grounding to this important area of research, we present a formal study of MRTA problems. A domain-independent taxonomy of MRTA problems is given, and it is shown how many such problems can be viewed as instances of other, well-studied, optimization problems. We demonstrate how relevant theory from operations research and combinatorial optimization can be used for analysis and greater understanding of existing approaches to task allocation, and to show how the same theory can be used in the synthesis of new approaches.*

KEY WORDS—task allocation, multi-robot systems, coordination, utility

## 1. Introduction

Over the past decade, a significant shift of focus has occurred in the field of mobile robotics as researchers have begun to investigate problems involving multiple, rather than single, robots. From early work on loosely-coupled tasks such as homogeneous foraging (Matarić 1992) to more recent work on team coordination for robot soccer (Stone and Veloso 1999), the complexity of the multi-robot systems being studied has increased. This complexity has two primary sources: larger team sizes and greater heterogeneity of robots and tasks. As

significant achievements have been made along these axes, it is no longer sufficient to show, for example, a pair of robots observing targets or a large group of robots flocking as examples of coordinated robot behavior. Today we reasonably expect to see increasingly larger robot teams engaged in concurrent and diverse tasks over extended periods of time.

### 1.1. Multi-Robot Task Allocation

As a result of the growing focus on multi-robot systems, multi-robot coordination has received significant attention. In particular, multi-robot task allocation (MRTA) has recently risen to prominence and become a key research topic in its own right. As researchers design, build, and use cooperative multi-robot systems, they invariably encounter the fundamental question: which robot should execute which task in order to cooperatively achieve the global goal? By "task", we mean a subgoal that is necessary for achieving the overall goal of the system, and that can be achieved independently of other subgoals (i.e., tasks). Tasks can be discrete (e.g., deliver this package to room 101) or continuous (e.g., monitor the building entrance for intruders) and can also vary in a number of other ways, including time-scale, complexity, and specificity. We do not categorize or distinguish between different types of robotic tasks, although others have done so (see Section 2). Task independence is a strong assumption, and one that clearly limits the scope of our study. For example, we do not allow ordering constraints on a set of tasks; in general, we require that individual tasks can be considered and assigned independently of each other. This issue is addressed further in Section 7.2.

In this paper, we are concerned with methods for intentional cooperation (Parker 1998). In this model, robots cooperate explicitly and with purpose, often through task-related communication and negotiation. Intentional cooperation is clearly not a prerequisite for a multi-robot system to exhibit coordinated behavior, as demonstrated by minimalist or

939

emergent approaches (Deneubourg, Theraulaz, and Beckers 1991). In such systems, individuals coordinate their actions through their interactions with each other and with the environment, but without explicit negotiation or allocation of tasks. An open question is which tasks (if any) require intentional cooperation. For example, cooperative box-pushing has been demonstrated using both emergent (Kube and Zhang 1993) and intentional (Parker 1998) techniques, and there remains significant debate as to the relative value of the two approaches.

However, emergent systems tend not to be amenable to analysis, with their exact behavior difficult, if not impossible, to predict. We assert that, as compared with emergent cooperation, intentional cooperation is usually better suited to the kinds of real-world tasks that humans might want robots to do. If the robots are deliberately cooperating with each other, then, intuitively we expect that humans can deliberately cooperate with them, which is a long-term goal of multi-robot research. Furthermore, intentional cooperation has the potential to better exploit the capabilities of heterogeneous robot teams. In this paper, the use of intentional cooperation is at the level of task allocation, and need not propagate to the level of task execution. Importantly, we do not prescribe or proscribe any particular method for implementing the details of a task. For example, if a foraging task is assigned to a team of robots because they are best fit for the job, they can execute the task in any way they wish, from probabilistic swarming to classical planning.

### 1.2. Toward Formal Analysis

The question of task allocation must be answered, even for relatively simple multi-robot systems, and its importance grows with the complexity, in size and capability, of the system under study. The empirically validated methods demonstrated to date remain primarily ad hoc in nature, and relatively little has been written about the general properties of cooperative multi-robot systems. After a decade of research, while cooperative architectures have been proposed, the field still lacks a prescription for how to design a MRTA system. Similarly, there has been little attempt to evaluate or compare the proposed architectures, either analytically or empirically.

In this paper we present a particular taxonomy for studying MRTA, based on organizational theory from several fields, including operations research, economics, scheduling, network flows, and combinatorial optimization. We show how this taxonomy can be used to analyze and classify MRTA problems, and evaluate and compare proposed solutions. For the simpler (and most widely studied) problems, we provide a complete analysis and prescribe provably optimal, yet tractable, algorithms for their solution. For more difficult problems, we suggest candidate approximation algorithms that have enjoyed success in other application domains. There are also some extremely difficult MRTA problems for which there do not currently exist good approximations; in such cases we provide formal characterizations of the problems but do not suggest how they should be solved.

Our approach is not meant to be final or exhaustive and indeed it has limitations. However, we believe that the ideas we present constitute a starting point on a path toward a more complete understanding of problems involving MRTA, as well as other aspects of multi-robot coordination.

## 2. Related Work

Research in multi-robot systems has focused primarily on construction and validation of working systems, rather than more general analysis of problems and solutions. As a result, in the literature, one can find many architectures for multi-robot coordination, but relatively few formal models of multi-robot coordination. We do not attempt here to cover the various proposed and demonstrated architectures. For a thorough treatment of implemented multi-robot systems, consult Cao, Fukunaga, and Kahng (1997) or the more recent work of Dudek, Jenkin, and Milios (2002). Each provides a taxonomy that categorizes the bulk of existing multi-robot systems along various axes, including team organization (e.g., centralized versus distributed), communication topology (e.g., broadcast versus unicast), and team composition (e.g., homogeneous versus heterogeneous). Rather than characterize architectures, we seek instead to categorize the underlying problems, although we do analyze and discuss several key architectures that solve those problems; see Section 6.

Formal models of coordination in multi-robot systems tend to target medium-scale to large-scale systems composed of simple, homogeneous robots, such as the CEBOTS (Fukuda et al. 1988). Agassounon and Martinoli (2002) explored the tradeoffs between using a coarse, macroscopic model of such systems and using detailed, microscopic models of the individuals. Lerman and Galstyan (2002) presented a physics-inspired macroscopic model of a cooperative multi-robot system and showed that it accurately described the behavior of physical robots engaged in stick-pulling and foraging tasks. This type of model is descriptive but not prescriptive, in that it does not guide the design of control or coordination mechanisms.

Although simple and elegant, such models are insufficient for domains involving complex tasks or requiring precise control. To study complex tasks, Donald, Jennings, and Rus (1997) proposed the formalism of information invariants, which models the information requirements of a coordination algorithm and provides a mechanism to perform reductions between algorithms. Spletzer and Taylor (2001) developed a prescriptive control-theoretic model of multi-robot coordination and showed that it can be used to produce precise multi-robot box-pushing. Mason (1986) had earlier applied a similar control-theoretic model to box-pushing with dexterous manipulators.

Relatively little work has been done on formal modeling, analysis, or comparison of multi-robot coordination at the level of task allocation. Chien et al. (2000) developed a baseline geological scenario and used it to compare three different planning approaches to coordinating teams of planetary rovers. Klavins (2003) showed how to apply the theory of communication complexity to the study of multi-robot coordination algorithms. Finally, Jennings and Kirkwood-Watts (1998) described the method of dynamic teams, concentrating on programmatic structures that enable the specification of multi-robot tasks.

Multi-robot systems can also be formally described by process models, such as Petri nets (Murata 1989) and partially observable Markov decision processes (Kaelbling, Littman, and Cassandra 1998), both of which are highly expressive. Unfortunately, such models tend to be too complex to be directly analyzed or solved, even for modest-sized systems. Another formal model is that of the hybrid system (Alur et al. 1995), which characterizes discrete systems operating in an analog environment. Hybrid systems can also become complex and are usually used to describe or control the behavior of a single robot, via a so-called three-layer architecture (Gat 1998).

Our goal in this paper is to fill a gap in the existing literature on multi-robot coordination. We neither construct a formal model in support of a particular coordination architecture, nor compare different architectures in a particular task domain. Rather, we develop a task- and architecture-independent taxonomy, based on optimization theory, in which to study task allocation problems.

## 3. Utility

To treat task allocation in an optimization context, one must decide what exactly is to be optimized. Ideally the goal is to directly optimize overall system performance, but that quantity is often difficult to measure during system execution. Furthermore, when selecting among alternative task allocations, the impact on system performance of each option is usually not known. Consequently, some kind of performance estimate, such as utility, is needed.

Utility is a unifying, if sometimes implicit, concept in economics, game theory, and operations research, as well as in multi-robot coordination. It is based on the notion that each individual can internally estimate the value (or the cost) of executing an action. Depending on the context, utility is also called fitness, valuation, and cost. Within multi-robot research, the formulation of utility can vary from sophisticated planner-based methods (Botelho and Alami 1999) to simple sensor-based metrics (Gerkey and Matarić 2002b). We posit that utility estimation of this kind is carried out somewhere in every autonomous task allocation system, for the heart of any task allocation problem is comparison and selection among a set of available alternatives. Since each system uses a differ-

ent method to calculate utility, we give the following generic and practical definition of utility for multi-robot systems.

It is assumed that each robot is capable of estimating its fitness for every task it can perform. This estimation includes two factors, which are both task- and robot-dependent:

1. 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);

2. 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 one 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 $R$. This results in a combined, nonnegative utility measure:

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

For example, given a robot $A$ that can achieve a task $T$ with quality $Q_{AT} = 20$ at cost $C_{AT} = 10$ and a robot $B$ that can achieve the same task with quality $Q_{BT} = 15$ at cost $C_{BT} = 5$, there should be no preference between them when searching for efficient assignments, for

$$U_{AT} = 20 - 10 = 10 = 15 - 5 = U_{BT}.$$

Regardless of the method used for calculation, the robots' utility estimates will be inexact due to 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 that we are incapable of improving it at the task allocation level. When we discuss "optimal" allocations, 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. This notion of optimality is analogous to that used in optimal scheduling (Dertouzos and Mok 1983).

It is important to note that utility is an extremely flexible measure of fitness that can encompass arbitrary computation. The only constraint on utility estimators is that they must each produce a single scalar value that can be compared for the purpose of ordering candidates for tasks. For example, if the metric for a particular task is distance to a location and a candidate robot employs a probabilistic localization mechanism, then a reasonable utility estimate might be to calculate the

distance to the target using the center of mass of the current probability distribution. Other mechanisms, such as planning and learning, can likewise be incorporated into utility estimation. Regardless of the domain, it is vital that all relevant aspects of the state of the robots and their environment be included in the utility calculation. Signals that are left out of this calculation but are taken into consideration when evaluating overall system performance are what economists refer to as "externalities" (Simon 2001) and their effects can be detrimental, if not catastrophic.

## 4. Combinatorial Optimization

Before entering into a discussion of task allocation problems as being primarily concerned with optimization, it will be necessary to provide some theoretical background. The field of combinatorial optimization provides a set-theoretic framework, based on "subset systems", for describing a wide variety of optimization problems (Ahuja, Magnanti, and Orlin 1993):

DEFINITION 1.   (Subset System) A subset system $(E, F)$ is a finite set of objects $E$ and a non-empty collection $F$ of subsets, called independent sets, of $E$ that satisfies the property that if $X \in F$ and $Y \subseteq X$ then $Y \in F$.

That is, any subset of an independent set is also an independent set. A general maximization problem can be defined in the following way:

DEFINITION 2.   (Subset Maximization) Given a subset system $(E, F)$ and a utility function $u : E \rightarrow \mathbb{R}_+$, find an $X \in F$ that maximizes the total utility:

$$u(X) = \sum_{e \in X} u(e). \qquad (1)$$

The elements of $F$ are usually not given directly, or at least are inconvenient to represent explicitly. Instead, it is assumed that an "oracle" is available that, given a candidate set $X$, can decide whether $X \in F$. The job of such an oracle, given a proposed solution, is to verify the feasibility of that solution. For many problems, this verification is computationally trivial when compared to the complexity of the optimization problem.

Given a maximization problem over a subset system, one can define algorithms that attempt to solve it. Of particular interest is the canonical Greedy algorithm (Ahuja, Magnanti, and Orlin 1993):

ALGORITHM 1.   (The Greedy algorithm)

1. Reorder the elements of $E = \{e_1, e_2, \ldots, e_n\}$ such that $u(e_1) \geq u(e_2) \geq \ldots \geq u(e_n)$.

2. Set $X := \emptyset$.

3. For $j = 1$ to $n$:
      if $X \cup \{e_j\} \in F$ then $X = X \cup \{e_j\}$.

This algorithm is an abstraction of the familiar and intuitive Greedy algorithm for solving a problem: repeatedly take the best valid option. While the Greedy algorithm performs well on some optimization problems, it can do quite poorly on others. In particular, it performs well on certain subset systems that can be further classified as "matroids":

DEFINITION 3.   (Matroid) A subset system $(E, F)$ is a matroid if, for each $X, Y \in F$ with $|X| > |Y|$, there exists an $x \in X \setminus Y$ such that $Y \cup \{x\} \in F$.

That is, given two independent sets $X$ and $Y$, with $X$ larger than $Y$, $Y$ can be "grown" by adding to it some element from $X$. With respect to the current discussion, an equivalent definition of a matroid is that a subset system $(E, F)$ is a matroid if and only if the Greedy algorithm optimally solves the associated maximization problem (Korte and Vygen 2000). In the parlance of algorithmic analysis, matroids satisfy the Greedy-choice property, which is a prerequisite for a Greedy algorithm to produce an optimal solution (Cormen, Leiserson, and Rivest 1997). Matroids are of particular interest precisely because their associated optimization problems are amenable to the Greedy solution.

While the Greedy algorithm does not optimally solve every maximization problem, it is useful to know how poor the Greedy solution can be. For such purposes it is common to report a "competitive factor" for the suboptimal algorithm. For a maximization problem, an algorithm is called $\alpha$-competitive if, for any input, it finds a solution whose total utility is never less than $1/\alpha$ of the optimal utility.

## 5. A Taxonomy of MRTA Problems

We propose a taxonomy of MRTA problems based on axes laid out below. Our goals here are twofold: (1) to show how various MRTA problems can be positioned in the resulting problem space; (2) to explain how organizational theory relates to those problems and to propose solutions from the robotics literature. In some cases, it will be possible to construct provably optimal solutions, while in others only approximate solutions are available. There are also some difficult MRTA problems for which there do not currently exist good approximations. When designing a multi-robot system, it is essential to understand what kind of task allocation problem is present in order to solve it in a principled manner.

We propose the following three axes for use in describing MRTA problems.

- **Single-task robots (ST)** versus **multi-task robots (MT)**: ST means that each robot is capable of executing at most one task at a time, while MT means that some robots can execute multiple tasks simultaneously.

- **Single-robot tasks (SR)** versus **multi-robot tasks (MR)**: SR means that each task requires exactly one

robot to achieve it, while MR means that some tasks can require multiple robots.

- **Instantaneous assignment (IA)** versus **time-extended assignment (TA)**: IA means that the available information concerning the robots, the tasks, and the environment permits only an instantaneous allocation of tasks to the robots, with no planning for future allocations; TA means that more information is available, such as the set of all tasks that will need to be assigned, or a model of how tasks are expected to arrive over time.

We denote a particular MRTA problem by a triple of two-letter abbreviations drawn from this list. For example, a problem in which multi-robot tasks must be allocated once to single-task robots is designated ST–MR–IA.

These axes are not meant to be exhaustive, but to allow for a taxonomy that is both broad enough and detailed enough to meaningfully characterize many practical MRTA problems. Furthermore, this taxonomy will often allow for a prescription of solutions. The following sections present the combinations allowed by these axes, discussing, for each, which MRTA problem(s) it represents and what organizational theory pertains. Section 7 treats some important MRTA problems that are not captured by this taxonomy.

### 5.1. ST–SR–IA: Single-Task Robots, Single-Robot Tasks, Instantaneous Assignment

This problem is the simplest, as it is actually an instance of the optimal assignment problem (OAP; Gale 1960), which is a well-known problem that was originally studied in game theory and then in operations research, in the context of personnel assignment. A recurring special case of particular interest in several fields of study, this problem can be formulated in many ways. Given the application domain of MRTA, it is fitting to describe the problem in terms of jobs and workers.

DEFINITION 4. (Optimal Assignment Problem) Given $m$ workers, each looking for one job and $n$ prioritized jobs, each requiring one worker. Also given for each worker is a nonnegative skill rating (i.e., utility estimate) that predicts his/her performance for each job; if a worker is incapable of undertaking a job, then the worker is assigned a rating of zero for that job. The goal is to assign workers to jobs so as to maximize overall expected performance, taking into account the priorities of the jobs and the skill ratings of the workers.

The OAP can be cast in many ways, including as an integral linear program (Gale 1960): find $mn$ non-negative integers $\alpha_{ij}$ that maximize

$$U = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} U_{ij} w_j \qquad (2)$$

subject to

$$\sum_{i=1}^{m} \alpha_{ij} = 1, \quad 1 \le j \le n$$

$$\sum_{j=1}^{n} \alpha_{ij} = 1, \quad 1 \le i \le m. \qquad (3)$$

The sum (2) is the overall system utility, while eq. (3) enforces the constraint of working with single-worker jobs and single-job workers (note that since $\alpha_{ij}$ are integers they must all be either 0 or 1). Given an optimal solution to this problem (i.e., a set of integers $\alpha_{ij}$ that maximizes eq. (2) subject to eq. (3)), an optimal assignment is constructed by assigning worker $i$ to job $j$ only when $\alpha_{ij} = 1$.

The ST–SR–IA problem can be posed as an OAP in the following way. Given $m$ robots, $n$ prioritized tasks, and utility estimates for each of the $mn$ possible robot–task pairs, assign at most one task to each robot. If the robots' utilities can be collected at one machine (or distributed to all machines), then a centralized linear programming approach, such as the Hungarian method (Kuhn 1955), will find the optimal allocation in $O(mn^2)$ time.

Alternatively, a distributed auction-based approach, such as the Auction algorithm (Bertsekas 1990), will find the optimal allocation, usually requiring time proportional to the maximum utility and inversely proportional to the minimum bidding increment. In order to understand such economically-inspired algorithms, it is necessary to consider the concept of linear programming duality. As do all maximum linear programs, the OAP has a dual minimum linear program, which can be stated as follows. Find $m$ integers $u_i$ and $n$ integers $v_j$ that minimize

$$P = \sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j \qquad (4)$$

subject to

$$u_i + v_j \ge U_{ij}, \ \forall i, j. \qquad (5)$$

The Duality theorem states that the original problem (called the primal) and its dual are equivalent, and that the total utility of their respective optimal solutions are the same (Gale 1960).

Optimal auction algorithms for task allocation usually work in the following way. Construct a price-based "task market", in which tasks are sold by brokers to robots. Each task $j$ is for sale by a broker, which places a value $c_j$ on the task. Also, robot $i$ places a value $h_{ij}$ on task $j$. The problem then is to establish task "prices" $p_j$, which will in turn determine the allocation of tasks to robots. To be feasible, the price $p_j$ for task $j$ must be greater than or equal to the broker's valuation $c_j$; otherwise, the broker would refuse to sell. Assuming that the robots are acting selfishly, each robot $i$ will elect to buy a

task $t_i$ for which its profit is maximized:

$$t_i = \text{argmax}_j \{h_{ij} - p_j\}. \tag{6}$$

Such a market is said to be at equilibrium when prices are such that no two robots select the same task.

At equilibrium, each individual's profit in this market is maximized. Furthermore, the profits made by the robots and the profits made by the brokers form an optimal solution to the dual of the OAP:

$$
\begin{aligned}
u_i &= h_{it_i} - p_{t_i}, \ \forall i \\
v_j &= p_j - c_j, \ \forall j.
\end{aligned}
\tag{7}
$$

Thus, the allocation produced by the market at equilibrium is optimal (Gale 1960).

In MRTA problems, separate valuations are not given in this manner, but only combined utility estimates for robot–task pairs. However, task valuations can be defined for the robots and brokers as follows:

$$
\begin{aligned}
h_{ij} &= \alpha_{ij} \\
c_j &= 0.
\end{aligned}
\tag{8}
$$

The solution to the corresponding dual problem then becomes

$$
\begin{aligned}
u_i &= \alpha_{it_i} - p_{t_i} \\
v_j &= p_j.
\end{aligned}
\tag{9}
$$

Note that setting $c_j$ to 0 implicitly states that the brokers always prefer to sell their tasks, regardless of how much they are paid. In other words, it is always better to execute a task than not execute it, regardless of the expected performance. In economic terminology, those are lexicographic preferences with regard to the tasks (Pearce 1999). Such preferences violate important assumptions concerning the nature of utility values that are made when building or analyzing general economic systems. Fortunately, in constructing the market corresponding to the ST–SR–IA problem, no assumptions are made concerning the robots' preferences, and so lexicographic preferences do not present a problem. On the other hand, the behavior of more complex, long-lived economies, such as the markets suggested by Dias and Stentz (2001) and Gerkey and Matarić (2002a), may depend strongly on the nature of the robots' preferences, especially if the synthetic economies are meant to interact with the human economy.

The two approaches (i.e., centralized and distributed) to solving the OAP represent a tradeoff between solution time and communication overhead. Centralized approaches generally run faster than distributed approaches, but incur a higher communication overhead. To implement a centralized assignment algorithm, $n^2$ messages are required to transmit the utility of each robot for each task; an auction-based solution usually requires far fewer (sometimes fewer than $n$) messages to reach equilibrium. With the addition of simple optimizations, such as buffering multiple utility values and transmitting them

in one message, this gap in communication overhead will only become apparent in large-scale systems. Furthermore, the time required to transmit a message cannot be ignored, especially in wireless networks, which can induce significant latency. Thus, for small-scale to medium-scale systems, say $n < 200$, a broadcast-based centralized assignment solution is likely the better choice. Not surprisingly, many MRTA architectures implement some form of this approach (Parker 1998; Castelpietra et al. 2001; Østergård, Matarić, and Sukhatme 2001; Werger and Matarić 2001; Weigel et al. 2001).

In order to determine its viability for solving MRTA problems, we implemented, in ANSI C, the Hungarian method[1] (Kuhn 1955). We tested on randomly generated symmetric assignment problems (i.e., problems where $m = n$) with uniformly distributed utilities, and found that the Hungarian method is easily fast enough to be used in the control loop in real-world MRTA domains. Using a Pentium III 700 MHz, problems with tens of robots and tasks can be solved in less than 1 ms and problems with 300 robots and tasks can be solved in less than 1 s.

### 5.1.1. Variant: Iterated Assignment

Few MRTA problems exhibit exactly the above one-time assignment structure. However, many problems can be framed as iterated instances of ST–SR–IA. Consider the cooperative multi-object tracking problem known as CMOMMT, studied by Parker (1999) and Werger and Matarić (2001), which consists of coordinating robots to observe multiple unpredictably moving targets. When presented with new sensor inputs (e.g., camera images) and consequent utility estimates (e.g., perceived distance to each target), the system must decide which robot should track which target.

The Werger and Matarić (2001) MRTA architecture, broadcast of local eligibility (BLE), solves this iterated assignment problem using the following algorithm:

ALGORITHM 2.   (BLE assignment algorithm)

1. If any robot remains unassigned, find the robot–task pair $(i, j)$ with the highest utility. Otherwise, quit.

2. Assign robot $i$ to task $j$ and remove them from consideration.

3. Go to step 1.

This algorithm is an instance of the canonical Greedy algorithm. The OAP is not a matroid (see Section 4) and so the Greedy algorithm will not necessarily produce an optimal solution. The Greedy algorithm is known to be 2-competitive for the OAP (Avis 1983), and thus so is BLE. That is, in the worst case, BLE will produce a solution whose benefit is half of the

---

1. The code for our implementation is available from http://robotics.stanford.edu/~gerkey.

optimal benefit. Exactly this algorithm, operating on a global blackboard, has been used in a study of the impact of communication and coordination on MRTA (Østergård, Matarić, and Sukhatme 2001). A very similar assignment algorithm is also used by the Botelho and Alami (1999) M+ architecture.

The Parker (1998) MRTA architecture L-ALLIANCE, which can also perform iterated allocation, learns its assignment algorithm from experience. The resulting algorithm is similar to, but potentially more sophisticated than, the Greedy algorithm. If well trained, the L-ALLIANCE assignment algorithm can outperform the Greedy algorithm (Parker 1994), but is not guaranteed to be optimal.

Another domain in which the iterated OAP arises is robot soccer. Since many of the robots are interchangeable, it is often advantageous to allow any player to take on any role within the team, according to the current situation in the game. The resulting coordination problem can be cast as an iterated assignment problem in which the robots' roles are periodically reevaluated, usually at a frequency of the order of 10 Hz. This utility-based dynamic role assignment problem has been studied by many (Stone and Veloso 1999; Castelpietra et al. 2001; Weigel et al. 2001; Emery, Sikorski, and Balch 2002; Vail and Veloso 2003).

It is common in the robot soccer domain for each robot to calculate its utility for each role and periodically broadcast these values to its teammates. The robots can then execute, in parallel, some centralized assignment algorithm. For example, the Castelpietra et al. (2001) assignment algorithm consists of ordering the roles in descending priority and then assigning each to the available robot with the highest utility. This algorithm is yet another instance of the Greedy algorithm. Vail and Veloso (2003) also employ the Greedy algorithm with fixed priority roles. Weigel et al. (2001) employ a similar but slightly more sophisticated algorithm that tries to address the problem of excessive role-swapping by imposing stricter prerequisites for reassignment. Among other things, the algorithm requires that both robots "want" to exchange roles in order to maximize their respective utilities, recalling the conditions for equilibrium in markets (see Section 5.1). However, the Weigel et al. (2001) algorithm is not guaranteed to produce optimal assignments of roles, a fact that can easily be shown by counterexample.

Since the number of robots involved in many iterated MRTA problems today is small ($n \leq 11$ for robot soccer, which is more than for most current multi-robot systems), $O(n^3)$ optimal assignment algorithms could easily replace the suboptimal ad hoc assignment algorithms that are typically used. As the performance results mentioned in the previous section show, the Hungarian method can be used to solve typical problems in less than 1 ms per iteration with the moderately powerful computers found on today's robots.

Since there is some additional cost for running an optimal algorithm (if only in the work involved in the implementation), one might ask whether the optimal solution provides a

sufficient benefit. For example, it is known that for arbitrary assignment problems, the Greedy algorithm's worst-case behavior is to produce a solution with half of the optimal utility. However, it is not known how the algorithm can be expected to perform on typical MRTA problems, which exhibit some structure and are unlikely to present pathological utility combinations. Anecdotal evidence suggests that the Greedy algorithm works extremely well on such problems. An interesting avenue of research would be to analytically determine how well the Greedy algorithm will perform on the kinds of utility landscapes that are encountered in MRTA problems.

### 5.1.2. Variant: On-line Assignment

In some MRTA problems, the set of tasks is not revealed at once, but rather the tasks are introduced one at a time. If robots that have already been assigned cannot be reassigned, then this problem is a variant of SR–ST–IA, known as on-line assignment (Kalyanasundaram and Pruhs 1993). Instead of being initially given, the robot–task utility matrix is revealed one column (or row) at a time. If previously assigned robots can be reassigned, then the problem reduces to an instance of the iterated SR–ST–IA problem, which can be optimally solved with standard assignment algorithms.

The MRTA problems solved by the MURDOCH system (Gerkey and Matarić 2002b), in which tasks are randomly injected into the system over time, are instances of the on-line assignment problem. The MURDOCH assignment algorithm can be stated as follows:

ALGORITHM 3.    (MURDOCH Assignment Algorithm)

1. When a new task is introduced, assign it to the most fit robot that is currently available.

This simple algorithm is yet another instance of the Greedy algorithm, and is known in the context of network flows as the Farthest Neighbor algorithm. Not surprisingly, the on-line assignment problem is not a matroid; the Greedy algorithm is known to be 3-competitive with respect to the optimal post hoc off-line solution. Furthermore, this performance bound is the best possible for any on-line assignment algorithm (Kalyanasundaram and Pruhs 1993). Thus, without a model of the tasks that are to be introduced, and without the option of reassigning robots that have already been assigned, it is impossible to construct a better task allocator than MURDOCH.

### 5.2. ST–SR–TA: Single-Task Robots, Single-Robot Tasks, Time-Extended Assignment

When the system consists of more tasks than robots, or if there is a model of how tasks will arrive, and the robots' future utilities for the tasks can be predicted with some accuracy, then the problem is an instance of ST–SR–TA. This problem is one of building a time-extended schedule of tasks for each

robot, with the goal of minimizing total weighted cost. Using the terminology of Brucker (1998), this problem is an instance of the class of scheduling problems

$$R \mid\mid \sum w_j C_j.$$

That is, the robots execute tasks in parallel ($R$) and the optimization criterion is the weighted sum of execution costs ($\sum w_j C_j$). Problems in this class are strongly $\mathcal{NP}$-hard (Bruno, Coffman, and Sethi 1974). Even for relatively small problems, the exponential space of possible schedules precludes enumerative solutions.

A means of treating ST–SR–TA is to ignore the time-extended component and approximate the problem as an instance of the ST–SR–IA problem (Section 5.1), followed by an instance of the on-line assignment problem (Section 5.1.2). For example, given $m$ robots and $n$ tasks, with $n > m$, the following approximation algorithm can be used:

ALGORITHM 4.    (ST-SR-TA approximation algorithm)

1. Optimally solve the initial $m \times n$ assignment problem.

2. Use the Greedy algorithm to assign the remaining tasks in an on-line fashion, as the robots become available.

The performance of this algorithm is bounded below by the normal Greedy algorithm, which is 3-competitive for on-line assignment. The more tasks that are assigned in the first step, the better this algorithm will perform. As the difference between the number of robots and the number of tasks that are initially presented decreases (i.e., $(n - m) \to 0$), performance approaches optimality, wherein all tasks are assigned in one step. Thus, although it is not guaranteed to produce optimal solutions, this algorithm should work well in practice, especially for ST–SR–TA problems with short time horizons.

Another way to approach this problem is to employ an iterative task allocation system, such as the Dias and Stentz (2001) price-based market. The robots would opportunistically exchange tasks over time, thereby modifying their schedules. This idea is demonstrated by the multi-robot exploration system described by Zlot et al. (2002). However, without knowledge of the exact criteria used to decide when and with whom each robot will trade, it is impossible to determine the algorithmic characteristics (including solution quality) of this method.

### 5.2.1. Variant: ALLIANCE Efficiency Problem

Parker (1995) formulated a related MRTA problem called the ALLIANCE efficiency problem (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 an instance of the class of scheduling problems:

$$R \mid\mid C_{max}.$$

Problems in this class are known to be strongly $\mathcal{NP}$-hard (Garey and Johnson 1978). Parker (1995) arrived at the same conclusion regarding the AEP, by reduction from the $\mathcal{NP}$-complete problem PARTITION.

To attack the AEP, Parker (1998) used a learning approach, in which the robots learn both their utility estimates and their scheduling algorithms from experience. When trained for a particular task domain, this system has the potential to outperform the approximation algorithm described above (but it is not guaranteed to do so).

### 5.3. ST–MR–IA: Single-Task Robots, Multi-Robot Tasks, Instantaneous Assignment

Many MRTA problems involve tasks that require the combined effort of multiple robots. In such cases, we must consider combined utilities of groups of robots, which are in general not sums over individual utilities; utility may be defined arbitrarily for each potential group. For example, if a task requires a particular skill or device, then any group of robots without that skill or device has zero utility with respect to that task, regardless of the capabilities of the other robots in the group. This kind of problem is significantly more difficult than the previously discussed MRTA problems, which were restricted to single-robot tasks. In the multi-agent community, the ST–MR–IA problem is referred to as coalition formation, and has been extensively studied (Sandholm and Lesser 1997; Shehory and Kraus 1998).

It is natural to think of the ST–MR–IA problem as splitting the set of robots into task-specific coalitions. A relevant concept from set theory is that of a set partition. A family $X$ is a partition of a set $E$ if and only if the elements of $X$ are mutually disjoint and their union is $E$:

$$
\begin{aligned}
y \bigcap z &= \emptyset \quad \forall y, z \in X, y \neq z \\
\bigcup_{x \in X} &= E
\end{aligned}
\tag{10}
$$

With the idea of partitions in mind, a well-known problem in combinatorial optimization called the (maximum utility) set partitioning problem (Balas and Padberg 1976) is relevant:

DEFINITION 5.    (Set Partitioning Problem (SPP)) Given a finite set $E$, a family $F$ of acceptable subsets of $E$, and a utility function $u : F \to \mathbb{R}_+$, find a maximum-utility family $X$ of elements in $F$ such that $X$ is a partition of $E$.

The ST–MR–IA problem can be cast as an instance of SPP, with $E$ as the set of robots, $F$ as the set of all feasible coalition-task pairs, and $u$ as the utility estimate for each such pair.

Unfortunately, the SPP is strongly $\mathcal{NP}$-hard (Garey and Johnson 1978). Fortunately, the problem has been studied in depth (Atamtürk, Nemhauser, and Savelsbergh 1995), especially in the context of solving crew scheduling problems for airlines (Marsten and Shepardson 1981; Hoffman and Padberg 1993). As a result, many heuristic SPP algorithms have been developed.

It remains to be seen whether such heuristic algorithms are applicable to MRTA problems. Some approximation algorithms, including those of Hoffman and Padberg (1993) and Atamtürk, Nemhauser, and Savelsbergh (1995), have been shown to produce high-quality solutions to many instances of SPP. Even with hundreds of rows/columns and using mid-1990s workstation-class machines, these algorithms require at most tens of seconds to arrive at a solution. With ever-increasing computational power available on robots, it seems plausible that SPP approximation algorithms could be used to solve small-scale and medium-scale instances of the ST–MR–IA problem. To this end, a potentially important question is whether and how these algorithms can be parallelized. Shehory and Kraus (1998) showed how to implement a parallel SPP algorithm for coalition formation in a multi-agent context. Another important point is that, in order to apply certain SPP algorithms to ST–MR–IA problems, it may be necessary to enumerate a set of feasible coalition-task combinations. In the case that the space of such combinations is very large, there is a need to prune the feasible set; pruning can take advantage of sensor-based metrics such as physical distance (e.g., if two robots are more than 50 m apart, then disallow any coalitions that contain them both).

### 5.4. ST–MR–TA: Single-Task Robots, Multi-Robot Tasks, Time-Extended Assignment

The ST–MR–TA class of problems includes both coalition formation and scheduling. For example, consider the problem of delivering a number of packages of various sizes from a single distribution center to different destinations. The number of packages and their destinations are known in advance, as is the size of each package, which determines the number of robots required to carry it. Given a pool of robots, the problem is to build a delivery schedule for the packages, while guaranteeing that a team of the appropriate size is assembled for each package.

To produce an optimal solution, all possible schedules for all possible coalitions must be considered. This problem is $\mathcal{NP}$-hard. If the coalitions are given, with no more than one coalition allowed for each task, the result is an instance of a multiprocessor scheduling problem:

$$MPTm \mid\mid \sum w_j C_j.$$

Even with two processors ($MPT2 \mid\mid \sum w_j C_j$), this problem is strongly $\mathcal{NP}$-hard (Hoogeveen, van del Velde, and Veltman 1994), as is the unweighted version ($MPT2 \mid\mid \sum C_j$;

Cai, Lee, and Li 1998). With three processors, the maximum finishing time version ($MPT3 \mid\mid C_{max}$) is also strongly $\mathcal{NP}$-hard (Hoogeveen, van del Velde, and Veltman 1994).

A means of treating ST–MR–TA is to ignore the time-extended component and approximate the problem as an instance of iterated ST–MR–IA. For this purpose, a Greedy approximation algorithm akin to the one given above for the ST–SR–TA problem can be employed. Unfortunately, the quality of such an approximation is difficult to determine. Another approach is to employ a leader-based mechanism to dynamically form coalitions and build task schedules for them, as described by Dias and Stentz (2002). However, the performance and overhead of this method will also be difficult, if not impossible, to predict without detailed information about the implementation (how many and which robots will be leaders, how does a leader select among candidate coalitions, how long do coalitions persist, etc.).

### 5.5. MT–SR–IA and MT–SR–TA: Multi-Task Robots, Single-Robot Tasks

The MT–SR–IA and MT–SR–TA problems are currently uncommon, as they assume robots that can each concurrently execute multiple tasks. Today's mobile robots are generally actuator-poor. Their ability to affect the environment is typically limited to changing position, so they can rarely execute more than one task at a time. However, there are sensory and computational tasks that fit the MT–SR–IA or MT–SR–TA models quite well.

Solving the MT–SR–IA problem is equivalent to solving the ST–MR–IA problem (see Section 5.3), with the robots and tasks interchanged in the SPP formulation. Likewise, the MT–SR–TA problem is equivalent to the ST–MR–TA problem (see Section 5.4). Thus, the analysis and algorithms provided for the multi-robot task problems also directly apply here to the multi-task robot problems.

### 5.6. MT–MR–IA: Multi-Task Robots, Multi-Robot Tasks, Instantaneous Assignment

When a system consists of both multi-task robots and multi-robot tasks, the result is an instance of the MT–MR–IA problem. For example, consider the allocation of surveillance tasks to a team of robots in an office building. Each robot continuously patrols a fixed portion of the building. Due to computational and/or sensory limitations, each robot can simultaneously detect only a limited number of environmental events (e.g., suspicious person, smoke, open door). Given a set of events to look for, and knowledge about where in the building each event is likely to occur, which robots should be tasked to look for each event?

A relevant concept from set theory is the set cover. A family $X$ is a cover of a set $E$ if and only if the union of elements of $X$ is $E$:

$$\bigcup_{x \in X} = E. \qquad (11)$$

As compared with a partition (see Section 5.3), the subsets in a cover need not be disjoint. A well-known problem in combinatorial optimization called the (minimum cost) set covering problem (Balas and Padberg 1972), is relevant:

DEFINITION 6.    (Set Covering Problem (SCP)) Given a finite set $E$, a family $F$ of acceptable subsets of $E$, and a cost function $c : F \to \mathbb{R}_+$, find a minimum-cost family $X$ of elements in $F$ such that $X$ is a cover of $E$.

The MT–MR–IA problem can be cast as an instance of the SCP, with $E$ as the set of robots, $F$ as the set of all feasible (and possibly overlapping) coalition-task pairs, and $c$ as the cost estimate for each such pair.

Although superficially similar to the SPP, the SCP is in fact a "distant relative", with the solution space of the SCP being far less constrained (Balas and Padberg 1976). The two problems are similar in that the SCP is also strongly $\mathcal{NP}$-hard (Korte and Vygen 2000).

Chvátal (1979) developed a Greedy approximation algorithm for the SCP. The competitive factor for this algorithm is logarithmic in the size of the largest feasible subset (i.e., $\max_{f \in F} |f|$), and the running time is polynomial in the number of feasible subsets (i.e., $|F|$). Bar-Yehuda and Even (1981) present another heuristic set covering algorithm, whose competitive factor is the maximum number of subsets to which any element belongs (i.e., $\max_{e \in E} |\{f \in F : e \in f\}|$), and whose running time is the sum of the sizes of the feasible subsets (i.e., $\sum_{f \in F} |f|$) (Korte and Vygen 2000).

The important trend to note is that these heuristic algorithms perform well when the space of feasible subsets is limited, and that they perform poorly in the most general case of the SCP, with all subsets allowed. For MRTA, this result suggests that such algorithms would best be applied in environments in which the space of possible coalitions is naturally limited, as is the case with heterogeneous and/or physically distantly separated robots. In the case of equally-skilled collocated robots, these algorithms would tend to run slowly and produce poor-quality solutions.

To our knowledge, set covering algorithms have not been applied to MRTA problems, and it is an open question as to whether such an application would be beneficial. However, Shehory and Kraus (1996) successfully adapted and distributed the Chvátal (1979) approximation algorithm for use in multi-agent systems, which suggests that SCP algorithms may indeed be viable for MRTA problems.

### 5.7. MT–MR–TA: Multi-Task Robots, Multi-Robot Tasks, Time-Extended Assignment

We can extend the surveillance domain described in the previous section by specifying that certain events need not be mon-

itored immediately or continuously, but according to some predefined schedule: for example, "the left wing of the building should be checked every hour for open doors". The result is an MT–MR–TA problem, which is an instance of a scheduling problem with multiprocessor tasks and multipurpose machines:

$$MPTmMPMn \, || \, \sum w_j C_j.$$

This problem is strongly $\mathcal{NP}$-hard, because it includes as a special case the strongly $\mathcal{NP}$-hard scheduling problem $MPT2 \, || \, \sum w_j C_j$. We are not aware of any heuristic or approximation algorithms for this difficult problem.

## 6. Analysis of Existing Approaches

Presumably because it is the simplest case of MRTA, the ST–SR–IA problem has received the most attention from the research community. Having developed a formal framework in which to study to this MRTA problem, we can now apply it to an analysis of some of the key task allocation architectures from the literature. In this section, six approaches to the ST–SR–IA problem are analyzed, focusing on the following three characteristics:[2]

1. computation requirements;

2. communication requirements;

3. solution quality.

These theoretical aspects of multi-robot coordination mechanisms are vitally important to their study, comparison, and objective evaluation, as the large-scale and long-term system behavior is strongly determined by the fundamental characteristics of the underlying algorithm(s). We can derive these characteristics for existing architectures by seeing them as solutions to the underlying utility optimization problems that we identified in our taxonomy. First, we explain the methodology used in the analysis.

### 6.1. Methodology

Computational requirements, or running time, are determined in the usual way, as the number of times that some dominant operation is repeated. For the MRTA domain, that operation is usually either a calculation or comparison of utility, and running time is stated as a function of $m$ and $n$, the number of robots and tasks, respectively. Since modern robots have significant processing capabilities on board and can easily work in parallel, in this analysis 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 each robot must select the task with the highest utility, then the running time is $O(n)$, because each robot performs $n$ comparisons, in

---

2. This analysis was originally presented in Gerkey and Matarić (2003).

parallel. Note that this analysis does not measure or consider the actual running time of the utility calculation, in large part because that information is not generally reported. Rather, it is assumed that the utility calculations are computationally similar enough to be meaningfully compared.

Communication requirements are determined as the total number of inter-robot messages sent over the network. In the analysis 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. Further, we assume that a perfect shared broadcast communication medium is used 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 $O(m)$, because each robot makes a single broadcast.

Solution quality is reported as a competitive factor, which bounds an algorithm's performance as a function of the optimal solution (Section 4). The competitive factor for an architecture is determined by mapping its task allocation algorithm onto the underlying assignment problem. For any given task allocation architecture, this mapping could be arbitrarily complex and not necessarily informative. Fortunately, existing MRTA architectures tend to implement either the Greedy algorithm or a close variant. By identifying the allocation algorithm as such, we can put a lower bound on its performance, and thus gain some insight into how the architecture can be expected to perform, independent of the particular application domain.

### 6.2. Results and Discussion

Six MRTA architectures that have been validated on either physical or simulated robots are analyzed. Three of the architectures solve the iterated assignment problem and the other three solve the on-line assignment problem. While there are a great many architectures in the literature, we have attempted to gather a set of approaches that is representative of the work to date.

Of the iterated assignment architectures, the first is ALLIANCE (Parker 1998), one of the earliest demonstrated approaches to MRTA. This behavior-based architecture allocates tasks by maintaining, for each robot, levels of impatience and acquiescence concerning the available tasks. These motivation factors are combined to form, in effect, a utility estimate for each (robot, task) pair. Another behavior-based architecture is BLE (Werger and Matarić 2001), which is a distributed version of the well-known Subsumption Architecture (Brooks 1986). As described in Section 5.1.1, BLE works as follows. At a fixed rate (1 Hz), the robots compute and broadcast to each other their utility estimates for all tasks; allocation is performed after each broadcast with the Greedy algorithm. Another architecture that employs the Greedy algorithm is M+ (Botelho and Alami 1999), whose use of auctions represents the first market-based approach to MRTA (or at least the first that was motivated with economic ideas). Reassign-

ment of tasks is allowed in all three architectures, although the frequency of reassignment may vary. For example, in BLE reassignment occurs almost continuously, but in M+ reassignment occurs only when a new task becomes available.

Of the on-line assignment architectures, the first is MURDOCH (Gerkey and Matarić 2002b), which uses a first-price auction to assign each task, and does not allow reassignment. As stated in Section 5.1.2, this approach is also an implementation of the Greedy algorithm. The two remaining architectures, from Dias and Stentz (2001) and Chaimowicz, Campos, and Kumar (2002), also assign tasks with first-price auctions, but allow (in some circumstances) later reassignment.

Tables 1 and 2 summarize the results for the iterated assignment architectures and on-line assignment architectures, respectively. Perhaps the most significant trend in these results is how similar the architectures look when examined in the manner. For example, the iterated architectures listed in Table 1, which assign all available tasks simultaneously, exhibit almost identical algorithmic characteristics. Only the ALLIANCE architecture (Parker 1998) shows any difference; in this case the decrease in communication overhead is achieved by having each robot internally model the fitness of the others, thereby effectively distributing the utility calculations. More striking are the results in Table 2, which lists architectures that assign tasks in a sequential manner: with respect to computational and communication requirements, these architectures are identical. In terms of solution quality, the approaches of Dias and Stentz (2001) and Chaimowicz, Campos, and Kumar (2002), which allow reassignment of tasks, can potentially perform better than MURDOCH.

These results are particularly interesting because they suggest that there is some common methodology underlying many existing approaches to MRTA. This trend is difficult or impossible to discern from reading the technical papers describing the work, as each architecture is described in different terms, and validated in a different task domain. However, with the analysis described here, fundamental similarities of the various architectures become obvious. These similarities are encouraging because they suggest that, regardless of the details of the robots or tasks in use, the various authors are all studying a common, fundamental problem in autonomous coordination. As a corollary, there is now a formal grounding for the belief that these ad hoc architectures may have properties that allow them to be generalized and applied widely.

Of course, the described analysis does not capture all relevant aspects of the systems under study. For example, in the ALLIANCE architecture, the robots' computational load is increased to handle modeling of other robots, but this analysis does not consider that extra load. Such details, which are currently not widely discussed in the literature, will likely become more important as the field moves toward improved cross-evaluation of solutions.

In addition to enabling evaluation, this kind of analysis can be used to explain why certain solutions work in practice.

**Table 1. Summary of Selected Iterated Assignment Architectures for MRTA**

| Name | Computation/ Iteration | Communication/ Iteration | Solution Quality |
|---|---|---|---|
| ALLIANCE[a] (Parker 1998) | $O(mn)$ | $O(m)$ | at least 2-competitive |
| BLE (Werger and Matarić 2001) | $O(mn)$ | $O(mn)$ | 2-competitive |
| M+ (Botelho and Alami 1999) | $O(mn)$ | $O(mn)$ | 2-competitive |

[a] In addition to solving the ST–SR–IA problem, the ALLIANCE architecture is also capable of building time-extended task schedules in order to solve a form of the ST–SR–TA problem (see Section 5.2.1).
*Note.* Shown here for each architecture are the computational and communication requirements, as well as solution quality.

**Table 2. Summary of Selected Online Assignment Architectures for MRTA**

| Name | Computation/ Task | Communication/ Task | Solution Quality |
|---|---|---|---|
| MURDOCH (Gerkey and Matarić 2002b) | $O(1)$ / bidder $O(n)$ / auctioneer | $O(n)$ | 3-competitive |
| First-price auctions (Dias and Stentz 2001) | $O(1)$ / bidder $O(n)$ / auctioneer | $O(n)$ | at least 3-competitive |
| Dynamic role assignment (Chaimowicz, Campos, and Kumar 2002) | $O(1)$ / bidder $O(n)$ / auctioneer | $O(n)$ | at least 3-competitive |

*Note.* Shown here for each architecture are the computational and communication requirements, as well as solution quality.

For example, the on-line assignment architectures listed in Table 2 are all economically-inspired, built around task auctions. The designers of such architectures generally justify their approach with a loose analogy to the efficiency of the free market as it is used by humans. With a formal analysis, it is possible to gain a clearer understanding of why auction-based allocation methods work in practice. Specifically, is well known that synthetic economic systems can be used to solve a variety of optimization problems. As explained in Section 5.1, an appropriately constructed price-based market, at equilibrium (i.e., when the prices are such that no two utility-maximizing robots would select the same task), produces optimal assignments. The previously described economically-inspired architectures approximate such a market to varying degrees.

## 7. Other Problems

Although the taxonomy given in the previous sections covers many MRTA domains, several potentially important problems are excluded. Next we describe some problem domains that are not captured by the taxonomy.

### 7.1. Interrelated Utilities

Consider the problem of assigning target points to a team of robots that are cooperatively exploring an unknown environment. Many targets (e.g., the frontiers of Yamauchi 1998) may be known at one time, and so it is possible to build a schedule of targets for each robot. Unfortunately, this problem is not an instance of ST–SR–TA, because the cost for a robot to visit target $C$ depends on whether that robots first visits target $A$ or target $B$. Instead, this problem is an instance of the multiple traveling salesperson problem (MTSP); even in the restricted case of one salesperson, MTSP is strongly $\mathcal{NP}$-hard (Korte and Vygen 2000). If, as is often the case with exploration, it is possible to discover new targets over time, then the problem is an instance of the dynamic MTSP, which is clearly at least as difficult as the classical MTSP.

Given the difficulty of the multi-robot exploration problem, it is not surprising that researchers have not attempted to solve it directly or exactly. A heuristic approximation is offered by Zlot et al. (2002), who use TSP heuristics to build target schedules and derive costs that are used in market-based task allocation architecture. When a robot discovers a new target, it inserts the new target into its schedule, but retains the option of later auctioning the target off to another, closer robot.

The multi-robot exploration problem is an example of a larger class of problems, in which a robot's utility for a task may depend on which other tasks that robot executes. These problems in turn form part of another, more general class of problems in which a robot's utility for a task may depend on which other tasks any robot executes. That is, each robot–

task utility can depend on the overall allocation of tasks to robots. Such interrelated utilities can sometimes be tractably captured with factored partially observable Markov decision processes (POMDPs), assuming that a world model is available (Guestrin, Koller, and Parr 2001).

For mobile robots, this situation can arise any time that physical interference contributes significantly to task performance. For example, consider a multi-robot resource transportation problem in which each robot must choose which of a predetermined number of source-sink roads to travel. The decision of which road to travel should take into account the congestion caused by other robots. Taking the position that interference effects are difficult or impossible to adequately model a priori, Dahl, Matarić, and Sukhatme (2002) developed a reinforcement learning approach to the multi-robot resource transportation problem. The robots do not communicate with each other directly, but rather through physical interactions, with each robot maintaining and updating an estimate of the utility for each available road. This approach was shown to produce higher-quality solutions than those produced without learning, and added no communication overhead.

### 7.2. Task Constraints

In addition to an assumption of independent utilities, our taxonomy also assumes independent tasks. There may instead be constraints among the tasks, such as sequential or parallel execution. In principle, each set of tasks with such constraints could be phrased as a single monolithic task that requires multiple robots. The allocation of these larger tasks could then be described by the presented taxonomy (e.g., ST–MR–IA). Unfortunately, the difficult problem of reasoning about task constraints is not removed, but simply shifted into the utility estimation for each potential multi-robot team. In general, our analysis will not suffice in the presence of constraints among tasks.

Although the topic of job constraints is addressed by the scheduling literature (Brucker 1998), the addition of such constraints generally increases problem difficulty, and tractable algorithms exist for only the simplest kinds of constraints. A possible way to approach this problem is with techniques for dynamic constraint satisfaction (Modi et al. 2001).

## 8. Summary

In the field of mobile robotics, the study of multi-robot systems has grown significantly in size and importance. Having solved some of the basic problems concerning single-robot control, many researchers have shifted their focus to the study of multi-robot coordination. There is by now a plethora of examples of demonstrated coordinated behavior in multi-robot systems, and almost as many proposed coordination architectures. However, despite more than a decade of research, the field so far lacks a theoretical foundation that can explain or predict the behavior of a multi-robot system. Our goal in this paper has been to provide a candidate framework for studying such systems.

The word "coordination" is somewhat imprecise, and has been used inconsistently in the literature. In order to be precise about the problem with which we are concerned, we defined a smaller problem: MRTA. That is, given some robots and some tasks, which robot(s) should execute which task(s)? This restricted problem is both theoretically and practically important, and is supported by the significant body of existing work that focuses on MRTA, in one form or another.

To date, the majority of research in MRTA has been experimental in nature. The standard procedure, followed by a large number of researchers, has been to construct a MRTA architecture and then validate it in one or more application domains. This proof-of-concept method has led to the proposal of many MRTA architectures, each of which has been experimentally validated to a greater or lesser extent, sometimes in simulation and sometimes with physical robots. These research efforts are undeniably useful, as they demonstrate that successful multi-robot coordination is possible, even in relatively complex environments. However, to date it has not been possible to draw general conclusions regarding the underlying MRTA problems, or to establish a prescriptive strategy that would dictate how to achieve task allocation in a given multi-robot system.

We view MRTA problems as fundamentally organizational in nature, in that the goal is to allocate limited resources in such a way as to efficiently achieve some task(s). In this paper we have shown how MRTA problems can be studied in a formal manner by adapting to robotics some of the theory developed in relevant disciplines that study organizational and optimization problems. These disciplines include operations research, economics, scheduling, network flows, and combinatorial optimization.

Using such connections to relevant optimization theory, we have presented in this paper a formal analysis of MRTA problems. We have provided characterizations of a wide range of such problems, in the larger context of a taxonomy. For the easier problems, we have provided provably optimal algorithms that can be used in place of commonly-employed ad hoc or Greedy solutions. For the more difficult problems, we have, wherever possible, provided suggestions toward their heuristic solution. Thus, this work can be used to aid further research into multi-robot coordination by allowing for the formal classification of MRTA problems, and by sometimes prescribing candidate solutions.

The presented MRTA formalism is very general, in that it relies only on domain-independent theory and techniques. Thus, for example, the taxonomy given in Section 5 should apply equally well in multi-agent and multi-robot systems. However, in exchange for such generality, this formalism is only capable of providing coarse characterizations of MRTA

problems and their proposed solutions. Consider the analysis showing that MURDOCH, as an implementation of the canonical Greedy algorithm, is 3-competitive for the on-line assignment problem. This kind of competitive factor gives an algorithm's worst-case behavior, which may be quite different from its average-case behavior. In this respect, the bounds established for existing MRTA architectures, in terms of computational overhead, communication overhead, and solution quality, are relatively loose.

One way to tighten these bounds is to add domain-specific information to the formalism. By capturing and embedding models of how real MRTA domains behave and evolve over time, it should be possible to make more accurate predictions about algorithmic performance. For example, while the classical theory of the OAP makes no assumptions about the nature of the utility matrices that form the input, MRTA problems are likely to exhibit significant structure in their utility values. Far from randomly generated, utility values generally follow one of a few common models, determined primarily by the kind of sensor data that are used in estimating utility. If only "local" sensor information is used (e.g., can the robot currently see a particular target, and if so, how close is it?), then utility estimates tend to be strongly bifurcated (e.g., a robot will have very high utility for those targets that it can see, and zero utility for all others). On the other hand, if "global" sensor information is available (e.g., how close is the robot to a goal location?), then utility estimates tend to be smoother (e.g., utility will fall off smoothly in space away from the goal). A promising avenue for future research would be to characterize this "utility landscape" as it is encountered in MRTA domains, and then classify different MRTA problems according to the shapes of their landscapes, and make predictions about, for example, how well a Greedy assignment algorithm should be expected to work, as opposed to a more costly optimal assignment algorithm.

## Acknowledgments

## References

Agassounon, W. and Martinoli, A. 2002. A macroscopic model of an aggregation experiment using embodied agents in groups of time-varying sizes. *Proceedings of the IEEE Conference on System, Man and Cybernetics (SMC)*, Hammamet, Tunisia, pp. 250–255.

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ.

Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. 1995. The algorithmic analysis of hybrid systems. *Theoretical Computer Science* 138(1):3–34.

Atamtürk, A., Nemhauser, G., and Savelsbergh, M. 1995. A combined lagrangian, linear programming and implication heuristic for large-scale set partitioning problems. *Journal of Heuristics* 1:247–259.

Avis, D. 1983. A survey of heuristics for the weighted matching problem. *Networks* 13:475–493.

Balas, E. and Padberg, M.W. 1972. On the set-covering problem. *Operations Research* 20(6):1152–1161.

Balas, E. and Padberg, M.W. 1976. Set partitioning: A survey. *SIAM Review* 18(4):710–760.

Bar-Yehuda, R. and Even, S. 1981. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms* 2:198–203.

Bertsekas, D.P. 1990. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces* 20(4):133–149.

Botelho, S.C. and Alami, R. 1999. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, pp. 1234–1239.

Brooks, R.A. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2(1):14–23.

Brucker, P. 1998. *Scheduling Algorithms*, 2nd edition. Springer-Verlag, Berlin.

Bruno, J.L., Coffman, E.G., and Sethi, R. 1974. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM* 17(7):382–387.

Cai, X., Lee, C.-Y., and Li, C.-L. 1998. Minimizing total completion time in two-processor task systems with pre-specified processor allocations. *Naval Research Logistics* 45(2):231–242.

Cao, Y.U., Fukunaga, A.S., and Kahng, A. 1997. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots* 4(1):7–27.

Castelpietra, C., Iocchi, L., Nardi, D., Piaggio, M., Scalzo, A., and Sgorbissa, A. 2001. Communication and coordination among heterogeneous mid-size players: ART99. *RoboCup 2000*, LNAI Vol. 2019, P. Stone, T. Balch, and G. Kraetzschmar, editors. Springer-Verlag, Berlin, pp. 86–95.

Chaimowicz, L., Campos, M.F.M., and Kumar, V. 2002. Dynamic role assignment for cooperative robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 293–298.

Chien, S., Barrett, A., Estlin, T., and Rabideau, G. 2000. A comparison of coordinated planning methods for cooperating rovers. *Proceedings of Autonomous Agents*, Barcelona, Spain, pp. 100–101.

Chvátal, V. 1979. A greedy heuristic for the set cover problem. *Mathematics of Operations Research* 4:233–235.

Cormen, T.H., Leiserson, C.E., and Rivest, R.L. 1997. *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Dahl, T.S., Matarić, M.J., and Sukhatme, G.S. 2002. Adaptive spatio-temporal organization in groups of robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 30–October 4, pp. 1044–1049.

Deneubourg, J.-L., Theraulaz, G., and Beckers, R. 1991. Swarm-made architectures. *Proceedings of the European Conference on Artificial Life (ECAL)*, Paris, France, pp. 123–133.

Dertouzos, M.L. and Mok, A.K. 1983. Multiprocessor on-line scheduling of hard-real-time tasks. *IEEE Transactions on Software Engineering* 15(12):1497–1506.

Dias, M.B. and Stentz, A. 2001. A market approach to multirobot coordination. Technical Report CMU-RI-TR-01-26, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Dias, M.B. and Stentz, A. 2002. Opportunistic optimization for market-based multirobot control. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 30–October 4, pp. 2714–2720.

Donald, B., Jennings, J., and Rus, D. 1997. Information invariants for distributed manipulation. *International Journal of Robotics Research* 16(5):673–702.

Dudek, G., Jenkin, M., and Milios, E. 2002. A taxonomy of multirobot systems. *Robot Teams: From Diversity to Polymorphism*, T. Balch and L. Parker, editors. A.K. Peters, Natick, MA, pp. 3–22.

Emery, R., Sikorski, K., and Balch, T. 2002. Protocols for Collaboration, Coordination, and Dynamic Role Assignment in a Robot Team. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 3008–3015.

Fukuda, T., Nakagawa, S., Kawauchi, Y., and Buss, M. 1988. Self-organizing robots based on cell structures – CEBOT. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Victoria, BC, Canada, pp. 145–150.

Gale, D. 1960. *The Theory of Linear Economic Models*. McGraw-Hill, New York.

Garey, M.R. and Johnson, D.S. 1978. "Strong" NP-completeness results: motivation, examples, and implications. *Journal of the ACM* 25(3):499–508.

Gat, E. 1998. Three-layer architectures. *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors. AAAI Press, Menlo Park, CA, pp. 195–210.

Gerkey, B.P. and Matarić, M.J. 2002a. A market-based formulation of sensor-actuator network coordination. *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, Palo Alto, CA, pp. 21–26.

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

Gerkey, B.P. and Matarić, M.J. 2003. Multi-Robot Task Allocation: Analyzing the Complexity and Optimality of Key Architectures. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, pp. 3862–3868.

Guestrin, C., Koller, D., and Parr, R. 2001. Multi-agent planning with factored MDPs. *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, pp. 1523–1530.

Hoffman, K.L. and Padberg, M.W. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Science* 39(6):657–682.

Hoogeveen, J., van del Velde, S., and Veltman, B. 1994. Complexity of scheduling multiprocessor tasks with prespecified processor allocations. *Discrete Applied Mathematics* 55:259–272.

Jennings, J.S. and Kirkwood-Watts, C. 1998. Distributed mobile robotics by the method of dynamic teams. *Distributed Autonomous Robotic Systems 3*, T. Luth, P. Dario, and H. Worn, editors. Springer-Verlag, New York, pp. 47–56.

Kaelbling, L.P., Littman, M.L., and Cassandra, A.R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1–2):99–134.

Kalyanasundaram, B. and Pruhs, K. 1993. Online weighted matching. *Journal of Algorithms* 14:478–488.

Klavins, E. 2003. Communication Complexity of Multi-Robot Systems. *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. Burdick, S. Hutchinson, and K. Goldberg, editors. Springer-Verlag, New York, pp. 275–292.

Korte, B. and Vygen, J. 2000. *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag, Berlin.

Kube, C.R. and Zhang, H. 1993. Collective robotics: from social insects to robots. *Adaptive Behavior* 2(2):189–219.

Kuhn, H.W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1):83–97.

Lerman, K. and Galstyan, A. 2002. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots* 13(2):127–141.

Marsten, R.E. and Shepardson, F. 1981. Exact solution of crew scheduling problems using the set partitioning model: recent successful applications. *Networks* 11:165–177.

Mason, M.T. 1986. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research* 5(3):53–71.

Matarić, M.J. 1992. Designing emergent behaviors: from local interactions to collective intelligence. *From Animals to*

*Animats 2, 2nd International Conference on Simulation of Adaptive Behavior (SAB-92)*, J.-A. Meyer, H. Roitblat, and S. Wilson, editors. MIT Press, Cambridge, MA, pp. 432–441.

Modi, J., Jung, H., Tambe, M., Shen, W.-M., and Kulkarni, S. 2001. A dynamic distributed constraint satisfaction approach to resource allocation. *Principles and Practice of Constraint Programming—CP 2001*, T. Walsh, editor. Springer-Verlag, New York, pp. 685–700.

Murata, T. 1989. Petri nets: properties, analysis, and applications. *Proceedings of the IEEE* 77(4):541–580.

Østergård, E.H., Matarić, M.J., and Sukhatme, G.S. 2001. Distributed multi-robot task allocation for emergency handling. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Wailea, Hawaii, pp. 821–826.

Parker, L.E. 1994. *Heterogeneous Multi-Robot Cooperation*, PhD thesis, MIT EECS Department.

Parker, L.E. 1995. L-ALLIANCE: A mechanism for adaptive action selection in heterogeneous multi-robot teams. Technical Report ORNL/TM-13000, Oak Ridge National Laboratory, Knoxville, TN.

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

Parker, L.E. 1999. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing* 5(1):5–19.

Pearce, D.W., editor. 1999. *The MIT Dictionary of Modern Economics*, 4th edition. MIT Press, Cambridge, MA.

Sandholm, T.W. and Lesser, V.R. 1997. Coalitions among computationally bounded agents. *Artificial Intelligence* 94(1):99–137.

Shehory, O. and Kraus, S. 1996. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. *Proceedings of the International Conference on Multi-Agent Systems (ICMAS)*, Kyoto, Japan, pp. 330–337.

Shehory, O. and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence* 101(1–2):165–200.

Simon, H.A. 2001. *The Sciences of the Artificial*, 3rd edition. MIT Press, Cambridge, MA.

Spletzer, J.R. and Taylor, C.J. 2001. A framework for sensor planning and control with applications to vision guided multi-robot systems. *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, Kauai, Hawaii, pp. 378–383.

Stone, P. and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence* 110(2):241–273.

Vail, D. and Veloso, M. 2003. Dynamic multi-robot coordination. *Multi-Robot Systems: From Swarms to Intelligent Automata*, Vol. II, A. Schultz et al., editors. Kluwer Academic, Dordrecht, pp. 87–98.

Weigel, T., Auerback, W., Dietl, M., Dümler, B., Gutmann, J.-S., Marko, K., Müller, K., Nebel, B., Szerbakowski, B., and Thiel, M. 2001. CS Freiburg: Doing the right thing in a group. *RoboCup 2000*, LNAI Vol. 2019, P. Stone, T. Balch, and G. Kraetzschmar, editors. Springer-Verlag, Berlin, pp. 52–63.

Werger, B.B. and Matarić, M.J. 2001. Broadcast of local eligibility for multi-target observation. *Distributed Autonomous Robotic Systems 4*, L.E. Parker, G. Bekey, and J. Barhen, editors. Springer-Verlag, New York, pp. 347–356.

Yamauchi, B. 1998. Frontier-based exploration using multiple robots. *Proceedings of Autonomous Agents*, Minneapolis, MN, pp. 47–53.

Zlot, R., Stentz, A., Dias, M.B., and Thayer, S. 2002. Multi-robot exploration controlled by a market economy. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 3016–3023.