A survey of multi-robot task allocation

Alejandro R. Mosteo, Luis Montano

Robotics, Perception and Real Time Group

Aragon Institute of Engineering Research (I3A).

University of Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.

Tel. +34-976762707, Fax +34-976762043, e-mail: {amosteo,montano}@unizar.es

Abstract

This report offers an overview of principal approaches to task allocation in the context of mobile robotics, with a strong focus in solutions used in service and field robotics. Popular methodologies are identified and their strengths and shortcomings discussed. Critical properties of algorithms are described and examined in recently proposed architectures. A focus on algorithmic complexity and quality bounds of solutions is kept through all the survey.

Keywords: mobile robotics, planning, task allocation, auctions, distributed systems.

1 Introduction

Robotic teams have several potential advantages over single-robot operations: some problems can be unsolvable for robots on their own, requiring cooperation of several agents; they may be able to complete a mission in less time; cooperative approaches that are more efficiently may exist; finally, they are more resistant to failure thanks to redundancy of components.

The use of multi-robot teams introduces new challenges, and in particular a critical one is how to best take advantage from the variety of available resources. Agents in the environment must communicate and coordinate to determine what jobs have to be done and who is best suited to perform them, according to some work objectives. Interaction with human observers or controllers may be also a factor to be integrated as seamlessly as possible.

Generalized use of wireless communications has broadened the interest in such technologies and has opened the gates for integration with other domains like mobile robotics. Affordable of commodity hardware in both of these fields has created a propitious climate for advancements in a common ground. This kind of research can be found under several labels like Mobile Ad-hoc NETworks (MANETs) or Networked Robotic Systems (NRSs). It may also be seen as a specialized branch of cooperative robotics where the communication is explicit via the wireless medium. Finally, it shares many elements of research with the field of Wireless Sensor Networks (WSNs). The distinguishing characteristics of each approach can be succinctly summarized as:

- MANET: Mobile nodes with ad-hoc routing.
- NRS: Mobile robots and static sensors wirelessly connected.
- WSN: Medium/large mesh networks typically using low-power nodes.

From these salient traits, overlapping research interests are natural.

The fields of application of such convergence of technologies are abundant and notable results can be already found. There are examples in cooperative mapping (Zlot et al., 2002; Burgard et al., 2005), autonomous surveillance of buildings (Gerkey, Thrun and Gordon, 2005), monitoring of forest areas (Lemaire, Alami and Lacroix, 2004) and teamwork in higher RoboCup leagues (Vail and Veloso, 2003), to cite a few. Like the already pervasive Internet, the combination of networked static appliances and mobile robots seems an inevitable path for modern society. This will require the definition of standards and common protocols for interaction between such devices to properly exploit its full potential. Resource allocation strategies making use of these standards will aim to improve group performance beyond that of its individual members in isolation.

This survey addresses the task allocation problem from a point of view of networked robotic teams, its objective being to optimize the utilization of the resources available. These resources can take the form of mobile robots with heterogeneous capabilities depending on the on-board sensors and actuators, fixed sensors already present in the environment, computational-only devices and in general any available networked device. The optimization criteria may be varied and tailored to the missions to perform. For example, one may want to minimize the time to explore some environment (Burgard et al., 2005), or distribute the robots in the best way to maximize the wireless coverage over some area (Batalin and Sukhatme, 2002; Gerkey, Mailler and Morisset, 2006), or establish a real-time wireless chain between two distant points to relay a video stream (Tardioli and Villarroel, 2007). Furthermore, conflicting objectives could arise in a same mission.

It has been pointed (Gerkey and Matarić, 2004a) that this field of research has several similarities with *Distributed Agents* of classical *Artificial Intelligence*. Also, it is not uncommon to see the terms *Agent* and *Robot* being used indistinctly, specially when not all the elements in the team have robotic characteristics (e.g. static sensors). Another inheritance from AI are Hierarchical Task Networks (HTN), used to represent richer levels of task semantics. It has been shown (Zlot and Stentz, 2005) that taking advantage of such information, i.e. addressing planning aspects together with allocation, can be advantageous.

In general, work that systematizes the resource allocation problem has been given preference in this survey. General theories, frameworks and architectures provide abstractions that can be reused across problem domains. This enables the use of generic optimizations, being thus beneficial to any problem that can be modeled with these generic tools. These factors are stressed in this survey whenever applicable.

2 Objectives And Challenges

There are several properties that are desirable in the allocation mechanism of a robotic team. Not all of them are easily achievable, and they often entail trade-offs.

- Mathematical soundness: a well understood underlying mathematical theory gives the advantages emanating from the properties of said theory. We may find quality bounds, known expected time of computation, rate of convergence to solution, and so on (Gerkey and Matarić, 2004b).

- Distributability: understood as the capability to disseminate the work among the available devices, it allows a higher usage ratio of available resources, increased computational capabilities, and faster results (Gerkey, Mailler and Morisset, 2006). It may make the system more vulnerable if there are no additional mechanisms providing redundancy, since the failure of any element might impair the proper working of the whole system.

- Decentralization: the elimination of centralizing elements or bottlenecks provides fault tolerance and resiliency (Gancet et al., 2005). Replicating functionality across entities is a guarantee against the failure of any single element. Making roles transient or transferable (Long, Murphy and Parker, 2003) is another way of achieving decentralization.

- Scalability: it is the property by which a system can properly function with arbitrary problem sizes. It is expected that the amount of networked elements will go up (Konolige et al., 2004) with the adoption of these technologies. Good scalability properties are thus of paramount importance in some domains. One way to achieve scalability is by means of *locality*: only locally available or nearby reachable information is needed for an agent decisional process (Kalra, Ferguson and Stentz, 2005). Local algorithms have the advantage of putting less stress in the network and are in general more scalable.

- Fault tolerance: correct operation when facing partial failures is an always desired property, and is even more relevant when the amount of hardware that is being managed is respectable and occupying a large environment. Seamless integration and removal of agents in the system is another tolerance aspect with implications (Konolige et al., 2004; Long et al., 2005) in real deployments.

- Flexibility: different service missions may be performed in a same environment by using heterogeneous or flexible robots (Sanfeliu, 2008). It is then interesting to invest into methods that allow an easy integration of new agents, goals or other related elements.

- Responsiveness: meaning low latency until achieving good results after (usually large) changes are introduced in the working environment or mission conditions, but also in response to task addition and removal (Dahl, Matarić and Sukhatme, 2003; Scerri et al., 2005; Ferreira, Boffo and Bazzan, 2008), it is directly related with the good performance of the allocation strategies in dynamic environments.

3 Definitions

This section explores some of the fundamental ingredients of allocation strategies, and how they affect the capabilities available when using them. Some of the points discussed herein are more thoroughly analyzed in (Gerkey and Matarić, 2004b).

3.1 Planning *vs* allocation

There is some ambiguity in the use of the terms *planning* and *allocation*. Drawing terminology from (Zlot and Stentz, 2005), some consider that *planning* is the answer to the question “what has to be done”, regardless of how this work will be distributed to the available agents. On the other hand, *allocation* is the answer to “who does it”. Other sources (Dias et al., 2005) prefer to refer to the first question as *task decomposition*, since generally the methodology involves iterative refinement of complex tasks into simpler ones. In that case, planning refers to all stages of the problem.

There are several methodologies for planning (Dias et al., 2005):

- Decompose and then allocate: in this approach (Caloud et al., 1990), the isolation of both stages can simplify the problem, often with the drawback that relevant available information during allocation is not fed back to the decomposition stage.

- Allocate and then decompose: complex tasks are first distributed to agents and they are responsible for the refinement of these tasks (Botelho and Alami, 1999). Further reallocation may be necessary after these refinements.

- Hybrid approaches: the most common approach, in which there is no rigid separation of the stages described (Zlot and Stentz, 2003). Since all relevant information can be considered at once, they are in general more powerful but also more complex.

3.2 Cost models

Several magnitudes have been used to characterize the tasks executed in the robotic domain. These are then used during the allocation or optimization process. The most common ones use these basic concepts:

- Cost: a characterization of the cost that it takes for a robot to execute a task. Examples are time to reach a goal, distance traveled, energy consumed, etc. (Lagoudakis et al., 2005). Special mechanisms may be necessary to model impossible (i.e. infinite cost) tasks.

- Fitness: a characterization of “how well” (Stone and Veloso, 1998) an agent can perform a task. Usually a normalized range is used, in which 0 naturally represents tasks undoable by an agent.

- Reward: a characterization of the gain of completing a task, most often found in market-based approaches (Stentz and Dias, 1999). It acts as an incentive to perform tasks.

- Priority: some characterization of the urgency of completing a task. Depending on how the system is designed, higher priority tasks can preempt all lesser tasks (Antonelli, Arrichiello and Chiaverini, 2005).

A combination of these is generally used to represent the 'value' of a task. The common term to refer to these combinations is *utility*. Examples of commonly found utilities are

$$Utility = Reward - Cost$$
$$Utility = Fitness - Cost$$

Another commonly used trait is to floor the utility so it can never be negative:

$$Utility = max(Reward - Cost, 0)$$

This gives two opposite ways of thinking:

- Minimization: when reasoning purely in terms of costs, higher values are worse and the objective is to minimize the costs resulting from the allocation. This is often found in routing approaches where the bulk of the cost is goal reaching (Lagoudakis et al., 2005).

- Maximization: when thinking in terms of utilities, higher values are better and the objective is one of maximization. These are more naturally found where the underlying problem is also of maximization, e.g. network quality coverage (Gerkey, Mailler and Morisset, 2006) or map accuracy (Yamauchi, 1998).

Another possibility is to associate values not to tasks, but to world states. This is notoriously the case in Markov decision processes (MDP) or partially observable Markov decision processes (POMDP) (Peshkin et al., 2000; Guestrin, Koller and Parr, 2001; Bernstein et al., 2002; Schesvold et al., 2003; Spaan, Gordon and Vlassis, 2006), that apply this well-known mathematical framework to the multi-agent domain. In this case, a discounted (over time) or total (when there are final states) reward is computed by exploring the state space.

3.3 Task models

In certain domains, tasks have a well-defined set of constraints (duration, starting and final conditions). These tasks are prone to sequential execution and the most common terminology used for them is usually simply *jobs* or *tasks*. In other domains (principally in team games) predefined positions such as “attacker” or “defender” are assigned to agents. These can be better referred as *roles* (Dias et al., 2005). As such, roles name a set of related behaviors or recipes for actuation. Behavior-based approaches are more suited to this usage, since they tend to make use of excitation/inhibition algorithms that result in some particular behavior (i.e. role) to be activated to temporarily govern the robot (Batalin and Sukhatme, 2002).

3.4 Execution models

Depending on robot capabilities, the execution of tasks can take several forms (Gerkey and Matarić, 2004b).

- Tasks can require a *single robot* or *multiple robots* to be executed.

- Robots can be able to just perform a *single task* or *multiple tasks* at the same time.

The common combination of tasks requiring single robots which are only able to execute a task at a time is known as *sequential execution model*.

Primitive tasks, as seen in the context of HTNs are defined as a task that cannot be further simplified. Usually, these tasks correspond to the single tasks in the sequential execution model. However, in domains with tasks requiring several robots (e.g. pushing of wide objects), they also correspond to these tasks (Lin and Zheng, 2005). A name found in the literature for this kind of task is *joint task* (Lemaire, Alami and Lacroix, 2004).

3.5 Task constraints

An allocation algorithm has to accommodate tasks carrying several kinds of restrictions on its execution. The most commonly found constraints are:

- Partial ordering, in which a task must be completed *before* or *after* a set of others (but not necessarily immediately so, since in this case there is no planning flexibility). An example would be opening a door before another task can be accomplished in another room.

- Time windows, in which a task must be completed in a given time frame, or before a certain deadline (Larsen, Madsen and Solomon, 2004).

- Coupling, in which two or more tasks must be executed at the same time (e.g. in coordinated box-pushing of wide objects).

- Incompatibility, in which executing one task may preclude or obsolete the execution of others.

As already said, a common and powerful tool that allows the expression of these restrictions are HTNs (Erol, Hendler and Nau, 1994). Problems may exhibit only a subset of these restrictions, simplifying the necessary framework.

3.6 Other constraints

The constraints identified in the previous paragraphs are intrinsic to the tasks describing the problem domain. There are, however, other kind of constraints that can adversely affect the execution of a mission. These constraints arise from the use of limited technologies, or real world constraints.

- Mobility interferences, due to narrow spaces in relation to robot size or numbers (Guerrero and Oliver, 2006; Zuluaga and Vaughan, 2008). This is a problem often neglected in multi-robot simulation that nonetheless becomes apparent in real experiments.

- Network range, due to limited coverage of infrastructure, ad-hoc devices with limited range or a need for line-of-sight (Kalra, Ferguson and Stentz, 2007). A problem that was bypassed initially by assuming ideal communications, it has become one of the critical points to address in large-scale deployments (Hsieh et al., 2008).

3.7 Optimization objectives

Sensible objectives for allocation optimization depend on the mission at hand. Here are described the most relevant ones which are commonly found in the literature. Note that a minimization when using costs has a dual problem of maximization when using utilities. Nonetheless, they are enumerated explicitly.

- *Minimize costs of the worst agent*, also known as the *MinMax* criterion. This criterion is of interest in time critical missions, since it gives the shortest mission execution timespan (Lemaire, Alami and Lacroix, 2004). Its focus is to optimize the execution of the worst performing agent.

- *Maximize utility of the worst agent*, also known as the *egalitarian* criterion in welfare related studies, is the dual optimization of the previous one. In this case we try to optimize the worst performing agent (Mouaddib, 2004).

- *Minimize the sum of individual costs*, also known as minimization of the *TotalSum* or *MinSum*. It is relevant in efficiency contexts, for example optimization of fuel usage in transportation or delivery situations.

- *Maximize the sum of individual utilities*, dual of the previous one, is specially interesting in contexts where finalization of tasks gives some reward, since it gives the best possible allocation in this respect. It is usually found in market-based approaches because of the direct translation of economic profit characteristics (Stentz and Dias, 1999). Stochastic models (Spaan, Gordon and Vlassis, 2006) also try to obtain the maximum utility over a period of time, finite or not.

- *Minimize the average cost per task*, or latency when cost equals time. This objective measures the average time since a task appears in the system until it is completed. It has relevancy in domains where task completion is more important than aggregated global costs, e.g. in rescue situations where time to victim location may be critical.

- *Maximize throughput*, which is the dual of the previous one, and often used in industrial, manufacturing processes.

There are interesting effects to be considered when using some of these criteria. For example, the MinMax objective is only concerned with the worst agent, and hence all costs below this threshold are “hidden” in the final result, unless somehow included in the optimization. By using a pure MinMax optimization process, grossly underperforming behaviors below the critical one may occur. Conversely, in a pure MinSum optimization, the bulk of the work may fall on a single agent, even if there is a number of idle robots. These issues, its root causes not always explicitly identified, are circumvented with additional elements in the cost functions, like the *equity coefficient* in (Lemaire, Alami and Lacroix, 2004) or the overlapping factor in (Simmons et al., 2000).

3.8 Optimization of multiple objectives

It is not frequent to optimize several discrete criteria at the same time. The more common approach is to reduce all interesting aspects to a single value of utility or cost, as previously exposed, and with the associated problem of information loss. (Amigoni and Gallo, 2005) is an example that specifically address this problem. Here, the several utility functions are kept apart, and the optimum is determined in terms of *Pareto optimality*. Pareto-optimal solutions have the property that no criterion can be improved without worsening another one. (This has been also called the *egalitarian* property (Mouaddib, 2004)). Problems satisfying certain conditions (Fudenberg and Tirole, 1991) only have one Pareto-optimal solution (or several that are equivalent). This is called *strong Pareto optimality*.

Pareto solutions are interesting because they simply bypass the problem of combining magnitudes which may have no common units or meaningful conversions between them, reducing the need for "tuning parameters". The problem remains that a Pareto solution does not guarantee any minimums, and thus can be very bad for some of the magnitudes being optimized.

4 Computational Complexity

This section explores the topic of algorithmic complexity in task allocation. Not always discussed when presenting an algorithm or framework, complexity is key to determine scalability properties for an algorithm. Also, pathological worst cases may be identified in which apparently fast algorithms might fail to provide a timely answer.

4.1 Deterministic approaches

The problem is of NP-hard nature (Gerkey and Matarić, 2004b) except for the simplest models. That is, there is no expectancy of finding scalable, quick algorithms to solve the complete problem to optimality. Hence, most practical solutions have to make compromises. Nonetheless, there are known exact results in the P realm worth to be mentioned, which follow.

Optimal Assignation Problem

When there is a maximum number of tasks at most equal to the number of agents (for example, in team games where there is a fixed number of roles to assign) and in a sequential execution context, the problem can be equated to the Optimal Assignment Problem (OAP). This problem can be solved in time $O(n^3)$ by means of the Hungarian Method (Kuhn, 1955) or linear programming. Also, a distributed algorithm using auctions can be constructed that converges to the optimal solution (Bertsekas, 1990) with rates directly proportional to the maximum utility and inversely proportional to the minimum bid increment.

When there are more tasks than agents, OAP can still be used in its iterative form: in each computation, the optimal solution is found with one task per agent, leaving excess tasks unassigned. Assuming enough time elapses between task completions, this iterative algorithm can be used to find a new assignation upon each task completion. However, this approach is unsuitable for tasks that have an implicit cost when prematurely switched (e.g. if a retooling is necessary in a machine, or a complex process of grasping is ongoing).

Experimental results in (Goldman, 2004) suggest that valid performance for real-time assignation using a centralized solver can be expected for up to hundreds of agents/tasks, obviously subject to improvement related with technological Moore's Law (Schaller, 1997) or other technological breakthroughs.

4.2 Greedy algorithms

Greedy algorithms can be trivially constructed in many situations for fast solution finding. The basic idea is to choose at each decision step the best available candidate. Only problems that satisfy certain conditions related with matroid theory (Korte and Vygen, 2000) will be solved to optimality by a greedy algorithm, and this is rarely the case in allocation contexts. It is interesting however to be able to characterize the quality of the greedy solution for a given problem. In this regard, it is useful the concept of α-competitiveness, which means that a given solution will never have an utility less than $1/\alpha$ the one of the optimal solution.

The greedy algorithm for the OAP is 2-competitive, or 3-competitive compared to the *post hoc* optimal solution if new tasks appear on-line and robots cannot be reassigned (in this later case, without a model of the new possible tasks this is the best possible solution (Kalyanasundaram and Pruhs, 1993)). As a consequence of the previous result, in the context of sequential task execution the greedy algorithm exhibits also a 3-competitive performance.

4.3 Brief note on NP-hard models

NP-hard problems cannot be solved efficiently in the general case. A study of models exhibiting this property is available e.g. in (Gerkey and Matarić, 2004b). Not cited there, but subject to the same problem of complexity is the general technique of Distributed Constraint Optimization (DCOP) (Yokoo, 2001), which can be found in task allocation in, for example, the work of (Scerri et al., 2005; Ferreira, Boffo and Bazzan, 2008).

4.4 Stochastic approaches

Comprised here are Markov Decision Processes (MDPs), Partially Observable MDPs (POMDPs), Stochastic Games (SG) and Partially Observable SGs (POSGs). Tentative approaches for tight coordination using the previous techniques are presented in §5.2. An study of complexity is offered in (Pynadath and Tambe, 2002) but suffice it to say that the complexity of these frameworks is humongous (in the order of NP-completeness for centralized approaches, and NEXP-completeness for distributed ones (Bernstein et al., 2002)), so approximation strategies are being actively researched. Two representative frameworks are PEGASUS (Ng and Jordan, 2000) and PERSEUS (Porta, Spaan and Vlassis, 2005; Porta et al., 2006).

4.5 Hierarchical Task Networks

HTNs have been extensively used for planning in artificial intelligence domains in the past. They have been imported to the robotic domain with success (Clement and Durfee, 1999; Goldman et al., 2000; Belker, Hammel and Hertzberg, 2003; Zlot and Stentz, 2003; Gancet and Lacroix, 2004; Zlot and Stentz, 2005). Next follows a description of its characteristics and advantages and a summary of its applications in multi-robot research. An in-depth analytical study of their properties can be found in (Erol, Hendler and Nau, 1994). Results in there show that, except for the simplest HTN (totally ordered tasks and no variables), finding valid plans may range from NP-complete to undecidable complexity.

HTNs are a particular kind of network, organized as a tree data structure. The most salient component of a HTN are tasks, that are linked by means of *parent-child* relationships. As such, sibling child tasks are a refined, more-detailed description of the execution of its parent. Tasks that are leaves are called *primitive* and completing execution of these tasks implies the completion of its parents. Tasks that are not leaves are called *abstract* or *compound* tasks.

The domain knowledge is embedded not only in the tasks but principally in the *methods*, which are algorithms that, given an abstract task, can generate the required child tasks to complete the abstract task. These children need not be primitive, but eventually some descendants have to be for the network to be executed, since agents are only able to execute primitive tasks.

Important benefits of HTNs are

- Simple visualization, that allows human operators to supervise and provide plans in an intuitive way.
- Generic mechanisms to manage domain specific knowledge, pluggable in the form of tasks and methods.

- Abstraction by means of top-bottom analysis given abstract goals, and reciprocally bottom-top construction starting at the agent capabilities.
- Abundance of research, implementations (Nau et al., 1999; Goldman, 2004; Nau et al., 2004) and applications.

Some examples of application follow. HTNs were used in (Zlot and Stentz, 2003; Zlot and Stentz, 2005) to allow complex auctions of not only single tasks but partial subplans, augmenting in a structured way the auction mechanism. They are used to autonomously generate executable plans bridging the gap between abstract, symbolic plans and the sensed data in (Gancet and Lacroix, 2004). In (Clement and Durfee, 1999) they are used for coordination and conflict avoidance at abstract levels of reasoning by using *inconditions*, that is, invariants that must hold *during* task execution. In (Goldman et al., 2000) the key ingredient is human interaction: HTNs are used as an effective way to quickly instruct teams of mobile robots, rapidly and accurately tailoring existing plans to novel situations.

5 Solution Models

This section presents different solutions to the task allocation problem, grouped by the kind of strategies used, in no particular order.

5.1 Centralized models

A solution is said centralized when a single element in the system is responsible for managing all the available resources. Their strong point is that they can use the best known algorithms and usually have more information available than distributed or local algorithms. This strength is in return burdened by the risk of losing contact with the controlling element, introducing a single point of failure. Scalability is jeopardized by the network load required to gather the required inputs and the complexity of the algorithms used. Usually, hybrid characteristics are present (Causse and Pampagnin, 1995) to minimize these problems.

Centralized algorithms in pure form are sometimes seen in real robotic teams, although they are not favored. (Brummit and Stentz, 1998) is an example of a purely centralized mission planner. In (Bowling, Browning and Veloso, 2004) an entire RoboCup team is controlled by a central coordinator, because information is acquired globally by means of a zenithal camera. Central algorithms are also useful during simulation stages of research since all the information is readily available (Rosencrantz, Gordon and Thrun, 2003; Amigoni and Gallo, 2005). Many stochastic frameworks also exhibit centralized traits since, for example, the optimal joint policy must be centrally computed (Gmytrasiewicz and Doshi, 2004) and then relayed to the agents.

Market-based approaches also advocate for temporal and localized centralization as a way to obtain better deals (Dias and Stentz, 2002). Temporal leaders are opportunistically used to improve performance, but in their absence the system would equally function properly.

5.2 Stochastic models

Probabilistic frameworks are a way to provide optimal control in tightly coupled domains. The main obstacle for their use is that they quickly become intractable even for small sized problems. For this reason, approximation techniques are actively being investigated (Ng and Jordan, 2000; Porta, Spaan and Vlassis, 2005).

In the framework of MDPs, there are cooperative agents with perfect knowledge of its environment and a probabilistic model of the effects of its actions. The unrealistic assumption of perfect world observability is lifted in partially observable MDPs, being replaced for some probabilistic model of the sensory inputs. When the environment includes adversarial elements or agents with unknown policies we call them stochastic games (SG) or partially observable SGs.

The principal advantage of these techniques is its promise of providing optimal control in tightly coupled missions with uncertain world dynamics and perceptions. Its drawbacks are, however, numerous: They are computationally very expensive; in most cases include some degree of centralization to compute joint policies; simulation results in very small sized worlds are not straightforward to carry to real-life missions; finding the optimal policies is an iterative process that can be slow and that may require repeated exposure to the environment in which the action takes place.

Other approaches found in task allocation are optimization methods with an stochastic step. Simulated annealing (Gerkey, Mailler and Morisset, 2006) and genetic algorithms (Alba and Dorronsoro, 2004) are two such methods.

5.2.1 Collaborative stochastic frameworks

The first advances (Peshkin et al., 2000) in multi-agent POMDP planning are of limited practical utility, being reduced to very simple models: usually small grid worlds are used, which translation to reality is not apparent. In (Guestrin, Koller and Parr, 2001) factored MDPs are used to make tractable otherwise intractably sized MDPs, and are demonstrated in very small simulated discrete worlds. (Pynadath and Tambe, 2002) provides a generic framework for the study of the complexity of solutions based in this paradigm. It is a good starting point to fathom the huge computational complexity of this kind of approach.

For the explained reasons, (PO)MDPs are more successful in environments where decisions are discrete and to be taken amongst few options. For example, a role assignation solution is presented in (Spaan and Groen, 2003), where the small number of roles and agents (three in each case) makes it suitable for a MDP approach. In (Schesvold et al., 2003) a combat unmanned air vehicle (UAV) has to choose between search or strike actions and a POMDP is used to compute the policy. An exact algorithm has been developed for the solving of POMDPs in (Hansen, Bernstein and Zilberstein, 2004), although it is practical for very small problems only. Preliminary results for the PERSEUS framework are shown in (Spaan, Gordon and Vlassis, 2006), where communication is explicitly integrated in the planning framework of a decentralized POMDP solved by approximation, although experimentation is limited to small simulated grid worlds.

5.2.2 Adversarial stochastic frameworks

Relevant concepts in the context of stochastic games are

- Nash equilibrium: all competing agents have one (or several) policies such no agent can improve its reward if the rest remain stationary in its policies.

- Rationality: if adversary policies are stationary, a learning algorithm will be rational if it converges to a best-response policy for those adversarial policies.

- Convergence: a learning algorithm has this property if it will necessarily converge to some final policy. It may depend in the algorithms used by adversarial agents.

In (Bowling and Veloso, 2002) results are presented to achieve both rationality and convergence and this is demonstrated in a small grid world. The Nash equilibrium objective is relaxed in (Gmytrasiewicz and Doshi, 2004), being argued that it is an incomplete assumption because no clear way to choose amongst several equilibria exists, and periods off-equilibrium must be managed anyway. Experimentation with real robots is presented in (Emery-Montemerlo et al., 2005), where a POSG is used for a pursuit problem using two robots. Clustering techniques are used to make the problem tractable.

5.3 Auction models

This section summarizes (or hints at, since there is a very large body of research) results achieved using auction techniques. These are characterized by the use of a network protocol for the purpose of some economic-like negotiation between agents. Sometimes the negotiation closely resembles a real auction of goods and these solutions can be also referred as being market-based. However, this is not always the case and the negotiation can follow other patterns, so the term auction-based is used for generality.

The simplest model for auction proposals is a three-step negotiation:

- Step 1: a task is published to the agents by some entity, usually called *auctioneer*. In many occasions the auctioneers are the agents themselves.
- Step 2: agents suited to the tasks reply to the auctioneer with a *bid* on the task.
- Step 3: the auctioneer awards the task to some agent, after evaluating the received bids.

Most variations specify additional rules or steps that provide some advantage over the basic algorithm, as explained later. The advantages of this approach are several and this is manifested in the abundance of research and experimentation with real robots:

- Simplicity: the idea behind most auction-based protocols is simple and intuitive. Implementation is also not difficult for the basic algorithms.
- Flexibility: as long as tasks can be represented by some utility value, the system is able to manage them.
- Expandability: the basic idea is prone to modifications for improved performance.
- Fault tolerance: as long as the seller keeps track of its sold tasks, they can be resold in the even of failure of the actual owner.

The basic idea behind most auction-based algorithms can be already found in (Smith, 1980) in the 80s, and even there the roots can be traced back in time to the 70s. In this seminal paper, nodes providing tasks to be solved are called *managers*, and nodes bidding for them are called *contractors*. The contract net is exemplified with distributed computers and sensors instead of mobile agents, but the ideas remain the same. (Wellman, 1997) provides a good argumentation for the use of market-aware software agents, which can be seen as the intermediate step towards proper robots. (Stentz and Dias, 1999) is a representative report where a free market is advocated to control mobile robots under the premise that by maximizing individual profits, the global plan profits are maximized. This is an exploratory paper where many ideas are hinted without further development: already there are the ideas of combinatorial auctions, opportunistic centralization and some forms of learning. A first simulated implementation appears in (Thayer et al., 2000) in a mission of multi-robot exploration.

Combinatorial auctions are implemented in (Hunsberger and Grosz, 2000) for the purpose of role assignation and synchronized task execution. In this case, agents bid for *combinations* of tasks they are able to carry. The auctioneer must then decide the best combination of bidders for the mission at hand. It is noted that this is a NP-hard problem, but manageable for bid numbers in the order of thousands.

Opportunistic optimization via leaders is studied in (Dias and Stentz, 2002). Here, a leader negotiates with groups of robots to centrally reason about their assigned tasks and reallocate them in more profitable assignations.

The first (to our knowledge) *embodied* implementation using real robots appears in (Gerkey and Matarić, 2002), where the terminology used is *publish/subscribe*. Proper working is demonstrated in a coordinated box-pushing task and in a randomly generated alarm-like uncoupled tasks scenario. Another robotic demonstration is shown in (Zlot et al., 2002) in a distributed mapping mission. The implementation used therein, named *TraderBots*, is subsequently improved in (Zlot and Stentz, 2003) to use HTNs as a way to distribute planning and manage combinatorial auctions. In these upgraded TraderBots, bids can be for task trees at any depth level, hence allowing for more solutions to be explicitly evaluated and bought. TraderBots are examined again in the light of three kinds of robot malfunction in (Dias et al., 2004) with encouraging results.

Tight coordination is summarily addressed in (Kalra and Stentz, 2003), and more exhaustively explored in (Kalra, Ferguson and Stentz, 2005) with a market-based framework named *Hoplites*. Passive and active coordination strategies are demonstrated in real robots performing a security sweep in a corridor with some obstacles.

A form of learning is introduced via *opportunity cost* in (Schneider et al., 2005). This opportunity cost reflects the expected earnings per second of a robot. This is used to modify the auction mechanism so robots try to maximize this opportunity cost. This has applications in time-discounted environments, where tasks lose value as time goes by.

In (Bererton et al., 2003), a departure from purely economically inspired background is found, where the auctions are designed to distributively solve an approximated MDP, thus uniting the auction and stochastic paradigms.

An excellent survey on market-based techniques is (Dias et al., 2005). Another milestone is the work in (Lagoudakis et al., 2004; Lagoudakis et al., 2005; Tovey et al., 2005), which addresses computational complexity and cost bounds in auction-based allocation for multi-robot routing.

5.3.1 Optimality

Early implementations lack a detailed study about the optimality of solutions found, besides the intuitive idea that robots try to maximize profit. Furthermore, without a proper characterization of auction timings and task commitments not much can be said about mathematical bounds, even if experimental results are satisfactory. This was partially addressed in the work (Gerkey and Matarić, 2003), where auctions from (Gerkey and Matarić, 2002) are characterized as a distributed implementation of a greedy assigner, consequently exhibiting 3-competitiveness.

Other implementations like (Zlot et al., 2002) do not strictly follow the greedy algorithm, allowing for the continuous bidding of already assigned tasks. This presumably improves efficiency, but it is not clear if new bounds can be established.

In (Lagoudakis et al., 2004) an auction-based method is presented that is 2-competitive in static environments. This is not surprising since this is the bound already known for greedy allocations in observable contexts. Further theoretical results about auction-based allocation for the vehicle routing problem appear in (Lagoudakis et al., 2005; Tovey et al., 2005).

5.4 Behavioral models

Behaviors are patterns of actuation that are embedded into agents and that are enabled or disabled in response to certain stimuli. When a behavior is activated, it will motivate the robot to perform certain preprogrammed actions. Usually several behaviors can be active at the same time, so they must have rules for prioritization or combination.

Ad-hoc solutions for some multi-robot domains have been proposed using behaviors. They are easily applicable to formations. A good example with explicit descriptions of behavior rules is found in (Balch and Arkin, 1998). Coverage is tackled in (Batalin and Sukhatme, 2002), where robot spreading is achieved with a repulsive behavior combined with an exploratory one that is attracted to unknown spaces. Potential field based reactions are used in a RoboCup setting in (Mitsunaga, Izumi and Asada, 2003). In (Hayes, McJunkin and Kosecka, 2003), the foraging task is performed using behaviors inspired by ant pheromone trails.

More relevant to resource allocation is the work (Parker, 1998), where the ALLIANCE architecture is presented, with a strong emphasis on fault tolerance. Low level behaviors implement the usual obstacle-avoidance techniques, while high level ones are used to perform the task allocation. The two critical elements are *impatience* –a robot gets impatient if he sees a task that nobody is executing, thus triggering its adequate behavior to perform it– and *acquiescence* –that makes a robot to relinquish a task if it detects that its performance is below expectancies. Robots broadcast periodically its current commitments to affect the behaviors of nearby teammates. L-ALLIANCE is an extension presented in (Parker, 1997) where the L stands for learning. The rates of impatience and acquiescence are dynamically adapted by reacting to the context and environment.

Another architecture using a behavior-based approach is Broadcast of Local Eligibility (BLE) (Werger and Matarić, 2000). In it, robots broadcast their fitness ("eligibility") for a task, and inhibit the behaviors of peers with less eligibility, thus claiming the tasks.

MONAD (Vu et al., 2003) aims to provide a flexible architecture with off-line scripting capabilities for team design and programming, while relying in behavior-based techniques for the on-line control of the team. One intended goal is to allow for quick implementation of different team control architectures. The architecture is demonstrated in a robotic soccer domain.

6 Architectures

There is notable early research in architectures for intelligent systems in, e.g., (Saridis, 1979), which advocates a three-level hierarchy based on the principle of *increasing precision with decreasing intelligence*. Task allocation would fit typically into the organization level therein described, above the coordination and execution levels. This section reviews in chronological order some relevant architectures which aim to provide a general framework for task allocation. They have been already mentioned in the respective general sections when appropriate. Optimality and network usage results are extracted from (Gerkey, 2003) or their presentation papers.

6.1 ALLIANCE (1997)

This behavior-based architecture has been described in §5.4 and exhibits 2-competitiveness in the case of instantaneous assignment (i.e., all tasks are known at execution start). Bandwidth usage is in the order of Robots times Tasks.

6.2 GRAMMPS (1998)

The GeneRAlized Mission planner for MultiPle mobile robotS is a "field-capable system which couples a general-purpose interpreted grammar for task definition with dynamic planning techniques" (Brummit and Stentz, 1998). It notably includes planners for the TSP and MTSP problems but, as these are intractable for large sizes, it also includes a simulated annealing route planner. It is thus an early example of hybrid approach, although it still depends on a Central Mission Planner.

6.3 Broadcast of local eligibility (BLE, 2000)

Seen in §5.4, this behavior-based architecture is 2-competitive when working with instantaneous assignments. Bandwidth usage is in the order of Robots times Tasks.

6.4 Murdoch (2002)

This first implementation in embodied robots of auction-based techniques has been described in §5.3. The quality of its produced allocations is 3-competitive for the on-line tasks problem (tasks that arrive at random after execution). Computational costs are very low and network resources used are lineal on the number of robots.

6.5 TraderBots (2002)

Seen in §5.3, this is probably the most representative implementation of early auction-based techniques. It is expected a 3-competitiveness or better, since the reallocation introduced can only best any allocation produced by Murdoch.

6.6 MONAD (2003)

Seen in §4.3, its main contribution is the flexibility of its scripting off-line description language that allows quick tailoring of new teams. However, it lacks (Vu et al., 2003) fault tolerance capabilities.

6.7 COMETS (2004)

This project exhibits a comprehensive architecture which makes use of four planning and allocation modules and five levels of decisional autonomy. The four modules can be switched progressively from centralized to distributed operation, which configures a incremental delegation of decisional capabilities from the operator to the agents (which are UAVs in this case). Also, a symbolic planner is used for reasoning capabilities.

The allocation scheme is derived from the Contract-Net protocol, from which a low computational complexity is inferred. However, in order to optimize the MinMax objective instead of the MinSum one, an *equity coefficient* is introduced. The authors do not discuss any competitiveness bounds, although they find that this coefficient improves the results for their particular objective.

6.8 Hoplites (2005)

This market-based framework, seen in §5.3, addresses domains in which actions of robots are tightly coupled. The uncertainty inherent in these tasks also necessitates persistent tight coordination between teammates throughout execution, that is, active and high-frequency communication. There is no complexity discussion in the original paper (Kalra, Ferguson and Stentz, 2005), and furthermore the planning components are left open by the authors, since they depend on the problem at hand. These issues are addressed in a later work (Kalra, Ferguson and Stentz, 2007).

6.9 PERSEUS (2005)

PERSEUS is a probabilistic framework which uses approximation techniques in order to solve a POMDP problem. The issues that these problems present for exact solving have been introduced in §5.2, and PERSEUS proposal is to use a randomized point-based value iteration algorithm. This requires random sampling and thus introduces uncertainties on the actual quality of the solution over the optimal policy, which are not discussed by the authors (Spaan and Vlassis, 2005), although they note several advantages of their approach over other approximations, which allow the use of a greater number of points in the belief set.

6.10 URUS (2008)

The Ubiquitous networking Robotics in Urban Settings (URUS) project (Sanfeliu, 2008) defines a standard set of services to be provided by any URUS robot. This allows the integration of any robot that implements them into the networked robotic system (NRS). The allocation architecture of the URUS project comprises two principal subsystems: a centralized (although replicated in each robot) simulated annealing planner, and an auction-based allocator.

The stochastic nature of the first subsystem precludes the guarantee of any quality bounds, although its complexity per step, due to the nature of the URUS missions, is constant, involving only task permutations. This algorithm is used as an opportunistic optimizer of known plans, and to quickly integrate large task sets into the system.

The auction-based allocator makes use of a similar approach to that of Murdoch or TraderBots, and hence 3-competitiveness is the expected quality bound. Its complexity per bid is linear on the number of tasks, although more expensive heuristics like 2-op movements (Reinelt, 1994) can be used when the robot plan is small.

7 Discussion

We have seen an overview of the principal aspects of multi-robot task allocation from a service robotics point of view. Firstly, common terminology and requirements have been presented. These requirements are often in conflict with each other, forcing solutions to make compromises in one or other direction, and to embrace hybrid or richer solutions.

Algorithmic complexity is one of the critical issues in task allocation, since most problems are of NP-hard nature. Distributed algorithms may also raise the complexity, or transfer it from computation to network load. Authors not always make an effort to identify the complexity of their solutions. Selected studies have focused in this shortcoming, identifying the nature of typical problem models. Consequently, known significant results have been highlighted.

Several main trends of solution have been described. Initial centralized planning approaches have given way to distributed ones over time, although many architectures retain centralized planners for opportunistic use. Stochastic frameworks start from a centralized learning process, so efforts have been devoted to their decentralization. In other direction, auction-based solutions are a strong current in task allocation, which started with simple market analogies to later incorporate sophistications like combinatorial auctions, hierarchical bidding, and learning of parameters that affect the bidding process. Finally, behavioral solutions have been proposed which are notable by their resiliency against individual robot failures. However, it has been noted that there is a functional equivalence between some behavioral and auction-based architectures.

Finally, a historical review of generic, reusable architectures for task allocation has been presented. We can differentiate between the ones devoted to proof-of-concept, which usually make use of a single algorithm or strategy to be demonstrated, and hybrid ones, in which several planning and allocation components are present. These tend to be more recent, as the field matures and simple novel ideas are exhausted.

8 Acknowledgements

This research was supported by the I3A Fellowship Program, the Network Robot Systems EURON II Network Research Atelier (NoE-507728 RA) and the project EXPRES (técnicas de EXPloración avanzadas en tareas de REScate, DPI2003-07986).

9 References

Alba, E. & Dorronsoro, B. (2004). *Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms*, Lecture Notes in Computer Science 3004: 11-20.

Amigoni, F. & Gallo, A. (2005). *A Multi-Objective Exploration Strategy for Mobile Robots*. In: *IEEE Int. Conf. in Robotics and Automation (ICRA'05)*, pp. 3861-3866.

Antonelli, G.; Arrichiello, F. & Chiaverini, S. (2005). *The null-space-based behavioral control for mobile robots*. In: *IEEE Computational Intelligence in Robotics and Automation (CIRA'05)*, pp. 15-20.

Balch, T. & Arkin, R. C. (1998). *Behavior-based Formation Control for Multi-robot Teams*, IEEE Transactions on Robotics and Automation 14(6).

Batalin, M. A. & Sukhatme, G. S. (2002). *Spreading Out: A Local Approach to Multi-robot Coverage*. In: *6th Int. Symp. on Distributed Autonomous Robotics Systems (DARS'02)*.

Belker, T.; Hammel, M. & Hertzberg, J. (2003). *Learning to Optimize Mobile Robot Navigation Based on HTN Plans*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*.

Bererton, C.; Gordon, G.; Thrun, S. & Khosla, P. (2003). *Auction Mechanism Design for Multi-Robot Coordination*. In: *Conf. on Neural Information Processing Systems (NIPS)*.

Bernstein, D. S.; Givan, R.; Immerman, N. & Zilberstein, S. (2002). *The Complexity of Decentralized Control of Markov Decision Processes*, Math. Oper. Res. 27(4): 819-840.

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

Botelho, S. & Alami, R. (1999). *M+: A Scheme for Multi-Robot Cooperation Through Negotiated Task Allocation and Achievement.*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'99)*, pp. 1234-1239.

Bowling, M.; Browning, B. & Veloso, M. (2004). *Plays as effective multiagent plans enabling opponent-adaptive play selection*. In: *Int. Conf. on Automated Planning and Scheduling (ICAPS'04)*.

Bowling, M. H. & Veloso, M. M. (2002). *Multiagent learning using a variable learning rate*, Artificial Intelligence 136(2): 215-250.

Brummit, B. & Stentz, A. (T. (1998). *GRAMMPS: A Generalized Mission Planner for Multiple Mobile Robots*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'98)*.

Burgard, W.; Moors, M.; Stachniss, C. & Schneider, F. (2005). *Coordinated Multi-Robot Exploration*, IEEE Transactions on Robotics 21(3): 376-378.

Caloud, P.; Cho, W.; Latombe, J. C.; Pape, C. L. & Yim., M. (1990). *Indoor automation with many mobile robots*. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'90)*.

Causse, O. & Pampagnin, L. H. (1995). *Management of a multi-robot system in a public environment*. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'95)*, pp. 2246.

Clement, B. J. & Durfee, E. H. (1999). *Top-down search for coordinating the hierarchical plans of multiple agents*. In: *3rd Conf. on Autonomous Agents*, pp. 252-259.

Dahl, T. S.; Matarić, M. J. & Sukhatme, G. S. (2003). *Multirobot task-allocation through vacancy chains*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, pp. 2293-2298.

Dias, M. B. & Stentz, A. (T. (2002). *Opportunistic Optimization for Market-Based Multirobot Control*. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'02)*.

Dias, M. B.; Zinck, M. B.; Zlot, R. M. & Stentz, A. (T. (2004). *Robust Multirobot Coordination in Dynamic Environments*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*.

Dias, M. B.; Zlot, R. M.; Kalra, N. & Stentz, A. (T. (2005). *Market-based multirobot coordination: a survey and analysis*, Carnegie Mellon Technical Report CMU-RI-TR-05-13.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J. & Thrun, S. (2005). *Game Theoretic Control for Robot Teams*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, pp. 1175-1181.

Erol, K.; Hendler, J. A. & Nau, D. S. (1994). *HTN Planning: Complexity and Expressivity*. In: *Conf. on Artificial Intelligence (AAAI'94)*, pp. 1123-1128.

Ferreira, P.; Boffo, F. & Bazzan, A. (2008). In: Jamali, N.; Scerri, P. & Sugawara, T. (Ed.), *Using Swarm-GAP for Distributed Task Allocation in Complex Scenarios*, Springer Berlin / Heidelberg.

Fudenberg, D. & Tirole, J., 1991. *Game theory*. MIT Press, Cambridge, MA.

Gancet, J.; Hattenberger, G.; Alami, R. & Lacroix, S. (2005). *Task Planning and Control for a multi-UAV system: architecture and algorithms*. In: *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*.

Gancet, J. & Lacroix, S. (2004). *Embedding heterogeneous levels of decisional autonomy in multi-robot systems*. In: *Int. Symp. on Distributed Autonomous Robotic Systems (DARS'04)*.

Gerkey, B. P. (2003). *On Multi-Robot Task Allocation*, University of Southern California.

Gerkey, B. P.; Mailler, R. & Morisset, B. (2006). *Commbots: Distributed control of mobile communication relays*. In: *AAAI Workshop on Auction Mechanisms for Robot Coordination (AuctionBots)*, pp. 51–57.

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

Gerkey, B. P. & Matarić, M. J. (2003). *Multi-robot task allocation: analyzing the complexity and optimality of key architectures.*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, pp. 3862-3868.

Gerkey, B. P. & Matarić, M. J. (2004a). *Are (explicit) multi-robot coordination and multi-agent coordination really so different?*. In: *AAAI Spring Symposium on Bridging the Multi-Agent and Multi-Robotic Research Gap*, pp. 1-3.

Gerkey, B. P. & Matarić, M. J. (2004b). *A formal analysis and taxonomy of task allocation in multi-robot systems*, Int. J. of Robotics Research 23(9): 939-954.

Gerkey, B. P.; Thrun, S. & Gordon, G. (2005). *Parallel stochastic hill-climbing with small teams*, Multi-Robot systems: from swarms to intelligent automata 3: 65-77.

Gmytrasiewicz, P. & Doshi, P. (2004). *A framework for sequential planning in multi-agent settings*. In: *Int. Symp. on Artificial Intelligence and Mathematics*.

Goldman, R. P. (2004). *Adapting Research Planners for Applications*. In: *Int. Conf. on Automated Planning and Scheduling (ICAPS'04)*.

Goldman, R. P.; Haigh, K. Z.; Musliner, D. J. & Pelican, M. J.S.. (2000). *MACBeth: A Multi-Agent Constraint-Based Planner*. In: *AAAI'00 Workshop on Constraints and AI Planning, tech. report WS-00-02, AAAI*, pp. 11-17.

Guerrero, J. & Oliver, G. (2006). *Physical interference impact in multi-robot task allocation auction methods*. In: *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*.

Guestrin, C.; Koller, D. & Parr, R. (2001). *Multiagent Planning with Factored MDPs*. In: *Advances in Neural Information Processing Systems (NIPS-14)*.

Hansen, E.; Bernstein, D. & Zilberstein, S. (2004). *Dynamic programming for partially observable stochastic games*. In: *National Conf. on Artificial Intelligence (AAAI'04)*.

Hayes, J.; McJunkin, M. & Kosecka, J. (2003). *Communication Enhanced Navigation Strategies for Teams of Mobile Agents*. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, pp. 2285-2290.

Hsieh, M. Ani.; Cowley, A.; Kumar, R. Vijay. & Taylor, C. J. (2008). *Maintaining network connectivity and performance in robot teams*, Journal of Field Robotics 25(1--2): 111-131.

Hunsberger, L. & Grosz, B. J. (2000). *A Combinatorial Auction for Collaborative Planning*. In: *Int. Conf. on MultiAgent Systems (ICMAS'00)*, pp. 151.

Kalra, N.; Ferguson, D. & Stentz, A. (2007). *A Generalized Framework for Solving Tightly-coupled Multirobot Planning Problems*. In: *Int. Conf. on Robotics and Automation (ICRA'07)*, pp. 3359-3364.

Kalra, N.; Ferguson, D. & Stentz, A. (T. (2005). *Hoplites: a market-based framework for planned tight coordination in multirobot teams*. In: *Int. Conf. on Robotics and Automation (ICRA'05)*, pp. 1182-1189.

Kalra, N. & Stentz, A. (T. (2003). *A Market Approach to Tightly-Coupled Multi-Robot Coordination: First Results*. In: *ARL Collaborative Technologies Alliance Symposium*.

Kalyanasundaram, B. & Pruhs, K. (1993). *Online weighted matching*, J. of Algorithms 14(3): 478-488.

Konolige, K.; Fox, D.; Ortiz, C.; Agno, A.; Eriksen, M.; Limketkai, B.; Ko, J.; Morisset, B.; Schulz, D.; Stewart, B. & Vincent, R. (2004). *Centibots: Very large scale distributed robotic teams*. In: *Int. Symp. on Experimental Robotics*.

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

Kuhn, H. W. (1955). *The Hungarian Method for the assignment problem*, Naval Research Logistic Quarterly 2(1): 83-97.

Lagoudakis, M.; Berhault, M.; Koenig, S.; Keskinocak, P. & Kleywegt, A. J. (2004). *Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation*. In: *IEEE/RSJ Int. Conf. on Intelligent Robotic Systems (IROS'04)*, pp. 698-705.

Lagoudakis, M. G.; Markakis, E.; Kempe, D.; Keskinocak, P.; Kleywegt, A.; Koenig, S.; Tovey, C.; Meyerson, A. & Jain, S. (2005). *Auction-Based Multi-Robot Routing*. In: *Robotics: Science and Systems*.

Larsen, A.; Madsen, O. B. G. & Solomon, M. M. (2004). *The A-Priori Dynamic Traveling Salesman Problem with Time Windows*, Transportation Science 38(4): 459-472.

Lemaire, T.; Alami, R. & Lacroix, S. (2004). *A Distributed Tasks Allocation Scheme in Multi-UAV Context*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*.

Lin, L. & Zheng, Z. (2005). *Combinatorial Bids based Multi-robot Task Allocation Method*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, pp. 1157-1162.

Long, M.; Gage, A.; Murphy, R. & Valavanis, K. (2005). *Application of the Distributed Field Robot Architecture to a Simulated Demining Task*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*.

Long, M. T.; Murphy, R. R. & Parker, L. E. (2003). *Distributed Multi-Agent Diagnosis and Recovery from Sensor Failures*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, pp. 2506-2513.

Mitsunaga, N.; Izumi, T. & Asada, M. (2003). *Cooperative Behavior based on a Subjective Map with Shared Information in a Dynamic Environment*. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, pp. 291-296.

Mouaddib, A.-I. (2004). *Multi-Objective Decision-Theoretic Path Planning*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, pp. 2814-2819.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Muñoz-Avila, H.; Murdock, J. W.; Wu, D. & Yaman, F. (2004). *Applications of SHOP and SHOP2*, IEEE Intelligent Systems 20(2): 33-41.

Nau, D. S.; Cao, Y.; Lotem, A. & Muñoz-Avila, H. (1999). *SHOP: Simple Hierarchical Ordered Planner*. In: *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 968-975.

Ng, A. Y. & Jordan, M. I. (2000). *{PEGASUS}: A policy search method for large MDPs and POMDPs*. In: *16th Conf. on Uncertainty in Artificial Intelligence (UAI'00)*, pp. 406-415.

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

Parker, L. E. (1997). *L-ALLIANCE: Task-Oriented Multi-Robot Learning In Behavior-Based Systems*. In: *Advanced Robotics, Special Issue on Selected Papers from IROS'96*, pp. 305-322.

Peshkin, L.; Kim, K.-E.; Meuleau, N. & Kaelbling, L. P. (2000). *Learning to cooperate via policy search*. In: *16th Conf. on Uncertainty in Artificial Intelligence (UAI'00)*, pp. 489-496.

Porta, J. M.; Spaan, M. T. J. & Vlassis, N. (2005). *Robot Planning in Partially Observable Continuous Domains*. In: *Robotics: Science and Systems*, pp. 217-224.

Porta, J. M.; Vlassis, N.; Spaan, M. T. & Poupart, P. (2006). *Point-Based Value Iteration for Continuous POMDPs*, J. Mach. Learn. Res. 7: 2329-2367.

Pynadath, D. V. & Tambe, M. (2002). *Multiagent teamwork: analyzing the optimality and complexity of key theories and models*. In: *1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pp. 873-880.

Reinelt, G., 1994. *The traveling salesman: computational solutions for TSP applications*. Springer.

Rosencrantz, M.; Gordon, G. & Thrun, S. (2003). *Locating moving entities in indoor environments with teams of mobile robotics*. In: *Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'03)*.

Sanfeliu, A. (2008). *Cooperative robotics in urban sites*. In: *Interantional Symposium on Network Robot Systems*.

Saridis, G. N. (1979). *Toward the realization of intelligent controls*, Proceedings of the IEEE 67(8): 1115-1133.

Scerri, P.; Farinelli, A.; Okamoto, S. & Tambe, M. (2005). *Allocating tasks in extreme teams*. In: *4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'05)*, pp. 727-734.

Schaller, R. (1997). *Moore's law: past, present and future*, IEEE Spectrum 34(6): 52-59.

Schesvold, D.; Tang, J.; Ahmed, B. M.; Altenburg, K. & Nygard, K. E. (2003). *POMDP Planning for High Level UAV Decisions: Search vs. Strike*. In: *16th Int. Conf. on Computer Applications in Industry and Engineering (CAINE'03)*.

Schneider, J.; Apfelbaum, D.; Bagnell, D. & Simmons, R. (2005). *Learning Opportunity Costs in Multi-Robot Market Based Planners*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, pp. 1163-1168.

Simmons, R.; Apfelbaum, D.; Burgard, W.; Fox, D.; Moors, M.; Thrun, S. & Younes, H. (2000). *Coordination for Multi-Robot Exploration and Mapping*. In: *National Conf. on Artificial Intelligence (AAAI'00)*.

Smith, R. G. (1980). *The Contract Net Protocol*, IEEE Transactions on Computers 29(12): 1104-1113.

Spaan, M. T. J.; Gordon, G. J. & Vlassis, N. (2006). *Decentralized planning under uncertainty for teams of communicating agents*. In: *Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'06)*, pp. 249-256.

Spaan, M. T. J. & Groen, F. C. A. (2003). *Team coordination among robotic soccer players*. In: *RoboCup 2002*, pp. 409-416.

Spaan, M. T. J. & Vlassis, N. (2005). *Perseus: Randomized Point-based Value Iteration for POMDPs*, Journal of Artificial Intelligence Research 24: 195-220.

Stentz, A. & Dias, M. B. (1999). *A Free Market Architecture for Coordinating Multiple Robots*.

Stone, P. & Veloso, M. M. (1998). *Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork*. In: *Agent Theories, Architectures, and Languages*, pp. 293-308.

Tardioli, D. & Villarroel, J. L. (2007). *Real Time Communications over 802.11: RT-WMP*. In: *Mobile Ad-Hoc and Sensor Systems*.

Thayer, S.; Digney, B.; Dias, M. B.; Stentz, A. (T.; Nabbe, B. & Hebert, M. (2000). *Distributed Robotic Mapping of Extreme Environments*. In: *SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*.

Tovey, C.; Lagoudakis, M. G.; Jain, S. & Koenig, S. (2005). *Generation of Bidding Rules for Auction-Based Robot Coordination*. In: *3rd International Multi-Robot Systems Workshop*.

Vail, D. & Veloso, M. (2003). *Dynamic multi-robot coordination*. In: Schultz, A. C.; Parker, L. E. & Schneider, F. E. (Ed.), *MultiRobot Systems: From Swarms to Intelligent Automata*, Springer.

Vu, T.; Go, J.; Kaminka, G. A.; Veloso, M. M. & Browning, B. (2003). *MONAD: a flexible architecture for multi-agent control.*. In: *Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'03)*.

Wellman, M. P. (1997). *Market-Aware Agents for a Multiagent World*. In: *8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 1.

Werger, B. B. & Matarić, M. J. (2000). *Broadcast of local eligibility: behavior-based control for strongly cooperative robot teams*. In: *4th Int. Conf. on Autonomous agents (AGENTS'00)*, pp. 21-22.

Yamauchi, B. (1998). *Frontier-based exploration using multiple robots*. In: *2nd Int. Conf. on Autonomous agents (AGENTS'98)*, pp. 47-53.

Yokoo, M., 2001. *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*. Springer-Verlag, London, UK.

Zlot, R.; Stentz, A.; Dias, M. B. & Thayer, S. (2002). *Multi-Robot Exploration Controlled by a Market Economy*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'02)*.

Zlot, R. M. & Stentz, A. (T. (2003). *Market-based Multirobot Coordination Using Task Abstraction*. In: *Int. Conf. on Field and Service Robotics*.

Zlot, R. M. & Stentz, A. (T. (2005). *Complex Task Allocation For Multiple Robots*. In: *IEEE Int. Conf. on Robotics and Automation (ICRA'05)*.

Zuluaga, M. & Vaughan, R. T. (2008). *Modeling multi-robot interaction using generalized occupancy grids*. In: *In Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA'08)*.