Task and Role Allocation Within Multi-Agent and Robotics Research

Adam Campbell and Annie S. Wu

May 17, 2007

1 Introduction and motivation

In the past couple of decades, multi-agent researchers and roboticists have focused on designing systems containing multiple, autonomous agents that work together to accomplish a common objective. In natural systems, we have seen how groups of animals can solve problems that could not be solved by solitary individuals [1, 27]. Because of the emergent behaviors within multi-agent systems, the whole is sometimes greater than the sum of their parts. Roboticists are interested in multi-robot cooperation because if intelligent teamwork can be seen amongst a group of robots, then in the future they can probably cooperate with humans also [7]. Also, because of the high level of redundancy and use of decentralized control, they are without a single point of failure and, thus, robust to complete crashes occurring when one part of the system breaks [4, 17, 24, 28, 29, 30]. Unfortunately, the benefits of using multiple agents to solve problems do not come free.

Researchers focused on multi-robot systems have found that robot-robot interference can cause task completion performance to decrease as more robots are added to the team [9, 14]. Along with the physical limitations, computational problems also arise in multi-agent and multi-robot systems. When searching through the multi-agent literature, four main problems are found. Two of the problems, role allocation and task allocation, are nearly exclusive to multi-agent systems, while the other two, task decomposition and action selection, are related to both multi and single-agent systems. Task decomposition is a divide-and-conquer approach to problem solving, as it involves breaking up large tasks into smaller, manageable subtasks. Task allocation is the problem of optimally assigning these subtasks to agents. Action selection involves determining which low level actions to take in order to complete an assigned task. Related to task allocation is the higher level problem of role allocation. Typically, roles define which tasks an agent should complete, and tasks influence the actions that an agent takes. Because action selection is a problem that concerned researchers focusing on single-robot systems, it will not be discussed here. An overview of action selection research can be found in [26, 11]. Similarly, task decomposition is a problem for both single-agent systems [] and multi-agent systems

[22, 15] and will not be discussed here. Task and role allocation have been two of the focuses of multi-agent and multi-robot research lately, but some work has confused the two problems, and there has been little work in figuring out the separation between these problems. This paper will attempt to define the two problems and then give a survey of the research related to them.

2 Background information

Because distributed teamwork is present in natural systems such as social insect groups [23, 1, 27], definitions of some key terms will be taken from the behavioral ecology literature. To try and promote consistency within the robotics, multi-agent, and biological literature, we will attempt to explain how various terms are used throughout the respective literatures. From the biology literature, we find Oster and Wilson defining a task as a "set of behaviors that must be performed to achieve some purpose of the colony" [23, p. 326]. Anderson and Franks, also biologists, use Oster and Wilson's task definition, but in the context of overall group fitness. They describe a task as being "an item of work that potentially makes a positive contribution, however small, to inclusive fitness" [1]. Anderson and Franks also mention subtasks, which are those sets of behaviors that are only rewarded when the full task is completed [1]. Breaking a problem into subtasks and then assigning those subtasks to agents of a team leads to the problem of optimal assignment. How can the tasks be assigned to agents so that the overall utility of the team is maximized? Some agents' skills may be well suited for some tasks and poorly suited for others. This is known as the task allocation problem within the multi-agent and robotics literature.

What, then, is the role allocation problem? Researchers have actually used the terms role and task interchangeably [8]. We believe that there is a definite difference between the two, and we hope to shed some light on those differences. Gerkey and Matarić use the two terms interchangeably in [7, 8], but do note that role usually has a more time-extended connotation whereas tasks are more transient in nature. We believe that task should be used as a unit of work, whereas role should be used to describe the part or character that an agent "plays" within the team. The role that an agent is currently in will most likely define which tasks the agent can perform well or can perform at all. Some of the confusion may be the result of a one-to-one mapping between role and task within certain problem domains. For example, if the goal of a team containing N members is to retrieve N labeled mines, then it is possible to assign a mine to each agent and then say something akin to: "The task of agent x is to retrieve mine y." It could also be presented in a different way: "The role of agent x is to be the *mine-y-retriever*." In the first statement, task is used to refer to the actual piece of work that is assigned to agent x, while role is used in the second to describe the "character" that agent x plays within the group. The distinction becomes more clear when a separate scenario is considered. Imagine the N agents being assigned to retrieve mines and protect their home base. In this case, the two roles are *forager* and *soldier*. The tasks that the agents work on are determined by the role that they are in. Because of the interrelationship between tasks and roles within the multi-agent settings, both role allocation and task allocation research is presented in this paper.

Nair *et al.* break up the decision making problems of cooperating agents into role-taking and role-executing problems [20, 21]. The role-taking problem is essentially that of role allocation, while role-executing is the problem of figuring out which action to take while in a particular role. Although one could argue that the role-taking problem could be encompassed by solving the role-executing problem by making one of the actions a role can take actually switch the role of the agent, Nair *et al.* make the distinction because doing so allows for the costs of role-executing and role-taking to be considered separately. Nair *et al.* show that, in general, the problem of finding a role-taking (or role allocation) policy is *NEXP-Complete* even when the role-executing policy is held constant [21].

In [3], Cao *et al.* discuss some of the major issues when designing teams of cooperative, mobile robots. Five research axes within the field are defined in the paper. These include group architectures, resource conflicts, origins of cooperation, learning, and geometric problems. Task decomposition and allocation were intentionally left out of their work because at the time of writing little research had been done in this area, the tasks that the cooperating robots were trying to solve were not that difficult, and the two problems are directly related to the group architecture that is being used. In the past decade the robotic systems have become much more advanced than their predecessors, and the problems have become more difficult. Also, a great amount of interest in multi-agent teamwork has been created by well known problems such as RoboCup Soccer [12]. Because of the vast amount of task and role allocation work that has been produced in the past decade, along with the new motivations for role differentiation created by problems such as RoboCup, we believe that a survey of the role and task allocation work is now appropriate. Other surveys related to cooperative mobile agents are Dudek's taxonomy for swarm robotic systems [4] and Panait and Luke's survey of machine learning within cooperative multi-agent systems [25].

3 Problem details

This section will discuss the work that has been done within role and task allocation. Research that contains overlap between the two will be discussed in detail, as one of the main purposes of this paper is to better define a separation between the problems.

3.1 Task allocation

Work on task allocation within multi-agents is a subset of the distributed problem solving work with in the Artificial Intelligence community. In [33], Reid and Davis describe task-sharing, where processing nodes decompose tasks into subtasks and then spawn those off to other nodes, which in turn, can decompose their task into subtasks, and so on. Difficulty arises when deciding which nodes should take which tasks so as to minimize processing time. Instead of processing nodes, we are interested in cooperative, autonomous agents, whether they be members of a simulated team in a video game [35] or physical robots demining a mine field [16].

Early multi-robot architectures that were focused on solving the task decomposition and allocation problems were designed to be flexible, fault tolerant, and decentralized [22, 15]. To determine task allocations, negotiations take place through bidding systems, where available agents place higher "bids" on tasks they are well suited to work on. Agents with higher bids are then assigned to these tasks. Tasks can be dynamically created, decomposed into smaller tasks, and assigned at run-time, which is needed for these systems due to the uncertainty with the environments. These early architectures paved the way for the current research.

As claimed by Gerkey and Matarić in [7], most of the task allocation work up to the point of the writing of their paper had been empirical and contained little theoretical foundation. In this work, Gerkey and Matarić present a formal framework for task allocation. Their taxonomy of the multi-robot task allocation problem uses three criteria, with each criteria containing two values, thus, creating eight classes of task allocation problems. The complexity of each classification is analyzed separately, so that once new problems arrive, they can be classified, and their theoretical complexity will be already known. Unfortunately, not all task allocation research falls neatly within one of these eight classifications, but the work done by Gerkey and Matarić does give a strong theoretical foundation for the problem.

Phrases other than task allocation have been used to refer to this problem. Task division is used in [32]. Division is used instead of allocation because the area in which the robots were foraging was divided into sections and each robot was responsible for foraging items within one specific section. Other multi-robot research concerned with task allocation include [24, 17]. Also, Gerkey's PhD dissertation focuses on this problem [6]. Because there has been so much work done on this problem, a majority of this paper will focus on the role allocation problem. The point here was to give the reader a feel for what task allocation is and the methods used to solve it, so that we can now describe how it relates to role allocation.

3.2 Role allocation

The word role is used in theater to denote the character or part that an actor plays. In sociology it is used to mean the "sets of actions taken within the pattern of some institution" [2], and in the multi-agent literature, a role has been used to refer to time extended tasks [8]. Role allocation is analogous to task allocation, with the difference being role assignment instead of task assignment. One may ask the following question: If task and role allocation are such similar problems, then why consider them separately? We believe that role allocation is a higher level problem, because an agent's assigned role will affect which tasks it can and cannot perform well. Furthermore, an agent's role may prevent it from performing a task at all. Gerkey and Matarić began to use the word role, as opposed to task, when they discussed teamwork in the RoboCup Soccer domain [8]. The sophistication of the problem requires coordinated, intelligent behavior, and thus, the use of specialized agents. As with any classification, there exist grey areas where it is not clear how to classify some items. For example, in Lebella *et al.*'s paper title *Efficiency and Task Allocation in Prey Retrieval* [14], the authors use the terms *loafer* and *forager* to refer to the two separate tasks of the agents, however, it is probably more appropriate to refer to these terms as roles, since they define the parts each agent plays within the team. Within the group of foragers, tasks such as *"search upper right quandrant of environment"* or *"retrieve mine at position X,Y"* could be allocated. This is to take nothing away from the work that was done by Lebella *et al.*, we just wish to promote consistency in usage of the terms role and task within the multi-agent context, and this example illustrates the relationship between the task and role allocation problems and shows how the terms may be confused.

In [21], Nair *et al.* present role reallocation, which is the reassigning of roles when failures occur or when new tasks are created. They use role allocation to refer only to initial role assignment. They make this distinction so the two problems can be analyzed separately. Because we are interested in work dealing with dynamic team behavior, we will use the phrase role allocation to refer to both the initial role assignment and reassignment problems. As with task allocation, researchers have used many different terms to describe role allocation. These include such terms as team selection [38] and team formation [20]. The remainder of this discourse will focus on the various issues of role allocation, and if need be, its relation to task allocation.

4 Team compositions

When making the distinction between homogenous and heterogeneous team compositions within the role allocation context, we are referring to both the sets of roles that the cooperating agents can be in, and their initial chances of being in those roles. The team composition of a multi-agent team is homogenous if each member of the team can assume any one of the same set of roles, and their initial chances of being in those roles are equal. Agents may differ in their communication abilities, processing power, and physical structures, but if they all have the same set of available roles and the same initial chances of being in one of those roles, then their team composition with regards to role allocation is considered homogenous. The initial chances of being in a role are determined by the designer of the system. For example, if one robot in a *soldier/forager* team is built to be durable and more powerful than the other members, then it would make sense to assign that agent to the *soldier* role, so the system designer could build the appropriate probabilities into the system. Robot swarms containing (nearly) identical robots are a good example of homogenous team composition [18]. In [18], an agent's role is determined by its position, relative to that of its teammates. Therefore, its initial position may affect its chances of being in a particular role. If the system designer does not make a conscious effort to alter these chances, then we still consider the team homogenous. The role allocation restrictions put forth by the designer may be beneficial because they reduce the number of possible role allocation combinations that can occur [38]. This reduction in complexity could come at the cost of flexibility. In the scenario presented above, where a *soldier/forager* team is gathering items while protecting a home base, the idea of giving the bigger agent a higher probability of being a soldier is reasonable, but if there is no chance for one of the weaker agents to assume the *soldier* role, then when the large individual breaks down, the who team may fail.

5 Role allocation paradigms

Role allocation is the problem of assigning roles to agents so as to maximize the overall utility of the group. With explicit role allocation, each agent has a set of roles that it can be assigned, and the problem is to find the best way to allocate these roles. This usually occurs through some algorithm where agents individually decide their role, or they use a negotiation process through communication with other agents. In either case, agents are explicitly deciding to assign or reassign their role. With implicit role allocation, the assigning of roles is much more subtle. In some of the implicit role allocation methods, the roles themselves are not predefined but are learned, thus making it difficult to tell when agents are even taking on different roles. We feel that it is important to make a distinction between these two approaches because the former lends itself to formal analysis, whereas the performance of the latter systems are usually measured experimentally.

5.1 Implicit role allocation

In [14], foraging robots have a probability that determines whether or not they will continue searching for prey when they return to the nest site. Robots adjust their foraging probability, and as it approaches zero they become less likely to forage. Robots remaining idle at the nest are referred to as *loafers* and benefit the group because having too many foragers can sometimes decrease efficiency due to increasing robot-robot interference [9]. No where in this system did the robot explicitly choose to become a *loafer* or *forager*, yet these two roles still exist. Through the interactions between the robots and their environment, the self organizing nature of this system creates two types of agents.

Another example of implicit role allocation is seen in the formation movement experiments of [28], where robots change their role based on their locations relative to that of their teammates. The strategy used to change roles is discovered by evolving neural network controllers for the robots. Implicit role allocation can also occur through the neuroevolution strategy presented in [35]. In this work, Stanley *et al.* extend NEAT [36, 34] so that evolution occurs in real-time within a video game. Individuals within the population are separated into various species based on differences in their neural network topologies. This protects individuals that may have evolved new network structures, but have yet to optimize them. It should be noted that even though individuals within the population are speciated when reproduction occurs, all individuals are cooperating on the same team. By allowing separate species to evolve within one population, members of different species are essentially evolving separately and thus have the opportunity to specialize and form different roles within the team.

Sometimes, the role that an agent takes on is dependent upon the agent, or agents, it is interacting with. In [39], a group of simulated members of a wolf pack interact with each other and with a user controlled wolf. When wolves interact, they can either act to enforce their dominance or act submissively. Over time, wolves begin to build social relationships with each other and a dominance hierarchy is formed. Although the focus of their research was not on teamwork, the method used to differentiate the status of agents could be used to form a managerial type hierarchy. Agents at the top of the heirarchy could become the leaders of the team and inform the lower members of what tasks to complete. Because the leader-follower relationship between agents may not be transitive, a complex network of relationships can emerge. This flexibility could be desired, or it could become a problem for defining roles. For example, if agent A is dominant over agent B, B is dominant over C, and C is dominant over A, then who should be the leader when choosing roles? If only two of the agents are interacting at any particular time, then the relationship is clear, but when the three interact simultaneously, a cycle is created and could cause undesirable behaviors. In most cases these types of problems are not found because the role an agent is in is explicitly defined.

One of the difficulties with implicit role allocation is figuring out a way to observe whether or not agents are even assuming different roles within a run. The authors of [28] had to remove robots one-by-one in order to determine the evolved function of the robots. Another drawback to implicit role allocation is that agents do not usually have the ability to communicate their roles to neighbors because they are not even aware of their current role. In Stanley *et al.*'s work on rtNEAT [36], the agents would have to become aware of their evolved, specialized abilities, and then communicate those abilities to other agents.

5.2 Explicit role allocation

In explicit role allocation, agents make the conscious decision to take on a particular role. This is usually decided through some sort of negotiation procedure. Problems that are solved through explicit role allocation have the benefit of being formally analyzed using methods found in [7] and [21]. Being able to formally analyze a problem allows experimenters and system designers to determine how close their role allocation algorithms come to producing the optimal answer. Formal analysis also allows one to get an idea of the performance versus efficiency tradeoffs for their particular system.

6 Measuring effectiveness of role allocation strategies

Although empirical analysis does show how one role allocation method compares to another within a specific domain, it does little to give a deep understanding of what the method can and cannot do, or what problems the method is and is not well suited for. A solid mathematical foundation for any science is needed in order to analyze and study the phenomena within that area.

As stated earlier, a taxonomy for the multi-robot task allocation problem can be found in [7]. This classification consists of three axes: single-task robots versus multi-task robots, single-robot tasks versus multi-robot tasks, and instantaneous assignment versus time-extended assignment. Single-task robots are those that can only execute one task at a time, while multi-task robots can execute multiple tasks simultaneously. Similarly, single-robot task problems are ones that require only one robot per task, whereas multi-robot tasks require a coalition of robots. Finally, instantaneous assignment differs from time-extended assignment because it lacks planning for future events. Using all combinations of these properties, eight types of multi-robot task allocation problems are defined. Gerkey and Matarić reduce these eight types of problems to well known problems in Computer Science so that problem complexity can be easily analyzed. For example, problems that involve agents that can only perform one task at a time, each task only requires one agent, and the agents cannot plan for the future can be reduced to the optimal assignment problem [5] and so can be solved in polynomial time. However, when the problem changes slightly, and tasks start to require more than one robot, the problems become *NP-Hard*. In [8], Gerkey and Matarić assume that agents' role utilities are independent of one another, and that the state of the art for solving problems with interrelated utility problems like this are Markov decision processes [40] which are too difficult to solve in a problem like RoboCup Soccer. Nair *et al.* actually tackle this problem by modelling role allocation and reallocation problems with distributed partially observable Markov decision processes (POMDPs) in [21].

7 Details of existing methods, models, and architectures

7.1 Negotiation

Some of the earliest work on task allocation is seen in Lueth and Laengle's presentation of the KAMARA multi-robot architecture [15], which utilizes a blackboard [10] communication mechanism where agents post bids on tasks contained within a mission set. The agent to first post its valuation on the task becomes a mediator and decides which agent is best fit for a particular task. Once the mediator has determined this, it assigns the appropriate agent to the task. It should be noted here, that the assignment of a mediator role to an agent is a form of role allocation. More information on negotiation methods of role and task allocation can be found in [17, 24].

7.2 Machine learning

Martinson and Arkin use Reinforcement Learning in order to try and solve the role allocation problem in a foraging task [16]. [19] show how a supervised learning technique can be used to learn role assignments, and show an example of their strategy in an automated steam condenser design experiment.

Neuroevolution [41], the process of evolving neural networks, is used in [28] to evolve a controller that is used separately by each robot trying to achieve the goal of coordinated movement. In this case, the roles themselves are an emergent property of the system, and not explicitly defined by the system designers. Another example of neuroevolution being used to implicitly evolve roles is seen in [35].

8 Applications

Although theoretical techniques to analyze of the complexity of role allocation algorithms are available [21], experiments qualitatively show the usefulness of particular algorithms and give a way to compare different approaches that do not lend themselves to easy theoretical analysis. Some problem domains were constructed to test one particular algorithm, while others, such as the RoboCup domains, have been used by many researchers.

8.1 RoboCup

One of the most well known domains of multi-agent and robotics research is the RoboCup Soccer domain [12]. RoboCup Soccer is attractive to researchers for several reasons. The first being that because it is so well known, and so much research has been done using the domain, it is easy to compare results from current research to results obtained by previous researchers. This benefit is not present when one produces their own, *ad hoc* domain. Another reason for its popularity is that the simulator can be downloaded for free by anyone, so researchers can spend time working on their algorithms, as opposed to wasting time constructing and designing the domain. For researchers interested in investigating role allocation methods, RoboCup soccer provides a dynamic, noisy environment where agents with incomplete knowledge must work independently, yet still collaborate in order to win their games. For role allocation work done with the RoboCup Soccer domain, see [8, 37].

Because of the small number of agents in the RoboCup Soccer domain, it might not be such an attractive domain for researchers concerned with testing their role allocation algorithms. RoboCup Rescue might be more appealing, as it consists of heterogeneous agents and a constantly changing, noisy environment [13]. Also, the need for automated disaster response teams could greatly reduce the amount of casualties that occur when they strike, so the benefits of research in this domain are numerous. For role allocation work done with the RoboCup Rescue domain, see [20, 21, 31].

8.2 Military applications

[16] gives an example of a scenario where a team of agents are trying to discover and transport mines back to a base station while enemy agents and terrain obstacles pose problems. In [21], the authors use their role allocation algorithm to give a solution the mission rehearsal domain. This domain consists of a helicopter team containing scouts and transport helicopters transferring cargo through a hostile area.

8.3 Foraging

The foraging problem is a general one and it encompasses other problems such as toxic waste cleanup, disaster response, and demining a battle field [16]. Researchers in the multi-agent community have usually focused on these types of problems within the context of task allocation. Foraging can be accomplished by a single individual, so it becomes easy to determine the changes in performance when more than one agent is used. An example of implicit role allocation in a foraging scenario is described in [14], where the roles of agents self organize themselves so as to minimize the robot-robot interference problem.

[24] presents a new task for testing coordination strategies within a multi-robot system. Various alarms are set off at random times, and it is the job of the robots to locate these alarms and perform some action on them. [17] uses the emergency handling problem to show that the best role allocation strategy depends on the amount of sensor and actuator noise within the system.

8.4 Localization and mapping

A swarm of robots are used to distributively map an environment in [29]. In order for this to occur, robots must be able to figure out their global coordinates within the map. In [29] this is accomplished by assigning robots to be in the landmark role. When in the landmark role, robots remain stationary and constantly broadcast their position and heading to neighbors.

8.5 Coordinated movement

In [28], a homogenous team of robots must coordinate their actions so that they all move together a certain distance. Robots are penalized for straying from the group and rewarded for travelling together. Agents assume different roles based on their relative locations within the team. Another example of agents changing roles based on their relative positioning within a group is seen in [18], where a swarm of robots must disperse throughout an environment while remaining communication connectivity. Robots on the boundaries of the group are considered *frontier* robots and lead the way for the *interior* and *wall* robots.

9 Conclusions

Having a group of autonomous agents cooperate and form and reform teams as their environment changes poses a difficult problem for artificial intelligence, multi-agent system, and robotics researchers. Natural systems, especially social insects, have shown that an extremely large number of individuals can accomplish what a single individual cannot. Large tasks can usually be decomposed into smaller, more manageable tasks which can then be allocated to available agents. The allocating of these tasks to agents in a way so that resource use is minimized is the task allocation problem. Research on task allocation methods for mobile robot teams has been going on for the past couple of decades. Tasks such as box pushing, foraging, and part assembly have shown benefits for using cooperative teams of agents. In recent years, an increase in problem complexity has created a demand for more sophisticated teamwork architectures. Agents in these systems can take on particular roles which end up defining their behaviors and which tasks they can and cannot complete. The problem of role allocation has been proven to be *NEXP-Complete* [21]. This high level of complexity will require future designers of multi-agent teams to develop extremely sophisticated algorithms that can deal with the uncertainties of the agents' environment. Humans and other animals are able to solve highly complex problems cooperatively through developed social relationships. It might be the case that robots will need to do the same thing. Robots will have to trust that the information that they are receiving from their robot teammates is correct, and they must be able to detect when failures have occurred. More importantly, humans interacting in human-robot teams will have to trust that their mechanic teammates are capable of sophisticated, adaptive behavior, especially if they are being relied on in military situations. One of the first steps towards this goal is demonstrating cooperation in tasks such as RoboCup Soccer and other benchmark domains that require high levels of cooperation. If it can be shown that robots can work together to solve complex tasks, human-robot teams will become more realizable.

References

[1] Carl Anderson and Nigel R. Franks. Teams in animal societies. *Behavioural Ecology*, 12(5):534–540, 2001.

[2] Eric Baumer and Bill Tomlinson. Institutionalization by habitualization and typification of roles in societies of autonomous agents. In *In Second NASA/JPL Workshop or Radical Agent Concepts (WRAC)*, 2005.

[3] Y. Uny Cao, Alex S. Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[4] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for swarm robots. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993.

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

[6] Brian P. Gerkey. *On Multi-Robot Task Allocation*. PhD thesis, University of Southern California Computer Science Department, August 2003.

[7] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, September 2004.

[8] Brian P. Gerkey and Maja J. Matarić. On role allocation in robocup. In *RoboCup 2003: Robot Soccer World Cup VII*. Springer-Verlag, 2004.

[9] Dani Goldberg and Maja J. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *AAAI/IAAI*, pages 637–642, 1997.

[10] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence*, 26(3):251–321, 1985.

[11] Mark Humphrys. Action selection methods using reinforcement learning. In P. Maes, M. Matarić, J. A. Meyer, J. Pollack, and S. W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 135–144, Cambridge, Massachusets, 1996. MIT Press, Bradford Books.

[12] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the 1st international conference on autonomous agents*, pages 340–347, Marina del Rey, California, United States, 1997.

[13] Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsuhi Shinjou, and Susumu Shimada. Robocup rescue: search and rescue in large-scale disasters as adomain for autonomous agents research. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 739–743, Tokyo, Japan, 1999.

[14] Thomas H. Labella, Marco Dorigo, and Jean-Louis Deneubourg. Efficiency and task allocation in prey retrieval. In *Proceedings of the 1st International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pages 274–289, 2004.

[15] Tim C. Lueth and Thomas Laengle. Task description, decomposition and allocation in a distributed autonomous multi-agent robot system. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 1516–1523, September 1994.

[16] Eric Martinson and Ronald C. Arkin. Learning to role-switch in multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 2727–2734, 2003.

[17] Maja J. Matarić, Gaurav S. Sukhatme, and Esben H. Østergaard. Multirobot task allocation in uncertain environments. *Autonomous Robots*, 14(23):255–263, 2003.

[18] James McLurkin and Jennifer Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed Autonomous Robotic Systems Conference*, June 2004.

[19] Prasad Nagendra, Lesser M.V., Lander V.R., and S. Learning organizational roles in a heterogeneous multi-agent system. *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 291–298, January 1996.

[20] Ranjit Nair, Milind Tambe, and Stacy Marsella. Team formation for reformation in multiagent domains like robocuprescue. In *Proceedings of the International Symposium on RoboCup*, 2002.

[21] Ranjit Nair, Milind Tambe, and Stacy Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, 2003.

[22] Fabrice R. Noreils. An architecture for cooperative and autonomous mobile robots. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2703–2710, May 1992.

[23] George F. Oster and Edward O. Wilson. *Caste and Ecology in the Social Insects*. Princeton University Press, Princeton, New Jersey, 1978.

[24] Esben H. Østergaard, Maja J. Matarić, and Gaurav S. Sukhatme. Distributed multi-robot task allocation for emergency handling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Wailea, Hawaii, October 2001.

[25] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[26] Lynne E. Parker. Adaptive action selection for cooperative agent teams. In *Proceedings of the second international conference on From animals to animats 2 : simulation of adaptive behavior*, pages 442–450, Cambridge, MA, USA, 1993. MIT Press.

[27] Lynne E. Parker. Current state of the art in distributed autonomous mobile robotics. In Lynne E. Parker, George Bekey, and Jacob Barhen, editors, *Distributed Autonomous Robotic System 4*, pages 3–12. Springer-Verlag, Tokyo, October 2000. ISBN: 4-431-70295-4.

[28] Matt Quinn, Lincoln Smith, Giles Mayley, and Phil Husbands. Evolving teamwork and role allocation with real robots. In *Proceedings of the 8th International Conference on The Simulation and Synthesis of Living Systems (Artificial Life VIII)*, 2002.

[29] Joseph A. Rothermich, M. Ihsan Ecemis, and Paolo Gaudiano. Distributed localization and mapping with a robotic swarm. In E. Sahin and W. Spears, editors, *Proceedings of the Swarm Robotics Workshop at the 8th International Conference on the Simulation of Adaptive Behavior SAB-04*, pages 58–69. Springer-Verlag, 2004.

[30] Paul E. Rybski, Amy Larson, Harini Veeraraghavan, Monica LaPoint, and Maria Gini. Performance evaluation of a multi-robot search & retrieval system: Experiences with mindart. Technical Report 03-011, Department of Computer Science and Engineering, University of Minnesota, MN, USA, February 2003.

[31] Paul Scerri, Alessandro Farinelli, Stephen Okamoto, and Milind Tambe. Token approach for role allocation in extreme teams: analysis and experimental evaluation. In *Proceedings of the 2nd IEEE International Workshop on Theory and Practice of Open Computational Systems*, 2004.

[32] Miguel Schneider-Fontan and Maja J. Matarić. Territorial multi-robot task division. *IEEE Transactions of Robotics and Automation*, 14(5), 1998.

[33] Reid G. Smith and Randall Davis. Frameworks for cooperation in distributed problem solving. SMC-11(1):61–70, January 1981.

[34] Kenneth O. Stanley. *Efficient Evolution of Neural Networks Through Complexification.* PhD thesis, Department of Computer Sciences, The University of Texas at Austin, August 2004.

[35] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Real-time neuroevolution in the nero video game. *IEEE Transactions on Evolutionary Computation*, 9(6):653–668, December 2005.

[36] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[37] Peter Stone and Manuela M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In *Agent Theories, Architectures, and Languages*, pages 293–308, 1998.

[38] Gil Tidhar, Anand S. Rao, and Elizabeth A. Sonenberg. Guided team selection. In *Proceedings of the 2nd International Conference on Multi-agent Systems (ICMAS-96)*, 1996.

[39] Bill Tomlinson. Social characters for computer games. *International Journal of Interactive Technology*, 2(2):101–115, 2005.

[40] D. J. White. *Markov decision processes.* John Wiley & Sons, Chichester, England, 1993.

[41] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9), September 1999.