

Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems

Lynne E. Parker

Distributed Intelligence Laboratory
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
parker@eecs.utk.edu

Abstract

This paper introduces the concepts of distributed intelligence, outlining the motivations for studying this field of research. We then classify common systems of distributed intelligence based upon the types of interactions exhibited, since the type of interaction has relevance to the solution paradigm to be used. We outline three common paradigms for distributed intelligence — the bioinspired paradigm, the organizational and social paradigm, and the knowledge-based, ontological paradigm — and give examples of how these paradigms can be used in multi-robot systems. We then look at a common problem in multi-robot systems — that of task allocation — and show how the solution approach to this problem is very different depending upon the paradigm chosen for abstracting the problem. Our conclusion is that the paradigms are not interchangeable, but rather the selection of the appropriate paradigm is dependent upon the specific constraints and requirements of the application of interest. Further work is needed to provide guidance to the system designer on selecting the proper abstraction, or paradigm, for a given problem.

Introduction to Distributed Intelligence

Distributed Intelligence refers to systems of *entities* working together to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn. Here, we define an *entity* as any type of intelligent process or system, including agents, humans, robots, smart sensors, and so forth. In these systems, different entities commonly specialize in certain aspects of the task at hand. As humans, we are all familiar with distributed intelligence in teams of human entities. For example, corporate management teams consist of leaders with particular specialties as Chief Executive Officer (CEO), Chief Operating Officer (COO), Chief Financial Officer (CFO), Chief Information Officer (CIO), and so forth. Oncology patient care teams consist of doctors that specialize in certain areas, such as surgical oncology, medical oncology, plastic and reconstructive surgery, pathology, etc. Distributed intelligence is also exhibited in military applications, such as special forces A-Teams, where team members

specialize in weapons, engineering, medicine, communications, and so forth. Another military example includes personnel on an aircraft carrier flight deck, who are segmented into the catapult crew, the landing signal officers, ordnance-men, plane handlers, etc. As humans have clearly learned, these teams can very efficiently solve complex tasks by making use of specialists who work together productively.

The objective of distributed intelligence in computer science (and related fields) is to generate systems of software agents, robots, sensors, computer systems, and even people and animals (such as search and rescue dogs) that can work together with the same level of efficiency and expertise as human teams. Clearly, such systems could address many important challenges, including not only urban search and rescue, but also military network-centric operations, gaming technologies and simulation, computer security, transportation and logistics, and many others.

As a research topic, the study of distributed intelligence has gained much popularity in recent years. Figure 1 shows data from the Web of Science resulting from a keyword search on the terms “distributed intelligence”, “distributed AI”, “distributed artificial intelligence”, “multiagent”, “multi-agent”, “distributed robot”, “multirobot”, and “multi-robot”. Each year’s results shows the number of publications that appeared containing these keywords. The search begins in year 1980 — the most recent year with no publications containing any of these keywords — up through 2006. (Presumably, the 2006 data is somewhat incomplete, as there is a time lag between the appearance of a publication and its inclusion in Web of Science.) While this is an admittedly incomplete survey of this area of research, the data does clearly show the significantly increasing interest in this research area, as investigators and application developers are recognizing the potential power of distributed intelligence.

What is the potential promise of distributed intelligence? Certainly, some applications can be better solved using a distributed solution approach — especially those tasks that are inherently distributed in space, time, or functionality. Further, if a system is solving various subproblems in parallel, then it offers the potential of reducing the overall task completion time. Any system consisting of multiple, sometimes

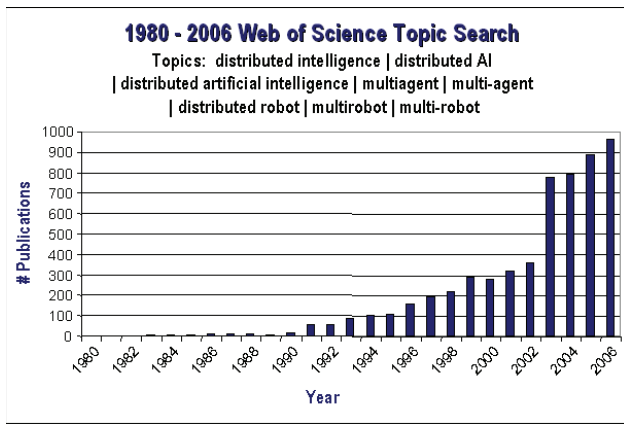


Figure 1: Web of Science data showing the number of publications appearing per year on topics related to distributed intelligence.

redundant, entities, offers possibilities of increasing the robustness and reliability of the solution, due to the ability for one entity to take over from another failing entity. Finally, for many applications, creating a monolithic entity that can address all aspects of a problem can be very expensive and complex; instead, creating multiple, more specialized entities that can share the workload offers the possibility of reducing the complexity of the individual entities.

Of course, these advantages of distributed intelligence are, to some extent, offset by some disadvantages. For example, even though the individual entity cost and complexity may be less, determining how to manage the complete system may be more difficult and complex, because of the lack of centralized control or of a centralized repository of global information. Further, distributed intelligent systems may require more communication to coordinate all the entities in the system. Increasing the number of entities can lead to increased interference between entities, as they must act without complete knowledge of the other entities' intents. Finally, systems of multiple entities will typically experience increased uncertainty about the state of the system as a whole.

Overall, however, as new solution approaches are being developed and validated, the research and user community is finding that it is often possible to realize the advantages of distributed intelligence systems while countering many of the possible disadvantages. The challenge is determining how best to properly design the system so as to achieve global coherence through the local interactions of individual entities.

The Domain Space of Distributed Intelligence

As researchers are discovering, there are many possible solution strategies, or paradigms, for achieving distributed in-

telligence. Not all of these paradigms are appropriate for all types of distributed intelligence. Thus, it is important to understand the various types of distributed intelligence that can occur in different application settings.

One important way to view the domain space in distributed intelligence is by understanding the types of interactions that can take place between the entities in the system. As illustrated in Figure 2, we find it instructive to view the types of interactions along three different axes — the *types of goals*, whether entities have *awareness of others* on the team, and whether an entity's actions *advance the goals of others* on the team. In terms of types of goals, we classify systems into two types — those in which each entity has *individual* goals, and those in which the entities have *shared* goals. For the *awareness of others* axis, we divide the systems into those that are *aware* and those that are *not aware*. By *aware* in this context, we refer to whether entities reason about the actions and intentions of their teammates. Robots that are *not aware* may sense the presence of local entities and move so as to maintain a certain distance, for example, but otherwise perform no other reasoning to understand the intent or future plans of the teammates. Often, these “unaware” systems operate based on the principle of *stigmergy*, in which communication between entities is not direct, but rather through changes made in the environment.

Finally, we segment systems into those in which an entity's actions do advance the goals of others on the team (*yes*) and those that do not (*no*). An example of an entity advancing the goals of others with its actions is a floor cleaning robot, as a member of a floor cleaning robot team. Each robot's actions of cleaning a bit of the floor are helpful to the other teammates, who do not have to repeat the floor cleaning in that particular spot.

Obviously, these segmentations of the domain space are approximate, yet we believe they are helpful in understanding the primary types of interactions that can occur in typical applications. Different areas of this subspace represent common types of interactions seen in systems of distributed intelligence. These common forms of interaction are:

- Collective
- Cooperative
- Collaborative
- Coordinative

In the following paragraphs we describe these types of interactions in more detail.

Perhaps the simplest type of interaction is the *collective* interaction, in which entities are not aware of other entities on the team, yet they do share goals, and their actions are beneficial to their teammates. An example of this type of interaction in multi-robot systems is the swarm robotics work of many researchers (e.g., (McLurkin 2004; Mataric 1995;

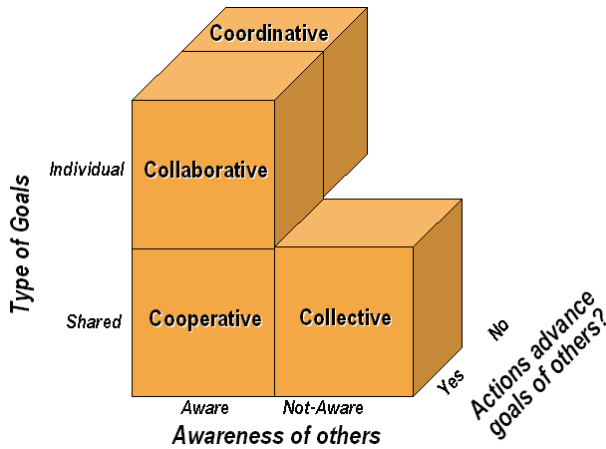


Figure 2: Categorization of types of interactions in systems of distributed intelligence.

Kube & Zhang 1993)). This work focuses on creating systems of robots that can perform biologically-relevant tasks, such as foraging, swarming, flocking, herding, formation-keeping, and so forth. Robots in these systems typically perform relatively simple local control laws which, when combined with larger numbers of robots, result in the global goal being achieved, often as an emergent property of the local interactions.

The second type of interaction is the *cooperative* interaction, in which entities are aware of other entities, they share goals, and their actions are beneficial to their teammates. In multi-robot systems, an example of this type of interaction is multiple robots working together and reasoning about each other's capabilities in order to accomplish a joint task, such as pushing a box (e.g., (Gerkey & Mataric 2002)), cleaning up a worksite (e.g., (Parker 1998)), performing search and rescue (e.g., (Murphy 2000)), or extra-planetary exploration (e.g., (Stroupe *et al.* 2006)). In these systems, robots may at times be working on different parts of the higher level goal, and thus may at times have to ensure that they share the workspace without interfering with each other. However, the majority of the work of the robots is focused on working together to achieve a common goal.

A third type of interaction in systems of distributed intelligence occurs when robots have individual goals, they are aware of their teammates, and their actions do help advance the goals of others. This part of the domain space is typically called *collaborative*, and is characterized by entities helping each other to achieve their individual, yet compatible, goals. While closely associated to the cooperative domain space, we make a distinction here to focus on the ability of entities to work together to help others better achieve their individual goals. In human research teams, we are familiar with the concept of collaboration, in which each person brings unique expertise that helps the team as a whole achieve a broader objective. Each team member has his/her

own goal of performing his/her own aspect of the research, but by working together with others with complementary expertise, each can help the other members better achieve their individual goals. Of course, most of these collaborations are also cooperative, and it is possible to turn a collaborative team into a cooperative team by simply viewing the team goals from a higher perspective. A multi-robot example of a collaborative team is a group of robots that each must reach specified goal positions that are unique to each member. If robots are unable to reach their goal positions independently, due to sensor limitations, they could work together with other robots by sharing sensory capabilities to help all team members reach their individual goal locations. This type of collaboration is sometimes called *coalition formation*, and has been illustrated in (Parker & Tang 2006; Vig & Adams 2006).

Finally, the fourth type of interaction relevant to distributed intelligence is what we call *coordinative*. In these systems, entities are aware of each other, but they do not share a common goal, and their actions are not helpful to other team members. In multi-robot systems, these situations often occur when robots share a common workspace. The robots must work to coordinate their actions to minimize the amount of interference between themselves and other robots. Multi-robot path planning techniques (e.g., (Kloder & Hutchinson 2006; Guo & Parker 2002)) or traffic control techniques (e.g., (Asama *et al.* 1991; Yuta & Premvuti 1992; Wang 1991)) are commonly used in these domains.

As a side note, we could have extended the third axis of our domain space to categorize systems based on whether they (1) *positively* affect the goals of other entities, (2) have *no effect* on the goals of other entities, or (3) *negatively* effect the goals of other entities. Then, we could create a new type of interaction in which entities have individual goals, they are aware of each other, but their actions have a negative effect on others' goals. This defines the *adversarial* domain, in which entities actively work against each other. In multi-robot systems, this topic is studied extensively in the multi-robot soccer application domain (e.g., (Kitano *et al.* 1997; Browning *et al.* 2005; Veloso, Stone, & Han 1999; Stone & Veloso 1999)). This form of interaction also has clear relevance for many military applications.

Paradigms for Distributed Intelligence

Just as there are many types of interactions in systems of distributed intelligence, there are also many paradigms for achieving distributed intelligence. Each paradigm abstracts the problem space in a different way, enabling the system designer to view the system from a perspective that sheds light on proper solution strategies. Often, these paradigms take inspiration from societies of insects, or societies of humans. Not all paradigms are appropriate for all types of interaction dynamics. In this section, we outline some of

the more common paradigms for distributed intelligence, especially focusing on their relevance to multi-robot systems. Note that a fundamental challenge in all of these paradigms is determining how best to achieve global coherence from the interaction of entities at the local level. By abstracting the problem in different ways, alternative solution strategies become apparent that can help address this challenge.

Three commonly used paradigms for building systems of distributed intelligence include:

- Bioinspired, emergent swarms paradigm,
- Organizational and social paradigms, and
- Knowledge-based, ontological, and semantic paradigms.

We discussed concepts of the *bioinspired, emergent swarms* paradigm in the previous section, as part of the description of collective interactions. In this paradigm, the need for communication between entities is greatly reduced by assuming the ability of the entities to sense relevant information in their local environments (i.e., stigmergy). The application requirements in these problems allow for simple action protocols, or control rules, that are identical on each entity, and that lead to the desired group behavior. An example local control rule under this paradigm that can cause all the agents/robots to aggregate (as in a swarm) is:

```
Aggregate:
  If agent is outside aggregation
    distance
  then turn toward aggregation
    centroid and go.
Else
  stop.
```

This is a powerful paradigm for those applications that require the same task to be performed across a distributed workspace, where the task does not require complex interactions of entities and all entities are interchangeable. Research challenges include developing tools that can predict the global behavior given a set of local control rules, as well as the inverse problem, in which we want to derive the local control rules, given a desired global behavior. This paradigm is relevant for many spatially distributed applications, including flocking, schooling, foraging, chaining, search, sorting, herding, aggregation, condensation, dispersion, containment, formations, harvesting, deployment, and coverage. However, other types of interactions require more complex solution paradigms.

Organizational and social paradigms are typically based on organizational theory derived from human systems. Knowledge from the fields of sociology, economics, and psychology, and related areas, have proven valuable for understanding how to create systems of intelligent artifacts that can work together to solve complex problems. In these ap-

proaches, agent/robot interactions are designed by modeling individual and group dynamics as part of an organization. These approaches reduce the communications requirements among entities by making use of models drawn from these fields. This type of approach is commonly used for *cooperative* and *collaborative* types of distributed intelligence.

Two examples of organizational theory applied to multi-robot systems are the use of *roles* and *value systems*, as well as the use of *market economies*. *Roles* are often used to divide the application into manageable tasks that can each be assigned to a different robot in the team. An easy division of work is achieved by assigning roles according to the skills and capabilities of the individual team members. For instance, in multi-robot soccer (Stone & Veloso 1998; Marsella *et al.* 1999; Veloso, Stone, & Han 1999), positions played by the different robots are often defined as roles, such as goal keeper, left defender, right defender, left forward, right forward, and so forth. The robot best suited, and perhaps in closest proximity, to the available roles/positions then selects to perform that role.

Market economies are used in multi-robot systems as a paradigm for *task allocation*, which we discuss further in the next section. In brief, task allocation is the problem of mapping tasks to robots, such that the most suitable robot is selected to perform the most appropriate task, leading to all tasks being satisfactorily completed. Market-based approaches to task allocation (e.g., (Dias *et al.* 2006; Zlot & Stentz 2006; Gerkey & Mataric 2002)) make use of the theory of market economies to determine how best to manage bids, how to handle multiple bids in parallel, how to consider multiple tasks at once, and so forth. More discussion on this topic is given in the next section.

A third paradigm commonly used for developing systems of distributed intelligence is the *knowledge-based, ontological, and/or semantic* paradigm. The focus in these approaches is on knowledge sharing between heterogeneous robots/agents, with the objective of easily allowing these entities to share and understand knowledge from disparate sources. Often, knowledge is defined as an ontology, which specifies a common vocabulary and semantics for the knowledge in the system. Such approaches require a language for representing knowledge, such as the Knowledge Interchange Format (KIF) (Genesereth & Fikes 1992), as well as a language for communicating knowledge, such as the Knowledge Query and Manipulation Language (KQML) (Finin, Labrou, & Mayfield 1995). This paradigm achieves communication reduction by making use of shared assumptions of vocabulary and semantics.

This type of paradigm can be used for many types of interactions, including cooperative, collaborative, and coordinative. However, while this paradigm has become perhaps the dominant paradigm in multi-agent systems, it is not commonly used in multi-robot systems, at least in the form of full-fledged ontologies. More than likely, this is because

physical robot systems are more challenged by noise and uncertainty in sensing and actuation, as well as low-bandwidth communications, limited power, and limited computation. As such, the limiting bottleneck in multi-robot systems is not typically the semantics of the shared knowledge, but rather dealing with these uncertainties. However, this does not mean that multi-robot systems do not use knowledge-based approaches. On the contrary, many approaches do model information about the system and about the teammates in order to more effectively cooperate, collaborate, and coordinate.

Contrasting Paradigms for a Typical Multi-Robot Challenge: Task Allocation

Having explored three common paradigms in systems of distributed intelligence, we now briefly compare and contrast these paradigms in their approach to a common challenge in multi-robot systems — that of task allocation. As previously introduced, task allocation arises in many multi-robot applications in which the mission of the team is defined as a set of tasks that must be completed. Each task can usually be addressed by a variety of different robots; conversely, each robot can usually work on a variety of different tasks. Independent tasks can be achieved concurrently, while dependent tasks must be achieved according to their interdependence constraints. Once the set of tasks has been defined, the challenge is to determine the preferred mapping of robots to tasks that optimizes some objective function. This is the task allocation problem. The general task allocation problem is known to be NP-hard (Gerkey & Mataric 2004), meaning that optimal solutions cannot be found quickly for large problems. Therefore, solutions to this problem are typically approximations that are acceptable in practice.

Let us examine how each of the paradigms we have discussed would handle the multi-robot task allocation problem. First, the bioinspired approach to task allocation would typically assume large numbers of homogeneous robots that are all interchangeable. In this situation, any robot that is available and senses the need for a task to be performed can select to perform that task (i.e., the task is allocated to that robot). Because of stigmergy, robots do not have to explicitly communicate in order to determine which task to undertake. Robots that fail can be replaced by any other available robot. If all robots operate under this principle, then the entire mission is typically accomplished.

Second, an organizational approach to task allocation would make use of roles, as we described previously for multi-robot soccer. Each role encompasses several specific tasks, and robots select roles that are best suited for their capabilities. In this case, robots need not be homogeneous, but instead can have a variety of different sensing, computation, and effector capabilities.

An alternative organizational approach to task allocation is the market-based approach also mentioned previously. In

these approaches, robots explicitly communicate to bid for tasks according to their expected contribution to those tasks. Assignments are typically made by greedily assigning each task to the robot that can perform it with the highest utility. The fundamental paradigm for interaction in this case is based upon the Contract Net Protocol (Smith 1980), which was the first to address the problem of how agents can negotiate to collectively solve a set of tasks. The use of a market-based approach specifically for multi-robot task allocation was first developed in the M+ architecture (Botelho & Alami 1999). In the M+ approach, robots plan their own individual plans for the task they have been assigned. They then negotiate with other teammates to incrementally adapt their actions to suit the team as a whole, through the use of social rules that facilitate the merging of plans.

Finally, the knowledge-based approach is also used for task allocation in multi-robot teams, through the modeling of teammate capabilities. Many variations are possible, such as in the ALLIANCE approach (Parker 1998), in which robots model the ability of team members to perform the tasks of the system by observing team member performance and collecting relevant task quality statistics, such as time to task completion. Robots then use these models to select tasks to perform that benefit the group as a whole. In this approach, explicit communication is not required for the selection of task assignments. Other techniques are also possible that make use of learned models of teammate capabilities.

As we see through these task allocation examples, a specific problem in multi-robot systems can be addressed in many different ways, based upon the paradigm selected for abstracting the problem at hand. Each paradigm has its own advantages and disadvantages, which may be specific to the application. The paradigms are therefore not interchangeable for many applications, with the most suitable approach depending upon the relevant constraints and requirements of the application.

Conclusions

In this paper, we have outlined aspects of the field of distributed intelligence, focusing on the types of interactions that can occur in such systems, and some common paradigms used to achieve distributed intelligence. To explore the challenges, we have used examples from the field of multi-robot systems to illustrate, compare, and contrast the alternative interactions and paradigms. The main message of these discussions is that the choice of paradigm is not always obvious, and is dependent upon the requirements of the application to be addressed. We also note that complex systems of multiple robots can make use of several different paradigms simultaneously. For example, a large-scale exploration, mapping, deployment, and detection problem, such as that described in (Howard, Parker, & Sukhatme 2006) can make use of an organizational paradigm to define roles for the high-level abstraction, an application-

specific knowledge-based approach for multi-robot mapping, a knowledge-based modeling approach for mobile network deployment, and a bioinspired approach for creating a mobile sensor network. The challenge as system designers is to create and make use of the appropriate paradigms that best address the specific constraints and challenges of the application at hand.

References

- Asama, H.; Ozaki, K.; Itakura, H.; Matsumoto, A.; Ishida, Y.; and Endo, I. 1991. Collision avoidance among multiple mobile robots based on rules and communication. In *Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems*. IEEE.
- Botelho, S., and Alami, R. 1999. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1234–1239. IEEE.
- Browning, B.; Bruce, J.; Bowling, M.; and Veloso, M. 2005. STP: Skills, tactics and plays for multi-robot control in adversarial environments. *IEEE Journal of Control and Systems Engineering* 219:33–52.
- Dias, B.; Zlot, R.; Kalra, N.; and Stentz, A. 2006. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE* 94(7):1257–1270.
- Finin, T.; Labrou, Y.; and Mayfield, J. 1995. Kqml as an agent communication language. In Bradshaw, J., ed., in *Software Agents*. MIT Press.
- Genesereth, M. R., and Fikes, R. E. 1992. *Knowledge Interchange Format, Reference Manual*. Computer Science Department, Univ. of Stanford.
- Gerkey, B. P., and Matarić, M. J. 2002. Sold! auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation* 18(5):758–768.
- Gerkey, B., and Matarić, M. J. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research* 23(9):939–954.
- Guo, Y., and Parker, L. E. 2002. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*.
- Howard, A.; Parker, L. E.; and Sukhatme, G. S. 2006. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment, and detection. *International Journal of Robotics Research* 25:431–447.
- Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawa, E.; and Matasubara, H. 1997. Robocup: A challenge problem of ai. *AI Magazine* 18(1):73–86.
- Kloder, S., and Hutchinson, S. 2006. Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics* 22(4):650–665.
- Kube, C. R., and Zhang, H. 1993. Collective robotics: From social insects to robots. *Adaptive Behavior* 2(2):189–219.
- Marsella, S.; Adibi, J.; Al-Onaizan, Y.; Kaminka, G.; Muslea, I.; and Tambe, M. 1999. On being a teammate: Experiences acquired in the design of RoboCup teams. In Etzioni, O.; Muller, J.; and Bradshaw, J., eds., *Proceedings of the Third Annual Conference on Autonomous Agents*, 221–227.
- Matarić, M. J. 1995. Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems* 16:321–331.
- McLurkin, J. 2004. Stupid robot tricks: Behavior-based distributed algorithm library for programming swarms of robots. In *M.S. Thesis, Massachusetts Institute of Technology*.
- Murphy, R. R. 2000. Marsupial robots for urban search and rescue. *IEEE Intelligent Systems* 15(2):14–19.
- Parker, L. E., and Tang, F. 2006. Building multi-robot coalitions through automated task solution synthesis. *Proceedings of the IEEE, special issue on Multi-Robot Systems* 94(7):1289–1305.
- Parker, L. E. 1998. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation* 14(2):220–240.
- Smith, R. G. 1980. The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* C-29(12).
- Stone, P., and Veloso, M. 1998. A layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence* 12:165–188.
- Stone, P., and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence* 110(2):241–273.
- Stroupe, A.; Okon, A.; Robinson, M.; Huntsberger, T.; Agazarian, H.; and Baumgartner, E. 2006. Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. *Autonomous Robots* 20(2):113–123.
- Veloso, M.; Stone, P.; and Han, K. 1999. The cmunited-97 robotic soccer team: Perception and multiagent control. *Robotics and Autonomous Systems* 29(2-3):133–143.
- Vig, L., and Adams, J. A. 2006. Multi-robot coalition formation. *IEEE Transactions on Robotics* 22(4):637–649.
- Wang, J. 1991. Fully distributed traffic control strategies for many-agv systems. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, 1199–1204. IEEE.
- Yuta, S., and Premvuti, S. 1992. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In *Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems*, 1566–1574. IEEE.
- Zlot, R., and Stentz, A. 2006. Market-based multirobot coordination for complex tasks. *International Journal of Robotics Research* 25(1):73–101.