8 *Multi-agents*

Chapter Objectives:

- Define the types of *control regimes*, *cooperation strategies*, and *goals* in multi-agents.
- Given a description of an intended task, a collection of robots, and the permitted interactions between robots, design a multi-agent system and describe the system in terms of heterogeneity, control, cooperation, and goals.
- Compute the *social entropy* of a team.
- Be able to program a set of homogeneous reactive robots to accomplish a foraging task.
- Describe use of social rules and internal motivation for emergent societal behavior.

8.1 Overview

This chapter explores artificial intelligence methods for coordinating and controlling collections of mobile robots working on completing a task. Collections of two or more mobile robots working together are often referred to as teams or *societies* of multiple mobile robots, or more concisely *multi-agents*.

SOCIETIES
MULTI-AGENTS

Multi-agent teams are desirable for many reasons. In the case of planetary explorers or removing land mines, more robots should be able to cover more area. Like ants and other insects, many cheap robots working together could replace a single expensive robot, making multi-agents more cost effective. Indeed, the term *swarm robots* is becoming popular to refer to large numbers

SWARM ROBOTS

of robots working on a single task. Another motivation of multiple robots is redundancy: if one robot fails or is destroyed, the other robots can continue and complete the job, though perhaps not as quickly or as efficiently. Rodney Brooks at MIT first proposed to NASA that teams of hundreds of inexpensive ant-like reactive robots be sent to Mars in a technical report entitled "Fast, Cheap and Out of Control"[30] in part because having many robots meant that several robots could be destroyed in transit or during landing without a real impact on the overall mission.

Multi-agent teams are becoming quite popular in robot competitions, especially two international robot soccer competitions: RoboCup and MIROSOT. In these competitions, teams of real or simulated robots play soccer against other teams. The soccer task explicitly requires multiple robots that must cooperate with each other, yet react as individuals.

Readers with a strong background in artificial intelligence may notice similarities between teams of mobile robots and teams of software agents ("webots" which search the web and "knowbots" which do data mining). Those similarities are not accidental; software and physical agents fall into a research area in Artificial Intelligence often referred to as *Distributed Artificial Intelligence (DAI)*. Most of the issues in organizing teams of robots apply to software agents as well. Arkin,[10] Bond and Gasser,[119] Brooks,[26] and Oliveira et al.[113] all cite the problems with teams of multiple agents, condensed here as:

DISTRIBUTED ARTIFICIAL INTELLIGENCE (DAI)

- *Designing teams is hard.* How does a designer recognize the characteristics of a problem that make it suitable for multi-agents? How does the designer (or the agents themselves) divide up the task? Are there any tools to predict and verify the social behavior?

- *There is a "too many cooks spoil the broth" effect.* Having more robots working on a task or in a team increases the possibility that individual robots with unintentionally *interfere* with each other, lowering the overall productivity.

INTERFERENCE

- *It is hard for a team to recognize when it, or members, are unproductive.* One solution to the "too many cooks spoil the broth" problem is to try engineering the team so that interference cannot happen. But this may not be possible for every type of team or the vagaries of the open world may undermine that engineering. To defend itself, the team should be capable of monitoring itself to make sure it is productive. This in turn returns to the issue of communication.

- *It is not clear when communication is needed between agents, and what to say.* Many animals operate in flocks, maintaining formation without explicit communication (e.g., songs in birds, signals like a deer raising its tail to display white, speaking). Formation control is often done simply by perceiving the proximity to or actions of other agents; for example, schooling fish try to remain equally close to fish on either side. But robots and modern telecommunications technology make it possible for all agents in a team to literally know whatever is in the mind of the other robots, though at a computational and hardware cost. How can this unparalleled ability be exploited? What happens if the telecommunications link goes bad? Cell phones aren't 100% reliable, even though there is tremendous consumer pressure on cell phones, so it is safe to assume that robot communications will be less reliable. Is there a language for multi-agents that can abstract the important information and minimize explicit communication?

- *The "right" level of individuality and autonomy is usually not obvious in a problem domain.* Agents with a high degree of individual autonomy may create more interference with the group goals, even to the point of seeming "autistic."[113] But agents with more autonomy may be better able to deal with the open world.

The first question in the above list essentially asks *what are the architectures for multi-agents?* The answer to that question at this time is unclear. Individual members of multi-agent teams are usually programmed with behaviors, following either the Reactive (Ch. 4) or Hybrid Deliberative/Reactive (Ch. 7) paradigms. Recall that under the Reactive Paradigm, the multiple behaviors acting concurrently in a robot led to an *emergent behavior*. For example, a robot might respond to a set of obstacles in a way not explicitly programmed in. Likewise in multi-agents, the concurrent but independent actions of each robot leads to an *emergent social behavior*. The group behavior can be different from the individual behavior, emulating "group dynamics" or possibly "mob psychology." As will be seen in this chapter, fairly complex team actions such as flocking or forming a line to go through a door emerge naturally from reactive robots with little or no communication between each other. But as with emergent behavior in individual robots, emergent social behavior is often hard to predict. Complete architectures for designing teams of robots are still under development; Lynne Parker's ALLIANCE architecture[114] is possibly the most comprehensive system to date. The whole field of multi-agents is so new that there is no consensus on what are the important dimensions,

EMERGENT SOCIAL BEHAVIOR

or characteristics, in describing a team. For the purposes of this chapter, *heterogeneity, control, cooperation,* and *goals* will be used as the dimensions.[117]

8.2 Heterogeneity

HETEROGENEITY

HETEROGENEOUS TEAMS

HOMOGENEOUS TEAMS

Heterogeneity refers to the degree of similarity between individual robots that are within a collection. Collections of robots are characterized as being either heterogeneous or homogeneous. *Heterogeneous teams* have at least two members with different hardware or software capabilities, while in *homogeneous teams* the members are all identical. To make matter more confusing, members can be homogeneous for one portion of a task by running identical behaviors, then become heterogeneous if the team members change the behavioral mix or tasks.

8.2.1 Homogeneous teams and swarms

Most multi-agent teams are homogeneous swarms. Each robot is identical, which simplifies both the manufacturing cost and the programming. The biological model for these teams are often ants or other insects which have large numbers of identical members. As such, swarms favor a purely reactive approach, where each robot operates under the Reactive Paradigm. Insect swarms have been modeled and mimicked since the 1980's. The proceedings of the annual conference on the Simulation of Adaptive Behavior (also called "From Animals to Animats") is an excellent starting point.

An example of a successful team of homogeneous robots is Ganymede, Io, and Callisto fielded by Georgia Tech. These three robots won first place in the "Pick Up the Trash" event of the 1994 AAAI Mobile Robot Competition,[129] also discussed in Ch. 5. Recall that the objective of that event was to pick up the most trash (coca-cola cans) and deposit it in a refuse area. The majority of the entries used a single agent, concentrating on model-based vision for recognizing trash, cans, and bins and on complex grippers.

The three identical robots entered by Georgia Tech were simple, both physically and computationally, and are described in detail in a 1995 *AI Magazine* article.[19] The robots are shown in Fig. 8.1, and were constructed from an Intel 386 PC motherboard mounted on a radio-controlled toy tracked vehicle. The robots had a miniature wide-angle video camera and framegrabber. The flapper-style grippers had an IR to indicate when something was in the gripper. The robots also had a bump sensor in front for collisions. The robots were painted fluorescent green.

Each robot was programmed with a sequence of simple reactive behaviors (renamed here for clarity), following the reactive layer of Arkin's AuRA architecture described in Ch. 4:

wander-for-goal This behavior was instantiated for two goals: trash and trashcan. The motor schema was a random potential field, the perceptual schema was color blob detection, where trash="red" and trashcan="blue."

move-to-goal This behavior also had two different goals: trash and trashcan. The motor schema was an attractive potential field, and the perceptual schema for the trash and trashcan were the same as in the wander-for-goal.

avoid-obstacle This behavior used the bump switch as the perceptual schema, and a repulsive field as the motor schema.

avoid-other-robots The three robots did not communicate with each other, instead using only the repulsive field created by avoid-other-robots to reduce interference. The motor schema was a repulsive potential field (linear dropoff), while the perceptual schema detected "green."

grab-trash The robot would move toward the trash until the perceptual schema reported that the IR beam on the gripper was broken; the motor schema would close the gripper and back up the robot.

drop-trash When the robot reached the trashcan with trash in its gripper, the motor schema would open the gripper and back up the robot, and turn 90 degrees.

8.2.2 Heterogeneous teams

A new trend in multi-agents is *heterogeneous teams*. A common heterogeneous team arrangement is to have one team member with more expensive computer processing. That robot serves as the team leader and can direct the other, less intelligent robots, or it can be used for special situations. The danger is that the specialist robot will fail or be destroyed, preventing the team mission from being accomplished.

One interesting combination of vehicle types is autonomous air and ground vehicles. Researchers as the University of Southern California under the direction of George Bekey have been working on the coordination of teams of ground robots searching an area based on feedback from an autonomous

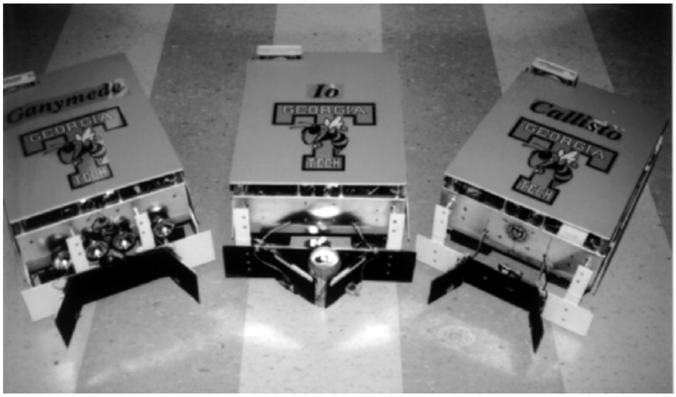
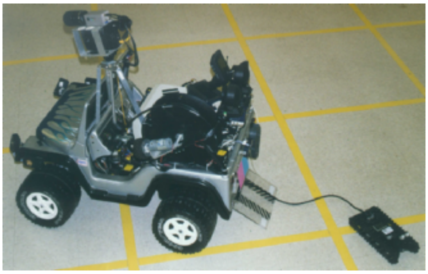


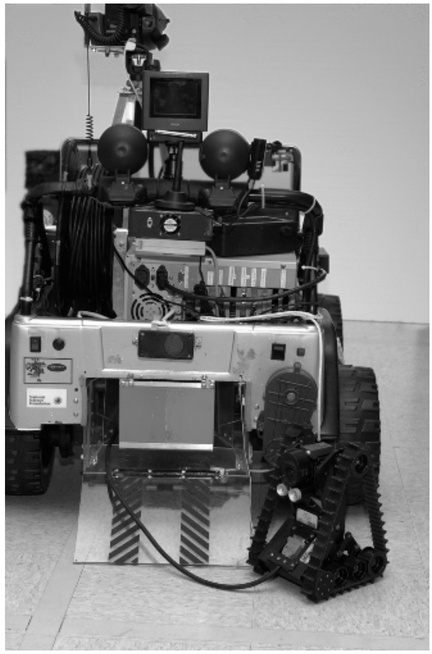
Figure 8.1 Georgia Tech's winning robot team for the 1994 AAAI Mobile Robot Competition, Pick Up the Trash event. (Photograph courtesy of Tucker Balch and AAAI.)

miniature helicopter. This combination permits the team to send a human observer a comprehensive view of a particular site, such as a hostage situation.

A special case of a cooperative, heterogeneous team of robots has been dubbed *marsupial robots.* The motivation for marsupial robots stemmed from concerns about deploying micro-rovers for applications such as Urban Search and Rescue. Micro-rovers often have limited battery power, which they can't afford to spend just traveling to a site. Likewise, micro-rovers may not be able carry much on-board processing power and need to have another, more computationally powerful workstation do proxy (remote) processing. A marsupial team consists of a large robot which carries one or more smaller robots to the task site, much like a kangaroo mother carries a joey in her pouch. Like a joey, the daughter robot is better protected in the pouch and can conserve energy or be recharged during transport. The mother can protect a delicate mechanism or sensor from collisions while it navigates through an irregular void. The mother can also carry a payload of batteries to recharge (feed) the daughter. It can serve as a proxy workstation, moving to maintain communications. The mother is likely to be a larger robot, while the daughter might be a micro-rover with sensors very close to the ground. The mother will have a better viewpoint and sensors, so in some circumstances it can communicate advice to the smaller daughter to help it cope with a "mouse's eye" view of the world. A teleoperator can also control



a.



b.

Figure 8.2 Two views of a marsupial robot team at University of South Florida. a.) Silver Bullet is the "mother" connected by an umbilical tether to a tracked chemical inspection robot Bujold, the "daughter." b.) Bujold exits from the rear of the jeep. (Photographs by Tom Wagner.)

the daughter more easily in some situations by looking through the mother's camera.

At this time, there appear to be only two physically realized implementations of autonomous marsupials: the University of South Florida teams, one of which is shown in Fig. 8.2, and the robots at the US Department of Energy's Idaho National Energy and Engineering Laboratory (INEEL).[3] The USF team is the only one where a mother robot carries a micro-rover inside the structure to protect it. The Mars Pathfinder mission is similar to a marsupial robot in that a micro-rover was transported to a mission site and the transport vehicle served as a support mechanism. However, our definition of marsupial assumes the mother is a fully mobile agent and can recover and retask the micro-rover.

8.2.3 Social entropy

The above examples show how different heterogeneous teams can be. One rough measure of the degree of heterogeneity is the *social entropy* metric created by Tucker Balch.[16] (Entropy is a measure of disorder in a system, especially in the sense of the Third Law of Thermodynamics. It was also adapted by Shannon for use in information theory to quantify the amount or quality of information in a system.) The point of social entropy is to assign a numerical value for rating diversity (or disorder) in a team. The number should be 0 if all team members are the same (homogeneous). The number should have the maximum value if all the team members are different. The number of team members which are different should make the overall number higher.

SOCIAL ENTROPY

To compute social entropy, consider a marsupial team $\mathcal{R}$ with a mother robot and three identical (hardware and software) micro-rovers. The formula for the social entropy, $Het(\mathcal{R})$, is:

$$Het(\mathcal{R}) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{8.1}$$

CASTES

There are two types of robots in the team, called *castes* or c: the mother and the daughters. Therefore $c = 2$. The term p_i is the decimal percent of robots belonging to caste c_i. If $i = 1$ for the mother, and $i = 2$ for the daughters:

$$\begin{aligned} p_1 &= \frac{1}{4} = 0.25 \\ p_2 &= \frac{3}{4} = 0.75 \end{aligned} \tag{8.2}$$

Substituting into Eqn. 8.1 (and remembering that $\log_2 n = \frac{\log_{10} n}{\log_{10} 2}$), the social entropy is:

$$\begin{aligned} Het(\mathcal{R}) &= -\sum_{i=1}^{c} p_i \log_2(p_i) \\ &= -(0.25 \log_2 0.25 + 0.75 \log_2 0.75) \\ &= -((-0.50) + (-0.31)) \\ &= 0.81 \end{aligned} \tag{8.3}$$

Now consider a case where the daughters are not identical. Suppose that one of the three micro-rovers has a different sensor suite and behaviors from the other two. In that case $c = 3$, where $p_1 = \frac{1}{4}$, $p_2 = \frac{2}{4}$, and $p_3 = \frac{1}{4}$. Substituting into Eqn. 8.1 yields 1.5. Since $1.5 > 0.81$, the marsupial team with the different daughter is more diverse that the marsupial team with all identical daughters.

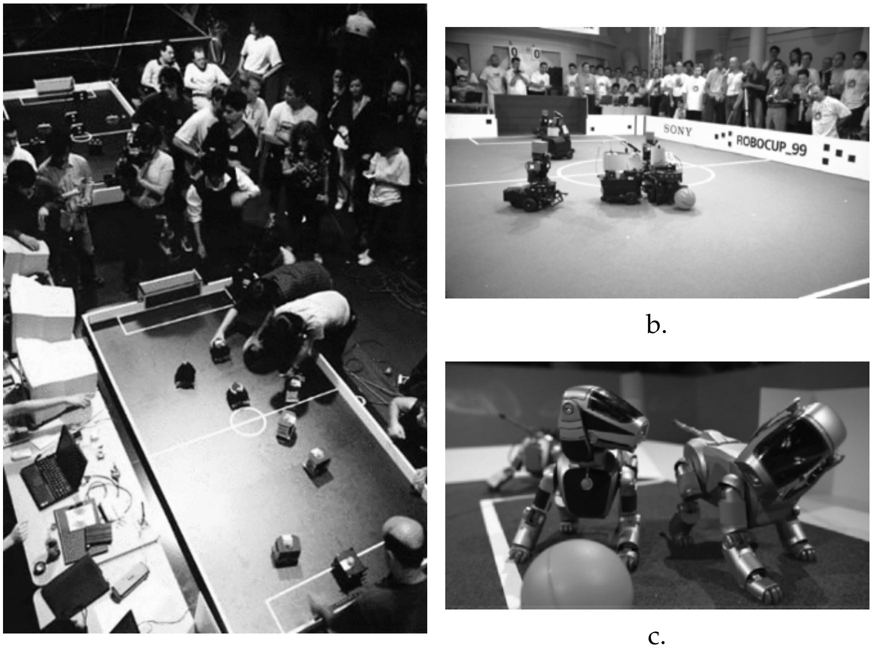
8.3 Control

CENTRALIZED CONTROL
DISTRIBUTED CONTROL

Control of multi-agents can fall in a spectrum bounded by *centralized control* and *distributed control* regimes. In centralized control, the robots communicate with a central computer. The central computer distributes assignments, goals, etc., to the remote robots. The robots are essentially semi-autonomous, with the centralized computer playing the role of a teleoperator in a teleoperated system. In distributed control, each robot makes its own decisions and acts independently. Of course, there is a range of regimes between fully centralized and fully distributed; the robots can interact with a central controller to receive new goals, then operate for the duration of the mission in a distributed manner.

Examples of full and partial centralized control can be found by comparing the RoboCup and MIROSOT robot soccer competitions. In those soccer competition events, teams of robots are controlled remotely by a central computer. In the small sized league of RoboCup and MIROSOT, teams of three, very small self-contained robots (7.5cm x 7.5cm x 7.5cm) play on a 130cm x 90cm arena with an orange golf ball serving as the miniature soccer ball. Each robot had a unique pattern of bright colors to make it visible from the overhead cameras, and the overhead camera is connected to a central processor. The robots communicate with the central processor over a radio link. In MIROSOT, the central processor commands each robot by supplying the direction to move. In RoboCup, the central processor can give either explicit directions or just locations of other robots and the ball, letting the robot's on-board behaviors generate the (one hopes) correct response. Fig. 8.3 shows a view of the small-sized league from the 1998 RoboCup World Cup.

MIROSOT robots are more drone-like than their RoboCup counterparts, since they are not required to carry any on-board sensing. They represent the extreme of centralized control, where everything must go through a single computer, much like the battle-droids in the *Star Wars* movie, *The Phantom Menace*. RoboCup robots are required to have some type of on-board



a. b. c.

Figure 8.3 RoboCup soccer competition: a.) Overhead view of the RoboCup playing field and teams for the small-sized league (overhead camera is not shown), b.) Mid-sized league, and c.) Legged league using Sony Aibo robots. (Photographs ©The RoboCup Federation 1999. All rights reserved.)

sensing for reflexive obstacle avoidance. In the RoboCup case, the robots must have a set of basic tactical behaviors (see Ch. 7), but may either receive strategic commands from the central computer or have on-board strategic behaviors. This type of control is conceptually equivalent to the Hybrid Reactive-Deliberative Paradigm, where the reactive layer physically resides on the robot and the deliberative layer resides on the central workstation.

Distributed control is more natural for soccer playing than centralized control, because each player reacts independently to the situation. An example of distributed control in robot soccer playing is the mid-sized league in RoboCup. Notice that in robot soccer the robots are inherently heterogeneous. Although they may be physically the same, each robot is programmed with a different role, most especially Goalie, Striker, and Defender.

Likewise, there are significant advantages to planning and learning strategies, two deliberative functions. Manuela Veloso and Peter Stone have used RoboCup as a test domain for research in deliberation.

8.4 Cooperation

COOPERATION
ACTIVE COOPERATION

Cooperation refers to how the robots interact with each other in pursuing a goal. Robots can show *active cooperation* by acknowledging one another and working together. Note that this does not necessarily mean the robots communicate with each other. For example, in robot soccer, one robot can pass the ball to another robot as part of an offensive play. The cooperation does not require communication—if a robot has the ball, can't see goal and can see team mate, then it passes to team mate, but this does require being aware of the teammates.

NON-ACTIVE COOPERATION

More often robots are programmed to exhibit *non-active cooperation*, whereby they individually pursue a goal without acknowledging other robots but cooperation emerges. The choice of cooperation schemes is often influenced by the sensory capabilities of the robots. Active cooperation requires that robot be able to distinguish its peer robots from other aspects of the environment. In the case of the Georgia Tech entry, each robot was covered in fluorescent green poster paper easily segmented as a color region. If the robots had not been green, they would have been treated as obstacles to be avoided. Non-active cooperation has attracted much interest in the robotics community because it requires very little sensing or behaviors.

PHYSICAL COOPERATION

RECONFIGURABLE ROBOTS

It is easy to think of cooperation in terms of robots working together on a task. Another aspect of cooperation is *physical cooperation*, where the robots physically aid each other or interact in similar ways. Marsupial robots are certainly a type of physical cooperation, especially during deployment and docking. An even more exciting type of cooperation occurs between *reconfigurable robots*. One of the first such systems was proposed by Toshio Fukuda, called CEBOT for "celluar robot system."[31] These are small identical robots that hook up to form a useful robot. Another aspect of reconfigurable robots is *cooperative mobility*, where one robot might come over and help another robot in trouble. Shigeo Hirose simulated robots which could link up with each other to gain more stability or traction in rough terrain.[67]

COOPERATIVE MOBILITY

8.5 Goals

The final dimension for characterizing a collection of multi-agents is how the robot works on a goal. If all the robots in the collection work on attaining the same explicit goal, then they are said to share a single goal, versus having individual goals.

An example of robots working a single goal is the winning team for the Office Navigation event in the 1996 AAAI Mobile Robot Competition.[78] The office navigation event had a robot that was supposed to search a series of rooms, find an empty conference room, and then go to a list of rooms where people were and tell them that a meeting was going to begin in the empty conference room. The event was originally conceptualized as a single agent task, but the SRI entry under the direction of Kurt Konolige consisted of three robots.[62] Each of the three robots ran the Saphira architecture (see Ch. 7) and were coordinated by a central workstation. While the robots were responsible for autonomous navigation, their goals were set by the central strategy agent. Even though they were navigating through different parts of the office maze, the robots were working on a single goal and the software agents on the central workstation were explicitly coordinating the actions of the robots. The robots were able to find an empty room and inform the attendees in 4 minutes and 30 seconds. The next best time was close to 10 minutes.

An example of purely reactive robots working on individual goals is a problem originally posed by Ron Arkin:[9] a group of robotic space "ants" foraging for mineral rich asteroids or Near Earth Objects (NEOs). If each robot in the group forages for its own asteroid, then they have individual goals. (Notice that a behavior that permits them to notice other robots and be repulsed will help disperse the robots.) If the robots are programmed so that they will all go to one specific asteroid, then they share a common goal.

Emergent cooperation is not the same thing as having a single goal. For example, suppose the robotic space ants are programmed to go to the nearest *non-moving* asteroid and bring it back to base. Each robot might have a set of behaviors: find-stationary-asteroid, move-to-asteroid, push-asteroid-to-home, and avoid-robots. The find-stationary-asteroid could be done with a random potential field (in 3 dimensions, of course). An attractive "asteroid-tropic" potential field could be used for the move-to-asteroid behavior. Likewise an attractive field could be used for the push-asteroid-to-home behavior, where the robot tries to stay behind the asteroid as it moves to home rather than avoid the asteroid. Avoid-robot could be done with a repulsive field. These behaviors give the robots individual goals, since there is no awareness of the goals of the other team members.

Now consider what happens when a robot ant encounters an asteroid it can't move. The robot stays there pushing. Eventually another robot will come along because the asteroid is not moving. As it is attracted to the "dark side" of the asteroid, it will come into range of the first robot. What happens? The avoid-robot behavior should be instantiated, causing the first robot to move over a bit. The second robot will also feel a repulsive force and slow down. As the first robot moves out of the way, the angle of repulsion changes, forcing the second robot to move sideways as well, as it continues to move to the asteroid. Together, the interaction between the two robots should cause them to naturally balance themselves behind the asteroid and push together. The point is that the robots were not explicitly directed to all work on the same NEO; they were each directed to find their own NEO, but circumstances led them to the same one.

8.6 Emergent Social Behavior

The examples of heterogeneity, cooperation, control, and goals give some hint of how an overall social behavior emerges from the actions of autonomous robots. The robot teams often are the result of extensive design efforts, where the teams aren't too large to interfere with each other, and are optimally sized for the particular task, etc. Many researchers are exploring the issues of what happens when the designer doesn't have a choice about the size of the robot population. How do social behaviors emerge in those cases? And how can social rules or conventions be established to make the team self-regulating and productive? This section summarizes two approaches: creating social rules for the robots to follow, and allowing internal motivation to cause the robots to adapt their behavior to problems.

8.6.1 Societal rules

Maja Mataric has focused her research on how group dynamics might emerge in herds of multiple agents operating under fully distributed control. She explored the impact of density and the impact of societal rules on overall team performance.[90] Each IS Robotics R2 robot was programmed with behaviors using the Subsumption architecture. She set up a scenario where up to 20 identical robots (now known as "The Nerd Herd") were given the same location as a goal. The goal, however, was on the other side of a partition with a narrow door, permitting only one robot to pass through the partition at a

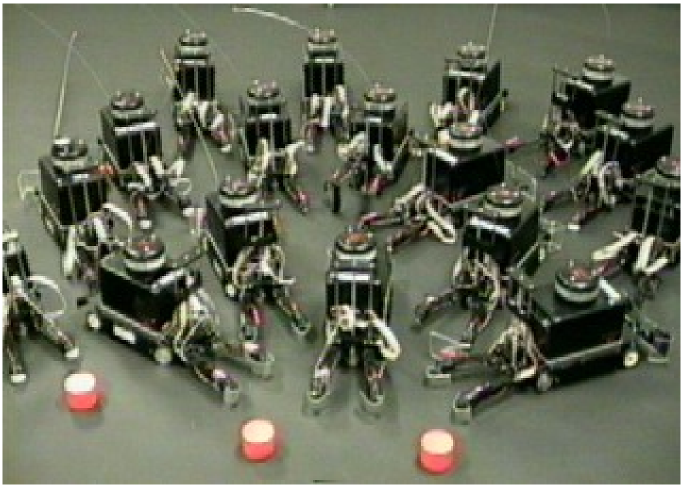


Figure 8.4 The Nerd Herd. (Photograph courtesy of USC Interaction Laboratory.)

time. The robots were placed randomly on the same side of the partition and started moving at the same time.

IGNORANT COEXISTENCE

In the first set of demonstrations, the robots functioned with *ignorant coexistence*. The robots coexisted in a team, but did not have any knowledge of each other. A robot treated another robot as an obstacle. Each robot had the equivalent of a move-to-goal and an avoid-obstacle behavior. Since robots were treated as obstacles, once the robots gathered at the opening, they spent most of their time avoiding each other. The team as a whole made slow progress through the door to the goal location. Worse yet, the larger the number of robots fielded, the larger the traffic jam, and the longer to get all the team members through.

INFORMED COEXISTENCE

In the second demonstration, *informed coexistence*, the robots were allowed to recognize each other and given a simple social rule governing inter-robot interactions. In addition to move-to-goal and avoid-obstacle, a third behavior was created for avoiding robots. If a robot detected another robot, it would stop and wait for time p. If the blocking robot was still in the way after p, the robot would turn left and then resume moving to the goal. The result of the new behavior was to reduce the traffic jams, and the group got through the door in about the same time as a single agent going back and forth through the opening 20 times.

INTELLIGENT COEXISTENCE

The real surprise came in the third demonstration, *intelligent coexistence*.

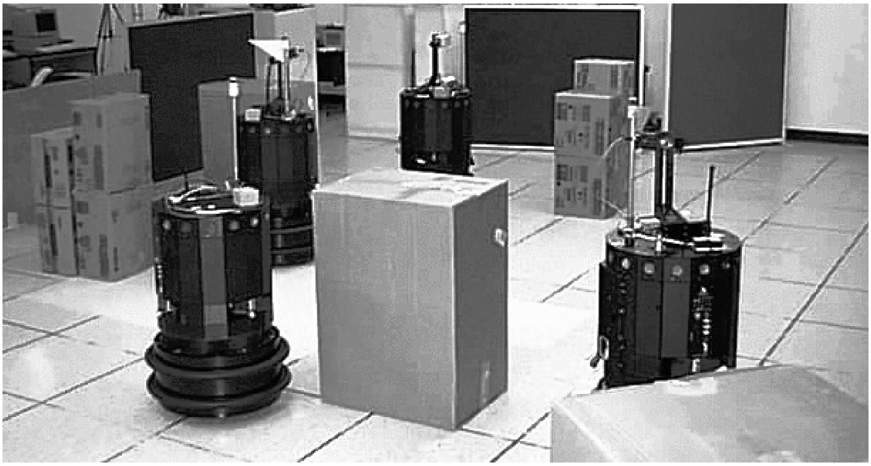


Figure 8.5 Robots cooperatively tracking an object under the ALLIANCE architecture. (Photograph courtesy of Oak Ridge National Laboratories.)

The social behavior for avoiding robots was replaced with another heuristic: the robots were repulsed from other robots, but as it moves away, it tries to move in the same direction as a majority of other robots. (Each robot broadcast its heading over a radio transmitter to compensate for the inability to recognize each other by vision or sonar, so that isn't considered communication.) As a result, the robots exhibited a flocking behavior and went through the door in single file! The need to go in the same direction created a tendency to form a line, while repulsion caused the robots to essentially create spaces for robots to merge into line. Together the two effects created a strong need to go through the door single file, even though there was no such explicit direction. Not only were traffic jams reduced, but the overall task was accomplished faster.

8.6.2 Motivation

In Mataric's work, the robots reduced interference through simple social rules with no communication, but the members of the team could not actively help out failed colleagues or change tasks dynamically. Lynne Parker has attempted to address the larger issues of robustness and fault tolerance with the ALLIANCE architecture,[114] an outgrowth of the Subsumption architecture. The central idea is that members in the team can either observe or "hear" the progress of others in teams, as well as their own progress. If they get frustrated with their own progress, they should stop what they're doing and move on to something else. Likewise, if a robot is free and another robot has been unable to accomplish a task, it should try to complete the unfinished task. This is particularly useful for tasks where there is a logical sequence of behaviors, where all of a particular task (like dusting) needs to be done for an area before the robots begin working on another task (e.g., sweeping). These changes in behaviors are regulated by a simple mechanism: *motivation*. The motivation of a robot to do a task is regulated by two internal motivations, *robot impatience* and *robot acquiescence*. The more frustrated a robot with gets with another robot's performance on t_i, the higher the impatience associated with that task t_i. Likewise, the more frustrated a robot gets with its own performance for a task, the higher the acquiescence. If the frustration threshold is exceeded, then the robot either takes over the unfinished task or abandons its current task and changes behavior.

MOTIVATION

ROBOT IMPATIENCE

ROBOT ACQUIESCENCE

Fig. 8.6 shows the time trace for an example of motivation for two space ants foraging for asteroids. (This example isn't really a sequential series of tasks in the manner used by ALLIANCE, but this conveys the elegance of motivation.) In this case, the reactive space ants have to either broadcast what they're doing or be able to perceive the other's progress. This makes it a bit different than the "no communication" approach. At time 0, both robots start by looking for asteroids. (We assume there is no frustration for the find task.) Both see asteroid A1, but Robot 1 is the first there. Robot 1 has now taken responsibility for Task 1 (T1), pushing A1 to home. Even though A1 is still stationary at time 3, Robot 2 does not join in as it would in the no-communication method. Instead, it begins to accrue impatience about T1. Once Robot 1 begins to push A1, it starts accruing frustration in the form of acquiescence. As with the no-communication example, a single robot cannot push the asteroid.

While Robot 1 is trying to push asteroid A1, Robot 2 sees and moves to asteroid A2. All the while its impatience over T1 is growing. At time 7, Robot 2 is trying unsuccessfully to push asteroid A2 (task T2) and its acquiescence counter is increasing. Also at time 7, Robot 2's patience with Robot 1 and task T1 has been exceeded. It pushes T1 onto its stack of things to do when it completes its curent task. Meanwhile, at time 9, Robot 1 gives up on T1. Although it is frustrated with Robot 2, it assumes that T2 is still under control and so begins to forage again. Finally, at time 10, the frustration over T2 reaches the limit and Robot 1 is free to help Robot 2.

time	Robot 1	Robot 2
0	find-stationary-asteroid	find-stationary-asteroid
1	sees A1	sees A1
2	move-to-asteroid(A1)	move-to-asteroid(A1)
3	arrives at A1	resumes find-stationary-asteroid
4	push-asteroid-to-home(A1) T1-acquiescence++	find-stationary-asteroid T1-impatience++
5	push-asteroid-to-home(A1) T1-acquiescence++	sees A2 T1-impatience++
6	push-asteroid-to-home(A1) T1-acquiescence++	move-to-asteroid(A2) T1-impatience++
7	push-asteroid-to-home(A1) T1-acquiescence++ T2-impatience++	push-asteroid-to-home(A2) T1-impatience>limit put T1 on stack T2-acquiescence++
8	push-asteroid-to-home(A1) T1-acquiescence++ T2-impatience++	push-asteroid-to-home(A2) A1-impatience++ T2-acquiescence++
9	T1-acquiescence>limit gives up on T1 find-stationary-asteroid T2-impatience++	push-asteroid-to-home(A2) T2-acquiescence++
10	T2-impatience>limit now attempts T2 move-to-asteroid(A2)	T2-acquiescence++
11	push-asteroid-to-home(A2) T2-acquiescence = 0	push-asteroid-to-home(A2) T2-acquiescence = 0
12	arrives at HOME	arrives at HOME

Figure 8.6 Example of how the internal motivation in ALLIANCE might be extended to work with two space ants.

8.7 Summary

In summary, many tasks favor the use of many cheap robots rather than a single expensive one. These collections of multiple robots are often referred to as multi-agents. Individual robots in a multi-agent team are generally programmed with behaviors, most often as purely reactive systems, but occasionally with a hybrid architecture. As with an overall behavior emerging on a single reactive agent, societies of reactive agents often exhibit an emergent societal behavior.

Multi-agent societies can be characterized according to where they fall on at least four dimensions (since multi-agent theory is relatively new, even for robotics, new dimensions may surface over time). Heterogeneity refers to whether the member robots are identical in software and hardware. Cooperation may be either active or non-active, while control may fall in the spectrum from fully distributed to fully centralized. A robot society may have a single, explicitly shared goal or each robot may have its own goal. When communication between agents is appropriate is a pervasive, open question.

From a practical side, the emphasis in multi-agents has been on how favorable group dynamics emerge from teams of homogeneous, purely reactive robots while operating under distributed control. Problems in emergent societal behaviors such as interference and the need to adapt to the open world can often been addressed by specifying social rules and internal motivation. However, more interest is emerging in robots that have either heterogeneous software or hardware capabilities, such as marsupial and reconfigurable robots. The diversity of a heterogeneous team can be captured somewhat by the social entropy metric.

8.8 Exercises

Exercise 8.1

Give three reasons why multi-agents are desirable. Describe the general attributes of applications which are well-suited for multi-agents, and give one example.

Exercise 8.2

Define the following:

a. heterogeneity

b. control

c. cooperation

d. goals

Exercise 8.3

Consider the example of space ants. What would happen if the first robot communicated with the other robots to recruit them to help move the asteroid? Would the behaviors or the goal structure necessarily change? Why or why not?

Exercise 8.4

Draw a FSA or write a script to coordinate the sequencing of the Pick Up the Trash behaviors for Io, Ganymede, and Callisto.

Exercise 8.5

Describe three approaches to societal behavior: social rules, internal motivation, and leadership.

Exercise 8.6

Were the behaviors for the Nerd Herd purely reactive? Why or why not?

Exercise 8.7 [*Programming*)]

Implement the space ant example with 3-5 robots capable of phototaxis and dead reckoning.

a. Multi-agent foraging. Start with only a phototropic and avoid-robot behavior, where a robot is an obstacle that isn't a light. The program will start with an empty world consisting of a light (you may need to make a "bigger light" by placing lights next to each other). Between 2 and 5 phototropic robots will be placed at different random starting locations in the world. Each will wander through the world, avoiding obstacles, until it comes to a light. Then it will move directly to the light (attractive field). If more than one robot is attracted to the same light, they should center themselves evenly around the light. Compare this with the program in Ch. 5 which had a single robot forage for lights. Which gets more lights faster?

b. Cooperating to bring the food home. Now add the push-to-home behavior where the robot wants to be on a straight line behind the light to home. What happens now?

Exercise 8.8 [*World Wide Web*]

Visit the RoboCup web site at www.robocup.org. Which team has performed the best over the past 3 years? Describe the multi-agent organization in terms of control and cooperation.

Exercise 8.9 [*Advanced Reading*]

Read Ed Durfee's humorous invited paper on DAI, "What Your Computer Really Needs to Know, You Learned in Kindergarten" (proceedings of the *Tenth National Conference on Artificial Intelligence*, 1992). For each of his ten issues ("Share Everything," "Play Fair," etc.), describe how this applies to robots. For each issue give an example of how it applies to a robot team described in this chapter.

Exercise 8.10 [*Advanced Reading*]

Read and summarize "Behavior-Based Formation Control for Multirobot Teams," by Tucker Balch and Ron Arkin in *IEEE Transactions on Robotics and Automation*, vol. 14, no 6., 1998.

Exercise 8.11 [*Science Fiction*]

Watch the movie "Silent Running" about a team of three mobile robots (Huey, Dewey, and Louie) working on a space station. Classify their teamwork in terms of heterogeneity, control, cooperation and goals.

8.9 End Notes

For further reading.
Chapter 9, "Social Behavior," of *Behavior-Based Robotics* by Ron Arkin has a detailed and comprehensive presentation of the ethology, philosophical considerations, and different robot architectures for multi-agents. It is well worth reading.

Swarms and flocks.
The references to swarm robots are too numerous to cite here; many papers explore details of insect behavior and coordination strategies as well as provide simulations. Jean-Louis Deneubourg has produced many interesting articles synthesizing insights form insect colonies into a form useful for programming mobile robots. As noted in *Behavior-Based Robotics*, Craig Reynolds' work in computer graphic simulation of flocks of in "Flocks, herds, and schools: A distributed behavior model," in *Computer Graphics*, 1987, showed how flocks emerge from simple, individual interactions.

"Fast, Cheap and Out of Control: The Movie."
The term "Fast, Cheap and Out of Control" later became the title of a 1997 award-winning documentary by Errol Morris on four men, including Rodney Brooks. The movie title implied that Morris saw human kind shifting from highly individualistic relations with the world developed over time (as seen by the lion tamer and topiary gardener) to decentralized, reactive mobs. Although the movie is not about robotics per se, it features interviews with Brooks and contains stunning shots of some of Brooks' robots walking over broken glass, shining like diamonds in the bright lights. Maja Mataric, one of Brooks' students at the time of the filming, can be seen in one of the shots wearing shorts.

Languages for multi-agents.
Researchers are beginning to work on languages for multi-agent coordination, including Holly Yanco and Lynn Stein at MIT.

Robot soccer.
There is some dispute over which competition was the first robot soccer competition: MIROSOT or RoboCup. RoboCup was originally proposed by Minoru Asada, a noted Japanese researcher in visually guided mobile robots, in 1995 for the 1997 IJCAI, giving researchers two years to prepare. Hiroaki Kitano has been responsible for much of the organization of the competition and funding from the Sony Corporation (among others). MIROSOT was started by Jong-Kwan Kim in Korea in 1996, a year after RoboCup was announced but a year before the first official RoboCup game.

Robot name trivia.
More robots are being named after women these days. The robots in Lynne Parker's laboratory at Oak Ridge National Laboratory are named after women pioneers in

computer science. Rogue at Carnegie Mellon University was named after the woman mutant from the *X-Men* comics (to complement the robot Xavier, named after the wheeled super hero). The robots in my lab at the University of South Florida are named after women science fiction authors. (Except one robot which was named after the Coors beer, Silver Bullet, while I was out of town.)

More robot name trivia.
The Nerd Herd consists of IS Robotics R2 robots which look like toasters. Brightly colored toasters, but toasters nonetheless. The 20 robots are named for things that come out of toasters, for example, Bagel.