Performance Evaluation of Multiple Robots in a Search and Retrieval Task

Paul Rybski, Amy Larson, Michael Lindahl, and **Maria Gini**
Department of Computer Science and Engineering
University of Minnesota
200 Union Street SE, Minneapolis, MN 55455
{rybski, larson, lindahl, gini}@cs.umn.edu

Abstract

In this paper we study the performance of multiple robots at a search and retrieval task. The robots have all the same capabilities and perform the same task without any explicit communication. The sensing capabilities of the robots are quite limited, yet the robots succeed at performing the task. We show how the performance is affected by the number of robots working in the same area, the distribution of the targets the robots are searching for, and the complexity of the environment.

Introduction

Cooperative multiple robots can often accomplish a task that is difficult, if not impossible, for a single robot. When approaching a problem involving coordinated behavior, the challenge arises out of determining the strategy that will maximize performance. Strategy considerations include: explicit versus implicit coordinated behavior, origin of control (central or decentralized), extent of communication, heterogeneous versus homogeneous abilities, and individual motivation (selfish or socialized). Despite the growing interest in multiple robots, only a handful of experimental studies exist which assess performance relative to cooperative and coordinative strategies.

For performance evaluation relative to strategy, the research questions we address in this paper are:

- How does the complexity of the environment affect performance? For instance, does the presence of extraneous obstacles suggest a different cooperation strategy?
- How does the number of robots operating in the same area affect performance? For instance, how does an increase in population increase the chances of unwanted interference. Mataric (Mataric 1995b) shows how the performance of a group of multiple robots performing a clean-up and collection task declines as the number of robots increases.
- How does the distribution of the targets in the environment affect performance? For instance, do simple reactive behaviors work best with random distribution of targets?

The tasks traditionally studied with multiple robots and behavior-based control include foraging which involves searching and retrieving items from a given area, box-pushing which involves moving an object between two locations, formation marching (Balch & Arkin 1995) which involves moving while maintaining a fixed pattern, and various forms of military surveillance (Parker 1996). Another classic task involves janitorial services (Parker 1996) where robots clean a room of an unfamiliar building by emptying the garbage, dusting the furniture, and cleaning the floor.

Here, we propose a task of search and retrieval whereby robots locate, collect, and return targets to a home base. Robots are homogeneous and perform independently with a localized goal of target retrieval without the aid of communication. (Robots and targets are shown in Figure 1.) The task is a simplified version of mine-field clearing where mines are localized using close-proximity sensors such as magnetometers, or of a find-and-rescue task where robots find and retrieve specific targets such as those dropped by air.

Due to the limited sensing capabilities of the robots used in these experiments, completion of the task is especially challenging. Despite these limitations, the robots are capable of successfully retrieving the targets, as we will show later in the paper.

Related Work

Most research with multiple robots has focused on collaborative work, and has taken various forms and approaches as detailed in the extensive survey by (Cao, Fukunaga, & Kahng 1997). The recent special issue of *Autonomous Robots* devoted to Robot Colonies and the book edited by Arkin and Bekey (1997) attests to the vitality of the field.

Most research in the achievement of a common goal using robot collaboration can be categorized into three broad groups:

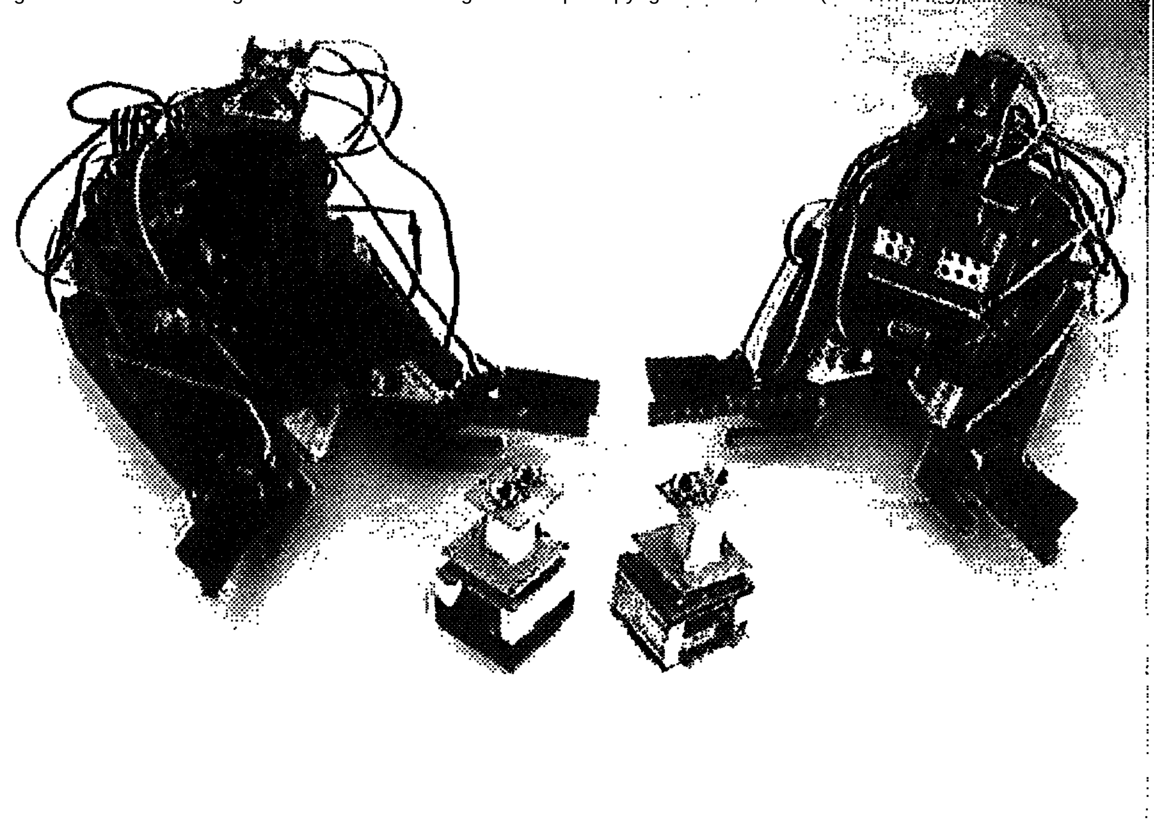

Figure 1: A picture of two robots and two targets

1. Problems where ***cooperation is essential to achieve the task***, such as cooperating to push a large, heavy object. Although each robot is capable of performing a useful task, the overall task accomplished by the group could not be achieved by a single robot. This kind of intentional cooperation has been demonstrated both without explicit communication (see, for instance (Sen, Sekaran, & Hale 1994)) and with communication (Rus, Donald, & Jennings 1995; Mataric 1997; Sasaki *et al.* 1995). Other problems where cooperation is essential are those faced by teams where each member plays a different role, as, for instance, players on a soccer team (Stone & Veloso 1997). Tambe (1997) has proposed methods for agent teams working in synthetic domains where individual team members have their own reactive plans, but are also given team plans that explicitly express the team's joint activities.

2. Problems where ***cooperation increases performance*** either by decreasing the time to complete the task, or by increasing the reliability. Sample tasks include cleaning up trash or mapping a large area. This type of cooperation depends most often on communication for information sharing (MacKenzie, Arkin, & Cameron 1997; Mataric 1997; Parker 1996; Schneider Fontan & Mataric 1996), but has also been demonstrated without explicit communication (Arkin 1992; Beckers, Holland, & Deneubourg 1994). However, even simple communication has been shown to increase substantially the performance of robots when foraging, consuming, and grazing (Balch & Arkin 1994). Alami (1995) has shown coordination of navigation of multiple robots through a plan-merging paradigm for the task of transporting containers in harbors.

3. Problems where ***cooperation emerges as a result of interactions***. A significant body of research in cooperative mobile robotics deals with the study of large numbers of homogeneous robots (swarms). When many simple robots are brought together, globally interesting behaviors can emerge as a result of the local interactions of the robots. A key research issue is determining the proper design of the local control laws that will allow the collection of robots to produce the desired behaviors. For instance, Mataric (1995a) describes group behaviors such as dispersion, aggregation, and flocking. Agah and Bekey (1995) studied the effect of com-

munication of simulated robot colonies in fetching and carrying tasks. Behaviors of colonies of insect species are called "eusocial" (McFarland 1994), to contrast them with "cooperative" behaviors shown by higher animals where cooperation is the result of interactions of selfish agents aimed at maximizing individual utility.

Many approaches have been proposed for designing mechanisms for agent interactions. Most mechanisms require explicit coordination and negotiation, but direct communication is often replaced by indirect communication via sensing or via the environment as, for instance, in (Arkin 1992; Beckers, Holland, & Deneubourg 1994).

Conventions are a straightforward way of achieving coordination in a multi-robot system. In general, designing all necessary conventions is difficult and perhaps intractable (Shoham & Tennenholtz 1995), but it has been shown that conventions can be reached without centralized control if agents interact and learn (Shoham & Tennenholtz 1992). Shoham proposes a set of social laws for a group of idealized mobile robots that allow them to move on a grid without colliding and interfering. The social laws are basically traffic laws that control how the robots move, what they should do when they get to junctions, etc. The assumption is that the robots are homogeneous and law abiding. Wang (Wang 1995), for instance, proposes several functional primitives for traffic control in distributed robotic systems.

Experimental Description

Many factors determine the effectiveness of a cooperative multi-robotic solution to a search and retrieval problem. Three such factors include the number of robots used, the physical distribution of the targets, and the obstacle density in the environment. The purpose of this paper is to study how the overall success of a robotic team is affected by altering these environmental factors.

The task set before the robots was that of simple search and retrieval. The robots start from a fixed location in the environment, search the area for targets and return them to the home base.

Experiments were run on different numbers of robots, ranging from one to four. Each experiment on a collection of robots was run four times and the results were averaged. The distribution of targets in the arena varied from a uniform spread to tightly clustered groups. Finally, some experiments added obstacles to the environment to impede the robots' paths. These obstacles occupied roughly 25% of the arena's area.

Two different environments were constructed for use in the experiments. The first was a circle four meters in diameter (Figure 2) and the second was an oblong area roughly seven meters long and six meters wide (Figure 3). For both sets of experiments we used a single landmark, a 100W incandescent light bulb. The robots used the landmark to search the arena and home in on the drop-off zone.

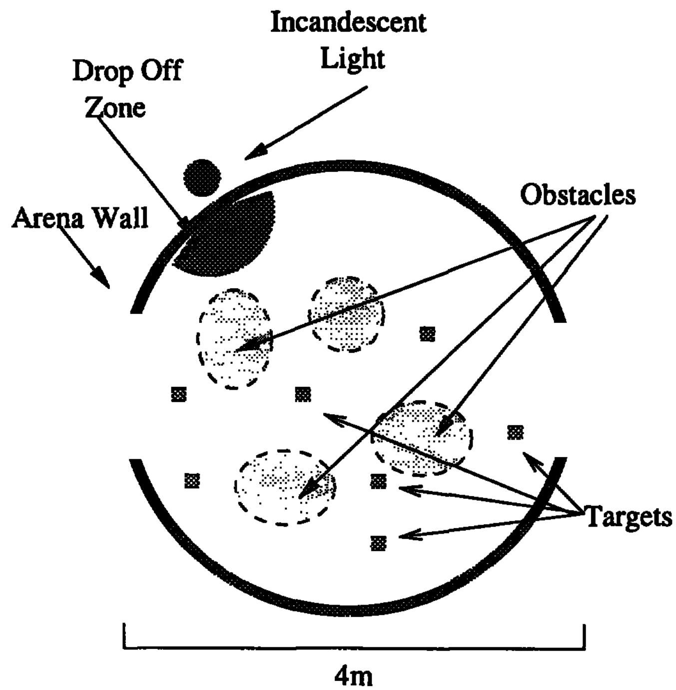

Figure 2: The small arena used for the experiments

For each of the experiments, robots were only allowed to gather targets for 10 minutes. The times at which the robots delivered a target to the landmark light bulb were recorded.

The targets can be detected at a range of about two to three feet with a 60 degree angle of sensitivity, so the robots must actively roam around the arena in order to locate them. The robots are currently not explicitly aware of each other's presence. If two collide, they simply treat each other like obstacles and position themselves along an unimpeded path.

The robots are programmed to only react to their immediate environment, thus they retain no internal map and use no previous knowledge about their environment. The robots started out at the drop-off zone and moved away from it, using it for very rudimentary tracking and homing.

Robotic Hardware

The robots are constructed out of LEGO Technic blocks. LEGOs were used because they are lightweight, strong, easy to work with, and are ideal for rapid prototyping and proof-of-concept designs. The University of Minnesota AIRVL has successfully been making use of LEGOs as a robot fabrication material for several years.

The chassis is a dual-treaded design, giving it holonomic mobility. To grasp a target, each robot is equipped with a single degree of freedom forklift-style gripper, capable of lifting a target off the ground to

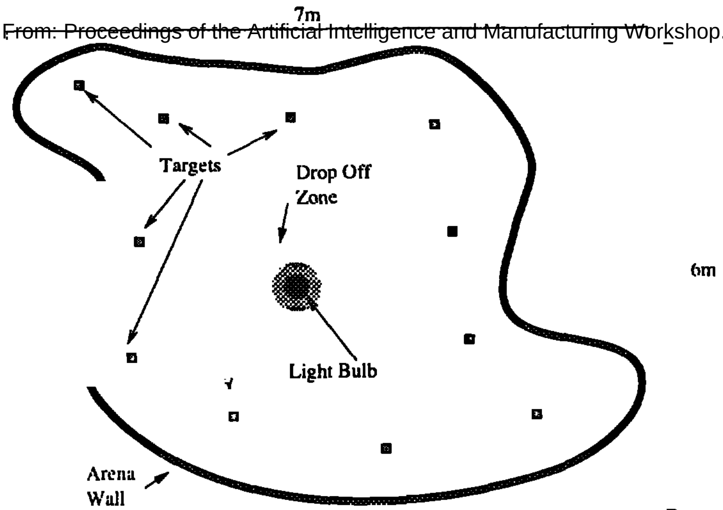

Figure 3: The large arena used for the experiments

transport it. A single infrared break-beam sensor is mounted between the tines of the gripper to let the robot know that it has a target in its grasp. The gripper is small enough so that only a target will trigger its grasp. Some details of the robot hardware are shown in Figure 4.

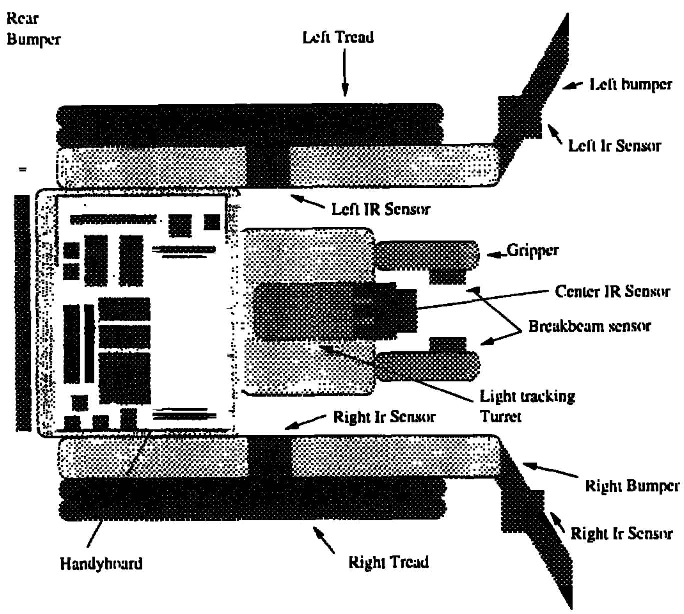

Figure 4: The robot design

For obstacle avoidance, a set of feelers/bumpers are located just beyond the front of the robots' treads. These bumpers register collisions from the front-left and front-right as well as from each side. They also serve as a funnel to guide targets into the gripper. A rear bumper allows the robot to detect collisions from behind.

The targets that the robots attempt to locate are also constructed out of LEGOs and transmit a continuous stream of 40 KHz infrared light. In order to detect the targets, the robots are outfitted with five photodetectors sensitive to this frequency and wavelength. Three of these detectors were fitted with lenses and are mounted on the front of the robot. The lenses decreased the angle of sensitivity and thus made the sensors very directional. This allows the robot to orient itself properly when attempting to grab a target. The remaining two sensors are mounted to detect targets on the sides of the robot.

In order to track the visible-light landmarks, a turret-mounted set of cadmium-sulfide (CdS) photosensitive resistors was used. The turret was located high enough to see over any obstacles and was thin enough not to obscure another robot's line of sight to a landmark.

The on-board computer was Fred Martin's Handyboard, a MC68HC11-based microcontroller with 32K of RAM (Martin 1998). The AIRVL uses this particular microcontroller almost exclusively in concert with its LEGO robots because of its expansive software libraries and ease of device interface capabilities. Development of the software for the Handyboard was done using Interactive-C (Wright, Sargent, & Witty 1996), a specialized subset of C which features libraries specialized for the Handyboard's hardware and more exotic facilities like explicit multi-tasking capabilities.

Robot software

Control of the robot is achieved through a finite-state machine (FSM) sequencer. In order to best solve the search and retrieval task, it was broken down into logical units where each unit was assigned to a single state of the FSM to solve it. Each state consists of a set of behaviors running in parallel, similar to a subsumption-style (Brooks 1986) method of programming. Each state's behavior is responsible for handling one segment of the robot's control code, mapping a sensor set to an actuator command. In order to resolve conflicts, each behavior is given a unique priority. When a behavior is activated by the triggering of a sensor, (such as when the collision detection behavior notices that a bumper has been depressed), it begins transmitting commands to the actuator control processes. Behaviors with higher priorities take precedence over behaviors with lower priorities. This ensures that certain actions which are more critical to the survival of the robot will be accomplished before others. However, some care is needed to insure that unintentional and unnecessary robot actions do not emerge from the complex interaction of these groups of behaviors.

As seen in Figure 5, State 0 is responsible for searching the arena for a target. This state consists of the following behaviors: Grab Target, Obstacle Avoidance, Seek Targets, and Search Arena. Grab Target is responsible for stopping the robot once a target enters its gripper. Obstacle Avoidance moves the robot out of the way when it collides with an obstacle or another robot.

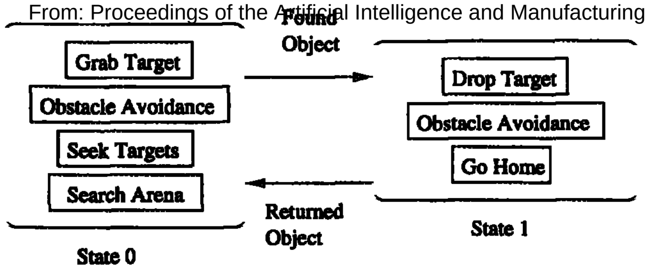

Figure 5: The finite state machine program

Seek Targets uses the front infrared detectors to line the robot up with a target for grasping. Search Arena uses the landmark as a reference for randomly navigating about the arena as well as rotating the robot to face a target when detected.

State 1 is responsible for taking the robot back to the landmark when it captures a target. This state consists of the following behaviors: Drop Target, Obstacle Avoidance and Go Home. Drop Target is responsible for dropping a target off when it gets to the appropriate zone. Obstacle Avoidance is used in the same fashion as in state zero. Finally, Go Home uses the landmark to point the robot in the direction of the drop-off zone.

Experimental Data

Several sets of experiments were run to analyze the effects of various environmental factors on the robot's performance. The first experiment analyzed how varying the number of robots affected the rate at which targets were gathered. This experiment took place in the smaller arena containing seven uniformly distributed targets and no obstacles. The average number of targets retrieved at the end of each minute was examined. These results are shown in Figure 6.

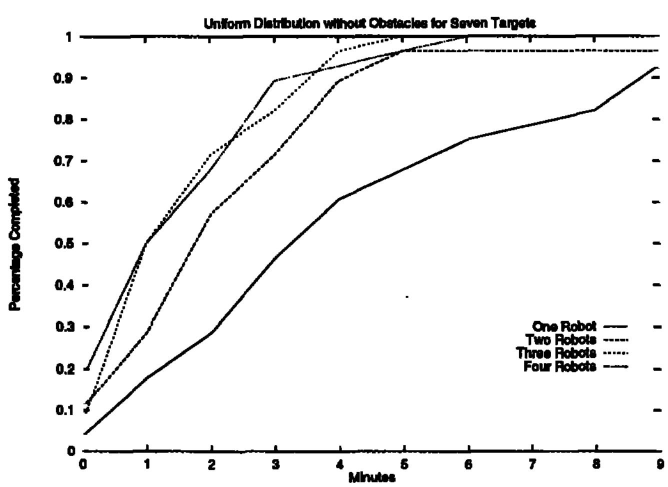

Figure 6: Performance in small arena, seven targets, no obstacles, one to four robots

The second experiment differed from the first only by the addition of obstacles in the arena. Again, performance was analyzed relative to the number of robots performing the task. These results are shown in Figure 7.

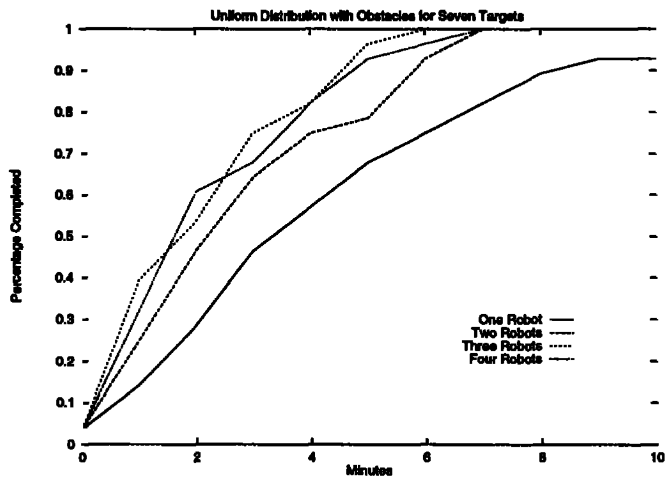

Figure 7: Performance in small arena, seven targets, obstacles, one to four robots

The final set of experiments analyzed how the robots performed in the larger arena. In these sets of experiments, the targets were placed in a uniform distribution and there were no obstacles. There were eleven targets used in this arena instead of seven to help decrease the sparseness between targets. These results are shown in Figure 8.

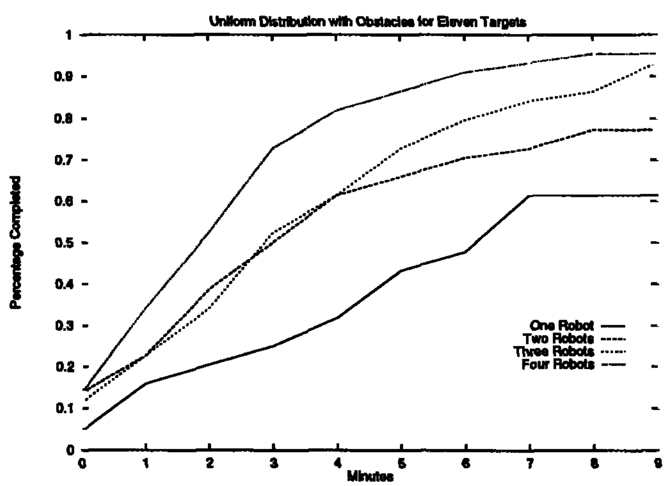

Figure 8: Performance in large arena, eleven targets, no obstacles, one to four robots

Finally, two experiments were run in which the targets were placed in small clusters instead of in a uniform distribution. The first experiment took place in the small arena with obstacles while the second experiment took place in the large arena with no obstacles. These experiments were only run with four robots.

In the first of these two experiments, the robots actually did better than in the uniform distribution. In the second, the robots did much worse than with the uniform distribution. See Figures 9 and 10.

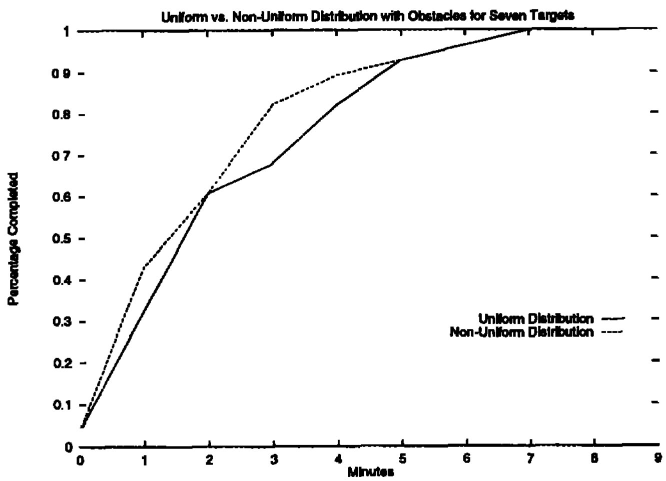

Figure 9: Performance of uniform vs. non-uniform distribution, seven targets, small arena, obstacles, four robots

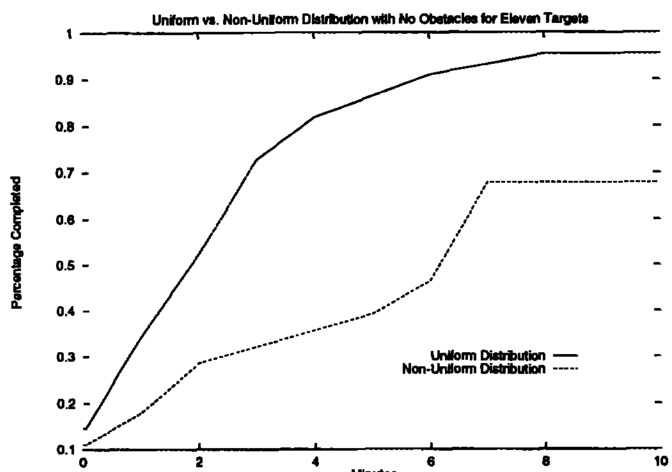

Figure 10: Performance of uniform vs. non-uniform distribution, eleven targets, large arena, no obstacles, four robots

Conclusions and Future Work

We have analyzed how the robot performance at a search and retrieval task is affected by environmental factors and by the number of robots.

The first thing to notice is how the number of robots affects the overall performance of the system. Increasing the number of robots definitely increased the total number of targets retrieved and the rate of retrieval. However, robot collisions and interference tended to slow the collection of targets down greatly. When two or more robots decided to try to grab the same target, they would usually spend a fair amount of time colliding with each other and keeping the target out of each other's grippers. Adding more robots didn't produce a linear or superlinear increase in performance. Continually increasing the number of robots produces only a diminishing increase in performance.

There are several differences between the seven target experiments with and without obstacles. First of all in the experiments with obstacles involving three and four robots, the robots took a little bit longer to retrieve the targets but eventually ended up doing just about as well as the experiment without obstacles. However, for the experiments that involved only one or two robots, the rate of collection and overall success rate went up.

The speedup for the experiments with the small number of robots can be attributed to the fact that so much of the arena was made inaccessible because of the obstacles. These obstacles decreased the overall area of the arena thus robots only had to search a very small amount of it before encountering a target. The same thing can be said for the experiments with the larger number of robots, but more interference caused the overall speed of collection to decline somewhat.

In the experiments that compared the uniform with the non-uniform distribution of targets in the large arena, the robots suffered a severe performance hit when all of the targets were placed in a single clump. There was a very interesting interaction between the robots and their environment with respect to the control code. Apparently, the positioning of the arena walls, the drop-off zone light bulb and the interactions of the robots' internal processes caused the robots to favor some sides of the arena more than others. This non-uniformity of the search pattern caused the robots to miss the majority of the targets.

This strange interaction of process timings, variables, and the environment is something that was not entirely unexpected but is something that may be impossible to completely weed out. Reactive programs of this nature quite often have unwanted side effects that are extremely difficult to remove without accidentally introducing others. It may be that such a complex system is nearly impossible to model properly because of its close coupling with the environment that it must interact with.

In our future work, we will study the effectiveness of providing additional and more detailed information to the robots. One consideration is whether performance would increase if the robots could record previous knowledge such as where they have searched and where they have found obstacles and/or targets. In these further experiments, the robots will use three landmarks to determine their X, Y positions and orientation (Cohen & Koss 1992). When returning to the drop-off zone with a target, the robots might pass other targets that they must leave behind because their gripper is full. The robots would remember their position at that point and would use the geometric information from three unique landmarks to return to that position after dropping off the target at the home base. We will analyze if the increased knowledge translates into increased performance.

In order to incorporate this additional information, a more deliberative architecture will most likely be required. Currently, the robots' programs are completely reactive in nature, and thus no time is spent analyzing anything beyond the immediate surroundings. Introducing a more abstract decision process will very likely introduce delay periods in which the robot must spend time reasoning about its environment. We hope that some middle ground can be discovered in which the robot can increase its rate of success by using more environmental knowledge while not decreasing its overall performance.

Acknowledgments

We wish to thank all the people who have helped in this project, in particular Sirintorn Clay, Anne Paxton, Adam Thorne, David Littau, Elaine Barrett, Doug Perrin, and Chris Smith.

This material is based upon work supported by the Defense Advanced Research Projects Agency, Electronics Technology Office ("Distributed Robotics" Program), ARPA Order No. G155, Program Code No. 8H20, Issued by DARPA/CMD under Contract #MDA972-98-C-0008

References

Agah, A., and Bekey, G. A. 1995. In a team of robots the loudest is not necessarily the best. In *Proc. IEEE Int'l Conf. on Systems, Man, and Cybernetics*, 3800–3805.

Alami, R.; Robert, F.; Ingrand, F.; and Suzuki, S. 1995. Multi-robot cooperation through incremental plan-merging. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2573–2579.

Arkin, R. C., and Bekey, G. A., eds. 1997. *Robot Colonies*. Kluwer.

Arkin, R. C. 1992. Cooperation without communication: Multi-agent schema based robot navigation. *Journal of Robotic Systems* 9(3):351–364.

Balch, T., and Arkin, R. 1994. Communication in reactive multiagent robotic systems. *Autonomous Robots* 1(1):27–52.

Balch, T., and Arkin, R. C. 1995. Motor schema-based formation control for multiagent robot teams. In *Proc. First Int'l Conf. on Multi-Agent Systems*, 10–16.

Beckers, R.; Holland, O. E.; and Deneubourg, J. L. 1994. From local actions to global tasks: Stigmergy in collective robotics. In *Artificial Life IV*, 181-189. MIT Press.

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

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

Cohen, C., and Koss, F. 1992. A comprehensive study of three object triangulation. In *Proceedings of the 1993 SPIE Conference on Mobile Robotics*, 95–106.

MacKenzie, D.; Arkin, R. C.; and Cameron, R. 1997. Specification and execution of multiagent missions. *Autonomous Robots* 4(1):29–57.

Martin, F. G. 1998. *The Handy Board Technical Reference.* MIT Media Laboratory, Cambridge, MA.

Mataric, M. J. 1995a. Designing and understanding adaptive group behavior. *Adaptive Behavior* 4:51–80.

Mataric, M. J. 1995b. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems* 16(2-4):321–331.

Mataric, M. J. 1997. Using communication to reduce locality in distributed multi-agent learning. In *Proc. Nat'l Conf. on Artificial Intelligence.*

McFarland, D. 1994. Towards robot cooperation. In *Proc. Simulation of Adaptive Behavior.*

Parker, L. E. 1996. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics* 10(6).

Rus, D.; Donald, B.; and Jennings, J. 1995. Moving furniture with teams of automonous mobile robots. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems.*

Sasaki, J.; Ota, J.; Yoshida, J.; Kurabayashi, D.; and Arai, T. 1995. Cooperating grasping of a large object by multiple mobile robots. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1205–1210.

Schneider Fontan, M., and Mataric, M. J. 1996. A study of territoriality: The role of critical mass in adaptive task division. In et al., P. M., ed., *From Animals to Animats 4, Fourth Int'l Conference on Simulation of Adaptive Behavior (SAB-96)*, 553–561. MIT Press/Bradford Books.

Sen, S.; Sekaran, M.; and Hale, J. 1994. Learning to coordinate without sharing information. In *Proc. Nat'l Conf. on Artificial Intelligence*, 426–431.

Shoham, Y., and Tennenholtz, M. 1992. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proc. Nat'l Conf. on Artificial Intelligence*, 276–281.

Shoham, Y., and Tennenholtz, M. 1995. On social laws for artificial agent societies: off-line design. *Artificial Intelligence* 73(1–2):231–252.

Stone, P., and Veloso, M. M. 1997. A layered approach to learning client behaviors in the robocup soccer server. *Journal of Applied Artificial Intelligence.*

Tambe, M. 1997. Implementing agent teams in dynamic multi-agent environments. *Applied Artificial Intelligence.*

Walker, A., and Wooldridge, M. 1995. Understanding the emergence of conventions in multi-agent systems. In *Proc. First Int'l Conf. on Multi-Agent Systems*, 384–389.

Wang, J. 1995. Operating primitives supporting traffic regulation and control of mobile robots under distributed robotic systems. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1613–1618.

Wright, A.; Sargent, R.; and Witty, C. 1996. *Interactive C User's Guide.* Newton Research Labs, Cambridge, MA.