

# **Multi-Target Observation: An Application of Multi-Robot Learning**

**Rachel Yeargin**

**Lynne Parker, Mentor**

Center for Engineering Science Advanced Research

Computer Science and Mathematics Division

Oak Ridge National Laboratory

Summer 2001

# Table of Contents

I	Abstract
II	Introduction
III	The Need for Robot Learning
IV	The CMOMMT Problem
V	Hand Generated Solution
VI	Learning the CMOMMT Application
	A. Lazy Learning and Q-Learning
	B. Pessimistic Algorithm
	C. Results of Learning Approach
VII	Conclusions and Future Research
VIII	References

## Acknowledgements

Special thanks to Lynne Parker for all of your patience and guidance throughout this project. To the Department of Energy and the Computer Science and Mathematics Division thanks for making this internship a possibility, it was a wonderful learning opportunity. This research was performed under the Research Alliance for Minorities

Program administered through the Computer Science and Mathematics Division, Oak Ridge National Laboratory. This Program is sponsored by the Mathematical, Information, and Computational Sciences Division; Office of Advanced Scientific Computing Research; U.S. Department of Energy. Oak Ridge National Laboratory is managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.

## **Abstract**

**In recent years, interest in the area of the performance of multi-robot teams in cooperative tasks has significantly increased. As a result, scientists have delved into a new realm of research and experimentation with multi-robot learning. This paper examines various types of robot learning and the benefits or challenges associated with each type. The Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) application is presented and viewed as a valuable testing domain in the field of multi-robot teams and cooperative learning. The hand-generated approach to this application (A-CMOMMT) will be used as a control model in our research into generating learning techniques that can improve upon this approach's results. The goal for this particular project is to improve the performance of a previous approach that uses lazy Q-learning and self-organizing maps. This project is designed to generate a learning algorithm that reaches or exceeds the performance of the hand-generated approach. We do this by introducing a component that allows the robots to not only be aware of the nearby targets, but also of the nearby robots and their actions. The ultimate and considerably broader goal of this research is to develop learning techniques that allow for a more generalized application of cooperative robotics to numerous real world problems.**

## **Introduction**

**In recent years, significant progress has been made in the field of cooperative robotics. As a result, the interest in the area of robot learning and its application to multi-robot teams has also grown. This growth is due greatly to the wide variety of problems that these multi-robot teams can be applied to. Probably the most common applications thus far have been in areas such as air fleet control, box pushing, predator/prey problems, and multi-robot soccer. All of these problems require a great deal of reliability and flexibility, which a multi-robot team provides. But only in the multi-robot soccer problem are the actions of one robot dependent upon the current actions of the other robots. These types of tasks are characterized as being inherently cooperative because the success of the team is dependent upon the combined actions of the robot team rather than the individualized tasks assigned to each robot. A problem similar to this one has been introduced in order to add to the already challenging domain of multi-robot learning. This problem is called the Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT). This application -although similar to the multi-robot soccer problem due to the fact that it is an inherently cooperative task- is different in the respect that it is necessary for the existence of scalability to very large numbers of robots.**

**The motivation behind using this application has been explained and tested in other work. So, in this paper, we examine various types of robot learning and the benefits and drawbacks of each type with respect to the CMOMMT application. We will also describe several approaches to the CMOMMT application that have been tested or are in the process of being tested in an attempt to surpass the results generated through the usage of the handed-generated approach to this problem. Finally, we will summarize our analytical results, discuss future work on this project,**

and draw conclusions about how our results may affect the overall study of learning in multi-robot teams.

## **The Need for Robot Learning**

Learning is often described as being one of the fundamentally necessary components of cooperative robotics. Compared to the concept of single robot learning, cooperative robot learning adds several complications such as a larger search space, the need for awareness of other team members, the ability of robots to analyze their behaviors with respect to the tasks given to the entire group, and the challenge of inherently cooperative tasks. Despite these complications in learning, multi-robot teams have significant advantages over a single robot performing a similar task. For example, a robot team can distribute actions allowing many robots to be in many different places at the same time. This team can also incorporate inherent parallelism in which many robots can do many different things simultaneously. Due to the fact that the simpler solution is almost always better, multi-robot teams can decompose certain types of problems and allocate them to various team members, thus eliminating the need for a comprehensive but difficult solution in a single robot.

Robot learning is and always will be extremely important to the success of multi-robot teams. A learning robot is one that can improve its behavior as a result of direct interaction with the environment. The ability of a robot to select an efficient behavior from a set of potential behaviors and automatically modify its behavior to improve its performance is imperative to the goals of the multi-robot teams. A significant long-term goal is for these robot systems to be able to perform their tasks over a long period of time without human interference. Learning makes this goal more of a reality.

There are two types of robot learning paradigms that are widely used, supervised learning and reinforcement learning. In supervised learning, the operator is required to define a set of situation-action pairs. In contrast, reinforcement learning generates the learning base through a combination of exploration and a reinforcement function. Generally, reinforcement learning involves an agent, which is the learner or the decision maker, the environment, which is everything that surrounds the agent, and the actions, which are the things that the agent can do. In the context of cooperative robotics the reinforcement function is designed to measure the performance of the whole team of robots and at the same time measure the performance of each individual robot. It is for this reason that reinforcement learning is generally the choice for application in real world problems.

Incorporating learning and adaptation into the field of autonomous robots will not only lessen the amount of difficulty in the initial programming, but also allow the robots to change their behavior over time as the world around them changes. It is here that applications such as CMOMMT are essential because they allow researchers to test their theories until a successful multi-applicable approach to robot team learning has been developed.

## CMOMMT Problem Description

The testing domain that we are studying in this problem the Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT) is defined as follows:

**S:** a two-dimensional, bounded, enclosed spatial region

**V:** a team of  $m$  robot vehicles,  $v_i, i = 1, 2, \dots, m$ , with 360 degrees field of view observation sensors that are noisy and of limited range

**O(t):** a set of  $n$  targets,  $o_j(t), j = 1, 2, \dots, n$ , such that target  $o_j(t)$  is located in region  $S$  at time  $t$ .

A robot  $v_i$  can be assumed to be observing a target whenever the target is within  $v_i$ 's sensing range. Define an  $m \times n$  matrix  $B(t)$  as follows:

$$B(t) = [b_{ij}(t)]_{m \times n} \text{ such that } b_{ij}(t) = \begin{cases} 1 & \text{if robot } v_i \text{ is observing target } \\ & o_j(t) \text{ in } S \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Then, the goal is to develop an algorithm, which we call *A-CMOMMT*, that maximizes the following metric  $A$ :

$$A = \sum_{t=1}^T \sum_{j=1}^n \frac{g(B(t), j)}{T}$$

where:

$$g(B(t), j) = \begin{cases} 1 & \text{if there exists an } i \text{ such that } b_{ij}(t) = 1 \\ 0 & \text{otherwise} \end{cases}$$

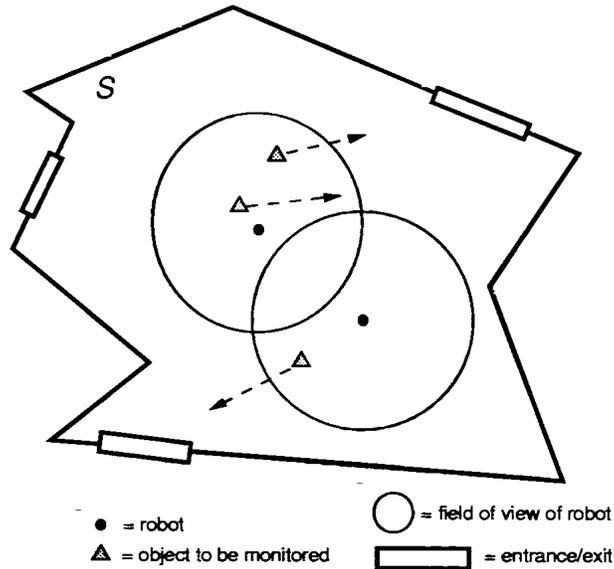
That is, the goal of the robots is to maximize the average number of targets in  $S$  that are being observed by at least one robot throughout the mission that is of length  $T$  time units. Additionally we define *sensor\_coverage*( $v_i$ ) as the region visible to robot  $v_i$ 's observation sensors, for  $v_i \in V$ . Then we assume that in general,

$$\bigcup_{v_i \in V} \text{sensor\_coverage}(v_i) \ll S.$$

In other words, the maximum area covered by the observation sensors of the robot team is much less than the total area that is to be observed. It can then be assumed that fixed robot sensing locations or paths will not be adequate and that robots must move dynamically as targets appear in order to maintain the target observations and to maximize the coverage.

This is the basic CMOMMT problem. There are however, an almost infinite number of ways to increase the difficulty of this problem. Due to the existent amount of possible variations on the dynamic and distributed sensory coverage numerous problems are possible. In addition, the relative numbers and speeds of the robots and the targets that they are tracking can vary as can the availability of inter-robot

communication. The robots also have heterogeneous tendencies due to the fact that they can differ in their sensing and moving capabilities. These complications are what make the CMOMMT application such a rich testing domain in almost all areas of cooperative robotics and robot learning.



## Hand-Generated Approach

A hand-generated solution to the CMOMMT problem has been developed in [15]. This solution performs well compared to various control groups that it has been compared to. It is called the A-CMOMMT and it has been implemented on both physical and simulated robot teams. The hand-generated solution allows the robots to use weighted local force vectors that attract them to nearby targets and repel them from nearby robots. The weights are computed in real time by a higher level reasoning system in each robot, and are based on the relative location of the nearby robots and targets.

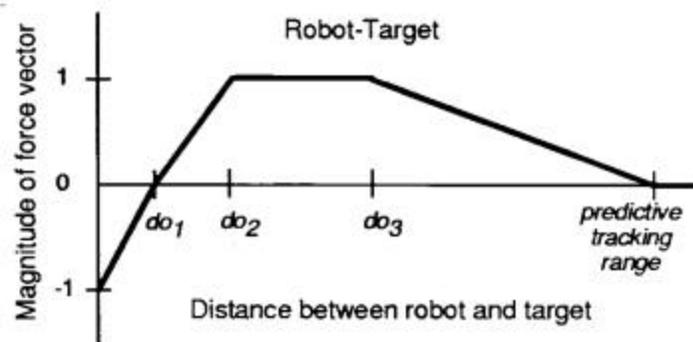
The local force vectors are calculated as follow. The magnitude of the force vector attraction of the robot  $v_i$  relative to target  $o_k$ , denoted  $|f_{ik}|$  for parameters

$0 < do_1 < do_2 < do_3$  is:

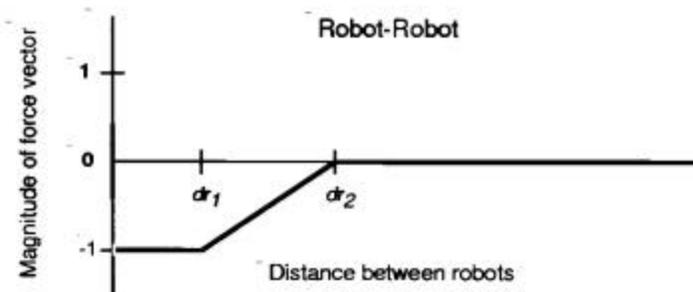
$$|f_{ik}| = \begin{cases} \frac{-1}{do_1} & \text{for } d(v_i, o_k) \leq do_1 \\ \frac{do_2 - do_1}{do_2 - do_1} & \text{for } do_1 < d(v_i, o_k) \leq do_2 \\ \frac{-do_2}{do_3 - do_2} & \text{for } do_2 < d(v_i, o_k) \leq do_3 \\ 0 & \text{otherwise} \end{cases}$$

Where  $d(a,b)$  returns the distance between the two robots and /or targets. The magnitude of the force vector repulsion of robot  $v_l$  relative to robot  $v_i$  denoted  $|g_{li}|$  for parameters  $0 < dr_1 < dr_2$  is:

$$|g_{li}| = \begin{cases} -1 & \text{for } d(v_l, v_i) \leq dr_1 \\ \frac{1}{dr_2 - dr_1} & \text{for } dr_1 < d(v_l, v_i) \leq dr_2 \\ 0 & \text{otherwise} \end{cases}$$



**Fig 2: Function defining the magnitude of the force vectors of nearby targets.**



**Fig 3: Function defining the magnitude of the force vectors of nearby robots.**

In the context of these two functions, the predictive tracking range is the range just beyond the sensing capability of the robots, but within which robot trackers should continue to adjust their motions if targets are nearby. The first function defines the relative magnitude of the attractive forces of a target within the predictive tracking range of a given robot. If a robot is too close to the target that it is tracking (distance  $< do_1$ ) then it is repelled from this target, thus minimizing the occurrence of collisions. It is then apparent that the preferred tracking range is between  $do_2$  and  $do_3$ . The second function defines the magnitude of the repulsive forces between the robots. The robots will strongly repel each other if they are too close together (distance  $< dr_1$ ).

In contrast, if the robots are far enough apart (distance  $> dr_2$ ) they will have no effect in force vector calculations.

Higher-level information can be very important in the improvement of robot team performance. Therefore, the hand-generated approach includes higher-level control to weight the contributions of each target's force field on the total computed field. This higher-level knowledge is expressed in a weight that reduces robot attraction to a nearby target if that target is within the field of vision of another robot. This idea helps to minimize the opportunities for targets to escape observation by reducing the overlap of the sensory areas of the robots.

The results of the A-CMOMMT approach can vary depending upon the number of robots and targets and the size of the work area. Through numerous simulations and physical robot experiments, it has been discovered that this algorithm performs best for a ratio of targets to robots greater than 1 to 2. It was also shown that in comparison to the local control only, random linear robot movement, and fixed robot positions that the A-CMOMMT performs significantly better.

## Learning the CMOMMT Application

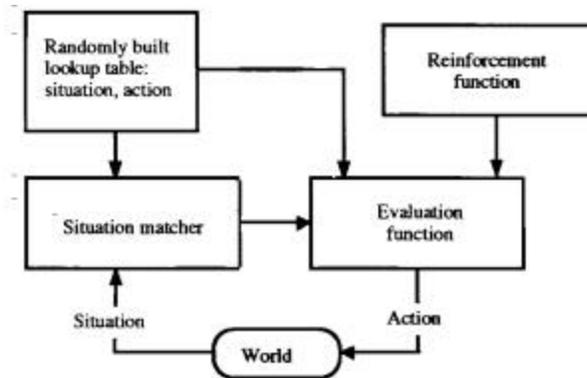
Despite the overall success of the A-CMOMMT approach to this problem, particular interest has been taken in improving upon these results with a learning-based approach that does not require the assumption of an *a priori* model. Several approaches have been developed with the objective of learning new cooperative behaviors and these approaches can be directly applied to the CMOMMT domain. One of these approaches, which was developed in [8], combines lazy learning with Q learning and a Pessimistic Algorithm that can compute a lower bound on the utility of executing an action for each team member. As expected, the challenges in this problem include a large search space size, the need for communication or awareness of other team members, and attempting to assign credit in an inherently cooperative problem.

### Lazy Learning and Q-learning

Lazy learning (instance based learning) greatly reduces the time required to build the bases that must be defined for each behavior in reinforcement learning. Lazy learning delays the usage of gathered information until it is needed. As a result, the same pool of information can be used for the synthesis of a variety of different behaviors. In relation to reinforcement learning, lazy learning builds a non-explicit model of the situation-action relation. It does this by sampling from the situation-action space by a random action selection policy, storing these events in memory, and then when needed, probing this memory for the best action. Coupling this to a reinforcement learning techniques such as Q-learning allows for a large reduction of necessary learning time.

Q-learning is an extremely useful reinforcement learning algorithm for an agent learning an action policy. It is based on the usage of a state-action table containing the gain that the agent obtains by executing an action from a specific state. The combination of these two learning types is called Lazy Q-learning.

In order to express a particular behavior the memory must be probed with the reinforcement function. The objective is to provide an approach to predicting the rewards for some state action pairs without generating them. The algorithm that accomplishes this objective works as follows. First, a situation matcher locates all the states within the memory that are within a given distance. Then, if the situation matcher has failed to find any nearby situations the action comparator will select an action at random. However, if one of these states is located, the action comparator will select, after careful examination, the action with the highest expected reward. This action is then executed and a new situation results. This requires that new situation-action pairs be added to the memory and that new Q-values be dynamically computed. In this lazy Q-learning the exploration phase is done only once. The information collected is then stored and used in future experiments.



**Fig 4: Lazy learning:** the randomly selected situation-action pairs in the lookup table are used by the situation matcher to select the action to execute in the current situation. The reinforcement function qualifies the actions proposed, helping to select the best one.

### The Pessimistic Algorithm

**The Pessimistic Algorithm** for the selection of the best action to execute for a robot in its current local situation is defined as follows: find the lower bounds of the utility value associated with the various potential actions that may be conducted in the current situation, then choose the action with the greatest utility. In general, a local robot situation is an incomplete observation of the state of the entire system.

Therefore, instead of completing the entire observation in an attempt to solve the problem, we simply rank the utility of the actions. If by using a unique instance of the memory, we can obtain the utility of the situation, then it is likely that the utility attributed to the local situation is due to the other robots actions. The probability of this occurrence decreases in proportion to the number of similar

situations. By taking the minimum utility value of this set of similar situations then there is no implication of losing targets.

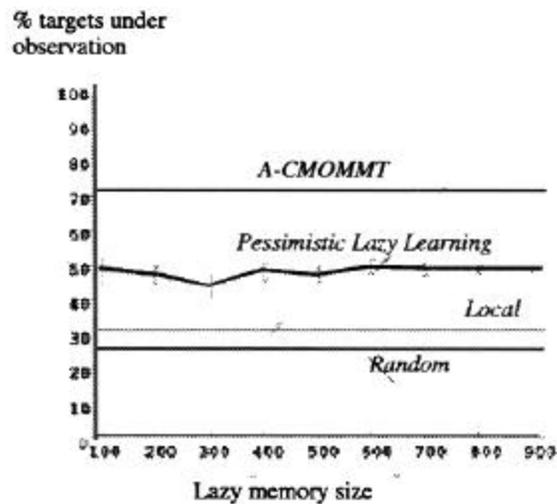
The Pessimistic Algorithm is defined as follows:

- Let  $M$  be the memory, a lookup table of situation-action pairs gathered during an exploration phase:  $M = [(s(1), a(1)), \dots, (s(t), a(t)), (s(t+1), a(t+1)), \dots]$ .
- Let  $sit$  be the current situation.
- Find  $S(sit)$ , the set of  $n$  situations of  $M$  similar to  $sit$ .
- Let  $S_{follow}(sit)$  be the set of situations that directly follows each situation of  $S(sit)$ .
- Compute the lower bound (LB) of the utility value (U) associated with each situation  $s(k) \in S_{follow}(sit)$ :  
-- $LB(s(k)) = \min(U(s(m)))$ , for  $s(m) \in S(s(k))$ , the set of situations similar to  $s(k)$ .
- Execute the action that should take the robot to the new situation  $s^*$  :  
 $s^* = \max(LB(s))$  and  $s \in S_{follow}(sit)$ .

There are numerous ways to calculate the utility  $U$  associated with the given situation. However, in this application, it can be the exact value of the reinforcement function for this particular situation-action pair, which is stored in the lookup table ( $M$ ) along with the number of targets under observation. The value used as a utility if one or more targets have been acquired since the previous situation is +1, if one or more targets have been lost it is -1, and it is 0 otherwise.

## Results of the Learning Approach

The efficiency of the Pessimistic Algorithm has been compared with the performances of the A-CMOMMT, a random action policy, and a user-defined non-cooperative policy. It is evident that there is a definite improvement of performance in the lazy Q-learning over the random selection policy. This simply proves that as a learning type, lazy-Q learning is very important. In comparison to the user-defined policy, the Pessimistic Algorithm was also more successful. One significant factor that could cause the variation in performance is the fact that in Q-learned behavior, there is far less rigidity than in the user-defined policy. Basically since the robots in Q-learned behavior are not center-of-gravity-oriented, they will exhibit a very erratic tracking path, moving however necessary to keep from losing the robot. As a result the surface area under observation per unit of time is much larger. Despite the lazy Q-learning's impressive performance in comparison to these two approaches, it was nowhere near as successful as the A-CMOMMT approach. Pessimistic Lazy Learning cannot even compete due to the fact that it does not take into account the location of neighboring robots or their actions. Therefore this approach will be revised to allow for the use of information on the location of neighboring robots.



**Fig 5: Performance of the Pessimistic lazy Q-learning approach compared to a random action selection policy, a user-defined non-cooperative policy, and the hand-generated solution A-CMOMMT.**

## Conclusions and Future Research

**Due to the promising future of the Pessimistic Lazy Learning approach to the Cooperative Multi-Robot Observation of Multiple Moving Targets, research is underway to improve upon the results exhibited in these past experiments. It can be concluded that by incorporating a function into the already existing learning algorithm that will take into account the locations and actions of the other robots that the results should be improved significantly. It is hopeful that these results will equalize and maybe even surpass the performance of the A-CMOMMT approach. In addition to providing better results, this approach will allow researchers to eliminate the assumptions of an a priori model that are existent in most of the current approaches to this problem.**

## References

- [1] L. E. Parker and C. Touzet. Multi-robot learning in a cooperative observation task. In L. E. Parker, G. Bekey, and J. Barhem, editors, *Distributed Autonomous Robotics Systems*, volume 4, pages 391-401. Springer, 2000.
- [2] L. E. Parker. A case study for life long learning and adaptation in cooperative robot teams. In *Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotics Systems II*, volume 3839, pages 92-101, 1999.
- [3] D. Aha, editor. *Lazy Learning*. Kluwer Academic Publishers.
- [4] T. Kohonen. *Self Organization and Associative Memory*. Springer, Berlin, Heidelberg, 1984. 3<sup>rd</sup> ed. 1989.
- [5] T. Kohonen. *Self Organizing Maps*. Springer Series in Information Sciences, Vol 30, 1995;
- [6] C. Touzet. "Bias Incorporation in Robot Learning," submitted for publication, 1998.
- [7] L. E. Parker. "Cooperative Robotics for Multi-Target Observation" *Intelligent Automation and Soft Computing* 5(1), 5-19, 1999.
- [8] C. Touzet, "Distributed lazy q-learning for cooperative mobile robots," submitted to *Autonomous Robots*, 1999.
- [9] M. Mataric. Learning in multi-robot systems. In Gahard Weiss and Sandip Sen, editors, *Adaptation and Learning in Multi-Agent Systems*. Springer, 1996.
- [10] L. E. Parker. "The Effect of Action Recognition and Robot Awareness in Cooperative Robotics Teams," Proc. IROS 95, Pittsburgh, PA 1995.
- [11] L. E. Parker. "Cooperative Motion Control for Multi-Target Observation," Proc. IROS 97, Grenoble, France 1997.
- [12] J. W. Sheppard and S. L. Salzberg. "A Teaching Strategy for Motion Based Control," *Lazy Learning*, D. Aha (ed), Kluwer Academic Publishers, 343-370, 1997.
- [13] C. Touzet. "Robot Awareness in Cooperative Mobile Robot Learning" *Autonomous Robots* Vol 8 No 1 January 2000.

**[14] C. Touzet. "Programming Robots with Associative Memories" IJCNN 99 Washington D.C. USA July 10-16 1999.**

**[15] L. E. Parker, C. Touzet, D. Jung "Learning and Adaptation in Multi-Robot Teams" Proc of Eighteenth Symposium on Energy Engineering Sciences, 2000 177-185.**

**[16] L.E. Parker "Behavior Based Robotics Applied to Multi-Target Observation in Intelligent Robots: Sensing Modeling and Planning" R. Bolles, H. Bunke, and H. Noltemeier (eds.) *World Scientific* 1997 pg 356-373.**