Chapter 4

Task and Reward

An objective of this research is the development of a methodology supporting investigation of the impact of reinforcement and task on robot team behavioral diversity and performance. This dissertation examines the relationships experimentally by varying the the independent variables (reinforcement and task) and evaluating their impact on the dependent variables (diversity, performance and learning rate). Characterization of the task and reward structure is crucial for establishing the experimental space of this research.

Task characterization is important because it helps answer questions regarding how the same type of reinforcement can lead to different performance and and diversity levels in different tasks. Without answering "how are soccer and foraging different?" for instance, we can't answer "why is diversity good in soccer but bad in foraging?"

A taxonomy of reward structure is also important. Research presented in later

chapters shows that performance and diversity in a learning team depend on the form of reinforcement used to train the robots. The results indicate that there are tradeoffs to consider in the selection of a reward function; for instance some functions provide quicker learning but slightly poorer performance. Without a characterization of the differences between reward functions, it would be impossible for a robot systems designer to consider these tradeoffs intelligently.

At present no taxonomies of task or reinforcement exist. To address this crucial gap a new system for characterizing multi-robot tasks and a taxonomy of reinforcement functions for multi-robot teams are presented. As well as aiding this investigation, these classifications are potentially useful for other researchers investigating multiagent robotics.

4.1 Characterization of multi-robot tasks

The reinforcement learning literature considers the task and the performance metric as one and the same. That view is adopted here; a robot team's mission is to maximize performance over time. But there are other components of a task as well. In addition to the performance metric, a task is further defined by the environment and robotic platform. Since the investigation focuses on links between task and diversity, we seek a classification identifying aspects of a task that may be correlated with a requirement for cooperation and/or diversity in a robot team.

The approach is to examine the task performance metric, task environment and robotic platform for constraints and features that distinguish various tasks from one another. The focus is on tasks involving the movement of robots or the movement of objects by robots. In a manner similar to Dudek's taxonomy of multi-robot systems [DJW93], a set of descriptive tags that identify salient features of a task are proposed. Although similar in style to Dudek's formulation, this classification focuses on *task* rather than robot.

The descriptive tags are summarized in Table 4.1 and explained in detail in the following subsections. Several example tasks are examined and classified using the descriptors at the end of this section.

Table 4.1: Summary of terms characterizing Multi-robot tasks.

Descriptor	Meaning
Time	
TIME_LIM	fixed time task
TIME_MIN	minimum time task
TIME_UNLIM	unlimited time task
SYNC	synchronization required
Subject of Action	
OBJECT_BASED	movement/placement of objects is important
ROBOT_BASED	movement/placement of robots is important
Limited Resources	
RESOURCE_LIM	limited external resources
ENERGY_MIN	minimum energy task
COMP_INT	team members compete with each other
COMP_EXT	team competes with external agencies
Movement	
CONVERGENCE	multiple robots converge to same position
COVERAGE	multiple agents spread apart
MOVEMENT_TO	movement to a position
MOVEMENT_WHILE	movement while maintaining position
Platform	
SINGLE_AGENT	a single agent can perform task
MULTI_AGENT	multiple agents are required
DISPERSED	agents must be geographically dispersed
SENSOR_COMPLETE	can sense all relevant features
SENSOR_LIM	world is partially observable
COMM	communication is required

4.1.1 Time

Most task definitions include a time constraint. The AAAI-98 "Find Life on Mars" task for instance, required robots to collect as many objects as possible in 10 minutes. For some tasks a robot's performance is evaluated by how long it takes to complete the task. This section considers various task time constraints.

Evaluations of robot systems are typically conducted in discrete experimental trials. `TIME_LIM` tasks are those where each trial runs over a fixed length of time, while `TIME_MIN` refers to tasks where performance is measured as the time required to complete a task. An example `TIME_LIM` task is "collect as many attractors as possible in 10 minutes." An example `TIME_MIN` task would be "collect five attractors as quickly as possible."

Some tasks do not have a time constraint; they are referred to as `TIME_UNLIM`. `TIME_UNLIM` tasks are either carried out indefinitely, or terminate on non-time-related criteria. An example indefinite time task might be "patrol the building for burglars" or "balance the pole." The soccer experiments explored later in this work (Chapter 7) are `TIME_UNLIM` since they do not end on a time limit, but rather when a total of number of goals are scored.

In addition to elapsed time constraints, some tasks include a synchronization requirement. One example is the task of pushing two buttons simultaneously. These are labeled `SYNC` tasks. Communication between agents is one way to accomplish `SYNC` tasks, but other mechanisms (such as clock synchronization), are also available.

4.1.2 The subject of action

Some tasks involve the placement or movement of an external object (e.g. the ball in soccer) while others concern the positioning of the robotic agent itself (formation). Tasks involving external objects are termed `OBJECT_BASED` tasks while those involving the agents themselves are `ROBOT_BASED` tasks.

4.1.3 Limited resources and competition

Many tasks, like foraging, involve limited environmental resources; as an agent performs the task it consumes the resource thus making the task more difficult. In multiagent systems, limited resources may force the agents on a team into competition

between themselves or with external agencies. In addition to external environmental resources the robot may also have to monitor *internal* resources, such as fuel.

Tasks that involve limited resources are termed RESOURCE_LIM. Robotic foraging is typically a RESOURCE_LIM task because there are only a fixed number of objects available for the robots to collect. When agents on a team compete with each other for a limited resource, the task is referred to as COMP_INT (internally competitive). Foraging is internally competitive since one agent's success in collecting an object, reduces the opportunity for others. When the agents compete with external forces, the task is COMP_EXT. Soccer is a COMP_EXT task because the agents on one team compete with the (external) agents on the other team for the opportunity to score goals.

RESOURCE_LIM refers specifically to resources external to the agent (e.g. attractors) versus internal resources (e.g. fuel). Tasks that call for fuel and/or energy conservation are referred to as ENERGY_MIN tasks. "Dock the boat using minimum fuel" is a TIME_UNLIM ENERGY_MIN task.

4.1.4 Movement

The coordinated movement of several robots is important in search, surveillance, grazing and cleaning tasks. Ali has developed a taxonomy of robotic movement tasks in his investigation of human control techniques for multi-robot teams [Ali97]. His terms, COVERAGE, CONVERGENCE, MOVEMENT_TO and MOVEMENT_WHILE are adopted here for the description of multiagent movement tasks.

Multi-robot movement tasks are broken into two primary classifications: COVERAGE and CONVERGENCE. In coverage tasks the robots spread out as much as possible. Examples include search, grazing and cleaning tasks. In convergence tasks robots gather together from dispersed locations. The description of movement tasks is further refined by whether the a robot should move *to* a position or move *while* maintaining a position. In MOVEMENT_TO tasks the robots move from their starting locations to a particular configuration, while in MOVEMENT_WHILE tasks the robots are to maintain a configuration while moving.

4.1.5 Platform dependencies

The capabilities and limitations of the robotic platform can significantly impact the space of potential solutions. A large multi-armed tank robot for example, may be able to individually accomplish many tasks that would otherwise require dozens of smaller robots. Other issues of robot platform, like sensor ranges, can affect the space of possible solutions as well.

Consider the task of transporting a cookie to a homebase. For humans, this is a single-agent problem: pick up the cookie, carry it home. But for ants the task assumes epic multiagent proportions. Tasks in which an individual agent is able to generate positive performance are referred to as `SINGLE_AGENT`. If it is unlikely that a single agent can perform suitably – that several agents are required – the task is `MULTI_AGENT`.

Tasks that require agents in several widely separated places at once are `DISPERSED`. Dispersion might be required because of size, reach, actuator or sensing limitations of individual robots on the team. An example of this type of task is one in which two buttons on opposite sides of a building must be pushed simultaneously (this is also a `SYNC` task). A task requiring continuous surveillance over areas larger than an individual robot's sensor range require dispersion as well. Soccer is a `DISPERSED` task since the ball can move rapidly from one part of the field to another; players must be dispersed to ensure effective play. Many `COVERAGE` tasks, like coordinated search, are `DISPERSED` because the agents must spread apart to ensure more complete sensor coverage. Formation tasks are not `DISPERSED` because, in formation, agents must be relatively close together with respect to the environment.

In some tasks for robots the agents are provided perfect, or nearly perfect sensing. In other words the agent knows, with complete accuracy, every aspect of the environment's state germane to the task. Many tasks examined in the reinforcement learning literature share this attribute (e.g. the acrobot [Boo97]). These tasks are referred to as `SENSOR_COMPLETE`. It is often impossible however for an individual agent to sense all aspects of the environment relevant to the task. Even if the robot has a comprehensive sensor suite, the sensors may be degraded in actual use by noise and occlusion. Such tasks are referred to as sensor limited or `SENSOR_LIM` tasks.

The decision problem for robots in a `SENSOR_LIM` task is *partially observable* [LCK95]. This means that many distinct situations may be perceived as equivalent

by the robot. For instance, as a foraging agent delivers an attractor, it might perceive itself as being in the same state whether it is 5 meters or 4 meters from the delivery area.

Finally, tasks that cannot be accomplished without communication are referred to as COMM tasks. The requirement for communication may be a consequence of sensor limitations. It can be argued for example, that with complete sensing, explicit communication is never required. SENSOR_LIM DISPERSED SYNC tasks however are likely to require communication as a means of achieving synchronization.

4.1.6 Classification of example tasks

The classification system outlined above is used in later chapters to classify tasks in the experimental investigation. The classification of tasks and the evaluation of diversity and performance of robot systems executing them will help identify which kinds of task are best served by diverse teams. Consider the following two tasks as examples of how classification works.

A foraging task might be expressed as "maximize the number of attractors delivered to homebase in 10 minutes." Or more explicitly as

$$\text{delivered}(A_i, t) = \begin{cases} 1 & \text{if object } A_i \text{ is in the delivery zone at time } t \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$$P = \sum_{i=1}^{N} \text{delivered}(A_i, t_0 + 10\text{min}) \tag{4.2}$$

where delivered(A_i, t) indicates whether object A_i is in the delivery area at time t, N is the number of objects available for collection. P is performance, the number of objects in the delivery area 10 minutes after the start of the experiment (t_0). The performance metric includes constraints on time (a 10 minute limit) and on the spatial arrangement of the collected objects (they must be in the delivery area). For this task there are no constraints on the location or number of robots. This foraging task is

- TIME_LIM because performance is measured over a fixed period,
- OBJECT_BASED since performance is based on the location of objects, not agents.,
- RESOURCE_LIM because as agents collect objects, the the availability of attractors is reduced.,

- COMP_INT because robots on a team compete for access to attractors among themselves,
- SINGLE_AGENT since an individual agent can perform positively, even though multiple agents may provide improved performance.

Now consider a formation maintenance task: "minimize total position error for four robots in a diamond formation," or mathematically,

$$\text{error}(R_i, t) = \mid \text{position}_i(t) - \text{desired_position}_i(t) \mid \tag{4.3}$$

$$P = - \sum_{\tau=t_0}^{t_0+10\text{min}} \{ \sum_{i=1}^{4} \text{error}(R_i, \tau) \} \tag{4.4}$$

where $\text{error}(R_i, t)$ is the formation position error for robot i at time t and P is the sum of errors of the four robots over 10 minutes. This task is distinguished from the foraging task above in that the positions of the agents themselves are important rather than the positions of objects to be manipulated. This task is

- TIME_LIM because the task is carried out over 10 minutes,
- ROBOT_BASED because performance depends on the location of agents, not objects,
- CONVERGENCE since the agents should maintain specific positions close to one another,
- MOVEMENT_WHILE because convergence should be maintained while the robots move,
- MULTI_AGENT since it implicitly requires four agents.

4.2 A taxonomy of reinforcement functions

The reinforcement function is usually closely coupled to the performance metric for a task. In fact many reinforcement learning investigations consider performance, task and reward as one and the same. Since learning agents strive to maximize the reward signal provided them, performance is maximized when their reward closely parallels performance. It is sometimes the case however, that agents cannot or should not be rewarded strictly according to overall system performance. Some examples include:

- The robot's sensors do not provide enough information for an accurate computation of performance.
- The delay in receiving a reward is too great; learning a sequential task is too difficult and/or takes too long.

- Performance depends on the actions of other agents over which the agent has limited knowledge and/or control.

As a result, the performance metric (task) and reward function are often quite different and must be treated separately. The following sections enumerate the various kinds of reward structure that have been developed (by the author and others) in response to the issues listed above. The taxonomy is summarized in Table 4.2. Each type has advantages and disadvantages; the challenge for the designer is to select a reinforcement function that most effectively balances these trade-offs.

Table 4.2: Summary the Multi-robot reward taxonomy.

Descriptor	Meaning
Source of reward	
INTERNAL_SOURCE	reward is internal based on sensor values
EXTERNAL_SOURCE	reward is generated by external agent
COMB_SOURCE	combined internal and external reward
Relation to performance	
PERFORMANCE	reward is tied directly to performance
HEURISTIC	reward based on intuition of state value
Time	
IMMEDIATE	immediate rewards are provided
DELAYED	reward is delayed
Locality	
LOCAL	individual agents receive unique rewards
GLOBAL	all agents receive identical reward signal
COMB_LOCALITY	combination of local and global
Continuity	
DISCRETE	several discrete reward values
CONTINUOUS	reward drawn from continuous interval

4.2.1 Source of reward

A reinforcement function is classified as INTERNAL_SOURCE or EXTERNAL_SOURCE depending on whether the learner computes a reward internally based on its own sensors, or the reward is computed externally by another agent. As an example of an

externally generated reward, consider a dog being trained by his master to fetch a ball. Reinforcement in this case is provided through verbal cajoling, e.g. "no!" "good dog," etc. Notice that the pet's reward is based on its sensing of the trainer's mood and *not* the location of the object to be fetched. Similarly we might train a robot by equipping it with "reward" and "punish" buttons to be activated by a trainer according to his evaluation of the desirability of the robot's behavior. This is the approach Yanco utilized in training a team of communicating robots [YS93].

External rewards might be useful in commercial intelligent systems applications where the manufacturer cannot anticipate how the consumer will use the product. It would be unreasonable for example, to assume that the purchaser of an intelligent water heater would be inclined (or able) to program a reward function for it. A more realistic interface is the provision of a "punish" button to be pressed when the heater fails to turn on and the user receives a a cold shower.

A potential problem with external reward systems however, is the possibility of a mis-match between the agent's sensors and those of the trainer. Suppose the water heater's owner takes showers later in the day on weekends than on weekdays. Unless the heater is equipped with a time of day *and* a day of week clock it will be impossible for it to differentiate between weekends and weekdays. The result being that the heater must remain on for the entire morning to avoid negative reinforcement.

In addition to the potential for sensor mis-match, external rewards complicate the learning problem for planning systems. If the reward is not provided to the planner as a function of sensor state and robot action, the robot must additionally learn a model of the external rewarding agent. To avoid this problem, all of the experiments in this research are conducted using internal reward functions.

There are some situations where it might be appropriate for an agent to take part of its reward from an external source and derive part of it internally. As an example, a dog may need to satisfy its hunger by eating (internal), but also wish to please its owner (external). A third descriptor `COMBINED_SOURCE` is used for reward structures that combine internal and external rewards.

4.2.2 Relation to performance

A reward function is defined as `PERFORMANCE` if and only if maximum reward implies optimal performance. As an example of a `PERFORMANCE`-based reinforcement function

consider the task of docking a boat in minimum time. Performance for this task is $P = -$elapsed_time. The function $R(t) = -1$ is an appropriate PERFORMANCE-based reward for the task because reward is maximized if and only if performance is maximized. If T is the number of steps required to complete the task, $\sum_{t=1}^{T} -1$ is maximized (least negative) when the task is completed quickly.

In many tasks a PERFORMANCE-based reward is *delayed*: it may take many steps for an agent to reach a non-negative reward. In the forage task for instance, measured performance is negative or zero until an attractor is delivered. This delay makes it more difficult for an agent to assign credit or blame to actions taken in the past than if immediate rewards are provided. Heuristic rewards can address this problem by providing more immediate feedback. HEURISTIC rewards are based on the programmer's intuition of the value of an agent's actions in particular states.

As an example of a heuristic reward consider reinforcement in a robot soccer task. A PERFORMANCE-based function rewards or punishes the agents only when a goal is scored. An alternative heuristic approach might reinforce the agents with a reward from -1 to 1 depending on whether the ball is close to their own team's goal (-1), the center of the field (0), or the opponent's goal (1).

While heuristic reward functions usually provide for quicker learning, problems include the possibility of local maxima and/or global maxima that do not correspond with optimal performance. The learning agents may maximize their reward without necessarily optimizing performance.

4.2.3 Time and continuity

As mentioned above, rewards are often DELAYED. This means it may require many (perhaps hundreds) of movement steps before an agent receives a positive reward. Reward functions that provide immediate feedback as to the utility of an action in a particular state are called IMMEDIATE. HEURISTIC rewards often seek to improve performance by by providing more immediate feedback than their PERFORMANCE-based counterparts.

There are some tasks that provide IMMEDIATE PERFORMANCE-based rewards as well. As an example, consider the task of maintaining a specific water level by opening and closing a valve. The difference between the current water level and the desired level (the error) can be utilized as an IMMEDIATE PERFORMANCE reward.

The error signal in water level is a CONTINUOUS reward signal because it is drawn from a continuous interval. Other reward functions provide DISCRETE values. As an example of DISCRETE reinforcement consider the reward for a foraging robot. A discrete function might provide a reward of -1 at every step except +1 when the robot delivers an attractor.

4.2.4 Locality

An important consideration for learning multi-robot teams is whether each agent should be rewarded individually or all agents should receive the same reinforcement. GLOBAL reinforcement refers to the case where a single reinforcement signal is simultaneously delivered to all agents, while LOCAL reinforcement rewards each agent individually. In the case of soccer, a global system would reward all team members when any one of them scores a goal. With local reinforcement only the agent that scored the goal would be rewarded. Global rewards correspond more closely with overall system performance, but they may not be appropriate for all tasks.

In a SENSOR_LIM task, it may not be feasible for each agent to monitor activities of all the others in order to compute the global reward. In this case, implementation of global reinforcement may require expensive communication hardware. Also, there may be a weak correlation between an individual agent's actions and the value of the global signal; an agent may receive a positive reward while executing the "wrong" action. This could occur in a globally-rewarded foraging task if two agents simultaneously deposit attractors – one in the correct spot but the other in the wrong spot. Since all agents are simultaneously rewarded for the first agent's successful delivery, the second agent may incorrectly infer it behaved correctly.

Local reinforcement has the advantage of a stronger correlation between reward and the individual's actions, potentially providing for faster learning. But agents using local rewards for training may not be able to optimize performance of the overall system. Some investigators try to balance these goals by combining or blending local and global rewards [Mat94]. These reward systems are referred to as COMBINED_LOCALITY.

The impact of local and global reinforcement on robot team diversity is explored in depth in the foraging, soccer and formation experiments described later in this dissertation (Chapters 6 7 and 8).

4.3 Discussion and summary

To facilitate the investigation of how task and reinforcement impact multi-robot system diversity and performance, new classifications of task and reinforcement are introduced.

Task classification is based on features of the performance metric, environment and robotic platform that differentiate one task from another. The characterization focuses on robot and object movement tasks, and on features of these tasks that may correlate with a need for cooperation or diversity. This classification system helps us identify why multi-robot performance and diversity might change for different tasks, even when the same type of reinforcement is utilized to train the agents.

The chapter also presents a new taxonomy of reinforcement function. Ideally, learning robots are trained using a reward signal that parallels the performance metric. It is not always feasible however to utilize performance-based rewards in a multi-robot system. Research presented later in the dissertation shows that the choice of reinforcement can impact learning rate, performance and diversity in different ways. A multi-agent reward structure is classified as

- `INTERNAL` or `EXTERNAL` depending on whether the robot generates its own reward signal or it is provided by an external agent,
- `PERFORMANCE` or `HEURISTIC` depending on whether the reward is closely tied to system performance or based on the programmers intuition,
- `IMMEDIATE` or `DELAYED` depending on whether the rewards are provided at each step, or only at the end of the task,
- `DISCRETE` or `CONTINUOUS` depending on whether the reward is drawn from a continuous or discrete distribution, and
- `LOCAL` or `GLOBAL` according to whether each agent is rewarded individually or all agents are trained with identical rewards.

This classification helps define the experimental space of the research. It will also benefit the field by helping designers more easily identify the tradeoffs they make in the design of reward functions used in their systems.