Optimal Behavior in Group Environments

A Thesis presented

by

Sanmay Das

to

Computer Science

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

April 2, 2001

Abstract

Rational, autonomous agents must be able to revise their commitments in the light of new opportunities. The SPIRE experimental system allows the study of intention reconciliation in team contexts. Previous work on SPIRE has established some basic results in terms of agent-group interaction under different environmental conditions. It has also presented several decision-making strategies that can be used by agents. Different decision-making strategies lead to optimal performance in different environments and communities of agents. Since agents cannot know the model of the environment and the agent community in advance, it is clear that they need to adapt to their environments and the agent communities they are part of in order to achieve optimal performance.

This thesis introduces a new model for the SPIRE system. The new framework simplifies the previous model of agent-group interaction in ways that enable mathematical specification and provide a basis for the study of learning. Within the new framework, the utility-based decision-making strategies used in SPIRE reduce to cutoff functions, which specify that an agent should default on its commitment to a group-related activity if and only if the value of an outside offer exceeds the value of the group-related activity by more than a certain threshold. This thesis introduces a novel algorithm called cutoff-learning that allows individual agents to learn near-optimal policies by searching through the space of cutoff values and examines the algorithm in contexts where it is used by a single agent and where it is used by multiple agents.

The results presented in this thesis confirm previous work in suggesting that an intermediate level of defaulting on group commitments is often optimal for homogeneous communities of agents in environments with built-in disincentives for defaulting. Experiments also demonstrate the free-rider effect in heterogeneous communities, where it is always individually optimal for an agent to default a large fraction of the time. The cutoff-learning algorithm allows agents to quickly learn policies that lead to good performance. When all agents use the algorithm simultaneously, it leads to behavior that is near-optimal globally, although the algorithm may be problematic when agents are not learning simultaneously and at comparable rates.

Acknowledgments

First, I would like to express my gratitude to my advisors, Prof. Barbara Grosz and Prof. Avi Pfeffer. Barbara stimulated my interest in Artificial Intelligence and Multi-Agent Systems two years ago and has been at the same time a stimulating person to learn from and an incredibly supportive advisor. Her wealth of knowledge and her ability to frame research problems in a larger context are inspiring. Avi has provided insight into many problems and issues, and has guided me in fascinating directions, always challenging me to tackle harder, more interesting problems. Working with Barbara and Avi has made this project a lot of fun.

Dave Sullivan introduced me to the code-base for the SPIRE system and has been terrific in providing help with both theoretical and practical issues. Many of Sarit Kraus' ideas and motivations for the DSCP in SPIRE have been incorporated into the new model presented in this thesis. Dave Sullivan, Alyssa Glass and Michael Epstein designed the original SPIRE system under the direction of Barbara and Sarit. Funding for SPIRE research has been provided by National Science Foundation grants IIS-9978343, IRI-95-25915, CDA-94-01024 and IIS-9907482.

I would also like to thank Jishnu Das for frequent discussions from the perspective of Economics. Adriana Karagiozova and Jishnu provided helpful comments on an earlier draft. Radu Iovita helped with last-minute proofreading. Members of the AI Research Group provided useful comments during the formative stages of the project. Thanks to Stuart Shieber for agreeing to read and evaluate the final product.

This project has been a long endeavor and I would like to thank my friends who have been there for me during this time. Specially Radu for making the last four years so much fun, Jeff for many late-night sanity checks, Nathaniel, Nick and Dino for being great roommates and JoRel for her support when I began working on SPIRE. Above all, I'd like to thank my brothers Saumya and Jishnu and parents Ranen and Veena for starting me on this journey and constantly giving me their love and support.

Contents

1 Introduction and Related Work **1**

1.1 Autonomous Agents in Collaborative Environments 1

1.2 Intention Reconciliation and SPIRE 3

1.3 Learning in Multi-Agent Systems 6

1.4 Contributions 8

1.5 Overview 9

2 The SPIRE Model **11**

2.1 The Initial Model 11

2.2 A New Model for Reputation, Group-Related Tasks and Outside Offers 13

2.2.1 The Market-Oriented Motivation 13

2.2.2 Group-Related Tasks and Outside Offers 14

2.2.3 Group and Individual Reputations 16

2.2.4 Typical Parameter Settings 18

3 Static Decision-Making Policies **19**

3.1 Decision-Making Using One Step Lookahead and Cutoff Functions 19

3.1.1 The Decision Making Function **DF**(δ) and the Cutoff Function **CF**(v) 20
3.1.2 The Function **DF**(δ) is Equivalent to a Cutoff Function 21
3.2 Experimental Methodology 22
3.3 Homogeneous Communities of Agents Using Cutoff Functions 23
3.4 Heterogeneous Communities of Agents Using Cutoff Functions 26
3.5 Discussion 27

4 Learning Decision-Making Policies 29
4.1 An Algorithm for Learning Cutoffs 29
4.2 Empirical Results 32
4.2.1 Methodology 32
4.2.2 A Single Learning Agent in Different Communities . . 32
4.2.3 Learning in Two Agent Settings 35
4.2.4 Performance of a Learning Agent Over Time 35
4.2.5 Learning When Other Agents Also Learn 38
4.3 Another Framework for the Learning Problem 40

5 Conclusions and Future Work 44

Chapter 1

Introduction and Related Work

1.1 Autonomous Agents in Collaborative Environments

Artificial Intelligence (AI) systems are becoming increasingly complex and robust. Within the last decade, a car has steered itself across most of the United States (Pomerleau, 1993) and the successful Deep Space One mission used AI technology for spacecraft control (Weld, 1999). In addition, there are now many opportunities for complex artificial agents to collaborate. The field of Distributed Artificial Intelligence (DAI) deals with the management of individual and group behavior of collections of interacting, autonomous agents. While software agents connected through the Internet have garnered much attention recently, and are certainly a very important example of collaboration among software agents, there are many other contexts in which software agents collaborate with each other and with people.

For example, in the landmark RoboCup soccer project (Kitano et al., 1997), teams of robotic soccer players compete against each other. In such an environment an agent must both collaborate with teammates and compete with adversaries. As a result of the success of the RoboCup soccer project, a more socially significant challenge application of Multi-Agent Systems has recently been proposed. The Robocup Rescue application centers around disaster rescue and recovery (Kitano, 2000). Designing a team of agents with different capabilities that must collaborate in a hostile environment like disaster recovery is a difficult task that provides a useful testbed for work in Multi-Agent systems.

The field of Multi-Agent Systems focuses on the group behavior of collections of autonomous, individually rational agents. Collaborative behavior in such a context involves interaction among the agents. Research on collaborative planning has established that commitment to the group activity is a defining characteristic of collaboration (Grosz et al., 2001, *inter alia*). For successful completion of group activities, individuals must be committed to performing the tasks they are themselves assigned, and they must believe that other agents will also complete their tasks. On the other hand, individually rational agents must be able to maintain some flexibility, consider opportunities that arise after commitment to a group activity, and choose their actions freely.

The benefits of allowing agents such autonomy and flexibility are clear. Consider a robotic agent in a Robocup-Rescue setting that has agreed to perform a particular action as part of a larger group task intended to recover valuable jewellery from underneath the rubble of an earthquake. The agent is proceeding towards the place it has committed to reach by a particular time. On its way, it comes across a survivor lying underneath the rubble.

The agent has the capabilities necessary to rescue the survivor, but doing so would necessarily involve reneging on its commitment to the group activity to recover jewellery. If the agent is incapable of reneging on its commitment to the group, it will be unable to rescue the survivor, which is obviously an undesirable outcome.

1.2 Intention Reconciliation and SPIRE

Sullivan et al. (1999), citing Grosz and Kraus (1996) and Bratman (1987), note that rational agents cannot adopt conflicting intentions. If an agent has committed to a group task (commitment involves adopting an intention to perform a task), it cannot also adopt an intention to perform some other activity that conflicts with its intention to perform the group-related activity, the individual task that is part of the group activity. Intention reconciliation is the process whereby an agent reconciles its intention to perform a group-related activity with other, potentially conflicting possibilities. Since the agent cannot commit to both the group-related activity and the other activity, it must decide whether to maintain its intention to perform the group-related task or renege on its commitment to the group task and adopt an intention to perform the other activity.

Several recent approaches to modeling the intention reconciliation problem use a market-oriented or contract-based framework. Sandholm and Lesser (1996) introduce the concept of *leveled commitment contracts* in which agents pay the rest of the group a predetermined penalty for defaulting on a group-related task. Sen and Biswas (1998) introduce a setup in which there is no direct market-mediation, but instead agents choose which others to work with based on previous experiences. Teague and So-

nenberg (2000) investigate the effects of different ways of imposing penalties for defaulting in the context of a "target-capturing" game.

The SPIRE (SharedPlans Intention Reconciliation Experiments) simulation system is designed to allow the study of the effects of different environmental parameters and agent decision-making procedures in a collaborative setting (Sullivan et al., 1999; Glass & Grosz, 2000; Sullivan et al., 2000; Grosz et al., 2001). The work presented in this thesis, and the SPIRE project in a larger context differ from the work mentioned above in some key respects. First, they explicitly examine the interaction between an agent and the entire group it is part of, rather than interactions between individual agents. Second, the SPIRE framework uses game-theoretic and decision-theoretic analyses and algorithms that work in a fairly general, domain-independent context. Furthermore, this work examines the behavior of agents that are tied into working in a particular group. Agents do not have the option of leaving the group. The best analogy for the SPIRE framework is that of a company or firm that gets tasks from the market and then subcontracts these tasks to its agents. The SPIRE system is intended to elucidate the problem of when it is appropriate for an agent to default on a commitment to such a *group-related task* to commit to a potentially more valuable *outside offer*.

In previous work on the SPIRE system, penalties for defaulting are imposed in three different ways. First, there is a penalty for reneging on a group-related activity that represents the cost to the group. This penalty is shared across the entire group. Second, a percentage of tasks is allocated such that agents who default less get more valuable tasks. Finally, it is also possible to impose a measure of self-imposed reluctance to default. This "good-guy" effect, modeled using "brownie points" is described in detail by

Glass (1999) and Glass and Grosz (2000).

The model presented in this thesis is broadly based on the original SPIRE framework (Sullivan et al., 1999), but represents a significant simplification of, and, in places, departure from, that model. The new model serves a twofold purpose. First, it is more amenable to mathematical analysis of baseline decision-making procedures such as never reneging on a commitment to the group, or always reneging when the outside offer gives the agent more utility than the group-related task. Second, it isolates the most important system parameters, which allows the study of agents that attempt to *learn* optimal behavior.

One of the key elements of the SPIRE framework is a precise mathematical notion of reputation for each agent. An agent's reputation is hurt whenever it defaults on a commitment to a group-related task, but previous "bad behavior" is gradually forgiven over time if the agent behaves well by not defaulting on commitments to group-related activities. The average utility of group-related tasks an agent receives is a function of its reputation. Further, in the model used for this work (following Grosz et al. (2001)), the values of tasks each agent receives are also dependent on the reputation of the group as a whole. Hence the behavior of each agent affects not only the utility it can expect to receive in the future from group-related activities, but also the utility each and every member of the group can expect to receive in the future from group-related activities.

Within this framework, simulations are performed in which agents interact repeatedly with the environment by choosing whether or not to default on commitments to group-related tasks. The framework allows for the simulation of heterogeneous communities of agents that have different decision making procedures. Agent performance is typically measured by average

income (or reward) received over the course of the simulation.

1.3 Learning in Multi-Agent Systems

Reinforcement learning is the field of AI that addresses the question of how an autonomous agent that senses and acts in an environment can learn to choose actions that maximize its utility[1]. The agent typically learns a *policy*, a mapping from states to actions that specifies the action the agent should take in any given state. Most work in reinforcement learning uses the *Markov Decision Process* (MDP) framework. An MDP specifies a set of states, a set of actions available to an agent, the probabilities of transitions between each pair of states given an action (the *transition model*), and the rewards associated with each state (the *reward model*). A critical effect of this definition is that the probability of a transition from one state to another depends solely on the current state and action, and not on the history of actions an agent has taken or states an agent has visited.

Most work on reinforcement learning is in the context of *fully observable*, *stationary* environments. A fully-observable environment is one in which an agent knows the state it is in at any given point in time. A stationary environment is one in which the parameters do not change over time. These parameters may be deterministic or probabilistic, but they must remain fixed.

In general, real-world environments are not fully observable. It is unrealistic to expect that an agent knows its exact position in the state-space at any point in time. A *reactive* or *memoryless* policy is sufficient in fully

[1]For a detailed exposition of the MDP framework, see Mitchell (1997) or Russell and Norvig (1995)

observable MDPs, but some form of memory is necessary to learn optimal policies in partially observable MDPs (POMDPs) (Meuleau et al., 1999).

Another property that makes learning about real-world multi-agent environments difficult is their potential non-stationarity[2]. If an agent is learning in an environment where its payoffs are affected by the behavior of other agents, the environmental parameters are significantly affected by the behavior of the other agents. This is not a problem when all the other agents are following static or fixed policies. However, if the other agents are also learning about the environment and modifying their strategies over time, none of the standard reinforcement learning algorithms are guaranteed to converge. *Stochastic games*, also known as *multi-agent MDPs*, are a natural extension of the MDP framework to multi-agent domains. Stochastic games (SGs), first introduced by Shapley (1953), have been studied in the economics and game theory communities. However, the initial work of Shapley, and further work such as that of Vrieze (1987), assumes a requirement even stronger than full observability. It requires that the agent know the transition and reward models, which is not the case in the environments studied here. Recently, reinforcement learning techniques have also been applied to SGs (Claus & Boutilier, 1998; Bowling & Veloso, 2000b).

Stone and Veloso (2000) divide multi-agent environments into four categories based on whether the agents involved comprise a homogeneous or heterogeneous community, and whether inter-agent communication is possible or not. This thesis is concerned with homogeneous and heterogeneous communities of interacting but non-communicating agents. In the model used in this work, the behavior of an agent affects other agents, but the effects are only felt through the impact of the agent's actions on group rep-

[2]Most of this paragraph is based on the presentation of Bowling and Veloso (2000a)

utation. I present an algorithm that allows a single agent to learn policies that perform well across different environments, and discuss its applicability to situations in which more than one agent is learning.

1.4 Contributions

This thesis presents a new model for the study of intention reconciliation. The framework builds on the previous model of agent-group interaction in the SPIRE system, but simplifies the model considerably and provides a framework for the study of learning. There is a significant reduction in the number of parameters used by the system, which makes the new model more conducive to analytical treatment. The new model isolates the most important aspects of agent-group interaction.

Within this model, the fully-informed decision-making strategy used by Grosz et al. (2001) reduces to a simple strategy agents can use to decide whether or not to default on a group commitment without using any information about the state of the world. This strategy involves simply maintaining a *cutoff value* and defaulting whenever the utility of the outside offer exceeds the utility of the group-related task by more than that value. I present empirical results for communities of homogeneous and heterogeneous agents using such cutoff functions. The results for homogeneous communities show that there is a cutoff that optimizes mean individual utility received by agents and that this is a global maximum.

It is unrealistic for designers of real-world multi-agent systems to evaluate many different decision-making procedures through extensive simulations to decide on the optimal procedure. Further, the composition of a group of agents or the exact parameters and model of the environment may

be unknown at the time agents are designed. It is necessary for agents to adapt successfully to different environments. The simplicity of the cutoff-based strategy motivates an algorithm for *learning* the optimal cutoff value for a single agent when the environment is partially observable (group and individual reputations are unknown). This thesis introduces an algorithm called cutoff-learning which allows individual agents with no knowledge of the model underlying the environment to quickly learn near-optimal cutoff functions. I present empirical evaluations of this algorithm in scenarios where a single agent is learning about a community of agents that use fixed decision-making functions. I also consider the optimality of this algorithm when multiple agents in the same community are using it for learning, and find that the algorithm produces nearly optimal global results when all agents use it simultaneously. This thesis also includes a discussion of the benefits and limitations of cutoff-learning and presents the learning problem in a general framework.

1.5 Overview

This thesis is organized as follows: Chapter 2 briefly reviews the SPIRE model as presented by Grosz et al. (2001) and then presents the new model, focusing on group-related task values, outside offer values and the evolution of individual and group reputations over time. Chapter 3 introduces the static or fixed decision-making policies agents can use, and presents empirical results for homogeneous and heterogeneous groups of agents. Chapter 4 focuses on the problem of learning how to act, first presenting an algorithm that allows a single agent to learn an optimal policy in a group consisting of agents using static decision making policies and then examining the stability

of this algorithm when more than one agent is learning. Chapter 5 presents conclusions and suggests directions for future work.

Chapter 2

The SPIRE Model

2.1 The Initial Model

In SPIRE, a group of agents works together on a set of group tasks[1]. An agent performs a group-related activity or group-related task (sometimes referred to simply as tasks) as part of a larger group task. Task assignment is done in blocks of weeks, where a week consists of 40 time slots. The *task density* is a parameter that reflects how many of the 40 time slots are actually occupied by a group task in a week. The utility or value of each task is drawn from a uniform distribution. Outside offers (referred to as offers) are generated a fixed percentage of the time and are offered to a random agent. The distribution that offer values are drawn from is also uniform and fixed so that the lowest possible offer value is higher than the highest possible task value. When an agent receives an offer for a time slot in which a task is already scheduled, it must decide whether to renege on the group-related task to accept the outside offer or not. Another agent may be

[1] This discussion follows Sullivan et al. (1999) and Grosz et al. (2001)

able to replace an agent that defaults on a commitment to a group-related activity.

The penalties and costs for defaulting are imposed in two different ways. First, there is a baseline cost for every default that represents the cost of finding a replacement. If a replacement is not found, the group has to pay a penalty proportional to the value of the group-related task. The second way in which penalties are imposed on agents who perform poorly is based on the individual reputation of each agent. Each week, a portion of the group-related tasks are assigned according to reputation, and agents with better reputations get higher valued tasks.

This assignment is done in two different manners in previous work on SPIRE (Grosz et al., 2001). One method of assigning tasks (known as the *Ranking Social Commitment Policy* or *RSCP*) to agents based on reputation involves ranking all the agents according to reputation and then distributing tasks according to this ranking. This method leads to a very difficult game-theoretic situation when agents try to estimate future income. If an agent wants to know how much its ranking will suffer from defaulting on a particular group-related activity, it has to take into account the possible behaviors of all other agents. This difficulty in estimation led to the development of the *Discounting Social Commitment Policy (DSCP)* model, in which the values of tasks an agent is assigned depend only on the agent's individual reputation and the reputation of the group as a whole. In this model, instead of having to predict the behavior of every member of the group, an agent only has to predict the behavior of the group as a single entity, reducing the game-theoretic situation to a much simpler two player game, and making the model of agent-group interaction easier to reason about.

Because of the many parameters and different effects involved in the original SPIRE model, it is difficult to reason about the causes of various effects. Further, recent work by Teague and Sonenberg (2000) suggests that different methods of imposing penalties like the two mentioned above, as well as the self-imposed reluctance of agents to default (as discussed in the "brownie point" model of Glass and Grosz (2000)) lead to similar defaulting dynamics in the group. Building on this and the absence of significant effects of "brownie points" in the DSCP, Grosz et al. (2001) present a simplified model of agent-group interaction that is based on a market-oriented framework and provides a testbed for examining basic theories about agent-group interaction in differently constituted agent communities, and for experiments on learning how to behave in complex, multi-agent environments.

2.2 A New Model for Reputation, Group-Related Tasks and Outside Offers

2.2.1 The Market-Oriented Motivation

Consider a group (for example a company or firm) that receives a certain number of tasks to be performed at each time period from some outside contracting agency. The group then distributes these tasks among its own workers who cannot refuse a task assignment at the time when tasks are assigned. However, the workers may receive outside offers that they can then take. If an outside offer conflicts with a group-related task that the agent is supposed to be performing, the agent may renege on the group-related task.

The market will be unwilling to pay as much to a group with a poor

reputation as it would pay to a group with a good reputation for performing tasks with equal values. A group can acquire a poor reputation because many of its employees renege on their group commitments on a regular basis, leading to tasks not being completed. In the same way, a group has the ability to improve its reputation and get paid more for completing an equivalent set of tasks. So a group has the ability to change the amount of income it makes from the tasks it receives.

Suppose the group receives an income I from a particular task performed by a member-agent. The group is unlikely to give the agent the entire sum I. Instead it will give the agent some proportion of I, keeping the rest for such things as operational expenses and overhead. The amount the group keeps is determined by the individual reputation of the agent within the group. The upper bound on what the agent can make from the task is, of course, I. It is important to note that, in this framework, the overhead the company keeps can be thought of as a cost associated with agent defaults. It is not intended to be a profit margin that the company management tries to maximize.

2.2.2 Group-Related Tasks and Outside Offers

An important modification from the original SPIRE system is that task values are now drawn from a normal distribution truncated at 0, instead of from a uniform distribution. Further, all tasks are discounted; this differs from the method used by Grosz et al. (2001) who only discount a fixed proportion of the tasks. When only a small proportion of group-related tasks are assigned based on reputation, observations of individual utility received tend to be damped by the randomly distributed tasks. This modification is

intended to remove the damping effect. It also serves in part to replace the concept of fixed group penalties associated with defaulting.

Group-Related Task Values

Values are drawn from a normal distribution with a specified mean $\mu = M_0$ and standard deviation σ. Whenever the value generated is below 0, the task value is set to 0. M_0 represents the average "market-value" of the task. Depending on the reputation of the group, the group could be offered more or less than M_0 to perform this task. Let group reputation be $\mathbf{GR}$, and let α be a parameter associated with $\mathbf{GR}$. Then, for performing a task with average market value M_0, the group will get $M_0 + \alpha\mathbf{GR}$. Now, the value of performing this task to an agent a is given by $M_0 + \alpha\mathbf{GR} + \beta\mathbf{IR}_a$, where $\mathbf{IR}_a$ is a measure of the individual reputation of agent a and β is an associated scaling parameter.

An explicit task-density parameter (with all scheduled tasks having positive values) becomes unnecessary. The task-density parameter arises naturally because an agent can be assumed to be free to take an outside offer without having to default during the time slots when it would receive a task with value 0. This can lead to situations where a task with value to the group has no value for a particular agent because of the $\beta\mathbf{IR}_a$ factor. It would be convenient to ensure that such situations do not arise, perhaps by checking for such situations and giving those tasks to agents for which this would not be the case. However, this is not a common enough occurrence to justify the added complexity.

Outside Offers

Outside offers are also drawn from a normal distribution. The distribution of offer values will typically have a lower mean and higher standard deviation than the distribution of task values. This model, in keeping with the SPIRE model presented by Grosz et al. (2001), ensures that outside offers are only attractive to agents a certain percentage of the time. Also, outside offer values sometimes exceed group-related task values significantly.

An outside offer is generated for each agent at each time slot. This is different from the original system in which offers were only generated a fixed proportion of the time. However, this "outside offer percentage" parameter is replicated in the new system because it is never advantageous for an agent to default on a group-related task unless the competing outside offer is more valuable than the value of the task to the agent. Thus for any choice of means and standard deviations for task values and offer values, a certain percentage of outside offers will be more valuable than the group-related tasks. This parameter is dependent on individual and group reputations and hence will not be exactly the same (although it is reasonably close) for different communities of agents.

2.2.3 Group and Individual Reputations

The week-based framework in which there were 40 time slots per week has been eliminated. In the old system, reputations only changed at the end of each week. In the new model, reputation changes are effectively instantaneous, since new task and offer values for the next time slot take into account the changed reputations. γ is defined as a parameter in the range $[0, 1]$ that allows "forgetting" of previous defaults, so that reputation can

improve over time.

Group reputation **GR** is defined as a value in the range $[-0.5, 0.5]$. The evolution of **GR** from one time period t to the next, $t+1$ is governed by the equation:

$$\mathbf{GR}^{t+1} = \gamma \mathbf{GR}^{t} + (1-\gamma)(-0.5\frac{num\text{-}defaults^{t}}{num\text{-}agents})$$

where $num\text{-}defaults^{t}$ represents the number of defaults that occurred in the entire group at time t. Thus, group reputation will be stable at around 0 if approximately half the members of the group default all the time, and will asymptotically approach 0.5 if none of the members ever default and -0.5 if all the members always default. This ensures that **GR** is robust to many different environments in terms of outside offer means and standard deviations.

The individual reputation of an agent a at time t, $\mathbf{IR}_a^t$ is defined in the range $[-1, 0]$ so that it always serves as a negative factor, and an agent cannot make more than its group was given for a particular task. **IR** evolves as follows:

$$\mathbf{IR}_a^{t+1} = \gamma \mathbf{IR}_a^{t} + (1-\gamma)(-default(a,t))$$

where $default(a,t)$ is an indicator function that is 1 if agent a defaulted at time t and 0 otherwise. This function is again symmetric and linear. It avoids some of the problems associated with nonlinear reputation functions. Nonlinear functions, for example sigmoids, are more useful in certain environments like the 40 time-slot based weeks in the original SPIRE system, because the change of reputation does not occur till the end of the week, and an agent has the opportunity to default multiple times in one week —

if the reputation function were linear in such a case, an agent that defaulted once would almost always default all the time, and this leads to extreme behaviors in most cases. In an immediate-update scenario like the one presented here, the forgetting factor is strong enough to allow different agents to have IRs in different ranges without seeing extreme behavior like always defaulting or never defaulting.

2.2.4 Typical Parameter Settings

In the experiments presented here, the task value mean M_0 is set to 1 and the task value standard deviation is also set to 1. The scaling parameter α associated with **GR** is set to 2, so that the group reputation shifts the task value mean between 0 and 2. This heavy weight on **GR** replaces to some extent the notion of group costs in the original SPIRE model. Outside offers come from a distribution with a lower mean (0) and a higher standard deviation (typically 3), so that outside offers that are more valuable than the group-related activity assigned to an agent at the same time do not arise frequently, but they have the potential to afford the agent considerable utility.

Chapter 3

Static Decision-Making Policies

3.1 Decision-Making Using One Step Lookahead and Cutoff Functions

In work on the DSCP in SPIRE, Grosz et al. (2001) use a decision-making function that assumes full knowledge of both the state (including the individual and group reputations) and the equations which determine the state transitions (the *transition model*). This chapter defines a similar decision-making function **DF** that is essentially a direct translation of the utility-based decision-making function in the DSCP to the new model. This informed decision-making function is provably equivalent to a decision-making function that does not need any information about the transition model or the state, but simply determines whether to default or not based on whether the difference in the outside offer value and the task value to the agent is greater than a specified cutoff. This chapter presents empirical results for

homogeneous and heterogeneous communities of agents using such "cutoff functions."

3.1.1 The Decision Making Function DF(δ) and the Cutoff Function CF(v)

The decision making function **DF**(δ) is based on one-step lookahead. The parameter δ specifies the weight an agent attaches to its estimate of future expected income. All the work presented in this thesis uses the simplifying assumption that agents are not aware of the finite horizon when their interaction with the group will come to an end. Therefore, agents always make the assumption that their interaction with the group will continue for an infinite number of time-steps.

DF(δ) computes future expected income (FEI) in the cases where an agent defaults and those in which it doesn't default. FEI does not take into consideration future outside offers, it only accounts for future group tasks. The expected income for the next week is computed for the above two cases. As discussed above, the agent knows the mean of the distribution that initial task-values are drawn from and the parameters α and β, along with $\mathbf{IR}_a^t$ and $\mathbf{GR}^t$. This income is then multiplied by a weighting factor based on δ. In the infinite horizon case we consider, this weighting factor is the sum of the infinite geometric progression $\delta, \delta^2, \delta^3, \ldots$, which is $\frac{\delta}{1-\delta}$. If the weighted difference between the estimated FEI given the agent does not default and the weighted FEI given that the agent does default is greater than the gain in CI that can be obtained by defaulting, the agent does not default, otherwise it does.

The cutoff function **CF**(v) for decision making is defined in a simple

manner. An agent using $\mathbf{CF}(v)$ as the decision function chooses to default whenever the difference between the outside offer value and the task value for a week is greater than v.

3.1.2 The Function $\mathbf{DF}(\delta)$ is Equivalent to a Cutoff Function

Suppose the task value for time period t is T and the offer value is O. To summarize the notation:

1. M_0 is the mean from which task values are drawn before factoring in reputations.
2. $\mathbf{GR}^t$ is the group reputation at time t.
3. $\mathbf{IR}_a^t$ is the individual reputation of agent a at time t.
4. δ is the parameter used by agents using **DF** to discount future expected income.
5. γ is the "forgetting" parameter in the evolution of **IR** and **GR**.

Define $\mathbf{IR}_a^{t+1}(nodef)$ to be the reputation of agent a if it does not default at time t and $\mathbf{IR}_a^{t+1}(def)$ to be the reputation if it does default at time t. Let FEI-NODEF be the weighted expected future income as defined by the function **DF** if the agent does not default at time t and let FEI-DEF be the weighted expected future income as defined by DF if the agent defaults at time t. Then:

$$\text{FEI-NODEF} = \frac{\delta}{1-\delta}(M_0 + \beta\mathbf{IR}_a^{t+1}(nodef) + \alpha\mathbf{GR}^t)$$

$$\text{FEI-DEF} = \frac{\delta}{1-\delta}(M_0 + \beta\mathbf{IR}_a^{t+1}(def) + \alpha\mathbf{GR}^t)$$

Therefore, the difference in FEI is given by

$$\text{FEI-DIFF} = \frac{\delta}{1-\delta}(\beta \mathbf{IR}_a^{t+1}(nodef) - \beta \mathbf{IR}_a^{t+1}(def))$$

Now, $\mathbf{IR}_a^{t+1}(nodef) = \gamma \mathbf{IR}_a^t$ and $\mathbf{IR}_a^{t+1}(def) = \gamma \mathbf{IR}_a^t + \gamma - 1$. Therefore:

$$\text{FEI-DIFF} = \frac{\delta\beta}{1-\delta}(\gamma \mathbf{IR}_a^t - \gamma \mathbf{IR}_a^t) - (1-\gamma)$$

$$\Rightarrow \text{FEI-DIFF} = \frac{\beta(1-\gamma)\delta}{1-\delta}$$

Therefore, if $O - T > \frac{\beta(1-\gamma)\delta}{1-\delta}$, the agent will choose to default. The function $\mathbf{DF}(\delta)$ is equivalent to the function $\mathbf{CF}(\frac{\beta(1-\gamma)\delta}{1-\delta})$.

Note that the exact form of the result presented here does not always hold, because if α and β are sufficiently high and $\mathbf{IR}_a$ and $\mathbf{GR}$ sufficiently low, the values of FEI-NODEF and FEI-DEF may go below 0 in the above analysis, while the agent will always assume that the minimum income it can get from group-related tasks is 0, not negative. However, this case does not arise for any of the parameter settings discussed in this thesis.

3.2 Experimental Methodology

Experiments reported in this chapter were all run using the same basic parameters. Unless otherwise specified, $\alpha = 2$; $\beta = 0.5$; the mean of the distribution group-related task values are drawn from, $M_0 = 1$; the standard deviation of this distribution, $\sigma_{\text{gt}} = 1$; the mean of the distribution offer values are drawn from, $\mu_{\text{oo}} = 0$; the standard deviation $\sigma_{\text{oo}} = 3$. Experiments were run for 1000 time periods, and the first 100 time periods were ignored in tabulating average incomes or rewards to allow the group and individual

reputations to stabilize. Because of the static decision-making functions, allowing only 100 periods for this stabilization and gathering data from 1000 weeks leads to results that are the same as those obtained by running for more time periods. Error bars represent 95% confidence intervals. The next two sections present experimental results.

3.3 Homogeneous Communities of Agents Using Cutoff Functions

These experiments investigate the behavior of communities of agents where all the agents used the same value of v for their cutoff function $\mathbf{CF}(v)$. Obviously, agents using a lower value of v will be less "socially responsible" in the sense used by Glass (1999), that is, they will tend to default more on their group commitments, because it requires a lower incentive for them to be willing to renege on the group.

Figure 3.1 shows this effect clearly. The percentage of defaults decreases as the cutoff value v increases, asymptotically approaching 0. Group task income is defined as the income the group gets from performing tasks. The value of group task income is closely related to the percentage of defaults, and one can see from figure 3.2 that it increases as the percentage of defaults decreases, asymptotically reaching the theoretical optimum, which occurs when none of the agents ever default on group commitments.

The most interesting result is shown in figure 3.3, which shows the mean individual income of group members as a function of cutoff values. Mean individual income is defined as the sum of the incomes an agent receives from group-related tasks and outside offers (keeping in mind that the agent receives less income from completing a group task than the group does).

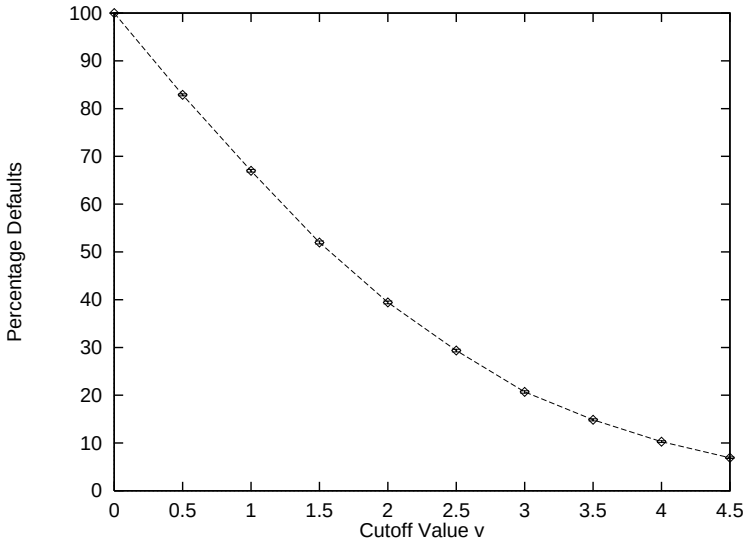

Figure 3.1: Percentage of outside offers accepted by agents, leading to defaults on group-related tasks, as a function of cutoff value

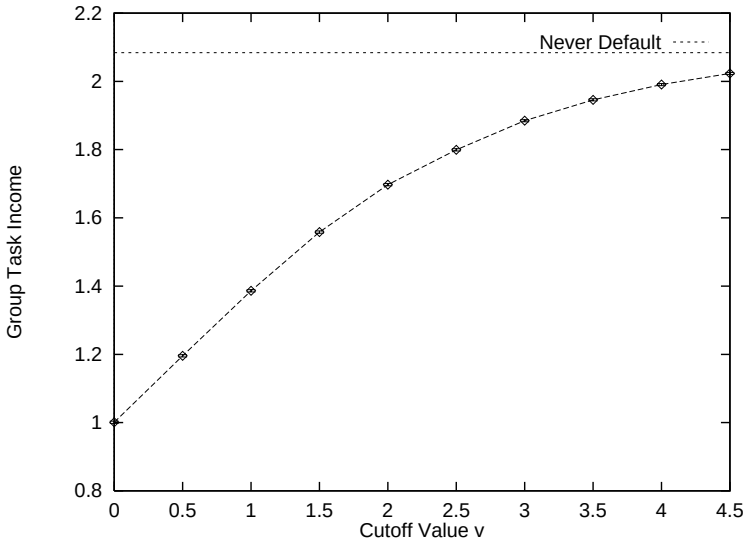

Figure 3.2: Group task incomes as a function of cutoff value

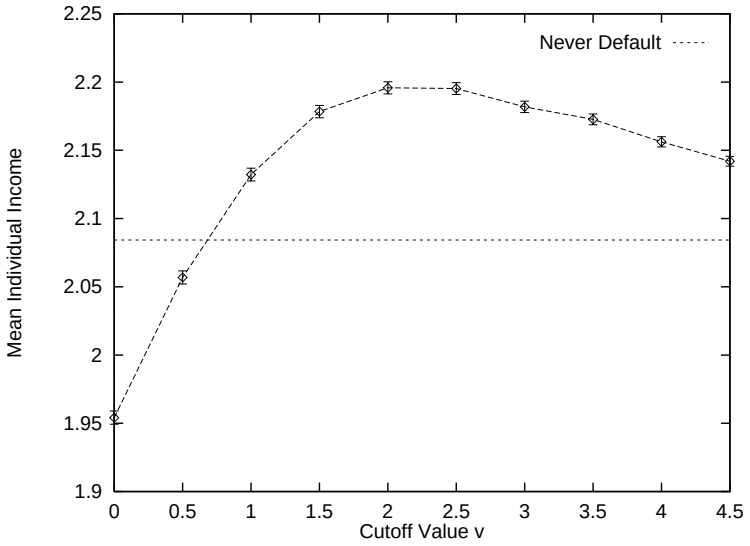

Figure 3.3: Mean individual incomes as a function of cutoff value

The optimal level of defaulting lies in between never defaulting and always defaulting. This not only corroborates the results of Grosz et al. (2001) and Glass and Grosz (2000) that an intermediate level of "social consciousness" is often optimal, it also shows that these results are robust across different models of agent-group interaction, and that they are a basic feature of domains like SPIRE in which there are environmental incentives not to default. In the agent-agent model of interaction of Teague and Sonenberg (2000), self-imposed reluctance to default (which is present in the brownie points model) leads to results similar to those produced by the presence of environmental incentives not to default. The results presented in this section extend this finding to the SPIRE model of agent-group interaction.

3.4 Heterogeneous Communities of Agents Using Cutoff Functions

This section examines the behavior of agents in mixed communities. For the sake of simplicity and establishing some basic results, the experiments focus on the case where a group consists of two types of agents, and the percentage of each type changes. We use representatives of "good" agents, those who are socially responsible and do not default much, and "bad" agents who default more frequently. For the purposes of comparison, two cutoff values are used such that homogeneous communities of agents using these two cutoffs perform at about the same level in terms of mean individual income. The two cutoff values are $v = 1.0$ (bad agents) and $v = 4.5$ (good agents). The good agents performed somewhat better in terms of MII when the communities were homogeneous.

One can see from figure 3.4 that the bad agents outperformed the good agents for any particular percentage of good agents. Therefore, it is individually optimal for any single agent to default frequently in any particular mix of good and bad agents. In fact, a single agent that always defaults (i.e. uses a cutoff of 0) in a community with 39 good agents performs even better than a single agent using a cutoff of 1 (having an MII of approximately 2.411 as opposed to 2.405 in simulations, although the difference is not statistically significant). However, one can also see from figure 3.4 that the presence of a higher percentage of good agents leads to a better outcome for every type of agent. Both good and bad agents perform better when there are more good agents present. If every single agent is bad and defaults a large fraction of the time, this is globally sub-optimal. These results are an example of what is known as the "free-rider" problem in game theory.

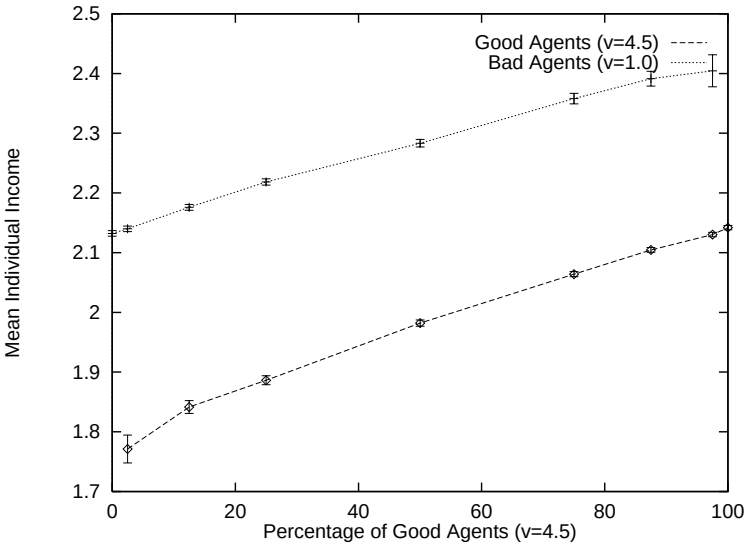

Figure 3.4: Mean individual incomes of good and bad agents that form a mixed community as a function of the percentage of agents that are good

3.5 Discussion

The results presented in this chapter indicate that cutoff functions represent a useful space of policies to study in the SPIRE framework. The smooth curve seen in figure 3.3 indicates that there exists a cutoff that globally maximizes mean individual income in homogeneous agent communities. It is also likely that there is a single best cutoff for an agent to use in any given community. While this cutoff may not represent the best policy if a different model of the evolution of reputations is used, the linearity of the reputation functions ensures that the cost of defaulting to an agent is independent of the value of the agent's individual reputation. Therefore, the decision of whether to default or not should be independent of all aspects except the difference in outside-offer and group-related task values.

The performance of an agent using a fixed policy is dependent on two

factors — the environment, referring to parameters such as the weights of **IR** and **GR** and the agent community, referring to the size of the group and the policies being used by other agents. Experiments show that there is no single policy an agent can follow that optimizes its performance across a range of different communities and environments in the SPIRE model. This motivates the study of learning in such environments. The next chapter investigates adaptive agents that learn cutoff policies based on the environments and agent communities they are part of.

Chapter 4

Learning Decision-Making Policies

4.1 An Algorithm for Learning Cutoffs

In reality, designers of multi-agent systems will not be able to run a large number of simulations to decide on optimal decision-making policies for individual agents. It is necessary for agents to adapt to the environments and communities within which they exist to optimize the utility they receive. This chapter investigates the design of a single agent that learns how to optimize its own reward in the context of any group and set of environmental parameters by learning the optimal cutoff to use. This section presents an algorithm that can be successfully used by a single agent in a community of agents that use static decision-making functions. Later sections examine the applicability of this algorithm when many agents use it simultaneously.

If we restrict the space of policies to the cutoff functions described in chapter 3, the problem can be tackled using a search through the space

of cutoff values. An agent accumulates data about the average utility it receives when using a particular cutoff v as the policy by using that policy for n weeks where n is the desired number of samples, and narrows its search to cutoffs that lead to higher average utilities. This is referred to as "playing the policy v". The agent searches in narrowing increments around the best cutoff found thus far. The question of determining the desired number of samples is a difficult one. For this thesis, the choice of the minimum and maximum number of weeks that an agent is allowed for collecting data on the average utility received when playing a particular policy is based on initial empirical evidence as to the success of different numbers of weeks. While we determine these numbers exogenously, specifying the minimum and maximum number of weeks of data collection, it is possible to use results from statistics to determine the number of samples to collect endogenously. This falls under the purview of *stopping problems*, discussed extensively by DeGroot (1970), and is a subject for future research.

The algorithm cutoff-learning proceeds in two phases. An agent starts by using a cutoff of 0 for 500 time periods and then switches to a cutoff of $0 + i$ where i is an increment. In the first phase of the algorithm, the agent increases its cutoff by twice the previous increment, and uses this cutoff for at least 500 time periods, and at most 1000. In the period when an agent has collected between 500 and 1000 samples, the agent performs a two-sided T-test (using a 0.05 significance level) to determine whether or not the average utility received is significantly different from the average utility received using the previous cutoff. When the difference is significant or when 1000 samples have been gathered, the agent checks which mean utility is greater. If the mean utility of the new cutoff is greater, the agent remains in stage 1, again increasing its cutoff by twice the previous increment i and gathering

data on average utility. If the mean utility of the new cutoff is lower than the mean utility of the previous cutoff, the agent sets m = previous cutoff and proceeds to the second phase of the algorithm.

The second phase of the algorithm is essentially a binary search around m. The agent halves the increment, setting $i = i/2$, and collects average utility data using cutoffs of $m - i$ and $m + i$. Again, data collection stops when it has either played the policy for 1000 time periods, or when it has played the policy for at least 500 time periods and the difference in mean utilities received by playing the current policy and playing the policy v is statistically significant. The value of m is then set to whichever of m, $m + i$ and $m - i$ yielded the highest mean average utility. The process is then repeated with the increment being halved again.

Theoretically, cutoff-learning is an appropriate algorithm for this domain for a number of reasons. It makes no assumptions about the environment or the nature of the community of agents the learning agent is part of. It performs an efficient search through the space of cutoffs, and rapidly narrows its focus to a region in the space of cutoff values that yields high utility. The agent can begin to perform well in terms of utility received at a very early stage while still improving its performance as time goes on. Further, the agent need not have a model of the environment. The algorithm does not require the agent to know individual or group reputations or their evolution through time. The agent needs very little information about the domain to perform well.

Table 4.1: Percentage of defaults by a learning agent in a community of agents using a fixed cutoff

Cutoff Used by Remaining Agents	Percentage Defaults
0	90.3 ± 0.7
1	87.8 ± 0.8
2	83.8 ± 1.0
3	86.9 ± 0.9

4.2 Empirical Results

4.2.1 Methodology

For most of these experiments, groups of 10 agents are considered. 9 of the agents use a specified cutoff. Simulations are run for 22,000 weeks (time periods) and data is collected from the last 2000. The increased number of weeks allows the performance of agents that learn to be measured after they have had enough time to learn a good policy. Separate experimental results show how the performance of a learning agent changes as the number of weeks increases. Error bars in graphs and plus-minus figures in tables represent 95% confidence intervals. Unless otherwise mentioned, it can also be assumed that $\alpha = 2$; $\beta = 0.5$; the mean of the distribution group-related task values are drawn from, $M_0 = 1$; the standard deviation of this distribution, $\sigma_{\text{gt}} = 1$; the mean of the distribution offer values are drawn from, $\mu_{\text{OO}} = 0$; and the standard deviation $\sigma_{\text{OO}} = 3$.

4.2.2 A Single Learning Agent in Different Communities

In these experiments, the community of agents is assumed to consist of nine agents playing a static cutoff policy (with each of them using the same cutoff) and one more agent which is either a learning agent or an agent using a

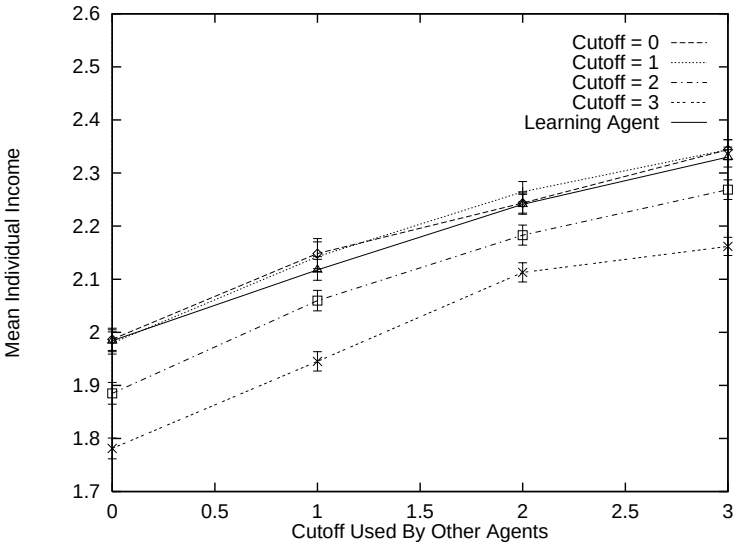

Figure 4.1: Mean individual incomes of single agents in communities of ten agents where the other agents use a fixed cutoff

cutoff. The performance of a single learning agent is compared with the performance of single agents using various different cutoffs given that the nine other members of the group continue to play their predetermined cutoffs. From the results on heterogeneous communities discussed in the previous chapter, it is reasonable to expect that it is always individually advantageous for a single agent to default a large proportion of the time in any given community. This expectation is verified by the experimental results presented in figure 4.1. Figure 4.1 also shows that the learning agent learns to default a large proportion of the time and achieves behavior competitive with agents using cutoffs of 0 and 1. Default percentages for the learning agent varied between 83% and 90% (see table 4.1). The learning agent performs slightly worse than agents using cutoffs of 0 and 1 because it continues to explore different cutoffs, some of which may be worse. However, the in-

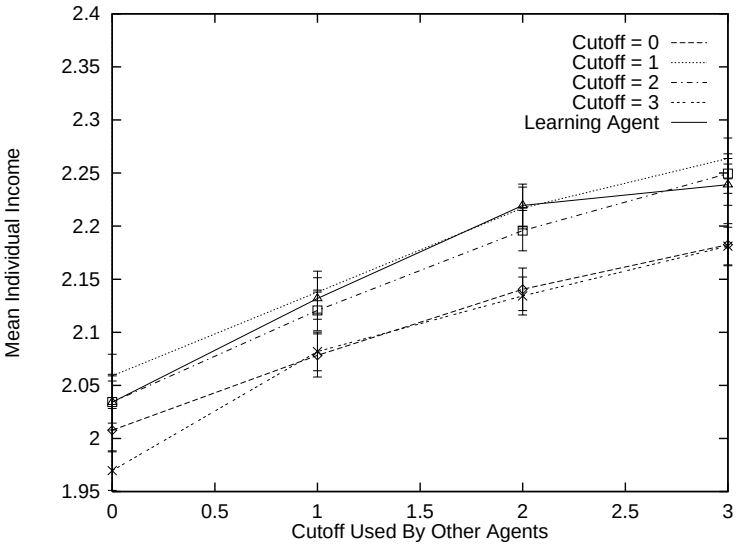

Figure 4.2: Mean individual incomes of single agents in communities of two agents where the other agent uses a fixed cutoff

crement becomes small enough after a sufficient number of weeks that the suboptimality is slight.

Table 4.2: Percentage of defaults by a learning agent in a community of two agents with the other agent using a fixed cutoff

Cutoff Used by Remaining Agents	Percentage Defaults
0	70.1 ± 1.4
1	55.6 ± 1.3
2	69.0 ± 1.2
3	75.5 ± 1.1

4.2.3 Learning in Two Agent Settings

The character of optimal behavior changes when we restrict our attention to groups consisting of two agents[1]. The effect of a single agent's behavior on group reputation is minimal in large communities of agents, because the group reputation update takes group size into account, penalizing a single default less when there are more members in the group and more opportunities for agents to default. Therefore, it is better for an agent to default less than it would in larger communities. It is unclear what kind of behavior is individually optimal in this setting. Figure 4.2 shows that always defaulting (having a cutoff of 0) is typically not a good policy in such an environment. Again, an agent using the cutoff-learning algorithm performs well in all cases, achieving high utilities. The default percentages for the learning agent are presented in table 4.2, and are lower than the default percentages for agents in larger groups.

4.2.4 Performance of a Learning Agent Over Time

Agents using the cutoff-learning algorithm perform well in the experiments reported above. The experiments reported in this section attempt to analyze the performance of agents using this algorithm as the number of weeks increases. This is important in the context of agent design. Unless the algorithm learns a good policy quickly and starts performing well reasonably early, it would be impractical to design agents using this algorithm in situations where group interactions are limited to a small number of time periods.

[1] It is quite likely that similar effects can be observed in larger communities by increasing the weight of **IR** and decreasing the weight of **GR**, which increases the individual disincentive for defaulting in any community and makes the behavior of the rest of the community less important than the behavior of the agent itself.

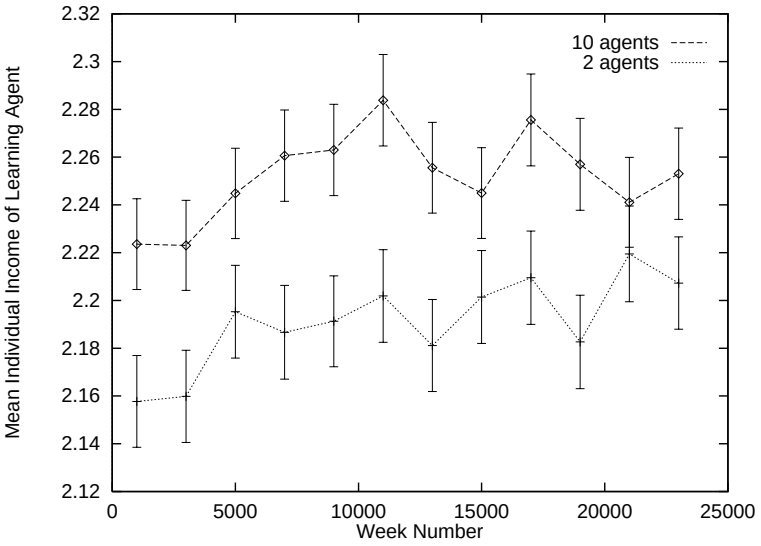

Figure 4.3: Mean individual incomes of a learning agent over different time slices in communities of 10 and 2 agents

In these experiments, a single learning agent was placed in communities where all other agents used cutoffs of 2. Two-agent and ten-agent communities were used. Performance was measured in slices of 2000 time periods, starting from 0-2000 and going up to 20000-22000. Results are reported in figure 4.3, with the 2000 week period designated on the X axis by its mean (for example, the mean individual income for the 0-2000 time-slice is reported on x=1000). Table 4.3 reports the percentage of defaults made by the learning agent over the time slices used in the experiments.

The results show that performance improves significantly for a certain amount of time and then tends to oscillate somewhat, while still staying relatively high. The cutoff-learning algorithm successfully learns different policies in ten and two agent situations, defaulting less in the two agent scenario, where defaulting is more expensive to the agent because of its

Table 4.3: Percentage of defaults by a learning agent in communities of two and ten agents over different time slices

Time Slice	% Defaults (10 agent community)	% Defaults (2 agent community)
0-2000	70.0 ± 1.2	62.2 ± 1.3
2000-4000	76.3 ± 1.3	66.5 ± 1.2
4000-6000	75.1 ± 1.2	56.4 ± 1.3
6000-8000	84.0 ± 0.9	72.0 ± 1.1
8000-10000	81.3 ± 1.1	57.6 ± 1.3
10000-12000	82.4 ± 1.0	63.2 ± 1.3
12000-14000	81.5 ± 1.0	71.9 ± 1.1
14000-16000	85.3 ± 0.9	64.5 ± 1.2
16000-18000	86.4 ± 0.9	68.1 ± 1.2
18000-20000	79.4 ± 1.0	65.2 ± 1.2
20000-22000	83.8 ± 1.0	68.9 ± 1.1
22000-24000	94.7 ± 0.6	56.7 ± 1.3

greater impact on group reputation. The variance in default percentage comes from two sources. First, there is a natural variance arising from the probabilistic nature of the values received from outside offers and from group-related activities. Second, the exploration of different cutoffs can lead to different results over slices of 2000 time periods because an agent may be exploring higher cutoffs in some of these slices and lower cutoffs in others.

The cutoff-learning algorithm learns a good cutoff to use without having to experiment with different cutoffs for a large number of weeks. This shows that the algorithm is useful both in terms of reaching a good cutoff value soon and in terms of asymptotic performance. However, asymptotic performance may be further improved if the algorithm is modified to store the best cutoff used so far in the simulation, and switch to using that cutoff when a certain number of weeks have passed and the agent can be fairly certain that it will not find a better cutoff.

4.2.5 Learning When Other Agents Also Learn

Algorithms that are successful in settings where a single agent is learning about its environment do not necessarily generalize to non-stationary environments, where multiple agents may be learning and changing their policies, which in turn could affect the environment they need to learn about. An agent cannot assume knowledge of what policies other agents are playing and of what policies they may play in the future.

When a single agent is learning how to act in a community of agents that use static decision-making policies, it can learn a very successful policy because the environment is stationary. The parameters that affect the utility an agent receives are fixed over time, because the other agents do not adapt to the learning agent's behavior. Further, as is shown in the previous chapter, behavior that is locally optimal for individual agents is often suboptimal for the entire group when all agents exhibit such behavior. For example, it is optimal for a given agent to default a large fraction of the time no matter what policies the other agents are following. However, if all the agents default a large fraction of the time, they do worse than if all of them default a smaller fraction of the time.

It turns out that when all the agents use the cutoff-learning algorithm, they tend to converge to a policy that is globally almost optimal. Optimality here is measured in terms of the results for communities of homogeneous agents using static decision-making functions, as discussed in chapter 3. In experiments performed with the same parameter settings as above and in communities of 10 agents all using cutoff-learning as their decision-making mechanism, the mean individual income for the agents was 2.175 ± 0.009, and agents defaulted on average $42.8 \pm 0.6\%$ of the time. This may seem

strange at first, because in any given large community of agents, it is always advantageous for a single agent to default a large percentage of the time if the behavior of the other agents is fixed. However, this result is explained by the fact that when all agents are using cutoff-learning, many of the cutoff changes are synchronous, and therefore agents are evaluating not only their own success when using a given cutoff, but also the average utility being received by all members of the group. Thus, an agent will never be able to find out that it could do better when using a lower cutoff when all other agents are using a higher cutoff.

A smarter agent may be able to exploit this tendency of the cutoff-learning algorithm and learn to take advantage of the social responsibility of other agents by playing the role of a "bad guy," defaulting a large percentage of the time and not suffering too much harm because the rest of the group behaves well. However, if all agents were smarter, they would start defaulting a large proportion of the time and the results would be globally sub-optimal. In most real-world multi-agent systems, it is impractical to assume that the interaction of each agent with the group starts simultaneously. If a single agent is added to an existing community of agents with no knowledge of the environment it is to be placed in, and all the agents that are already in the environment have already settled on cutoffs, the situation would be almost equivalent to introducing a single learning agent into a community of agents that are using a fixed cutoff. This agent would discover that defaulting a large proportion of the time was individually optimal. To prevent such exploitation of "good guys" who have mutually learned a globally optimal or near-optimal policy, it is necessary to use algorithms that are specifically geared for the multi-agent framework.

Such algorithms may need to learn mixed policies, in which states are

mapped to actions probabilistically. The major problem with multi-agent learning is that agents need to learn in a manner that is both rational and convergent. Bowling and Veloso (2000b) discuss an algorithm that fulfills these criteria by learning faster when an agent is performing poorly and slower when an agent is performing well. The study of multi-agent learning is still in its infancy and there are few theoretical performance guarantees. Applying a multi-agent learning technique to the SPIRE model is an important subject for future research.

4.3 Another Framework for the Learning Problem

Typically, agents that must both act and learn in an environment use Reinforcement Learning (RL) algorithms. RL algorithms are very successful in *stationary*, *fully-observable* MDPs. Stationarity is the property that environmental parameters do not change over time, and full observability is the property that an agent always has enough information to know its state completely. The generalization of MDPs to multi-agent (and usually non-stationary) scenarios is the Stochastic Game (SG) or multi-agent MDP. In an environment where multiple agents are learning, and their behavior in turn affects the environment, the assumption of stationarity does not hold and the usual RL algorithms are not guaranteed to converge.

The learning problem in SPIRE can be framed in the Markov Decision Process (MDP) framework. An MDP specifies a set of states S, a set of actions A available to the agent, a *transition model* M_{ij}^{a} that specifies the probability that an agent in state $i \in S$ will transition to state $j \in S$ given that it takes action a, and a *reward model* R_i that specifies the (perhaps probabilistic) reward for being in state i. A critical effect of this definition

is that the state an agent is in at time $t+1$ is solely dependent on the state it was in at time t and the action a it took at time t, and the reward is solely dependent on the state[2].

When a single agent is learning how to act in a community with other agents that are not learning, it should be able to use a standard reinforcement learning algorithm to learn a near-optimal policy. The purpose of the model-free Q-learning algorithm is to learn a *Q-function* through experience. The Q-learning algorithm learns the *Q-values* of state-action pairs, where the Q-value of a state-action pair is defined as the value of taking the action in that state. The Q-function is mathematically defined as follows. If an agent gets an expected reward of r when leaving a state s by taking an action a in an MDP governed by the reward model R_i and the transition model M_{ij}^b, then:

$$Q(s,a) = r + \sum_j M_{sj}^a (\max_b Q(j,b))$$

When an action a in state s takes an agent to state s', the update rule derived from this equation is:

$$Q(s,a) = (1-\alpha')Q(s,a) + \alpha'(r + \max_b Q(s',b))$$

where α' is a factor used to weight old and new estimates. Q-learning is very successful with small state-action spaces when it can use lookup tables. However, in the case of an infinite state-space like SPIRE, the lookup table needs to be replaced with a function approximator. Instead of doing lookups

[2]Some authors, for example Mitchell (1997) use a different formulation in which the reward function is also over state-action pairs. The two formulations are essentially equivalent.

on the table, values are given as input to the function approximator, which produces the output, and updates are made using a training rule. Function approximators serve an important purpose not only because they replace infinite lookup tables, but also because they allow generalization from states an agent has seen to states it has not seen. The performance of reinforcement learning techniques using function approximators varies widely (Smart & Kaelbling, 2000). Even in fairly simple cases, the replacement of a lookup table with a neural network in Q-learning has led to the failure of the learning algorithm (Moyan & Moore, 1995).

An alternative approach to Q-learning is to use Temporal Difference (TD) learning. TD-learning attempts to learn a *value function* for each state. The value of this function for a given state represents the value being in that state has for an agent. If the transition model is known, an agent can decide which action to choose in a given state by evaluating the expected rewards of each action and the value of each state it can transition to. Learning the value of a state is similar to learning a Q-function. The relevant update rule is:

$$V(s) = (1 - \alpha')V(s) + \alpha'(r + V(s'))$$

TD-learning often converges faster than Q-learning because propagation of punishments is not delayed by the presence of a maximization operator. TD-learning is a very useful approach when the transition model is known. In the SPIRE model, an agent knows the rewards for each of the actions it can take (the reward is the outside-offer value when the agent defaults and the group-task value when it does not). If an agent knows the transition model, it can decide whether to default or not by evaluating the values of

the states it would be in if it defaulted and if it didn't default. It should default if the sum of the value of the state reached by defaulting and the outside offer value exceeds the sum of the value of the state reached by not defaulting and the group-related task value.

Attempted implementations of Q-learning and TD-learning (with a known transition model) did not lead to good results. Various different state encodings were used, but the policies learned always led to agents defaulting frequently, even when such behavior was not optimal. It is unclear why these algorithms were unsuccessful. It is possible that the delayed reward for not defaulting is not being successfully incorporated. This could point to the need for a better exploration policy that allows all state-action pairs to be visited infinitely often in the limit. Alternatively, the problem could relate to function approximation, which is not always successful in reinforcement learning in continuous state-spaces. A third possibility is an implementation flaw.

Chapter 5

Conclusions and Future Work

This thesis introduces a new model of agent-group interaction that simplifies the original SPIRE model and makes it amenable to analysis. The new model also serves as a framework for studying the multi-agent policy learning problem in SPIRE. This model captures many of the essential elements of the intention-reconciliation process and the effects of environmental norms on agent behavior. Results with homogeneous and heterogeneous communities of agents using fixed policies reveal three important findings.

First, an intermediate level of "social consciousness" leads to optimal results in an environment with monetary disincentives for defaulting. This confirms the results of Glass and Grosz (2000) in the old SPIRE framework. Second, environmental disincentives for defaulting, such as a reduction in task-income due to a fall in reputation, can lead to results similar to those achieved using a combination of environmental disincentives and personal reluctance to default (like brownie points). This generalizes the work of

Teague and Sonenberg (2000) in an agent-agent interaction framework to the SPIRE framework of agent-group interaction. Third, results presented in this thesis demonstrate the "free-rider" effect in heterogeneous communities, where it is almost always optimal for any one agent to default very frequently, but if all the agents follow this policy, the results are globally suboptimal. Such behavior is a serious problem for designers of collaborative agent communities.

Analysis of the new model reveals that *cutoff functions* serve as appropriate policies in the SPIRE model. A cutoff function $\mathbf{CF}(v)$ simply specifies that an agent should default when the value of an outside offer exceeds the value of a group-related task by more than v. Optimal policies for an agent to use vary with the environment and the composition of the agent community an agent is placed in. For example, it is typically advantageous for an agent to default less in a smaller community, since each of its defaults affects the group reputation more. A novel algorithm called cutoff-learning, which performs a search through the space of cutoff values, allows single agents to learn good policies (cutoff values) rapidly over a wide range of environments and agent communities. When all the agents in a community use cutoff-learning together, they also learn near-optimal policies. However, it is likely that the algorithm will not lead to such desirable behavior when all agents do not start the learning process simultaneously (and hence change cutoffs simultaneously), because then the agents would not be evaluating a policy based on the utility arising from *all* agents playing that policy.

Replacements constitute an important issue for future work on SPIRE. The new model detailed here does not take into account the fact that agents defaulting on commitments could be replaced by other agents who are free at that particular time. This situation is easily modeled in terms of reputations

in the new framework, because the individual reputation of the defaulter should suffer, while the reputation of the group should not, given that the group managed to get the task done. It would be interesting to see if the effect of replacement agents is a major issue or if it is second-order and does not change the findings presented in this thesis significantly.

Another possible direction for investigation in the new model is the behavior of agents across different sets of environmental parameters. The overall framework and the evolution of reputations is designed to be robust over a wide range of environmental settings, but further empirical investigation would be useful.

There are many open questions in terms of learning in the SPIRE model. The question of when to stop collecting data in the cutoff-learning algorithm is important and there are models in statistics which may prove useful. Another question is when to stop exploring the space of cutoffs and use the best cutoff found so far. A third problem is to stop the algorithm from "taking a wrong turn" and exploring parts of the policy space that are worse than others.

A reinforcement-learning based approach should be reasonably successful in the SPIRE framework, at least in the case of a single-agent learning about its environment. The success of such an algorithm would also allow the multi-agent learning problem to be approached in terms of existing work on learning in multi-agent MDPs. If agents could learn mixed policies, which map states to actions probabilistically, this might succeed in preventing "good" agents from being exploited.

Another issue to be tackled in the reinforcement-learning framework is that of observability. How much information should agents have about the state of the world? One question to explore in this direction is that of

whether agents who do not have knowledge of **IR** or **GR** can successfully build models of these two factors and use them to improve prediction capabilities. It would be significant if such agents perform better than those who simply learn a cutoff.

The new SPIRE model is a robust, simple model for agent-group interaction. This model should scale well to various domains and can serve as a useful initial testing framework for designers of systems of autonomous, collaborative agents who must reconcile intentions. This thesis begins to investigate the difficult problem of learning in the SPIRE model, and the work presented here should provide a foundation for more work on learning in settings where agents must reconcile intentions.

Bibliography

Bowling, M., & Veloso, M. (2000a). *An analysis of stochastic game theory for multiagent reinforcement learning* (Technical Report CMU-CS-00-165). Carnegie-Mellon University.

Bowling, M., & Veloso, M. M. (2000b). Rational learning of mixed equilibria in stochastic games. Unpublished, under submission.

Bratman, M. (1987). *Intention, plans and practical reason.* Cambridge, MA: Harvard University Press.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Articial Intelligence (AAAI-98).* AAAI Press/MIT Press.

DeGroot, M. H. (1970). *Optimal statistical decisions.* New York: McGraw-Hill.

Glass, A. (1999). Creating socially conscious agents: Decision-making in the context of group commitments. Senior Honors Thesis, Harvard College, Cambridge, MA.

Glass, A., & Grosz, B. J. (2000). Socially conscious decision making. *Proceedings of the Agents2000 Conference.*

Grosz, B., & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence, 86.*

Grosz, B. J., Kraus, S., Sullivan, D., & Das, S. (2001). The influence of social norms and social consciousness on intention reconciliation. In submission.

Kitano, H. (2000). Robocup rescue: A grand challenge for multi-agent systems (invited talk). *Proceedings of ICMAS-2000.*

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., & Osawa, E. (1997). Robocup: The robot world cup initiative. *Proceedings of The First International Conference on Autonomous Agents (Agents-97).* The ACM Press.

Meuleau, N., Peshkin, L., Kim, K., & Kaelbling, L. (1999). Learning finite-state controllers for partially observable environments. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI-99.*

Mitchell, T. M. (1997). *Machine learning.* WCB/McGraw-Hill.

Moyan, J. A., & Moore, A. W. (1995). Generalization in reinforcement learning: Safely approximating the value function. *Advances in Neural Information Processing Systems: Proceedings of the 1994 Conference.*

Pomerleau, D. (1993). *Neural network perception for mobile robot guidance.* Kluwer Academic Publishers.

Russell, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach.* Prentice Hall.

Sandholm, T., & Lesser, V. (1996). Advantages of a leveled commitment contracting protocol. *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96).*

Sen, S., & Biswas, A. (1998). Effects of misconception on reciprocative agents. *Proceedings of the Second International Conference on Autonomous Agents (AGENTS-98).* ACM Press.

Shapley, L. (1953). Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America.*

Smart, W. D., & Kaelbling, L. P. (2000). Practical reinforcement learning in continuous spaces. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000).*

Stone, P., & Veloso, M. (2000). Multiagent systems: A survery from a machine learning perspective. *Autonomous Robotics*, *8.*

Sullivan, D., Glass, A., Grosz, B., & Kraus, S. (1999). Intention reconciliation in the context of teamwork: an initial empirical investigation. In M. Klusch, O. Shehory and G. Weiss (Eds.), *Cooperative information agents iii, lecture notes in articial intelligence*, vol. 1652, 138–151. Springer-Verlag, Berlin.

Sullivan, D., Grosz, B., & Kraus, S. (2000). Intention reconciliation by collaborative agents. *Proceedings of ICMAS-2000.*

Teague, V., & Sonenberg, L. (2000). Investigating commitment flexibility in multi-agent contracts. *Proceedings of the 2nd Workshop on Game-Theoretic and Decision-Theoretic Agents (GTDT-2000), associated with ICMAS-2000.*

Vrieze, O. (1987). Stochastic games with finite state and action spaces. No. 33. CWI Tracts.

Weld, D. (1999). Recent advances in AI planning. *AI Magazine*.