Coordination Issues in Cooperative Decentralized Decision Problems

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Yijia Zhao

December

2013

APPROVAL SHEET

The dissertation

is submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

AUTHOR

The dissertation has been read and approved by the examining committee:

Peter Beling

Advisor

Stephen Patek (advisor)

Alfredo Garcia

Randy Cogill

Zongli Lin

Accepted for the School of Engineering and Applied Science:

James H. Aylor

Dean, School of Engineering and Applied Science

December

2013

Coordination Issues in Cooperative Decentralized Decision Problems

A Dissertation

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy (Systems Engineering)

by

Yijia Zhao

December 2013

Approval Sheet

This dissertation is submitted in partial fulfillment of the requirements for the degree

of

Doctor of Philosophy (Systems Engineering)

Yijia Zhao

This dissertation has been read and approved by the Examining Committee:

Peter Beling, Dissertation Adviser

Stephen Patek, Dissertation Adviser

Alfredo Garcia, Committee Chair

Randy Cogill, Committee Member

Zongli Lin, Committee Member

Accepted for the School of Engineering and Applied Science:

Dean, Dean, School of Engineering and Applied Science

December 2013

Abstract

Decentralized control of complex engineering systems has the potential to provide the kind of security, reliability and scalability a centralized control scheme may be unable to offer in a highly dynamic setting. Much of the previous work in the area of decentralized control of a team of cooperative agents focuses on devising centrally computed policies that can be implemented in a distributed fashion to optimize team performance. In this thesis, we study decentralized decision problems each agent must formulate and solve in order to compute its own policy independently. A significant challenge in enabling the team of cooperative agents to work together efficiently is resolving coordination dilemmas associated with the presence of multiple optimal courses of actions. We aim to resolve these coordination dilemmas without assuming the presence of a centralized decision maker, the ability to negotiate or share intentions among the team members, or a consistent internal representation of the multiagent system.

Markov decision processes and its generalizations serve as the foundation in the study of single agent control. Decentralized control of cooperative agents is often framed as a multiagent extension of MDP, or as an identical interest stochastic game. Finding a solution for both involves solving stage games that are identical interest strategic games. Coordination dilemmas arise when multiple pareto-optimal Nash equilibria exist; solving these stage games thus reduces to an equilibrium selection problem. We propose a new solution concept as an equilibrium selection rule for a class of symmetric identical interest games where players are rewarded for commonality in their actions. The solution concept is endogenously salient and operates under the principle that no arbitrary decisions are allowed. We develop a linear time heuristic that 1) is theoretically guaranteed to compute the solution concept under certain conditions; 2) is shown to be successful with overwhelming likelihood in practice.

Next, we consider a decentralized path planning problem for team Bayesian search. A team of agents is tasked with making observations in a search area where an unknown number of targets exist. Each agent must formulate and solve a decentralized planning problem to compute its future actions. This planning problem is formulated as a partially observed Markov decision problem whose objective function is evaluated based on the assumption that all agents will use the same mixed strategy policy. We propose three dynamic programming heuristics for this planning problem-each can be used by agents in a decentralized fashion to compute an individual policy. The heuristics are designed such that all will arrive at the same policy as long as they use the same heuristics. The resulting policies are evaluated empirically in two instances of the team Bayesian search problem where resolving coordination dilemmas stemming from multiple optimal courses of actions is critical.

Acknowledgement

The dissertation process has been a long and winding journey, and there are many people I have become deeply indebted to along the way. First and foremost I would like to offer my sincerest gratitude toward my advisors Professors Peter Beling and Stephen Patek for their mentorship and guidance throughout the years. They have been instrumental in directing me from my initial interest on the topic to the final thesis. Both have shared valuable insights that helped me to move forward whenever I was stuck. This dissertation would not have been possible without their tireless encouragement - especially when I was overwhelmed with self-doubt.

I want to thank the other committee members, Professors Alfredo Garcia, Randy Cogill, and Zongli Lin, for their continued support throughout this process. Their valuable questions and insights have greatly benefited my work. I am truly grateful to Dean Pamela Norris whose support and encouragement was critical during the final stage.

I want to thank many people who have helped me during my graduate studies at the University: the Systems Engineering staff, Jill Bratton, Jennifer Mueller, Jayne Weber for their help navigating the administrative aspects of the entire doctoral process; my officemates and friends, Kaushik Sinha, Zhijiang Shen, Himanshu Gupta, Kangyuan Zhu, Emma Murray and Kanshukan Rajaratnam, with whom I have had so much fun discussing our research projects and other topics.

Finally, I would like to thank my family. I want to thank my parents for their unconditional love and never-ending support throughout my life, and my entire extended family who is always there for me when I need them. My husband has been a tireless cheerleader on this academic journey and is the best partner one could ask for in life. Last but certainly not least, I want to thank my son for teaching me what is truly important in life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With research advancements in robotics and software agents, multiagent systems have received much attention in the past two decades [1, 2]. Multiagent systems consist of multiple interacting intelligent agents that are able to perceive their environment through sensors and exercise control over their behaviors. Robotic soccer, search and rescue [3, 4], intruder capture [5], automated driving, and information systems are just some of the applications of a multiagent system.

In a cooperative multiagent system, a team of autonomous agents work to accomplish a common task. Decentralized control of such systems, where each agent is independently responsible for choosing its future action, is ideally suited to situations where communication is limited or altogether prohibited. It allows the system to be flexible in terms of its composition. We can envision scenarios where autonomous agents join together in an *ad hoc* fashion to compete for resources (*e.g.* network load balancing, routing) or accomplish a common task (robotic cop). It adds the scalability and modularity required in large and complex engineering systems (*e.g.* the Internet, smart electric grid). Furthermore, a decentralized control scheme enables the system to be more robust in the sense that performance degradation due to component failure tends to be much more graceful without centralized control serving as a single point of failure. This is highly desirable in civil applications where reliability is directly linked to economic outcomes. In military applications, this is absolutely necessary in order to have a system design that is secured against enemy attacks that may identify and destroy the centralized control and thus paralyze the entire system.

We study cooperative decentralized decision problems where a team of autonomous agents work toward a common goal. Each decision maker independently makes its own decision about its future actions, however, the common reward it receives and what state the system transits to depend on the actions of every independent decision maker in the system. On one hand, a full range of results developed in the area of automatic control and operations research can serve as a basis for research in decentralized control. On the other hand, adopting these results which are traditionally limited to situations where a single or centralized decision maker exists proves to be very challenging and therefore interesting. A central challenge in enabling a team of agents to work toward a common goal is the coordination of agents, and we will illustrate the type of coordination issues we are investigating with the following motivating examples.

1.1 Motivating Examples

1.1.1 The Dial-Wait Problem

Suppose two people, Al and Betty, are engaged in a telephone conversation and the line is cut in the middle of the conversation. Both Al and Betty wish to resume their conversation as quickly as possible, but who should call whom? Let us adopt a discrete-time model for the process by which Al and Betty reestablish connection, and let us assume that both parties require the same amount of time to either dial a number or to wait for a return call (per stage). There is clearly a coordination failure to address:

1. If both Al and Betty redial, then both will receive busy signals and the line will not be reconnected.

2. If both Al and Betty wait for the other to call, then again the line will not be reconnected.

3. Only if exactly one of them places the call will the connection be made.

Thus, unless Al and Betty have previously established in advance a protocol for who will dial and who will wait, they must play out a sequence of actions (dialing and/or waiting) for the connection to be remade. What complicates the problem is that Al and Betty do not have the opportunity to communicate with one another in order to establish the best way to proceed.

Even though the Dial-Wait problem is very simple, it sheds light on a number of interesting issues. First, observe that if either player arbitrarily decides to implement a deterministic sequence of actions, say $\{W, D, D, W, D, W, ...\}$, where "W" corresponds to "wait" and "D" corresponds to "dial", then the other player could have also selected the same sequence $\{W, D, D, W, D, W, ...\}$. This would result in the connection never being reestablished. While it seems unlikely that Al and Betty would make the same arbitrary choice, this is an example of the worst case outcome.

In contrast, randomized strategies make a lot of sense. For example, if Al implements a strategy of dialing with probability $p \in [0, 1]$ at each dial-wait opportunity, independently of his or Betty's actions at previous stages, and if Betty similarly dials with probability $q \in [0, 1]$ at each stage, the expected number of stages until the connection is reestablished works out to be $1/(p(1 - q) + (1 - p)q)$. Note that if Al and Betty choose $p = 0$ and $q = 1$, respectively, then they reconnect in one stage and therefore obtain the best possible outcome. What complicates matters is that another solution achieves the same result, namely $p = 1$ and $q = 0$. If Al and Betty can not agree on which one of these two solutions to pick, they wind up achieving the worst

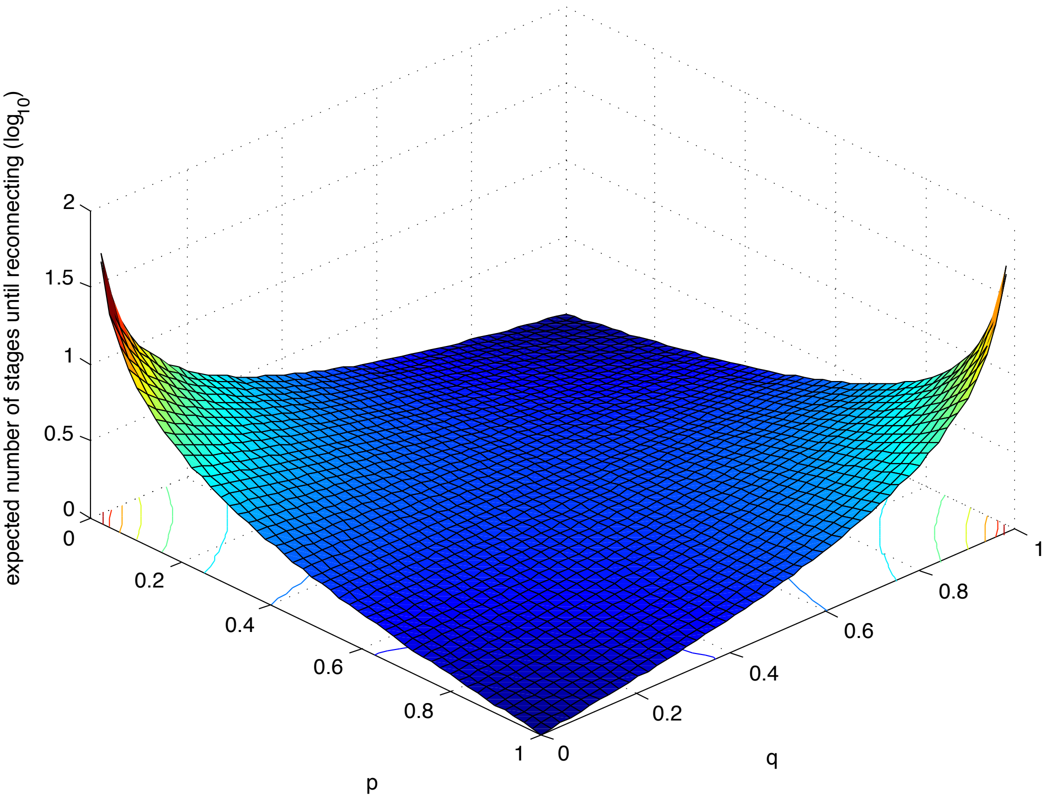


Figure 1.1: The Dial-Wait Problem: expected number of stages to reestablish the connection.

possible outcome: either $(p, q) = (0, 0)$ or $(p, q) = (1, 1)$, for which the connection is never reestablished.

	Dial	Wait
Dial	0,0	1,1
Wait	1,1	0,0

Figure 1.2: The Dial-Wait Problem as a strategic game.

If we analyze this example in game-theoretic terms, we can formulate the Dial-Wait problem as a 2-player 2-action strategic game as shown in Figure 1.2. We observe that both of the best-case solutions $(p, q) = (1, 0)$ and $(p, q) = (0, 1)$ are Nash equilibria in this strategic game, since neither player can deviate unilaterally to achieve a higher payoff. There is also a third Nash equilibrium solution with a lower payoff of 0.5 where each player dials with probability 1/2. Note that when one player chooses to dial with probability 1/2, he is guaranteed to receive the 0.5 payoff regardless of the action of the other player. Assuming that Al and Betty will only play actions that are consistent with Nash equilibrium solutions, then which equilibrium should they choose? Should they choose one of the Nash equilibria with the highest payoff 1? If so, which one? If Al and Betty disagree on the latter question then they experience the worst case outcome. Should Al and Betty settle on the mixed strategy equilibrium $(p, q) = (.5, .5)$? In what sense would this be rational?

1.1.2 A Decentralized Team Search Problem

We can illustrate coordination issues in decentralized cooperative decision problems with a more visual example by considering a search problem represented by the network in Figure 1.3. Each node in the network represents a search region. Let us assume that traversing an edge takes one time period, and the objective of the problem is to maximize the number of nodes visited before the end of the time horizon. Consider a two-agent scenario in which agent A is located at node 1 and agent B is at node 3, with one remaining time period. In such a situation, the objective will be maximized if one agent visits node 2 in the single remaining time period, while the other agent visits node 4. The difficulty is that there are two distinct ways to achieve optimality: (Alternative 1) agent A visiting node 2 and agent B visiting node 4 and (Alternative 2) agent A visiting node 4 and agent B visiting node 2. Note that if agent A chooses to pursue optimality through Alternative 1, then agent A will move to node 2. If simultaneously agent B elects to pursue optimality through the equally attractive Alternative 2 then both agents will end up at node 2, and optimality will be lost. We call this type of situation a *coordination dilemma*.

Coordination dilemmas would not be troublesome if there existed a centralized decision maker (perhaps one of the agents) who could implement an optimal policy by dictating the course of action that each agent was to follow. This scheme may be

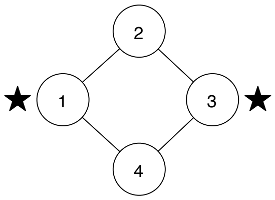


Figure 1.3: Example of potential coordination dilemmas

undesirable in applications where reliability is crucial as the centralized decision maker readily serves as a single point of failure. Furthermore, a centralized decision maker could potentially be hacked or hijacked and used to the advantage of adversaries.

Coordination can also be achieved through communication. One might imagine the agents announcing and negotiating their intentions to the point where each agent could predict with certainty the other's actions. If communication is limited or costly, it may be that neither is feasible. Allowing this line of communication also poses security threats as future locations of the agents can be easily predicted by adversaries who can eavesdrop on the agents.

Finally, we can enable coordination through the use of conventions or protocols that are established beforehand to handle all conceivable coordination dilemmas. We use social convention every day to avoid coordination failure in real life. Its use ranges from the simplest and most mundane tasks such as how we greet each other to avoiding potential congestions and accidents when driving. In engineering systems, one can design and implement, for example, a shared lexicographical ordering in which all joint actions are generated by all agents. In case of tie breaking, the first optimal joint action on the list can be adopted by everyone and therefore coordination can be achieved easily. Indeed, a lexicographic protocol can be constructed to handle all possible coordination confusions that may appear in this small problem. Generally speaking, establishment of an all powerful protocol or convention for multiagent systems is a complex process that scales very poorly with problem size. It is also difficult to include all possible scenarios beforehand.

More importantly, protocols or conventions must depend on the consistent ordering and labeling of agents, their actions and/or system states. For instance, a protocol or convention might state that, among other rules for the example in Figure 1.3, the agent currently at the lower numbered node will visit the lower number node next, *i.e.* the agent at node 1 will go to node 2 and the agent at node 3 will go to node 4. While both agents perceive the same situation, their internal representation of the situation is individually created and therefore the labeling of regions and indeed of the agents themselves may very well differ. If the labeling of the regions is not consistent among the two agents, it is possible that they disagree on who is at the lower numbered node. Consequently, both of them can arrive at the same node and result in a suboptimal outcome. This inconsistency leads to a coordination failure, even though both agents follow the same previously established protocol that was specifically designed to aid in the achievement of coordination. Not assuming the availability of consistent labeling in the design of decentralized planning algorithms maximizes flexibility in team composition, as new agents can join freely and current agents can depart with no ill effect.

1.2 Thesis Statement

In this thesis, we study the decentralized control of a team of cooperative autonomous agents with a common objective. Each agent independently makes the decision about which action to take in the future, however the common reward all agents receive and the future state of the overall system is determined by everyone's actions jointly. As we have illustrated with the two motivating examples above, a significant challenge in enabling the team of cooperative agents to work together efficiently is resolving coordination dilemmas associated with the presence of multiple optimal courses of actions. We aim to resolve these coordination dilemmas without assuming the presence of a centralized decision maker, the ability to negotiate or share intentions among the team members, or a consistent internal representation of the multiagent system.

Markov decision processes (MDP) and its generalizations serve as the foundation in the study of single agent control. Decentralized control of cooperative agents is often framed as a multiagent extension of MDP, or as an identical interest stochastic game. Finding a solution for both involves solving stage games that are identical interest strategic games. Coordination dilemmas arise when multiple pareto-optimal Nash equilibria exist; solving these stage games thus reduces to an equilibrium selection problem. We propose a new solution concept as an equilibrium selection rule for a class of symmetric identical interest games where players are rewarded for commonality in their actions. The solution concept is endogenously salient and operates under the principle that no arbitrary decisions are allowed. We develop a linear time heuristic that 1) is theoretically guaranteed to compute the solution concept under certain conditions; 2) is shown to be successful with overwhelming likelihood in practice.

Next, we consider a decentralized path planning problem for team Bayesian search, as an example of multi-stage cooperative decentralized decision problems. A team of agents is tasked with making observations in a search area where an unknown number of targets exist. Each agent must formulate and solve a decentralized planning problem to compute its future actions. This planning problem is formulated as a partially observed MDP whose objective function is evaluated based on the assumption that all agents will use the same mixed strategy policy. We propose three dynamic programming heuristics for this planning problem - each can be used by agents in a decentralized fashion to compute an individual policy. The heuristics are designed such that all will arrive at the same policy as long as they use the same heuristics. The resulting policies are evaluated empirically in two instances of the team Bayesian search problem where resolving coordination dilemmas stemming from multiple optimal courses of actions is critical.

1.3 Dissertation Organization

This thesis is organized into five chapters. In Chapter 2, we first provide an overview of MDP, multiagent extensions of MDP, and stochastic games. We review some previously proposed approaches to resolving coordination issues stemming from the existence of multiple optimal joint actions. In Chapter 3, we propose a new solution concept for a class of symmetric identical interest games that satisfy a certain set of assumptions. For this class of games, we formally define equivalence in actions and build the solution concept of a natural solution based upon it. We show that static agreement games (coordination games) belong to this class of games and demonstrate the proposed linear time heuristic algorithm on these games. In Chapter 4, we turn our attention to decentralized sequential decision making and study a decentralized planning problem for Bayesian team search. We provide mathematical formulations of the decentralized decision problem and the centralized version of the planning problem before proposing three heuristics for decentralized policy computation. Empirical evaluations of policies produced by these heuristics along with the centralized optimal policy are provided to demonstrate effectiveness and show the differences in heuristics. Finally we summarize and touch on future research directions in Chapter 5.

Chapter 2

Background

2.1 Single Agent Decision Problems

Traditionally, Markov Decision Process (MDP) serves as the foundation for research in the control of a single autonomous agent.

Definition 1 (Markov Decision Process). *A Markov Decision Process (MDP) is defined as a 4-tuple* (S, A, R, T)*, where* S *is the set of system states,* A *is the set of actions,* $R : S \times A \to \mathcal{R}$ *is the reward or payoff function, and* $T : S \times A \times S \to [0, 1]$ *is the system transition function. For all* $s, s' \in S$*,* $a \in A$*,* $R(s, a)$ *is the reward of taking action* a *in state* s *and* $T(s, a, s')$ *is the probability of reaching state* s' *when taking action* a *in state* s*.*

Generally, we also add a time horizon element to MDP. For example s^t denotes the system state at time t and r^t denotes the reward at time t. A deterministic policy or strategy of an agent is a mapping from set of states to set of actions. A randomized or mixed policy is a mapping from set of states to set of probability distributions over actions. In MDP, there always exists an optimal stationary policy that is deterministic. Therefore we generally limit our attention to deterministic policies and define a policy for MDP as $\pi : S \to A$. The objective is to find a feasible π such that a reward function will be optimized if policy π is followed. For infinite horizon MDPs, a common reward function uses the total discounted expected reward. $v_\pi(s)$, the total discounted expected reward if policy π is followed starting in initial state s, is defined as follows:

$$v_\pi(s) = \sum_{t=0}^{\infty} \beta^t E(r^t \mid \pi, s^0 = s), \tag{2.1}$$

where $\beta \in [0,1]$ is the time discount factor. Another common choice is the average expected reward. A feasible policy π^* is optimal if $v_{\pi^*}(s) \geq v_{\pi'}(s)$ for all $s \in S$ and all feasible policies π'.

It is well-known that any policy π^* is optimal if and only if for all states $s \in S$ it satisfies the following system of Bellman's equations based on principles of optimality:

$$v_{\pi^*}(s) = \max_a [R(s,a) + \beta \sum_{s' \in S} T(s,a,s') v_{\pi^*}(s')]. \tag{2.2}$$

Therefore computing an optimal policy reduces to solving the system of equations in (2.2). Two standard techniques are value iteration and policy iteration. In value iteration, the algorithm starts with randomly assigned values v^0 for all $s \in S$. We iteratively update v^{t+1} as

$$v^{t+1}(s) = \max_a [R(s,a) + \beta \sum_{s' \in S} T(s,a,s') v^t(s')]. \tag{2.3}$$

The sequence of functions v^t converges to the optimal value v_{π^*} in the limit and the actions that maximize the right hand side of (2.3) form the optimal policy π^*. Note that when multiple actions achieve the maximum, any one of them can be chosen to form the optimal policy. This is however no longer the case in decentralized multi-agent control settings where coordination among the decision makers is essential to avoid incoperating incompatible optimal actions into the optimal policy.

In policy iteration, an initial policy π' is chosen arbitrarily. During each iteration, $\pi = \pi'$ and the following computations are carried out for each state s:

$$v_\pi(s) = R(s, \pi(s)) + \beta \sum_{s' \in S} T(s, \pi(s), s') v_\pi(s'), \tag{2.4}$$

$$\pi'(s) = arg \max_a [R(s, a) + \beta \sum_{s' \in S} T(s, a, s') v_\pi(s')]. \tag{2.5}$$

Each iteration produces a policy with improving values until no further improvements are possible. At that point, the resulting policy is guaranteed to be optimal. Again when multiple actions satisfy the right hand side of Equation (2.5), any of these actions can be successfully incorporated into the resulting policy.

In many potential applications for MDP, the decision maker does not have access to the exact value of the current state. For example, sensors used for measurement may be inaccurate. When the state of the system is not known exactly but rather only a noisy observation of the true value of the state is available, the control problem is typically formulated as a partially observable Markov decision process (POMDP), a generalization of MDP [6].

Definition 2 (Partially Observable Markov Decision Process)**.** *A partially observable Markov decision process (MDP) is defined as a 6-tuple* (S, A, R, T, Ω, O)*, where* S *is the set of system states,* A *is the set of actions,* $R : S \times A \rightarrow \mathcal{R}$ *is the reward or payoff function, and* $T : S \times A \times S \rightarrow [0, 1]$ *is the system transition function,* Ω *is the finite set of observations, and finally* $O : A \times S \times \Omega \rightarrow [0, 1]$ *is the observation function.*

An information vector for a POMDP includes a complete history of observations and actions until the current decision period. Instead of searching for an optimal policy that maps the system state to an optimal action, solving an POMDP involves recasting it as a fully observable MDP with a state space that consists of all possible information vectors. Sufficient statistics which ideally reflect all relevant information about the process at a specific time, but are of a more manageable dimension than the information vector are often used in place of the information vector. The probability distribution of states conditional on the information vector often serves as a sufficient statistic. Value iteration and policy iteration can be used to solve POMDP exactly, although often times the complexity of computing an exact solution is prohibitive, even for a small problem.

2.2 Multiagent Extensions of MDP and POMDP

A decentralized decision process can be formulated either as a multiagent version of a MDP or as a stochastic game. As we shall see from their definitions they are in fact closely related to each other. There is no standard approach to extending fully observable MDP to the multiagent setting (see [7, 8, 9] for some previously proposed approaches). In a typical decentralized multiagent extension of a fully observable MDP, a group of agents collectively control the decision process. Each agent makes its own individual decision on which individual action to take and the system transits based on the joint action of all individual actions. A common reward is received by all agents based on this joint action as well. We follow the definition given in [10] and define a Multi-agent Markov Decision Process (MMDP) as follows:

Definition 3 (MMDP)**.** *A Multi-agent Markov Decision Process (MMDP) is a 5-tuple* $(S, N, \{A_i\}_{i \in N}, R, T)$*, where* S *is the set of system states,* $N = \{1, \ldots, n\}$ *is the set of agents,* A_i *is the set of actions for agent* i*,* $R : S \times A_1 \times \cdots \times A_n \to \mathcal{R}$ *is the reward or payoff function, and* $T : S \times A_1 \times \cdots \times A_n \times S \to [0, 1]$ *is the system transition function.*

We let $A = \times_{i \in N} A_i$ be the set of joint actions and use a without a subscript to denote a joint action $a = (a_1, \ldots, a_n)$ in A. An individual policy $\pi_i : S \to \Delta(A_i)$ is a mapping from the set of states to the set of probability distributions over actions. π_i is deterministic if $\Delta(A_i)$ consists of only probability distributions that place probability 1 on a single action in A_i. Otherwise we say that π_i is mixed or randomized. Here we no longer limit our attention to only deterministic policies. Let $\pi = (\pi_1, \ldots, \pi_n)$ be a joint policy, as in the case of MDP, we can define the reward function associated with joint policy π as follows:

$$v_\pi(s) = \sum_{t=0}^{\infty} \beta^t E(r^t | \pi, s^0 = s), \tag{2.6}$$

where $\beta \in [0, 1]$ is the time discount factor. A feasible joint policy π^* is optimal if $v_{\pi^*}(s) \geq v_{\pi'}(s)$ for all $s \in S$ and all feasible joint policies π'. The goal of a control algorithm in MMDP is to find individual policies that together would form an optimal joint policy.

Given that MDP has well-understood algorithms that produce optimal policies, it is tempting to think that extending these techniques to decentralized decision processes would be straightforward. While in limited circumstances these techniques can lead to an optimal joint policy, the decentralized nature of decision making here makes applying value iteration or policy iteration successfully quite challenging. The main difficulty lies with cases where multiple incompatible optimal joint policies exist. Take for example the value iteration update in (2.3) and modify a to mean a joint action in the setting of MMDP. While all agents can compute the value function $v^{t+1}(s)$, when multiple joint actions maximize the right hand side it is not clear how decentralized decision makers can agree on which maximizing joint action to choose. It is not difficult to see that each agent can choose a different optimal joint action and incorporate the individual action prescribed by said joint action into its individual policy, and the resulting joint policy may not produce the actual function value every agent has in mind. These types of coordination failures can occur during each update in value or policy iteration.

Many different approaches to extend POMDP to include a team of agents have been proposed [11, 12, 13]. Typical formulations have agents make their own observations of the system, and future rewards and observations depend on the joint actions of agents. Observations may or may not be shared, depending on the specific communication setup of the formulation. Much of the research in decentralized POMDP assumes an optimal policy is computed by a centralized decision maker and that policy is distributed among the agents for execution [14, 15, 16, 17]. The primary concern in terms of agent coordination is how each agent can act optimally in the collective sense without knowing exactly what the other agents have just observed. In other words, during the execution of the team task, each agent is fully aware of the policy or decision rule the other agents are operating under since each is equipped with the common policy predetermined by a centralized decision maker. However, each agent does not have an accurate picture of what the other agents are basing their decision on and this can lead to coordination failure.

While this line of research is very challenging, and there is a wide array of applications that can benefit from various proposed results, we want to emphasize here that what we are primarily interested in is the study of decentralized decision making. In other words, we are interested in the development of algorithms that can be used by individual agents to independently compute a policy. We can certainly include the assumption that observations are never shared or shared with a time delay in our problem setting, however, the same kind of coordination issues due to multiple optimal joint actions will still be present. In order to isolate the coordination issues we are interested in from the coordination issues stemming from partial information, we assume that individual observations are shared among the team of agents in the multiagent POMDP we investigate in Chapter 4.

2.3 Connection to Stochastic Games

Game theory deals with interactions among multiple decision makers. In fact, MMDP and multiagent extensions of POMDP are closely related to stochastic games [18] and partially observable stochastic games. In a stochastic game, players play a sequence of one-shot strategic games. After concurrently and independently playing each game, players receive payoffs specified by the payoff structure of the strategic game and proceed to the next game whose identity depends on the current game and the players' actions. Stochastic games are played in the game theoretically noncooperative setting, *i.e.* agents cannot form an enforceable agreement. More formally, we present the standard definition of stochastic games as follows [19]:

Definition 4 (Stochastic Games)**.** *A stochastic game is a tuple* $(S, N, \{A_i\}_{i\in N}, \{R_i\}_{i\in N}, P)$, *where* S *is the set of system states,* N *is the set of players,* A_i *is the set of actions for player* i, $R_i : S \times A_1 \times \cdots \times A_n \to \mathcal{R}$ *is the reward or payoff function for player* i, *and* $P : S \times A_1 \times \cdots \times A_n \times S \to [0, 1]$ *is the transition function that specifies the system dynamics.*

Each state is associated with a N-player stage game in strategic form where the set of actions for each player is A_i and the payoffs are specified by reward/payoff function R_i. MMDP is essentially a stochastic game where $R_i = R_j$ for all $i, j \in N$, *i.e.* MMDP is an identical interest stochastic game. Stochastic games provide the most general setting for considering decentralized decision processes because they can be used to model situations where decision makers have competing or complementary objectives. An individual policy or strategy $\pi_i : S \times A_i \to [0, 1]$ for agent i is a mapping from states to probability distributions over its individual actions. We let $\pi = (\pi_1, \ldots, \pi_n)$ be a joint policy of all agents. Π_i denotes the set of all feasible individual policies for agent i and Π be the set of all feasible joint policies. We define a reward function for each agent i as follows:

$$v_i(s, \pi_1, \ldots, \pi_n) = \sum_{t=0}^{\infty} \beta^t E(r_i^t | \pi_1, \ldots, \pi_n, s_0 = s). \tag{2.7}$$

In a strategic game, a Nash equilibrium is a joint strategy such that no single agent can attain a higher individual payoff by unilaterally deviating from that strategy [20]. Using a common game theory notation, we let π_{-i} be the joint strategy of all the agents except i. Nash equilibrium for a stochastic game can be similarly defined as follows:

Definition 5. *In a stochastic game, a Nash equilibrium is a joint strategy* $(\pi_1^*, \ldots, \pi_n^*)$ *such that for all* $s \in S$, $i \in N$, $\pi_i^{'} \in \Pi_i$,

$$v_i(s, \pi_i^*, \pi_{-i}^*) \geq v_i(s, \pi_i^{'}, \pi_{-i}^*) \tag{2.8}$$

A stochastic game can have multiple Nash equilibria. While Nash equilibrium policy should be the goal, with the exception of some special subclasses of stochastic games it is generally not clear which Nash equilibrium should be included in the Nash equilibrium policy. In identical interest or fully cooperative stochastic games, *i.e.* MMDPs, finding a Pareto-optimal Nash equilibrium maximizing payoffs for all players and therefore the objective for optimization is well-defined. When multiple Pareto-optimal Nash equilibria exist in a fully cooperative stochastic game, coordination failures can occur when players carry out actions prescribed by incompatible policies. Incompatibility of policies are defined as follows:

Definition 6. *We say that in a fully cooperative stochastic game, two policies* π *and* $\pi^{'}$ *are incompatible if there exists a state* $s \in S$ *such that* $v(s, \pi) = v(s, \pi^{'})$ *and a set of agents* $B \subset N$ *such that* $v(s, \{\pi_i\}_{i \in B}, \{\pi_j^{'}\}_{j \in N-B}) < v(s, \pi)$.

2.4 Previous Work

In this section, we discuss some of the common approaches to resolving coordination issues due to multiple optimal joint policies in existing literature. In the last decade or so, there has been a great deal of research effort devoted to formulating and solving multiagent versions of MDP and POMDP. However, as previously mentioned, much of this research assumes that the policy computation is carried out by a centralized planner. Agents receive this pre-established policy and execute it in a distributed fashion during run-time. Agents generally are not assumed to have the most up to date local information other agents are basing their actions on (according to the commonly known policy) and therefore must compensate for this uncertainty over information with intelligent guesses (belief state modeling is commonly used toward that end).

Even when policy computation is carried out in a decentralized fashion, the type of coordination issue we investigate is not commonly recognized in the control literature. It is perhaps not surprising that from a control point of view it is instinctive to think homogeneous agents can reach the same decision independently. The following is representative of the assumptions typically made about agents' ability to coordinate:

> "We assume that each agent behaves rationally and has the same mind power, *i.e.*, they will independently (without any communication) reach the exactly same conclusion given a common problem as such as solving a Markov decision process (MDP). This implies that all know the current global state (perfect coordination). This is because in such case all agents are now presented with the same decision problem (given global state, global reward function, and a common start condition), thus they will independently solve the decision problem, reaching the exactly same decision - which is an optimal decision, and each agent then implements the local part of this decision. Note that all this is done in an independent fashion." [7]

In one of the first works framing MDP in the decentralized setting, Boutilier proposed the MMDP framework and pointed out the substantial coordination challenges if one were to extend single MDP solution techniques to this new problem [10]. Treating MMDP as a stochastic game, there may be multiple optimal joint actions in a stage game. Without resorting to the use of a centralized coordinator or negotiation, Boutilier proposed forming a smaller identical interest strategic form game whose action set involves precisely those individual actions that are part of an optimal joint action. Coordinating on one of the optimal joint actions therefore is reduced to an equilibrium selection problem. He suggested that either conventions that rely on lexicographical ordering of agents and actions or learning algorithms can be used as a selection tool. A simple reinforcement learning algorithm was proposed and tested.

In [21] and [22] exact and approximating dynamic programming algorithms using generalized belief state are developed for partially observable stochastic games (POSG). Each agent maintains a belief over the underlying state as well as policies of other agents. The algorithms involve agents forming and solving strategic games in parallel during each stage of the dynamic programming process. To solve these strategic form games, iterated elimination of dominated strategies is proposed. Coordination issues arise when the iterated elimination of dominated strategies does not produce a game with an unique Nash equilibrium. The authors propose standard equilibrium selection rules as a way to resolve coordination dilemmas but fail to recognize the limitations of these standard selection rules.

In [23], decentralized decision making in a team of robots is modeled as a POSG. Similar complexity issues exist for solving POMDP exist in POSG and therefore solving for exact solutions is generally intractable. They propose an algorithm in which each agent approximates the POSG with smaller but related Bayesian games. Each agent uses the alternating-maximization algorithm (holding still actions of all agents' but one and finding the best-response action) to find the Bayesian Nash equilibrium. Since the equilibrium produced by the alternating-maximization algorithm is only guaranteed to be a local optimum, random restarts are used to decrease the likelihood of being stuck at a pareto-dominated equilibrium. Since all of this is done independently by each agent in parallel, it is essential that agents can formulate the same Bayesian games and coordinate on the same equilibrium. Toward this end, a synchronized random number generator is proposed to ensure coordinated restarts during the execution of the alternating-maximization algorithm. However, it is not clear how tie-breaking is handled when multiple best-response actions exist. A similar approach using synchronized random number generation is proposed in the online planning algorithm for decentralized POMDP when communication is limited [24]. Here, predetermined tie-breaking rules (*i.e.* convention) are used when multiple best-response actions exist during the alternating-maximization process.

Recognizing the limitations of various equilibrium selection rules, Gmytrasiewicz and Doshi propose a new framework for decentralized POMDP called interactive POMDP (I-POMDP) as a decision theoretical alternative to game theoretical approaches involving Nash equilibrium selection [12]. Belief state is used to model not only the physical environment but also the other agents preferences, capabilities, and beliefs. Each agent's belief is a probability distribution over states of the environment and the models of other agents which include their observations and decision rules that map observations to actions. Belief update involves updating possibly infinitely nested beliefs and therefore approximation methods are generally employed. One such approximation method is to consider only finite nestings in the belief update. It is shown that value iteration converges in finitely nested I-POMDPs. When multiple optimal actions exist for another agent, it is assumed that each optimal action will be played with equal probability.

In the remainder of this section, we discuss how coordination issues are handled in multiagent reinforcement learning. Reinforcement learning enables an autonomous agent to obtain optimal behavior through repeated interaction with the environment. Rather than requiring that reward structure and system dynamics be known before optimal control can be planned, optimal policy is learned through repeatedly taking an action and observing its consequences. Algorithms such as Q-learning are effective at learning an optimal policy in MDP [25, 26]. Much effort has been made to extend reinforcement learning to the multiagent setting [27], however applying existing learning techniques faces the same type of coordination challenges we described in the motivating examples.

In the basic Q-learning algorithm, a function Q is defined for each state and action pair:

$$Q^*(s,a) = R(s,a) + \beta \sum_{s' \in S} T(s,a,s') v_{\pi^*}(s'), \tag{2.9}$$

where $Q^*(s,a)$ is the total discounted reward of taking action a in state s and thereafter following the optimal policy π^*. Notice that by Bellman's equation (2.2), we have $v^*(s) = \max_a Q^*(s,a)$, and therefore the optimal policy associated with $v^*(s)$ can be identified by finding the actions that maximize $Q^*(s,a)$. The problem is then reduced to computing the function $Q^*(s,a)$ instead of searching for the optimal value of $v^*(s)$ directly. To achieve that goal, Q-function updates proceed as follows:

$$Q^{t+1}(s,a) = (1-\alpha^t)Q^t(s,a) + \alpha^t(r^t + \beta(\max_{a'}[Q^t(s',a')])) \tag{2.10}$$

where r^t and s' are the reward and state the system transits to after taking action a' in state s. $\alpha^t \in [0,1)$ denotes the learning rate. Watkins and Dayan [25] showed that the sequence $Q^t(s,a)$ in (2.10) converges to $Q^*(s,a)$ under the following assumptions: 1) the learning rate α^t should take decreasing values such that $\sum_{t=1}^{\infty} \alpha^t = \infty$ and $\sum_{t=1}^{\infty}$; 2) each state and action pair (s,a) is visited an infinite number of times. Singh [28] showed that Q-learning converges to optimal Q-function value and optimal policy when GLIE (greedy in the limit with infinite exploration) exploration is used. An exploration/exploitation strategy is GLIE if 1) each action is executed infinitely often in every state that is visited infinitely often; 2) in the limit, the learning policy is greedy with respect to the Q-value function with probability 1. Examples of a GLIE exploitation strategies include ϵ-greedy exploration and Boltzmann exploration.

After convergence is reached, the optimal policy π^* can be found by letting $\pi^*(s) = arg\max_a Q^*(s, a)$. When Q-learning is used to learn an optimal policy in MMDP, coordination dilemmas arise when multiple actions achieve the optimal Q-function values. In applications to stochastic games, it is generally agreed upon that the optimal policy the learning algorithms converges to should be a Nash equilibrium policy. However, incompatible Nash equilibrium policies are often present in both the entire stochastic game and stage games. Indeed, the convergence of Q-function value is relatively easily accomplished especially in identical interest stochastic games while the convergence in policy is difficult to achieve because of coordination failure.

In Nash Q-learning proposed by Hu and Wellman [29], the optimal Q value of taking an action is assumed to be the discounted sum of current reward and future rewards given that a Nash equilibrium strategy for each stage game is followed by all agents thereafter. In order to compute a Nash equilibrium for the stage games, each agent must be able to first form the stage game whose payoffs are specified by every agent's Q-function values. This is accomplished by having each agent maintain a model of every other agent's Q-functions. Convergence conditions include the two basic assumptions about infinite sampling and decaying of the learning rate. These conditions are necessary but far from sufficient. Note that every agent has a model of all agents' Q-functions and must update these Q-functions during each iteration. Every agent's model of Q-functions shares the same value by default at $t = 0$. Assuming that at time t, they are still identical and therefore all agents recognize the same stage game $(Q^t_1(s'), \ldots, Q^t_n(s'))$. If there exists a single Nash equilibrium for the stage game $(Q^t_1(s'), \ldots, Q^t_n(s'))$, then all agents can compute the same equilibrium and therefore update Q-functions in a way that in iteration $t+1$ everyone's model Q-functions remain the same. However, when there are multiple Nash equilibria for the stage game, if agents are not able to coordinate on the same equilibrium value in Q-function update then their Q-function models will diverge from that point on. This leads to the break down of Nash Q-learning as convergence in Q-function value can not be guaranteed. The following additional convergence conditions are therefore proposed to ensure the convergence of Q-function values:

- Every stage game $(Q_t^1(s), \ldots, Q_t^n(s))$ for all t and s has a global optimal point and agents' payoffs in this equilibrium are used to update their Q-functions.

- Every stage game $(Q_t^1(s), \ldots, Q_t^n(s))$ for all t and s has a saddle point and agents' payoffs in this equilibrium are used to update their Q-functions.

A joint action of a strategic game is a saddle point if it is a Nash equilibrium and each agent will receive a higher payoff when at least one of the other agents deviates from this joint action. All saddle points of a game are equivalent in their values. Note in particular that the same condition has to hold for all stage games for Nash Q-learning to converge, i.e. the choice of global optimal point and saddle point has to be consistent throughout the learning process. These conditions ensure that the values of the Nash equilibrium selected by all the agents are the same and leads to the convergence of Q-function value. Even when this restrictive condition is met, there is still no guarantee that convergence in policy will result unless agents can coordinate on the same equilibrium. Hu and Wellman propose that Nash equilibrium be chosen based on its expected reward or the fixed order in which it is generated by a common algorithm. Coordination is therefore resolved by using a common list and pre-established convention.

It is generally very difficult to find stochastic games that satisfy the strict convergence conditions proposed for Nash Q-learning. In order to relax the convergence conditions, Littman [30] proposed the use of adversarial equilibrium and coordination equilibrium in Friend-or-Foe Q-learning. An adversarial equilibrium is a Nash equilibrium in which no player is hurt by changes by other players (either jointly or individually). A coordination equilibrium is a Nash equilibrium in which all players attain their highest possible reward. An adversarial equilibrium always exists in two player zero-sum games while a coordination equilibrium always exists in identical interest games. Whenever a stage game has a coordination (adversarial) equilibrium, all of them will have the same value. These facts can be used to provide a less restrictive convergence condition. Two versions of Friend-or-Foe Q-learning exist. Of particular interest to us is the use of the coordination equilibrium and this is called Friend Q-learning. The Q-function is updated with the current reward plus the discounted future reward assuming all agents follow the coordination equilibrium strategy from there on. Again, several coordination equilibria of the same value can exist for a stage game. Even though Q-functions may converge to the optimum value the learned policies may not be optimal due to agents using incompatible equilibria. Littman does not propose a fixed order generation mechanism to resolve coordination dilemmas but simply observes that for some games even though convergence occurs the resulting policy is suboptimal. Littman shows that Friend-Q learns the value for a Nash equilibrium policy if the game has a coordination equilibrium for the entire game, however this does not guarantee a Nash equilibrium policy will result from the Q-function.

While Nash Q-learning and Friend-or-Foe Q-learning employ a Nash equilibrium solution concept in the computation of Q-function, Correlated-Q Learning [31] uses correlated equilibrium in much the same way. Nash equilibria in general can be difficult and costly to compute. Correlated equilibrium has the advantage that the set of correlated equilibria is a convex polytope and therefore they can be more easily computed using linear programming. Just like methods using Nash equilibria, the equilibrium selection problem is still central to coordination as there can exist more than one correlated equilibrium in any stage game. When multiple equilibria with the same payoff exist, a centralized mechanism is proposed to select the same correlated equilibrium.

In Optimal Adaptive Learning (OAL) [32], equilibrium selection is achieved using a model-based technique called Biased Adaptive Play (BAP). After each iteration of Q-function update, a virtual game (VG) for state s is constructed such that payoff of a joint action a is 1 if $Q(s, a)$ is within ϵ_t of the optimal Q value of taking any action in state s. Payoff is set to 0 for all other joint actions. If this VG is weakly acyclic, *i.e.* there exists a directed path leading from any vertex to a sink in its best response graph, Adaptive Play (AP) proposed by Young [33] will converge to a strict deterministic Nash equilibrium with probability 1. However, there is no guarantee that a VG is always weakly acyclic, and strict deterministic Nash equilibrium do not always exist for VG. The authors propose modifying the VG and AP as follows. Each VG is constructed as above with the addition of a biased set D which includes all of the joint actions with payoff 1. Let SP_t be the set of k samples from the most recent m joint actions at time t. While both AP and BAP use sampled recent plays to compute the presumed stationary mixed strategy of other agents, they differ when the best response (BR) set to this mixed strategy contains more than one element. In AP each agent will randomize uniformly among the elements in the BR set. BAP proceeds exactly like AP except for when the following BAP conditions are met:

1. There exists a joint action $a' \in D$ such that for all $a \in SP_t, a_{-i} = a'_{-i}$,

2. $D \cap SP_t \neq \emptyset$.

When BAP conditions are met, there exists a joint action in both SP_t and D such that it is a best response to every element in SP. The most recently sampled such action in SP_t is chosen and individual action is carried out accordingly. This makes the selection of Nash equilibrium deterministic and ensures the coordination on the same equilibrium among all agents. The result is that with probability 1 BAP converges to either a Nash equilibrium in D or a strict Nash equilibrium in VG. Since VG is constructed such that D contains all the ϵ-optimal joint actions, coordination on ϵ-optimal joint actions is achieved. It is shown theoretically that for any identical interest game OAL converges to an optimal Nash equilibrium with probability 1 under conditions that are easily met. While OAL has a strong theoretical guarantee, in practice BAP conditions are not easily met therefore coordination on the same equilibria may be slow. Consequently, convergence in policy may also be slow.

Finally, Rmax is a model-based deterministic learning algorithm originally proposed for MDP and later extended to identical interest stochastic games in [34] and fixed-sum stochastic games in [35]. It is model-based in the sense that each agent maintains and updates its own model of the reward structure and transition structure of the game and optimizes with regard to this model during each iteration. It is deterministic in the sense that the order in which joint actions are explored is fixed according to a common list all agents are given. Rmax is shown to converge to near optimal value in identical interest stochastic games in polynomial time in terms of the problem parameters. When choosing between multiple optimal joint actions, tie breaking is based on the order joint actions appear on the common list. The authors recognize that this is only feasible with agents sharing common labelings of agents and their actions. When this common labeling is not available, it is proposed that agents can learn the labelings during an preliminary order exploration phase. For example, if common labeling of the agents is available, labeling of each agent's actions can be learned as follows: each agent will play its actions one after another until it returns to its first action. A lexicographic ordering over the joint actions can then be produced. A more difficult scenario is when agents do not have a commonly known labeling of the agents. In this case, during the order exploration phase each agent will randomly select an ordering over the agents and carry out Rmax based on this randomized ordering. Rmax is carried out with a sufficiently long time so that with high probability a near optimal reward is produced, provided an identical ordering was chosen by all agents. This is repeated for a sufficiently large number of trials so that with high probability agents will choose the same ordering during one of the trials. Once all trials are completed, each agent selects the ordering with the best reward in the order exploration phase and executes the policy generated with that ordering for a number of steps. After each step, the reward is checked against the reward obtained during order exploration. If the difference is significant, with high probability the orderings used by agents are not identical and agents will move to the next best ordering. It is worth noting that in theory any existing solution technique for single agent MDP can be extended to the multiagent setting if agents can successfully learn common labelings. The authors do not provide experimental results showing how efficiently labels can be learned in practice.

Chapter 3

Natural Solutions

Successful application of single decision control algorithms to decentralized control of cooperative agents often requires the agents to solve identical interest strategic games. The success of these control algorithms depends on the agents' ability to coordinate on the same pareto-optimal Nash equilibrium. Research has focused on either endogenous qualities of various Nash equilibria or indigenous qualities that are derived from the fact that learning algorithms are shown to converge to these equilibria. In this chapter, we consider a class of symmetric games of identical interests where multiple pareto-optimal Nash equilibria are present. We argue that arbitrary actions are neither optimal nor rational, and a "natural solution" can be defined without arbitrary actions on the players' part. We will illustrate the concept for specific examples and discuss computational issues associated with this solution concept.

3.1 Coordination in Strategic Games

Game theory has been used to both explain interactions among presumed rational decision makers and prescribe strategic decision making in a wide array of economic and social situations. While game theory was initially proposed to solve fixed sum games in which opponents' interests are diametrically opposed, game theorists soon turned their attention to games that model situations where decision makers may benefit from cooperation. In the Prisoners' Dilemma depicted in Figure 3.1, for instance, the best outcome for the players as a whole is that neither confesses. However, individually, each has the incentive to confess regardless of what the other player chooses. This results in the worst outcome for the players as a group.

	Don't Confess	Confess
Don't Confess	3,3	0,4
Confess	4,0	1,1

Figure 3.1: Prisoners' Dilemma

Thomas Schelling, in his seminal work The Strategy of Conflict [36], considered cases where two individuals must coordinate on the same action in order to receive a common positive reward. It is hard for us to believe 50 years later, but before Schelling's groundbreaking work these types of identical interest coordination games were generally considered to be uninteresting or unproblematic. For instance, none other than the great Luce and Raiffa [37] claimed that any group of players "which can be thought of as having a unitary interest motivating its decisions can be treated as an individual in the theory." They insisted that solving games in which all players are equipped with identical preference order over outcomes is trivial.

In The Strategy of Conflict, Schelling described informal experiments where he asked people to choose between Heads or Tails, name a number in a series of numbers, or select a place to meet in a given city. Two people who agree on the same choice would each receive a common positive (though hypothetical) payoff regardless of which specific choice they agree on. Otherwise they would receive a zero payoff. In most instances, people seem to have an uncanny ability to coordinate on the same choice without prior communication. When 42 people were asked to choose between Heads or Tails, 36 chose Heads. The number 1 is by far the most popular number in the number naming experiment. Grand Central Station is the most common answer for a meeting place in New York City. This leads Schelling to argue that some strategy combination or outcome has properties of prominence or conspicuousness. He termed such a strategy combination a *focal point*. Lewis [38] coined the term *salience* to describe the property of an outcome that is “unique in some way that the subjects will notice, expect each other to notice.” Although the payoff associated with an outcome can be a source of its salience, what Schelling was primarily investigating with his experiments was the salience associated with decision or action labels. People gravitate toward decisions and actions whose labels are salient, and this salience is often rooted in common experience or cultural background of the players. When a “salient” solution presents itself [39, 40], all players instinctively focus on a specific equilibrium, often the option that is closer, easier, brighter, or something cognitively distinct.

Many have attempted to explain how salience came to be. One such theory of particular interest is the theory of team reasoning [41, 42]. An individual that team reasons asks ”what should we do?” rather than ”what should I do?” There are many versions of team reasoning theory, but roughly speaking each player will choose the team optimal joint action profile and act out its part accordingly. When people attempt to coordinate they often choose the action they perceive as most likely chosen by others rather than choosing their preferred action. While team reasoning does not apply directly to the coordination problems we study in this chapter, it nonetheless is in the same spirit as other parts of the dissertation.

In the absence of consistent common labels, salience derived from labels cannot be depended upon as the basis for decision making. Instead, decisions have to be made based solely on the utility or the payoff of the strategic game, and this is in fact more inline with the traditional game theoretic approach. We will formally define strategic games and introduce the solution concept of Nash equilibrium.

Definition 7. *A strategic game consists of*

- *a finite set of N players*
- *for each player $i \in N$ a nonempty set of actions A_i*
- *for each player $i \in N$ a preference relation $\succsim_i$ on $A = \times_{j \in N} A_j$.*

Generally the preference relation $\succsim_i$ is represented by a payoff function $u_i : A \to \mathcal{R}$ where $u_i(a) \geq u_i(b)$ if and only if $a \succsim_i b$. We can then refer to a strategic game with the 3-tuple $\langle N, (A_i), (u_i) \rangle$. The Nash equilibrium has emerged as the dominant solution concept for strategic games. It captures a steady state in which no single player can benefit by unilateral deviation. In a Nash equilibrium, each individual player's action is best-response given the action profile of the other players' actions.

Definition 8. *A Nash equilibrium of a strategic game $\langle N, (A_i), (u_i) \rangle$ is a profile $a^* \in A$ of actions with the property that for every player $i \in N$ we have*

$$u_i(a^*_{-i}, a^*_i) \geq u_i(a^*_{-i}, a_i) \text{ for all } a_i \in A_i.$$

The assured existence of a Nash equilibrium in mixed strategies is one of the key results that helped to shape the field in its early days [43]. However, in application its non-uniqueness proves to be problematic if it was to be used to aid coordination. Consider the game of Stag Hunt in Figure 3.2. Two individuals go on a hunting trip together. Each can choose to hunt a stag or a hare. However, they will only successfully hunt the more valuable stag if they cooperate. There are two Nash equilibria in this game: when both hunt the stag or when both hunt the hare. While one can argue the (2,2) joint action is the one players should coordinate on, if one player believes the other player will choose to hunt the hare then the only rational thing to do would be to hunt the hare too.

In response, various equilibrium refinement and selection schemes have been proposed. Deductive selection methods focus on reasoning and the inherent property

	Stag	Hare
Stag	2,2	0,1
Hare	1,0	1,1

Figure 3.2: Stag Hunt

of a specific Nash equilibrium. Examples of this include payoff dominance and risk dominance [44]. A Nash equilibrium weakly payoff dominates another Nash equilibrium if each player's payoff in the former is at least as good as his payoff in the latter. A Nash equilibrium piecewise risk dominates another if adherents of the former do better than adherents of the latter against players that play each Nash equilibrium with equal probability. The ability of these deductive methods to narrow the field down to a single Nash equilibrium is, however, generally limited. Furthermore, they often lead to conflicting outcomes. In the game of Stag Hunt, payoff dominance will lead to (2,2) and risk dominance prefers (1,1). Inductive selection methods use adaptive dynamics such that players through trial and error can hopefully converge to a single equilibrium. The most famous example of this is Fictitious Play [45, 46, 47].

Many of the proposed equilibrium selection mechanisms above seek to resolve a natural tension between objective individual self-interest and uncertainty about how the game will be played, and this tension seems generally unavoidable in non-cooperative games. However, for identical interest games, where self interest is synonymous with group interests, this tension is perhaps easier to resolve. We seek to exploit symmetry and common interests in deriving a new solution concept that is both technically precise and also endogenously salient. Furthermore, its guaranteed uniqueness ensures coordination when all players choose to take their respective actions accordingly.

3.2 Defining Natural Solutions

Definition 9. *A strategic game $\langle N, (A_i), (u_i) \rangle$ is a symmetric game of identical interest if*

- *$A_i = A_j$ for all players i and j*
- *$u_i(y^1, \dots, y^N) = u_j(y^1, \dots, y^N)$ for all players i and j when the group plays the profile $y^1, \dots, y^N$*
- *the common payoff is the same for any permutation of $y^1, \dots, y^N$.*

We can denote the common set of actions with A and the common payoff function with $u : A^N \to \mathcal{R}$. It is a well-known fact that symmetric strategic games generally have symmetric mixed strategy equilibria $\mathbf{x} = (x, \dots, x)$, where x is a mixed strategy over A [48, 49].

Assumption 1. *The common set of pure strategies A is finite, and the payoff function is such that given a subset of actions $G \subseteq A$, there is a* unique *symmetric mixed strategy Nash equilibrium $\mathbf{x}(G) = (x(G), \dots, x(G))$ for which the support of each player's equilibrium is precisely G. In addition, using $v(G)$ to denote the expected payoff ("value") associated with the equilibrium $\mathbf{x}(G)$, the following properties hold:*

P.1 *Given $G_1 \subseteq G_2 \subseteq A$, then*

$$v(G_1) \geq v(G_2), \tag{3.1}$$

and the inequality is strict if G_1 is a strict subset of G_2.

P.2 *Given disjoint G_1, G_2, G_3 (all subsets of A), then*

$$v(G_1) = v(G_2) \iff v(G_1 \cup G_3) = v(G_2 \cup G_3). \tag{3.2}$$

For convenience, we refer to any subset of actions $G \subseteq A$ as an *action group.* Also, we let $X(G)$ denote the set of all mixed strategies x whose support is precisely G. One implication of Assumption 1 is that a specific (unique) mixed strategy equilibrium $x(G)$ is implied by the decision to (i) put positive probability on each of the actions

$a \in G$ and (ii) put zero probability on all of the actions $b \in A \setminus G$. Thus, the problem of selecting a symmetric mixed strategy equilibrium is in a sense equivalent to the problem of selecting an action group.

The solution concept we propose for games that satisfy Assumption 1 is tightly coupled to the payoffs that can be achieved when individual players make no arbitrary decisions about what actions to play. Some additional notation will be helpful. For any $a \in A$, let the function $\tilde{u}_a : A^{N-1} \mapsto \Re$ be defined by

$$\tilde{u}_a(y^2, \ldots, y^N) = u(a, y^2, \ldots, y^N). \tag{3.3}$$

We can interpret $\tilde{u}_a$ as the payoff function for the game that is defined by Player 1 unilaterally declaring his intent to play $a \in A$. We are now equipped to define a notion of equivalence between actions.

Definition 10 (Equivalent Actions)**.** *Two distinct actions a and a' are* equivalent*, denoted $a \leftrightarrow a'$, if there exists a bijective function $\phi_{a,a'} : A^{N-1} \mapsto A^{N-1}$ such that*

$$\tilde{u}_a(\alpha) = \tilde{u}_{a'}(\phi_{a,a'}(\alpha)), \qquad \forall\, \alpha \in A^{N-1}. \tag{3.4}$$

In other words, two actions are equivalent if, after they are selected by Player 1, they offer the Players 2 through N the same opportunities to receive payoffs, i.e. if the games defined by $\tilde{u}_a$ and $\tilde{u}_{a'}$ are equivalent. We now proceed to define an important building block for our solution concept.

Definition 11 (Atomic Action Groups)**.** *An action group G is* atomic *if for all $a \in G$*

1. *the actions a and a' are equivalent (i.e. $a \leftrightarrow a'$) for all a' in G, and*
2. *the actions a and b are not equivalent for any $b \in A \setminus G$.*

For singleton action groups $G = \{a\}$, the first requirement above holds vacuously, though the second may not. As a convention, we do not consider the empty set $\emptyset$ to be atomic. The definitions above imply that G is atomic if (i) for any action $a \in G$ the $(N-1)$-player game defined by $\tilde{u}_a$ is equivalent to the $(N-1)$-player game defined by $\tilde{u}_{a'}$ for any other $a' \in G$ *and* (ii) G contains all such actions.

Definition 12 (Proper Action Groups)**.** *An action group G is* proper *if (i) it is a union of atomic action groups and (ii) it is such that if $F \subseteq G$ is an atomic action group then G contains all atomic action groups $H \subseteq A$ such that $v(H) = v(F)$. An action group is* improper *if it is not proper.*

Note that an atomic action group F is itself proper only if it is the unique atomic action group with the value $v(F)$. Strict subsets of atomic action groups are improper. Any action group involving a strict subset of an atomic action group is improper. By convention, the empty set is improper. The full set of actions A is itself necessarily proper.

Proper action groups have the following properties:

Proposition 1. *The union of two proper action groups is also itself proper.*

Proof. Let F and G be proper action groups. It is easy to see that $F \cup G$ is the union of atomic action groups. Suppose that the union $F \cup G$ is not proper, then by definition it must be the case that there exists an atomic action group $H \subseteq F \cup G$ and an atomic action group $H' \subseteq A \setminus (F \cup G)$ such that $v(H) = v(H')$. Since H is atomic, either $H \subseteq F$ or $H \subseteq G$. Without loss of generality let $H \subseteq F$. $H' \subseteq A \setminus (F \cup G)$ implies $H' \subseteq A \setminus F$. Since $v(H) = v(H')$ then by definition F cannot be proper. This contradicts our initial assumption and therefore the union must itself also be proper. □

Proposition 2. *If F and G are proper action groups and $G \subseteq F$, then $F \setminus G$ is also proper.*

Proof. To show that $F \setminus G$ is the union of atomic action groups, let a be an action in $F \setminus G$ and b be any action that is equivalent to a. Since F is proper, we have

$b \in F$. Additionally, b can not be contained in the proper group G since otherwise a will also be contained in G. Therefore $b \in F \setminus G$ and $F \setminus G$ is a union of atomic action groups. It remains to prove that $F \setminus G$ also satisfies the second part of the definition. Suppose this is not the case and therefore there exists an atomic action group $H \subseteq F \setminus G$ and an atomic action group $H' \subseteq A \setminus (F \setminus G)$ such that $v(H) = v(H')$. $A \setminus (F \setminus G)) = (A \setminus F) \cup G$, so H' is either a subset of G or a subset of $A \setminus F$. H' is a subset of G implies that G is not proper. If H' is a subset of $A \setminus F$, then F is not proper. Both cases contradict our initial assumption and therefore $F \setminus G$ must be itself proper. □

Definition 13 (Natural Action Groups)**.** *An action group G is* natural *if it is proper and has the property that for any proper action group $F \subseteq G$ there is no proper action group $H \subseteq A \setminus G$ such that $v((G \setminus F) \cup H) = v(G)$. If an action group is not natural, then we refer to it as* unnatural.

Thus, to be "natural" an action group G must be proper and must also be such that no subset of G that is proper can be replaced by a proper action group that is disjoint to G. Note that the full set of actions A is itself natural vacuously since there are no proper action groups $H \subseteq A \setminus A$. Observe also that if G is natural and $H \subseteq A \setminus G$ is such that $v(H) = v(G)$, then H cannot be proper. Indeed, if $G \cap H = \emptyset$, then H would be a disjoint proper action group whose value is the same as G, and this would contradict the fact that G is natural. More generally, we have the following proposition.

Proposition 3. *If G is a natural action group and action group $H \neq G$ is such that $v(H) = v(G)$, then H cannot be proper (and thus cannot be natural).*

Proof. Let G and H be as stated in the proposition. As we have already observed, if G and H are disjoint, then H cannot be proper. To address the remaining case in which $G \cap H \neq \emptyset$, let us suppose to the contrary that H is proper. The fact that

$v(G) = v(H)$ implies through property **P.1** that neither G nor H is nested within the other. Thus, since $G \neq H$, it must be true that $G \setminus H$, $H \setminus G$, and $G \cap H$ are all nonempty. Now let a be an action in the intersection $G \cap H$. Since both G and H are proper, they must both contain all actions $b \in A$ that are equivalent to a, and thus $G \cap H$ is a union of atomic action groups. In addition, for any atomic action group $F \subseteq G \cap H$, the fact that G and H are both proper implies that they both contain all atomic action groups with value equal to $v(F)$. Thus, $G \cap H$ must itself be proper. Consequently, $G \setminus H$ and $H \setminus G$ must also be proper. Now, since G is natural and $G \setminus H$ and $H \setminus G$ are proper, we have that

$$\begin{aligned} v(G) &\neq v((G \setminus (G \setminus H)) \cup (H \setminus G)) \\ &= v(H) \end{aligned}$$

which is a contradiction. Thus, H cannot be proper. □

By Proposition 3, if G is natural, there can be no other natural action groups $H \neq G$ with the same value as G. In other words, there is no remaining ambiguity about how to achieve the value of a natural action group. Consequently, we are motivated to make the following definition.

Definition 14 (Natural Solutions)**.** *A natural action group G^* for a given strategic game is the natural solution of the game if all other natural action groups G are such that $v(G) < v(G^*)$. We use v^* to denote the value of the natural solution $v(G^*)$.*

From our earlier observations, A itself is always natural, and, from property **P.1**, A is the the only action group with value less than or equal to $v(A)$. Since there can be only finitely many distinct natural action groups, a natural solution must exist.

The following proposition describes a convenient equivalent characterization of natural action groups.

Proposition 4. *A proper action group G is natural if and only if for any proper action group $F \subseteq G$ and any proper action group $H \subseteq A \setminus G$ it is true that $v(F) \neq v(H)$.*

Proof. By property **P.2**, the existence of action groups $F \subseteq G$ and $H \subseteq A \setminus G$ such that $v(F) = v(H)$ is equivalent to $v((G \setminus F) \cup F) = v((G \setminus F) \cup H)$. Thus, the hypothesis that $v(F) \neq v(H)$ for all pairs of proper action groups $F \subseteq G$ and $H \subseteq A \setminus G$ is equivalent to the hypothesis that G is natural. □

Clearly, for any game satisfying Assumption 1 the natural solution is a (possibly mixed) Nash equilibrium. Thus, we may regard "natural-ness" as an equilibrium selection mechanism, in the same vein as payoff dominance [44]. What we achieve in selecting a natural solution is a form of uniqueness: to paraphrase Proposition 3, *If G is the natural solution, then any other action group H with equivalent value cannot be proper, meaning that H must be comprised of some but not all elements of an atomic action group. Moreover, the natural solution G^* is the natural action group that offers the highest expected payoff.*

Assumption 1 certainly is key in deriving our main results. The assumption begins by requiring that a unique mixed strategy equilibrium be associated with the resolve (on the part of *all* players) to put positive measure on any action group $G \subseteq A$. Note that the existence of such a mixed strategy equilibrium is clear from [48, 49, 43], and the uniqueness requirement is what makes this an assumption. Property **P.1** requires that the equilibrium payoff associated with an action group G becomes strictly worse as new actions are added, i.e. played with positive probability. This property is the driving force behind Proposition 3 and also the assured existence of a natural solution. In a sense, **P.1** creates an essential tradeoff between (i) the value that can be achieved by all players agreeing on particular actions and (ii) the cost of uncertainty about which action to choose. Property **P.2** is a more technical requirement and is used mainly in validating the test for natural-ness in Proposition 4. Games that reward commonality in action selection tend to satisfy property **P.1**. In the next section, we illustrate this for a class of static "agreement" games, where positive reward is associated only with *every* player agreeing on an action.

3.3 Static Agreement (SA) Games

Definition 15. *(Static Agreement Games) A static agreement game is an N-player symmetric game defined by a common set of actions $A = \{a_1, a_2, \ldots, a_n\}$ and a payoff vector $u = (u_{a_1}, u_{a_2}, \ldots, u_{a_n})' > 0$ such that u_{a_i} is the common payoff if all players select the same action $a_i \in A$ and the common payoff otherwise is zero.*

Note that when there is a unique maximum u^* among the payoffs $\{u_{a_1}, u_{a_2}, \ldots u_{a_n}\}$, then it is reasonable to take the unique Nash equilibrium that achieves the value of u^* as an "optimal solution". All players should put probability 1 on the corresponding action in A. However, when the maximum payoff is not uniquely achievable, then what constitutes a reasonable solution becomes much less clear. Unfortunately, existing equilibrium selection mechanisms, such as the payoff and risk dominance criteria of [44], which are designed to identify pure strategy equilibria, do not provide a clear answer.

3.3.1 Verifying Assumption 1 for SA Games

We now show that static agreement games satisfy Assumption 1. To simplify notation, let $v(x)$ denote the expected payoff associated with all players using the same mixed strategy x, i.e.

$$v(x) = \sum_{i=1}^{n} u_{a_i} x_{a_i}^{N}. \tag{3.5}$$

Similarly, let

$$v(x, \bar{x}) = \sum_{i=1}^{n} u_{a_i} x_{a_i}^{N-1} \bar{x}_{a_i}. \tag{3.6}$$

denote the expected payoff given that $N-1$ players agree on $x \in X$ and a single player deviates by choosing $\bar{x} \in X$.

Now given an action group $G \subseteq A$, consider the mixed strategy

$$\begin{aligned} x(G) &= k_G \cdot (1_{a_1 \in G} \cdot u_{a_1}^{-1/(N-1)}, 1_{a_2 \in G} \cdot u_{a_2}^{-1/(N-1)}, \\ &\quad \ldots, 1_{a_n \in G} \cdot u_{a_n}^{-1/(N-1)})' \in X(G), \end{aligned} \tag{3.7}$$

where

$$k_G = \frac{1}{\sum_{a \in G} u_a^{-1/(N-1)}} \tag{3.8}$$

is a normalizing constant and $1_{a_i \in G}$ is an indicator variable that evaluates to one if $a_i \in G$ and zero otherwise.

Lemma 1. *Let G be an action group for a static agreement game. The mixed strategy profile $\mathbf{x}(G) = (x(G), \ldots, x(G))$, having value*

$$v(G) = v(x(G)) = k_G^{N-1}, \tag{3.9}$$

is the unique symmetric mixed strategy Nash equilibrium in $X(G)$

Proof. We first show that $\mathbf{x}(G)$ is a Nash equilibrium. Note that

$$\begin{aligned} v(x(G), \bar{x}) &= \sum_{i=1}^{n} u_{a_i} \left(k_G \cdot 1_{a_i \in G} \cdot u_{a_i}^{\frac{-1}{N-1}} \right)^{N-1} \bar{x}_{a_i} \\ &= k_G^{N-1} \sum_{i=1}^{n} 1_{a_i \in G} \cdot \bar{x}_{a_i} \\ &\leq k_G^{N-1} \\ &= v(x(G)), \end{aligned}$$

where the third line holds with equality when $\bar{x} \in X(G)$. Since no individual player can deviate from $\mathbf{x}(G)$ to obtain a higher expected payoff, $\mathbf{x}(G)$ is a Nash equilibrium.

To establish uniqueness, suppose that $(x, \ldots .x) \in X(G)$ is a Nash equilibrium. If we list the elements of G as $a_{(1)}, \ldots, a_{(|G|)}$, it must be the case that $u_{a_{(i)}} x_{a_{(i)}}^{N-1} = u_{a_{(i+1)}} x_{a_{(i+1)}}^{N-1}$ for $i = 1, \ldots, |G| - 1$. To see this, suppose without loss of generality that $u_{a_{(i)}} x_{a_{(i)}}^{N-1} > u_{a_{(i+1)}} x_{a_{(i+1)}}^{N-1}$, then any individual player can improve its payoff by playing action $a_{(i)}$ with probability $x_{a_(i)} + x_{a(i+1)}$ and $a_{(i+1)}$ with probability zero. These equalities along with the requirement that $x_{a_{(1)}} + \cdots + x_{a_{(|G|)}} = 1$ defines system of linear equations that can only be satisfied by one vector in $X(G)$, namely $\mathbf{x}(G)$. □

Lemma 2. *Let $G_1, \ldots, G_m$ be mutually disjoint action groups for a static agreement game. Then,*

$$v(\cup_{i=1}^{m} G_i) = \frac{1}{[\sum_{i=1}^{m} (v(G_i))^{-1/(N-1)}]^{N-1}} \tag{3.10}$$

Proof. Since

$$(v(G_i))^{-1/(N-1)} = \left[\sum_{a \in G_i} u_a^{-1/(N-1)} \right], \quad i = 1, \ldots, m,$$

we have that

$$\begin{aligned} v\left(\cup_{i=1}^{m} G_i\right) &= \frac{1}{\left[\sum_{a \in G} u_a^{-1/(N-1)}\right]^{N-1}} \\ &= \frac{1}{\left[\sum_{i=1}^{m} \sum_{a \in G_i} u_a^{-1/(N-1)}\right]^{N-1}} \\ &= \frac{1}{\left[\sum_{i=1}^{m} (v(G_i))^{-1/(N-1)}\right]^{N-1}}. \end{aligned}$$

□

Some easy consequences of Lemma 2 are the following.

Corollary 1. *Let G_1 and G_2 be action groups for a static agreement game such that $G_1 \subseteq G_2$. Then,*

$$v(G_1) \geq v(G_2), \tag{3.11}$$

and the inequality is strict if G_1 is a strict subset of G_2.

Corollary 2. *Let G_1, G_2, and G_3 be mutually disjoint action groups for a static agreement game. Then,*

$$v(G_1) = v(G_2) \quad \Longleftrightarrow \quad v(G_1 \cup G_3) = v(G_2 \cup G_3). \tag{3.12}$$

Corollary 3. *Let $G_1, G_2, \dots, G_m$ be mutually disjoint action groups with equal value, i.e. $v(G_1) = v(G_2) = \dots = v(G_m) = \kappa$. Then,*

$$v\left(\cup_{i=1}^{m} G_i\right) = \frac{\kappa}{m^{N-1}}. \tag{3.13}$$

Corollaries 1 and 2, along with Lemma 1, imply that static agreement games satisfy the requirements of Assumption 1, and thus the solution concept of natural solutions applies.

We point out that for any $a \in A$, the value of the singleton action group $\{a\}$ is u_a. Thus, if a is such that $u_a \geq u_{\bar{a}}$ for all $\bar{a} \in A$, then u_a is the highest value any action group can have. On the other hand, from Corollary 1, thinking of A itself as an action group, $v(A)$ is the smallest value that an action group can have and no other action group can achieve that value.

Note also that an action group $G \subseteq A$ is atomic if all of the actions that it contains have individually equivalent payoffs, and no other actions $b \notin G$ have the same payoff as those represented by G. In particular, G is not atomic if it involves some, but not all, actions a that achieve a particular value.

3.3.2 Examples

We now illustrate our solution concept in the context of some specific examples.

Example 1. Consider the two-player SA game defined by the payoff vector

$$u = (4, 4, 2, 2).$$

Here, the only atomic action groups are $G_{1-2} = \{a_1, a_2\}$ and $G_{3-4} = \{a_3, a_4\}$. Since $v(G_{1-2}) \neq v(G_{3-4})$, both are proper. A itself is also proper. All three proper action groups are natural. In particular, the action group G_{1-2} is natural despite the fact that is has the same value as $G_3 = \{a_3\}$ (and as $G_4 = \{a_4\}$) – the action groups G_3 and G_4 are not proper.

Example 2. Consider the two-player SA game defined by the payoff vector

$$u = (4, 4, 6, 6, 6, 8, 8, 8, 8).$$

Here, using the same notation as in the preceding example, the action groups G_{1-2}, G_{3-5}, and G_{6-9} are atomic. However, since $v(G_{1-2}) = v(G_{3-5}) = v(G_{6-9})$, none of the atomic action groups are individually proper. Similarly, any pair of atomic action groups is improper. The only proper action group is A itself, which is also natural.

Example 3. Consider the two-player SA game defined by the payoff vector

$$u = (3, 6, 6, 6).$$

Here, G_1 and G_{2-4} are the atomic action groups. The proper action groups are G_1, G_{2-4}, and A itself. Note that the action group G_1 is natural since the only disjoint proper action group G_{2-4} has a different value. (It is important to note that G_1 is natural despite the fact that other disjoint action groups have the same value – all such disjoint action groups, i.e. G_{2-3}, $G_{2,4}$, and G_{3-4}, fail to be proper.) The action group G_{2-4} is also natural since its value is not the same as the disjoint proper action group G_1. Finally, the third (and final) natural action group for this game is A itself.

Example 4. Consider the two-player SA game defined by the payoff vector

$$u = (3, 6, 12, 12, 18, 18, 18)$$

for which the atomic action groups are G_1, G_2, G_{3-4}, and G_{5-7}. The action group G_1 is proper since no other atomic action groups have the same value. Any action group involving some but not all of G_2, G_{3-4}, and G_{5-7} is improper. On the other hand G_{2-7} and A itself are proper. Since $v(G_1) \neq v(G_{2-7})$, all three proper action groups are natural.

Example 5. Consider the two-player SA game defined by the payoff vector

$$u = (2, 2, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6)$$

for which the atomic action groups are G_{1-2}, G_{3-4}, G_{5-9}, and G_{10-12}. The proper action groups are $G_{1-2,5-9}$, $G_{3-4,10-12}$ and A itself. Since $v(G_{3-4,10-12}) > v(G_{1-2,5-9})$, all three proper action groups are natural.

Example 6. Consider the two-player SA game defined by the payoff vector

$$u = (8, 8, 8, 8, 2, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6)$$

for which the atomic action groups are G_{1-4}, G_5, G_{6-7}, G_{8-12}, and G_{13-15}. The proper action groups are $G_{1-7,13-15}$, G_{8-12} and A itself. Since $v(G_{8-12}) > v(G_{1-7,13-15})$, all three proper action groups are natural.

Example 7. Consider the two-player SA game defined by the payoff vector

$$u = (2, 2, 8, 8, 12, 12, 12, 5, 5, 5, 5, 5, 6, 6, 6)$$

for which the atomic action groups are G_{1-2}, G_{3-4}, G_{5-7}, G_{8-12} and G_{13-15}. The proper action groups are G_{13-15}, $G_{1-2,8-12}$, $G_{1-2,8-15}$, G_{1-12}, A, G_{3-7}, and $G_{3-7,13-15}$, of which only $G_{1-2,8-12}$, A, and $G_{3-7,13-15}$ are natural.

Example 8. Consider the two-player SA game defined by the payoff vector

$$u = (18, 18, 18, 12, 12, 6, 2, 4, 8, 16, 32, 32),$$

for which the only natural action group is A itself (even though there are many proper action groups).

Table 3.1: Natural Solutions for the Examples

Example	Proper Action Groups	Natural Action Groups	Natural Solution
1	G_{1-2}, G_{3-4}, A	G_{1-2}, G_{3-4}, A	G_{1-2}
2	A	A	A
3	G_1, G_{2-4}, A	G_1, G_{2-4}, A	G_1
4	G_1, G_{2-7}, A	G_1, G_{2-7}, A	G_1
5	$G_{1-2,5-9}$, $G_{3-4,10-12}$, A	$G_{1-2,5-9}$, $G_{3-4,10-12}$, A	$G_{3-4,10-12}$
6	$G_{1-7,13-15}$, G_{8-12}, A	$G_{1-7,13-15}$, G_{8-12}, A	G_{8-12}
7	G_{13-15}, $G_{1-2,8-12}$, $G_{1-2,8-15}$, G_{1-12}, A, G_{3-7}, $G_{3-7,13-15}$	$G_{1-2,8-12}$, A, $G_{3-7,13-15}$	$G_{3-7,13-15}$
8	too many to list	A	A

Summary results for Examples 1-8 are shown in Table 3.1. All of the examples point to an important property of "natural solutions," namely the discontinuity of the solution. For instance, in Example 1, the natural solution is G_{1-2} with value $v(G_{1-2}) = 2$. By infinitesimally reducing the payoff associated with action a_2, the natural solution becomes G_1, with value 4. Thus, the concept itself is inherently extremely sensitive to small variations in payoffs.

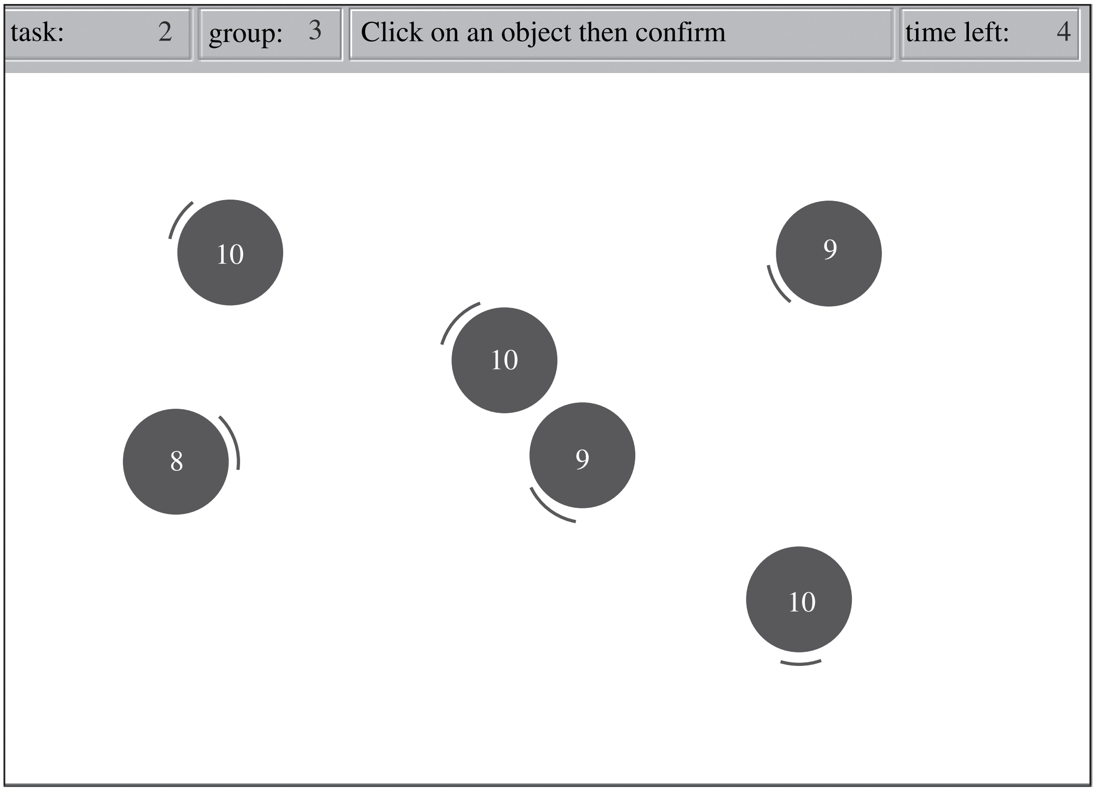
3.3.3 Descriptive Power of Natural Solution

While our interest in the development of natural solution, a new game theoretical solution concept, mainly lies in its prescriptive power, it is worthwhile to briefly discuss whether it has any descriptive power as a solution concept. In other words, is the kind of rationalization we proposed in the definition of natural solution evident when real human beings make decisions when coordination dilemmas are present?

While searching for an explanation for salience of focal points, Bardsley etc. [50] conducted experiments during which test subjects were asked to play repeated coordination games. The number tasks they designed are akin to the SA games. Examples of displays (how number tasks are presented to test subjects) are shown in Figure 3.3 and 3.4. In Figure 3.3, the discs containing numbers bounce around randomly on the computer screen. The lines next to the discs indicate movement and are not actually shown to test subjects. In the coordination game, two test subjects were each asked to choose one object. They are told that if the same object is chosen by both, then both will receive the number of points indicated on that object. Otherwise neither of them receives anything. The presentation of the number task is designed such that the payoffs alone are meant to influence decisions. In other words, the actions are presented in a way such that it is impossible for the test subjects to derive salience from their presentations alone.

In one set of experiments, the array of points carried by the set of objects are as follows:

- Type 1: $\{10, 10, 10, \mathbf{9}\}$, $\{10, 10, 10, 10, 10, \mathbf{9}\}$, $\{10, 10, 10, \mathbf{9}, 8, 7\}$, $\{10, 10, 10, 9, 9, \mathbf{8}\}$, $\{10, 10, 10, 10, \mathbf{9}, \mathbf{9}\}$,



task: 2 group: 3 Click on an object then confirm time left: 4

Figure 3.3: Example of display for number tasks. Reprinted from "Explaining focal points: Cognitive hierarchy theory versus team reasoning, " by Bardsley, N., Mehta, J., Starmer, C., and Sugden, R, 2010, The Economic Journal, 120: 40-79. Reprinted with permission.

- Type 2: $\{\mathbf{10}, 9\}$, $\{\mathbf{10}, \mathbf{10}, \mathbf{10}, 9, 9, 9\}$, $\{\mathbf{10}, \mathbf{10}, \mathbf{10}, \mathbf{10}, \mathbf{10}, 1\}$.

The natural solutions are indicated with the bold face above. Test subjects (university students) were able to select the natural solution overwhelmingly in all of these coordination games except for the last one under each type. There far more subjects still choose the natural solution. Notice for each array of points, there is a unique atomic action group with maximum group value. None of them involves two atomic action groups with the same group value, for which selecting the natural solution involves another layer of reasoning on the test subject's part. It is uncertain how many layers of reasoning a typical human being is mentally capable of or motivated enough to even attempt.

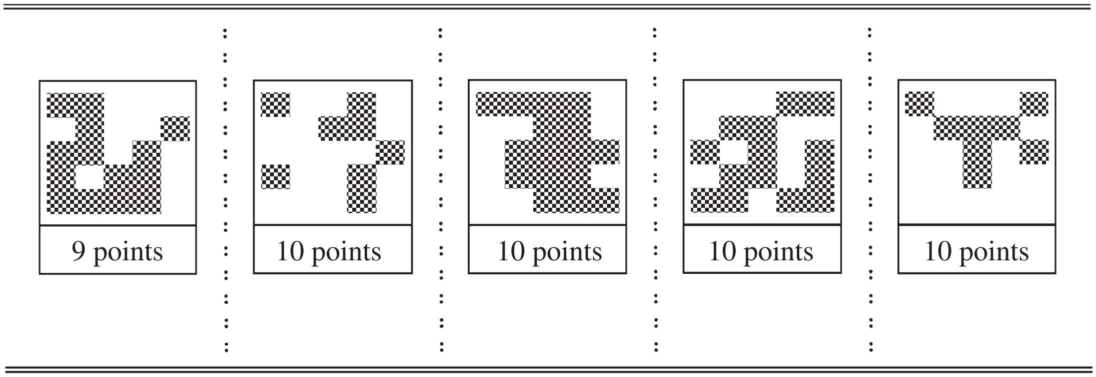


Figure 3.4: Example of display for number tasks. Reprinted from "Explaining focal points: Cognitive hierarchy theory versus team reasoning, " by Bardsley, N., Mehta, J., Starmer, C., and Sugden, R, 2010, The Economic Journal, 120: 40-79. Reprinted with permission.

3.4 Computational Issues

As discussed in the previous section, a natural solution is guaranteed to exist under Assumption 1. Clearly, for any instance of the game the computation can be done via brute force enumeration of all the natural action groups. We conjecture that computing a natural solution is NEXP-complete, and the worst-case complexity of finding an exact solution is potentially prohibitive. It is therefore worthwhile to consider whether there are efficient heuristics that can produce the natural solution almost consistently. In this section, we propose a heuristic that we refer to as the "parallel reduction algorithm (PRA)". We investigate when PRA is guaranteed to produce the natural solution, what happens when it fails to find the natural solution, and then demonstrate empirically its effectiveness on SA games.

3.4.1 The Parallel Reduction Algorithm (PRA)

The Parallel Reduction Algorithm involves aggregating unions of atomic action groups of like value in a stagewise process. We use the term "parallel" to indicate that possible unions of like-valued action groups are aggregated simultaneously, as opposed to for instance only aggregating the like-valued action groups with highest value. PRA is presented in Algorithm 1.

Algorithm 1 Parallel Reduction Algorithm (PRA)

```
Initialization
Given a symmetric identical interest game with action set $A$ and payoff vector $u$.
Partition $A$ into atomic action groups

        $\{G_{\alpha_1^1}, G_{\alpha_2^1}, \ldots, G_{\alpha_{m_1}^1}\},$

where $v(G_{\alpha_i^1}) \geq v(G_{\alpha_{i+1}^1})$, for $i = 1, \ldots, m_1 - 1$.
Recursion
In the $k$-th iteration:
if $v(G_{\alpha_1^k}) > v(G_{\alpha_2^k})$ then
    stop and output $G_{\alpha_1^k}$ as the solution.
else
    aggregate action groups to obtain a coarser partition

        $\{G_{\alpha_1^{k+1}}, G_{\alpha_2^{k+1}}, \ldots, G_{\alpha_{m_{k+1}}^{k+1}}\},$

where (i) each $G_{\alpha_i^{k+1}}$ is an exhaustive union of $k$-th stage action groups with like
value and (ii) $v(G_{\alpha_i^{k+1}}) \geq v(G_{\alpha_{i+1}^{k+1}})$, for $i = 1, \ldots, m_{k+1} - 1$.
end if
```

Proposition 5. *If PRA terminates within $k = 2$ stages, then it produces a natural solution.*

Proof. The first round of parallel reduction results in the partitioning of A into atomic action groups

$$\{G_{\alpha_1^1}, G_{\alpha_2^1}, \ldots, G_{\alpha_{m_1}^1}\},$$

where we may assume that $v(G_{\alpha_i^1}) \geq v(G_{\alpha_{i+1}^1})$, for $i = 1, \ldots, m_1 - 1$. PRA terminates at this point with $G_{\alpha_1^1}$ as the "answer" if $v(G_{\alpha_1^1}) > v(G_{\alpha_2^1})$. Would this answer be correct? Termination implies that $G_{\alpha_1^1}$ is necessarily proper; it would in fact have to be natural, since (i) there are no strict subsets of $G_{\alpha_1^1}$ that are proper and (ii) any other disjoint proper action group would have to have value less than $v(G_{\alpha_1^1})$. Thus, the "answer" would indeed be the natural solution of the SA game.

If PRA fails to terminate in the first iteration, then it must have been the case that $v(G_{\alpha_1^1}) = v(G_{\alpha_2^1})$, and the second stage of reduction results in a new, coarser partition of A:

$$\{G_{\alpha_1^2}, G_{\alpha_2^2}, \ldots, G_{\alpha_{m_2}^2}\},$$

where we may assume that $v(G_{\alpha_i^2}) \geq v(G_{\alpha_{i+1}^2})$, for $i = 1, \ldots, m_2 - 1$. Note that by the definition of PRA each $G_{\alpha_i^2}$ is an exhaustive union of like-valued atomic action groups and is thus necessarily proper. PRA terminates with $G_{\alpha_1^2}$ as the "answer" if $v(G_{\alpha_1^2}) > v(G_{\alpha_2^2})$. To see if this answer would be correct, suppose that the termination condition is satisfied. Note that no strict subset of $G_{\alpha_1^2}$ can be proper, thus, to prove that $G_{\alpha_1^2}$ is natural, it remains to show that no disjoint proper action group has the same value. We now establish that this is the case. First, note that any action group that involves a part, but not all of $G_{\alpha_i^2}$ for any i, cannot be proper. Thus, proper action groups that are disjoint to $G_{\alpha_1^2}$ must involve unions of $G_{\alpha_i^2}$ for $i > 1$. Since $v(G_{\alpha_1^2}) > v(G_{\alpha_i^2})$ for $i > 1$, all such unions must have strictly lower value, and $G_{\alpha_1^2}$ and must be natural. To see that $G_{\alpha_1^2}$ is the natural solution, we must show that all other natural action groups have lower value. In order for a group to be a natural action group, it must first be proper. Since any action group that involves a part but not all of $G_{\alpha_i^2}$ for any i cannot be proper, any proper action groups must involve unions of $G_{\alpha_i^2}$. This implies that its value is less than or equal to the value of $G_{\alpha_1^2}$. If the value is less than the value of $G_{\alpha_1^2}$, then even if it is a natural action group it cannot be the natural solution. If it has the same value as $G_{\alpha_1^2}$, then it must be the case that it is $G_{\alpha_1^2}$ itself, otherwise the termination condition would not be have been met. Thus we have concluded that $G_{\alpha_1^2}$ is the natural solution. □

Lemma 3. *Let $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ be the set of partitions during k-th stage of PRA. For all $k \geq 2$, if no strict subset in $\{G_{\alpha_1^{k+1}}, G_{\alpha_2^{k+1}}, \ldots, G_{\alpha_{m_{k+1}}^{k+1}}\}$ is natural, then no strict subset in $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ is natural.*

Proof. For all $k \geq 2$, each non-empty subset in $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ is a proper action group. By Proposition 4, for any subset in $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ to be natural it has to include all or none of the elements in $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ with the same group value $v(G_{\alpha_i^k})$. Since in PRA each $G_{\alpha_i^{k+1}}$ is an exhaustive union of k-th stage action groups with the same group value, any natural subset of $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ corresponds to a subset in $\{G_{\alpha_1^{k+1}}, G_{\alpha_2^{k+1}}, \ldots, G_{\alpha_{m_{k+1}}^{k+1}}\}$. Since no strict subset in $\{G_{\alpha_1^{k+1}}, G_{\alpha_2^{k+1}}, \ldots, G_{\alpha_{m_{k+1}}^{k+1}}\}$ is natural, no strict subset in $\{G_{\alpha_1^k}, G_{\alpha_2^k}, \ldots, G_{\alpha_{m_k}^k}\}$ is natural. □

Proposition 6. *If PRA terminates with full support, then it produces the natural solution.*

Proof. Suppose PRA terminates with full support in the T-th iteration. The T level partition must be a single action group $\{G_{\alpha_1^T}\}$. Let the T-1 level partition be $\{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\}$. It must be the case that $v(G_{\alpha_1^{T-1}}) = v(G_{\alpha_2^{T-1}}) = \ldots, = v(G_{\alpha_{m_{T-1}}^{T-1}})$. If $T = 2$, then $\{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\}$ are atomic action groups all with the same group value. By the definition of proper action groups, A is the only proper action group hence the only natural action group. PRA in this case produces the natural solution since it terminates with the action group A. If $T = 1$ the case is trivial.

Now consider the remaining case of when $T \geq 3$, *i.e.* $T - 1 \geq 2$. Each $G_{\alpha_i^{T-1}}$ is a proper action group, and consequently, any subset of $\{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\}$ is also proper as it is the union of proper action groups. Let F be any strict subset of $\{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\}$, and without loss of generality let $G_{\alpha_1^{T-1}} \subset \{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\} \setminus F$. Clearly F contains at least a proper action group with the same value as $G_{\alpha_1^{T-1}}$ which is outside of F. By Proposition 4, F then cannot be natural. Therefore no strict subset of $\{G_{\alpha_1^{T-1}}, G_{\alpha_2^{T-1}}, \ldots, G_{\alpha_{m_{T-1}}^{T-1}}\}$ can be natural. If $T > 3$, by repeat application of Lemma 3, we can conclude that no strict subset of $\{G_{\alpha_1^2}, G_{\alpha_2^2}, \ldots, G_{\alpha_{m_2}^2}\}$ is natural. Note that this also holds true when $T = 3$. Observe that by definition of proper action groups, every proper action group is a subset of $\{G_{\alpha_1^2}, G_{\alpha_2^2}, \ldots, G_{\alpha_{m_2}^2}\}$. Since no strict subset of $\{G_{\alpha_1^2}, G_{\alpha_2^2}, \ldots, G_{\alpha_{m_2}^2}\}$ is natural, no proper action group other than the entire set $\{G_{\alpha_1^2}, G_{\alpha_2^2}, \ldots, G_{\alpha_{m_2}^2}\}$ can possibly be natural. Therefore A is the only natural group and hence the natural solution for the game. Therefore, in this case PRA also produces the natural solution when it terminates with the action group A.

□

One implication of Proposition 5 is that the parallel reduction algorithm will always correctly identify natural solutions in games with three or fewer distinct atomic action groups. Unfortunately, PRA fails to produce the natural solution in general. To see this, consider the two-player SA game of Example 8 in Section 3.3.2, defined by the payoff vector

$$u = (18, 18, 18, 12, 12, 6, 2, 4, 8, 16, 32, 32).$$

This game reduces by PRA as follows[1]:

$$\begin{aligned}
u^0 &= (18, 18, 18, 12, 12, 6, 2, 4, 8, 16, 32, 32)\\
u^1 &= (6, 6, 6, 2, 4, 8, 16, 16)\\
u^2 &= (2, 2, 4, 8, 8)\\
u^3 &= (1, 4, 4)\\
u^4 &= (1, 2),
\end{aligned}$$

Terminating with G_{8-12}, whose value is $v(G_{8-12}) = 2$. While G_{8-12} is proper, it is unnatural since it has the same value as the disjoint proper action group G_7. It

[1] In implementing the PRA algorithm "by hand" we often find it convenient to not reorder action groups according to value.

turns out for this game that the only natural solution is A itself, whose value is $v^* = v(A) = 2/3$. (Interestingly, PRA produces a natural action group for the SA game defined by $u = (6, 6, 6, 2, 4, 8, 16, 16)$, i.e. u^1 above.)

While the action group produced by PRA is not guaranteed to be the natural solution of the game (indeed, as shown in the above example it may not even be natural), we can show that its value is at least as good as the value of the natural solution of the game.

Given a symmetric identical interest strategic game $\langle N, (A), (u)\rangle$ that satisfies Assumption 1, we can define a new symmetric identical interest strategic game $\langle N, (G), (u)\rangle$ for any $G \subseteq A$. It is easy to verify that this new game also satisfies Assumption 1 and therefore natural solution is well-defined in this new game. For notational simplicity, we denote $\langle N, (G), (u)\rangle$ with $g(G)$.

Lemma 4. *Given a strategic game $\langle N, (A), (u)\rangle$ and suppose PRA terminates after T iterations. Let G be any T level partition i.e. $G = G_{\alpha_i^T}$ for $i = 1, 2, \ldots, m^T$ and $H \subseteq G$. Then H is an atomic action group in g(G) if and only if H is an atomic action group in the original game.*

Proof. Let $g(A)$ be the original strategic game. Let $H \subseteq G$ be an atomic action group in $g(G)$. Then by definition, all actions in H are equivalent and no action in $G \setminus H$ is equivalent to any action in H. To show that H is also an atomic action group in $g(A)$, it remains to show that no action in $A \setminus H$ is equivalent to any action in H. Suppose this is not the case, and there exists an action $a \in A \setminus H$ such that a is equivalent to any action in H. Since H contains all actions in G equivalent to any action in H, it must be true that $a \notin G$, *i.e.* $a \in A \setminus G$. During the initialization phase of PRA, all actions equivalent to a are grouped into a single partition. This single partition is never split during subsequent iterations of PRA, therefore it is either entirely contained in G or disjoint from G. If it is entirely contained in G, then $a \in G$. If it is disjoint from G, then every action equivalent to a is also in $A \setminus G$ and consequently $H \in A \setminus G$. Either case leads to a contradiction, and therefore H must be an atomic action group in $g(A)$.

To show the other direction, let $H \subseteq G$ be an atomic action group in $g(A)$. By definition all actions in H are equivalent and no action in $A \setminus H$ is equivalent to any action in H. This means no action in $G \setminus H$ is equivalent to any action in H, and by definition H is an atomic action group in $g(G)$. □

Lemma 5. *Given a strategic game $\langle N, (A), (u) \rangle$, suppose PRA terminates after T iterations with $T \geq 2$. Let G be any T level partition i.e. $G = G_{\alpha_i^T}$ for $i = 1, 2, \dots, m^T$ and $H \subseteq G$. Then H is a proper action group in $g(G)$ if and only if H is a proper action group in the original game.*

Proof. If $H \subseteq G$ is a proper action group in $g(G)$ then by definition 1) H is union of atomic action groups in $g(G)$; 2) if $F \subseteq H$ is an atomic action group in $g(G)$, then H contains all atomic action groups in $g(G)$ with the same value as F. By Lemma 4, H is the union of atomic action groups in $g(A)$. We just have to show that for each $F \subseteq H$, where F is an atomic action group in $g(A)$, H contains all the atomic action groups in $g(A)$ of the same value as F. Suppose this is not true, and there exists $F \subseteq H$, with F an atomic action group in $g(A)$, $K \subseteq A \setminus H$, and K an atomic action group in $g(A)$, such that $v(F) = v(K)$. Since $F \subseteq G$ and F is an atomic action group in $g(A)$, by Lemma 4 F is also an atomic action group in $g(G)$. If $K \subseteq G$, then K is also an atomic action group in $g(G)$. This implies H cannot be a proper action group in $g(G)$, thus $K \nsubseteq G$. By virtue of how PRA operates, all atomic action groups in $g(A)$ with the same value are grouped into the same partition in the second stage, and this partition will either be entirely contained in G or disjoint from it. Since both F and K are atomic action groups in $g(A)$ with $v(F) = v(K)$, F and K are either both in G or disjoint from it. Since we have already established that K is not in G, it must be the case that both F and K are disjoint from G, but this contradicts the fact that F is in G. Therefore H must be a proper action group in $g(A)$.

To prove the other direction, suppose H is a proper action group in $g(A)$. H is the union of atomic action groups in $g(A)$. Since these atomic action groups are contained in G, by Lemma 4, H is the union of atomic action groups in $g(G)$. Now suppose H does not satisfy the second part of the definition of proper action groups, then there exists $F \subseteq H$, with F an atomic action group in $g(G)$, $K \subseteq G \setminus H$, and K an atomic action group in $g(G)$, such that $v(F) = v(K)$. F and K are both subsets of G and therefore are atomic action groups in $g(A)$ by Lemma 4. This together with the fact that $K \subseteq A \setminus H$ contradicts the assumption that H is a proper action group in $g(A)$. This concludes our proof. □

Proposition 7. *Given a strategic game $\langle N, (A), (u) \rangle$, suppose PRA terminates after T iterations with $T \geq 2$. $G_{\alpha_1^T}$, the solution produced by PRA, has group value at least as good as the value of the natural solution of the game.*

Proof. Consider the strategic game $g(G_{\alpha_1^T})$. If we apply PRA to this new game the algorithm should proceed in the same way as it did when it was applied to the original game. Therefore PRA should terminate with full support in $g(G_{\alpha_1^T})$. Given Proposition 6, we know that $G_{\alpha_1^T}$ is the natural solution of $g(G_{\alpha_1^T})$. If a group $F \subset G_{\alpha_1^T}$, F a proper action group in $g(G_{\alpha_1^T})$, is natural for the new game, then we know $v(F) > v(G_{\alpha_1^T})$. This contradicts the fact that $G_{\alpha_1^T}$ is the natural solution in $g(G_{\alpha_1^T})$, therefore, no $F \subset G_{\alpha_1^T}$ such that F is a proper action group in $g(G_{\alpha_1^T})$ is natural in $g(G_{\alpha_1^T})$. Let $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$ denote the set of second stage partitions that make up $G_{\alpha_1^T}$, then it must be the case that no strict subset of $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$ is natural in $g(G_{\alpha_1^T})$.

Next we show that no proper group in $g(A)$ containing a strict subset of $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$ (and not $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$) can be natural in $g(A)$. Suppose this is not true and let $\bar{G}$ be a natural group in $g(A)$ containing a strict subset of $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$ (and not $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$). Let $F = G_{\alpha_1^T} \cap \bar{G}$. $G_{\alpha_1^T}$ is the union of proper action groups in $g(A)$, so it is also a proper action group in $g(A)$. Therefore, F, the intersection of two proper action groups in $g(A)$, is a proper action group in $g(A)$. By Lemma 5, F is also a proper action group in $g(G_{\alpha_1^T})$ and thus F cannot be natural in $g(G_{\alpha_1^T})$. This means there exists $K \subset F$, K proper action group in $g(G_{\alpha_1^T})$, and $H \subseteq G_{\alpha_1^T} \setminus F$, H proper action group in $g(G_{\alpha_1^T})$, such that $v(K) = v(H)$. Since $(G_{\alpha_1^T} \setminus F) \cap \bar{G} = \emptyset$, $H \subseteq G_{\alpha_1^T} \setminus F$ implies $H \subseteq A \setminus \bar{G}$. Note that both K and H are proper action groups in $g(A)$. Therefore $\bar{G}$ cannot be natural in $g(A)$ by definition and this shows that no proper group in $g(A)$ containing strict subset of $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$ (and not $\{G_{\alpha_i^2}\}_{G_{\alpha_1^T}}$) can be natural in $g(A)$.

This means any natural group in the original game has to either contain $G_{\alpha_1^T}$ or be disjoint from it. If a natural group contains $G_{\alpha_1^T}$, then its value is at most $v(G_{\alpha_1^T})$. Equality holds only when that natural group is $G_{\alpha_1^T}$ itself. Similar analysis as we did for $G_{\alpha_1^T}$ can be applied to all other $G_{\alpha_i^T}$ with $i > 1$. If a natural group is disjoint from $G_{\alpha_1^T}$, the highest value of that natural group is $v(G_{\alpha_2^T})$. Since $v(G_{\alpha_2^T}) < v(G_{\alpha_1^T})$, we conclude that the value of the natural solution in the original game is at most $v(G_{\alpha_1^T})$ with equality only possible when $G_{\alpha_1^T}$ is natural. □

3.4.2 Empirical Evaluation

PRA is guaranteed to correctly identify natural solutions under the conditions described in Proposiition 5 and Proposition 6. Given its lack of guarantee in general, we are interested to see just how often PRA fails in practice. Toward this end, we applied PRA to a set of SA games that are specially designed to exaggerate PRA's chances of failing. Each SA game is generated as follows: 10 integers are uniformly and randomly chosen with replacement from the set of integers $\{1, 2, \ldots, 16\}$. x copies of the chosen integer are added to the set of payoffs, where x is uniformly chosen from $\{1, 2, \ldots, 10\}$. All experiment parameters are empirically tested to maximize the probability that PRA will fail to terminate within 2 stages.

In total, 10^7 SA games are generated. We apply PRA to this set of SA games and compare the action group produced by PRA to the natural solution of each game. The result is quite stark: the number of times PRA fails to identify the natural solution is consistently less than 20 out of 10^7.

3.4.3 Discussion

The fact that PRA may fail to produce a natural solution raises some interesting points. By definition, PRA always terminates finitely with a mixed strategy Nash equilibrium. As we have demonstrated in Proposition 7, when PRA does not terminate with the natural solution, the Nash equilibrium it produces has value strictly better than the value of the natural solution. Moreover, PRA is "robust" in the sense that whenever independent players implement the algorithm for a game it will always produce the same set of actions (and corresponding mixed strategy Nash equilibrium) regardless of how the players have labeled the actions. Consequently, we could drop the notion of "natural solutions" altogether and adopt PRA as a purely algorithmic approach to equilibrium selection.

However, for a number of reasons, a purely algorithmic mechanism for equilibrium selection can be unsatisfying. First, PRA is not the only algorithm that can unambiguously identify mixed strategy Nash equilibria. For example, any algorithm that always produces the action group A itself as the answer can be interpreted as one that robustly selects Nash equilibria, whereas clearly PRA may not always elect to put positive measure on all actions. Indeed, there are many robust algorithms for selecting action groups, all of which may produce distinct solutions, and consequently the equilibrium selection problem is "pushed up" one level to a problem of *algorithm selection*.[2] Another problem with a purely algorithmic response to equilibrium selection is the fact that, unless the algorithm happens to produce natural solutions, the

[2] Perhaps it is possible to define a solution concept in terms of an optimization over equilibria that can be robustly computed, i.e. out of all robust algorithms that can be applied to a particular game, hopefully there is one that uniquely produces an equilibrium with highest value, and we would call that equilibrium the solution to the game. Of course, there are problems with this approach if the maximum is not unique.

resulting solution can be such that no player is motivated to implement the solution. Consider again Example 8 where the PRA algorithm outputs the action group G_{8-12} with value $v(G_{8-12}) = 2$. In this case, a rational player may well consider playing the proper action group G_7, which as a singleton offers the same value as the five-action group G_{8-12}.

Chapter 4

Decentralized Planning in Bayesian Team Search

4.1 Introduction

We now turn our attention to a problem of decentralized planning in Bayesian team search. Single agent search and hypothesis testing problems have been considered extensively in the literature [51, 52, 53]. With the development of intelligent mobile sensing platforms, there has been a great deal of interest in enabling a team of autonomous agents to carry out a common search task. With limited communication capability, it is often assumed that agents do not share their individual observations perfectly or in a timely fashion. This poses one of the main difficulties in extending existing results in the area of partially observed Markov decision processes (POMDP) from the single agent Bayesian search problem to team Bayesian search. However, even if we assume that observation sharing is perfect and happens in a timely manner, decentralized planning meant to be carried out by individual team members separately still engenders very substantial issues of coordination.

4.2 Problem Formulation

4.2.1 Basic Setup and Bayesian Hypothesis Testing

Consider a finite-horizon formulation of the search problem, in which the fixed amount of time available for the task is discretized into $N+1$ equal time periods $t = 0, 1, \ldots, N$. The search area is modeled as a undirected graph $G = (V, E)$ with $|V| = v$ and $|E| = e$ such that the nodes represent the search regions. It is assumed that given a pair of nodes in V, they are adjacent in G if and only if agents can travel between them in exactly one time period. We assume that each search region is large enough in footprint for agents to loiter over the region indefinitely and therefore self loops are allowed in the graph G.

A set of mutually exclusive and all inclusive hypotheses is defined for each node. Without loss of generality, we consider the simple case of two hypotheses defined for each node $i \in V$.

- Hypothesis $\mathcal{H}_0^i$ states that there exist one or more targets at node i.

- Hypothesis $\mathcal{H}_1^i$ states that no targets exist at node i.

The finite set of possible observations is the same for all nodes at all times. We denote this set by Z. Initially, all agents are given the prior probability that $\mathcal{H}_0^i$ is true for all $i \in V$. In each time period, each agent makes a decision about which node to visit next based on the priors and all past and current observations made by all agents. At the end of period N, for each node a single hypothesis is accepted among the hypotheses defined for that node. The other hypotheses are rejected. Node i is classified to be containing one or more targets if $\mathcal{H}_0^i$ is accepted and it is classified to contain no targets otherwise. Since the total number of targets contained in the search area is unknown, the following assumption holds for our problem.

Assumption 2. *For any $i, j \in V$ with $i \neq j$, hypothesis $\mathcal{H}_0^i$ and H_0^j are independent of each other.*

As an immediate consequence of Assumption 2, $\mathcal{H}_1^i$ and $\mathcal{H}_1^j$ are independent of each other for all distinct pairs of nodes i and j. Given that all agents are homogenous, while time varying conditions such as weather, camouflage or movements on the ground may affect observations, we assume that the observations made of the same region during the same time period are identical. More formally, following assumptions hold throughout the chapter.

Assumption 3. *All observations made about a single node during a single time period have the same value. Furthermore, they are counted as one single common observation of that value.*

Assumption 4. *All observations made about a single node during different time periods are conditionally independent and identically distributed regardless of whether they are made by the same agent or different agents.*

Finally, f^i is the probability density function over Z if $\mathcal{H}_0^i$ is true and g^i is the probability density function over Z if $\mathcal{H}_1^i$ is true. Let p^i be the prior conditional probability that $\mathcal{H}_0^i$ is true and $\{z_0, \dots, z_m\}$ be a set of independent and identically distributed observations taken at node i. For $k = 1, \dots, m$, $\hat{p}^i(k)$, the posterior conditional probability that $\mathcal{H}_0^i$ is true given observations z_0 through z_k can be computed iteratively with Bayesian Theorem as follows:

$$\begin{aligned}
\hat{p}^i(0) &= \frac{p^i f^i(z_0)}{p^i f^i(z_0) + (1 - p^i) g^i(z_0)}, \\
\hat{p}^i(k) &= \frac{\hat{p}^i(k-1) f^i(z_k)}{\hat{p}^i(k-1) f^i(z_k) + (1 - \hat{p}^i(k-1)) g^i(z_k)}.
\end{aligned}$$

We can deduce from the above equations that for $k = 0, \ldots, m$,

$$\hat{p^i}(k) = \frac{p^i \prod_{l=0}^{k} f^i(z_l)}{p^i \prod_{l=0}^{k} f^i(z_l) + (1 - p^i) \prod_{l=0}^{k} g^i(z_l)}. \tag{4.1}$$

4.2.2 The Need for Randomized Policy

In order to have a well-defined mathematical formulation of the decentralized planning problem, we must resolve the question of whether the policies should be the same for all agents. The answer appears to be yes, in the sense that if the positions of any two agents were exchanged at any point during the search process we should rationally expect the search to continue as if the exchange had not been made at all. Without prior agreement or protocol as to the roles that each will play, the agents have no rational basis for developing anything but identical policies because all have identical capability, information, and intention. Yet it is clear that if the agents adopt identical deterministic policies the performance of the group on many problems may be quite poor.

Consider, as an extreme example, a problem in which the agents all start out at the same location. If each agent is limited to deterministic action choices, the agents will move as a body through the search area, and so collectively perform no better than any one of the agents could have done entirely on its own as multiple visits at the same time only produce one observation. Indeed, in general it is clear that any deterministic policy adopted by all the agents will produce a collective policy that would rank among the worst choices for a centralized planner. The situation for decentralized planning improves dramatically if we allow agents to adopt policies that involve a random choice of action. With such randomization, an agent can follow the same policy as the other agents and yet act differently than them, at least some of the time.

4.2.3 Decentralized Planning

Now we describe a framework in which randomization over individual maneuvers is allowed in a policy. Each agent makes a control decision on the basis of its current position (which limits where it can visit next) and the set of common information shared by all agents. The agent's current position at time k is denoted by $x(k)$. For each node $j \in V$, $y_j(k)$ is the number of agents at that node at time k, and $z_j(k) \in Z$ is the common observation made about node j at time k. Here, we will make a slight modification of the set Z by adding a NULL observation to the set. A NULL observation will be made at node j at time k if and only if $y_j(k) = 0$, *i.e.* no agent is visiting node j at the time. To simplify notation, let

$$\begin{aligned} y(k) &= (y_1(k), \ldots, y_v(k)), \\ z(k) &= (z_1(k), \ldots, z_v(k)). \end{aligned}$$

We can then refer to $y(k)$ as the location vector and $z(k)$ as the observation vector at period k. We write the system state as $\xi(k) = (x(k), \alpha(k))$, where $\alpha(k) = (y(k), z(0), \ldots, z(k))$ is the common information shared by all agents. Let p^j be the prior conditional probability for node j and $p^j(\alpha(k))$ be the posterior conditional probability that H_0^j is true given the current state $\alpha(k)$. Since the sequence of observations at each node i satisfies Assumption 3, we can simply apply Equation (3.1) to compute the posterior $p^j(\alpha(k))$:

$$p^j(\alpha(k)) = \frac{p^j \prod_{l=0}^k f^j(z_j(l))}{p^j \prod_{l=0}^k f^j(z_j(l)) + (1 - p^j) \prod_{l=0}^k g^j(z_j(l))}.$$

Let L_0^j and L_1^j be the cost incurred respectively when $\mathcal{H}_0^j$ and $\mathcal{H}_1^j$ are falsely accepted. We can think of L_0^j as the cost associated with false alarms and L_1^j the cost associated with misses. The total final Bayes risk of the final state $\alpha(N)$ is defined as follows:

$$R(\alpha(N)) = \sum_{j \in V} \min\{(1 - p^j(\alpha(N)))L_0^j, p^j(\alpha(N))L_1^j\},$$

where a classification decision at each node is made to minimize the Bayes risk at that node.

In order to allow randomization in the decision making, the control vector $\underline{u}(k)$ is defined as a v-dimensional probability vector where the jth component of $\underline{u}(k)$ corresponds to the probability that the agent visits node j next. Let $\mathcal{U}(\xi(k))$ be the set of all possible nodes the agent can travel to in the next time period given $\xi(k)$, *i.e.* $\mathcal{U}(\xi(k)) = \{j \in V | (x(k), j) \in E\}$. $\underline{u}(k)$ has the following properties:

1. If $j \notin \mathcal{U}(\xi(k))$, then $\underline{u}^j(k) = 0$,
2. For all $j \in \{1, \ldots, v\}$, $0 \leq \underline{u}^j(k) \leq 1$,
3. $\sum_{j \in \mathcal{U}(\xi(k))} \underline{u}^j(k) = 1$.

A mapping $\underline{\mu}_k$ from $S(k)$, the set of possible states in time k, is admissible if and only if $\underline{\mu}_k(\xi(k))$ satisfies the properties outlined above. A policy $\phi = \{\underline{\mu}_0, \ldots, \underline{\mu}_{N-1}\}$ is admissible if and only if $\underline{\mu}_k$ is admissible for all k.

In order to evaluate a particular admissible policy ϕ, it is important to recognize that the evaluation must be carried out under the assumption that all agents implement the policy ϕ. Under this assumption, $\alpha(k)$'s are random variables with distribution defined through the system transition equation

$$\alpha(k+1) = \mathcal{G}(\alpha(k), \underline{\mu}_k(\cdot, \alpha(k))). \tag{4.2}$$

To illustrate how the system behaves according to $\mathcal{G}$, first observe that $z_j(k+1)$ is distributed according to either f^j or g^j when $y_j(k+1) > 0$ and takes on the value

NULL with probability 1 otherwise. We will now focus exclusively on the location vector $y(k+1)$, and for this purpose it is best to look at the following simple example. Suppose that $y(k) = (1,0,0,1)$ and $\underline{\mu}_k(1,\alpha(k)) = \underline{\mu}_k(4,\alpha(k)) = (0,1/2,1/2,0)$. It is clear that $y(k+1)$ takes on the value $(0,1,1,0)$ with probability 1/2, $(0,2,0,0)$ with probability 1/4, and $(0,0,2,0)$ with probability 1/4.

Let $G_\phi(\alpha(N))$ denote the expected cost of policy $\phi = \{\underline{\mu}_0, \ldots, \underline{\mu}_{N-1}\}$ starting at initial state $\alpha(0)$.

$$G_\phi(\alpha(0)) = E[R(\alpha(N))], \tag{4.3}$$

where the system dynamic is described by Equation (4.2) and the expectation is taken over the random variables $\alpha(k)$. Here the randomness of $\alpha(k)$ results from the uncertainty in the observations as well as the uncertainty in the locations. The objective of a decentralized planning algorithm is to compute a policy that would minimize the expected cost as expressed in Equation (4.3).

4.3 Decentralized Planning Algorithms

In this section we introduce three decentralized planning heuristics for the Bayesian team search problem. All are designed to be run independently by all agents and serve to resolve complex situations where agents perceive equivalent but incompatible opportunities in conducting their search of the environment. We do not assume that agents have the ability to control or predict with certainty the actions of any other agent.

While the goal of this section is to develop *decentralized* planning heuristics, it is convenient to begin with a formal description of how the search planning problem would be formulated and solved from a centralized perspective. The centralized version of the planning problem differs from the decentralized problem in that a single decision maker has the authority to control all agents at each stage of the process.

Coordination failures cannot take place and only deterministic actions are needed. (When, from a given state of the process, the centralized planner identifies two or more equal-value action profiles for the agents, the planner will make an arbitrary selection and command all agents accordingly.)

4.3.1 Centralized Planning Problem As A Starting Point

The centralized version of the planning problem is an partially observed Markov decision process, as follows. The state of the system at time k is $\alpha(k) = (y(k), z(0), \ldots, z(k))$ where $y(k), z(0), \ldots, z(k)$ are defined in the same way as in the decentralized formulation. The set of all possible states for time period k is denoted by $A(k)$. Let $u(k)$, a vector of dimension v, be the control vector at time k. Then $u^j(k)$, jth component, is the number of agents at node j at time $k+1$. The agent location component of the system transition is deterministic in the sense that $y(k+1) = u(k)$ with probability 1. Let $U(\alpha(k))$ be the set of all possible control vector given the current system state $\alpha(k)$. A mapping μ_k from $A(k)$ to $\cup_{\alpha(k)\in A(k)} U(\alpha(k))$ is said to be admissible if $\mu_k^j(\alpha(k)) \in U(\alpha(k))$ for all $\alpha(k) \in A(k)$. A policy $\psi = \{\mu_0, \ldots, \mu_{N-1}\}$ is admissible if and only if μ_k is admissible for every $k = 0, 1, \ldots, N-1$.

Given an initial state $\alpha(0)$ and an admissible policy $\psi = \{\mu_0, \ldots, \mu_{N-1}\}$, the states $\alpha(k)$ are random variables with distribution defined through the system equation

$$\alpha(k+1) = \mathcal{F}(\alpha(k), \mu_k(\alpha(k))), \qquad k = 0, 1, \ldots, N-1. \tag{4.4}$$

The expected cost of ψ starting at $\alpha(0)$ is

$$F_\psi(\alpha(0)) = E[R(\alpha(N))]. \tag{4.5}$$

In Equation (4.5), the expectation is taken over the random variables $\alpha(k)$. Here the randomness stems from uncertainties associated with observations at time k alone.

Let Ψ be the set of all admissible policies, then an optimal policy ψ^* associated with a given initial state $\alpha(0)$ is one that minimizes the cost function F_ψ, i.e.

$$\psi^* = \arg\min_{\psi \in \Psi} F_\psi(\alpha(0)). \tag{4.6}$$

The optimal cost starting from state $\alpha(0)$ is therefore $F_{\psi^*}(\alpha(0))$. In fact, a policy that is optimal for each possible initial states can be found using standard dynamic programming methods.

Observe that one or more distinct joint individual maneuvers can lead to the same control vector u. This is precisely the situation we described in Example 1.3. The reason why we can simply specify the control vector as we do here is because the existing centralized decision maker is presumed to have the authority to force all agents to adopt the same joint individual maneuvers. For the same reason, agents are able to implement the same policy if multiple optimal policies exist as well.

This partially observed stochastic optimization problem can be solved by using the standard value iteration technique with the following dynamic programming recursion:

$$\begin{aligned} J_N(\alpha(N)) &= R(\alpha(N)), & (4.7)\\ J_k(\alpha(k)) &= \min_{u(k) \in U(\alpha(k))} E_{\alpha(k+1)}[J_{k+1}(\mathcal{F}(\alpha(k), u(k))] \quad \forall\, k = 0, \ldots, N-1. & (4.8) \end{aligned}$$

The expectation in Equation (4.8) is taken over the set of all possible $\alpha(k+1)$'s, however as we noted above the randomness of $\alpha(k+1)$ stems entirely from the randomness in observations $z(k+1)$. Therefore, the expectation here is really taken over the set of all possible observations $z(k+1)$ given the current posterior probabilities $p^i(\alpha(k))$ for all $i \in V$. The set of optimal controls given state $\alpha(k)$ and time k include all of the control vectors that achieve $J_k(\alpha(k))$, and we denote this set by $U_k^*(\alpha(k))$. The optimal policy obtained with this dynamic programming algorithm states that when the system state is $\alpha(k)$ at time k agents should implement the same centrally determined control vector from $U_k^*(\alpha(k))$. Furthermore, when a chosen control vector is achievable with multiple distinct joint individual maneuvers, the agents can successfully implement the chosen control vector by carrying out its individual maneuver accordingly. Function J is often referred to as the cost-to-go function. It represents the expected cost of following an optimal policy from this point on. For a given initial state $\alpha(0)$, $J_0(\alpha(0))$ is the unique optimal expected misclassification cost.

4.3.2 Decentralized Uniform Heuristic

It is clear that with individual labeling (*i.e.* each agent has its own internal representation of the world it perceives), each agent is capable of formulating the same centralized planning problem up to permutation of nodes and agents. Furthermore each agent is able to solve the centralized planning problem with the dynamic programming value iteration recursion in Equation (4.8). More specifically, each agent is capable of taking $\xi(k)$ and splitting it into $x(k)$ and $\alpha(k)$ and then finding a set $U_k^*(\alpha(k))$ that achieves the minimum in that recursion. What is problematic in the decentralized setting is the lack of means to guarantee that: 1) agents can agree on a single optimal action in $U_k^*(\alpha(k))$ when the optimal set has cardinality greater than one; 2) agents can agree on a single joint individual maneuver when a particular optimal action (number of agents at each location in the next period) can be achieved by multiple distinct joint individual maneuvers. When agents choose different optimal actions in $U_k^*(\alpha(k)$ or adhere to distinct joint individual maneuvers that accomplish the same future location vector, their collective behavior may very well be different from the optimal behavior prescribed by any optimal policy for the centralized planning problem. Consequently, the cost-to-go function value is no longer an accurate reflection of the cost of following an optimal policy. Therefore, additional care must be taken in order to successfully apply dynamic programming techniques in the decentralized setting.

Since every joint individual maneuver achieving a control vector in $U^*(\alpha(k))$ has the same cost-to-go, preferences based on that alone would dictate that each agent chooses each joint individual maneuver with equal probability. Without consistent labeling of the joint individual maneuvers (and control vectors) in $U^*(\alpha(k))$, perhaps this is the best the agents can achieve in the decentralized decision setting. We argue that this also adheres to the "no arbitrary decision" principle we operate under throughout this thesis. This suggests a uniform probability distribution over the set of optimal joint individual maneuvers in a policy. Given that the cost-to-go function values must reflect the risk and cost of coordination via this uniform randomization procedure, the cost-to-go function must be modified accordingly in the dynamic programming recursion.

Given $x(k)$, $\alpha(k)$ and u, the control vector containing the number of agents at each node in the next time period, one can find the set of joint individual maneuvers that achieve the control u. Toward this end, the agent labels itself as agent 1 and arbitrarily labels the others. With this labeling in mind, one can compute n-dimensional vectors such that 1) the ith element is agent i's individual maneuver (choice of node to visit next); 2) moves by all agents will together implement u. In the worst case, this can be done with enumeration of all possible joint individual maneuvers. We will denote this set with $M(x(k), \alpha(k), u)$. Define

$$M_k^*(x(k), \alpha(k)) = \cup_{u \in U_k^*(\xi(k))} M(x(k), \alpha(k), u). \qquad (4.9)$$

In other words, given appropriate cost-to-go functions J, $M_k^*(x(k), \alpha(k))$ contains all the joint individual maneuvers that achieve the minimum in the centralized dynamic programming recursion for time period k. Let $\mu_k^j(x(k), \alpha(k))$ be the probability of visiting node j next when the agent randomly selects a single joint individual maneuvers from $M_k^*(x(k), \alpha(k))$ with equal probability. It can be computed as

$$\mu_k^j(x(k), \alpha(k)) = \frac{\sum_{m \in M_k^*(x(k),\alpha(k))} 1_m^j}{|M_k^*(x(k), \alpha(k))|} \tag{4.10}$$

for every $k = 0, \ldots, N-1$. Here 1_m^j takes the value 1 if $m_1 = j$ and 0 otherwise. Let μ_k^j's be the basis forming $\phi = \{\underline{\mu}_0, \ldots, \underline{\mu}_{N-1}\}$, the policy produced by our first decentralized planning heuristic. This heuristic would compute a policy in which each agent would randomly choose any joint individual maneuver in $U_k^*(\alpha(k))$ with equal probability and implement the individual maneuver prescribed by that particular joint individual maneuver.

Together, Equations (4.9) and (4.10) transform a set of deterministic centralized control vectors into a single decentralized randomized control vector. We call this transformation the uniform randomization procedure. For notational simplicity, from now on we represent this transformation with the function $\mathbf{U}$. Decentralized Uniform Heuristic can be summarized in Algorithm 2.

Algorithm 2 Decentralized Uniform Heuristic

```
$\bar{J}_N^1(\alpha(N)) = R(\alpha(N))$.
For $k = N-1, \ldots, 0$,
    $\bar{U}_k^*(\alpha(k)) = \arg\min_{u(k)} E[\bar{J}_{k+1}(\mathcal{F}(\alpha(k), u(k))],$    (4.11)
    $\bar{\mu}_k(x(k), \alpha(k)) = \mathbf{U}(x(k), \bar{U}_k^*(\alpha(k))) \quad \forall\ x(k) \in V,$    (4.12)
    $\bar{J}_k(\alpha(k)) = E_{\alpha(k+1)}[\bar{J}_{k+1}(\mathcal{G}(\alpha(k), \bar{\mu}_k(\cdot, \alpha(k))))].$    (4.13)
```

In Equation (4.13), $\mathcal{G}(\alpha(k), \bar{\mu}_k(\cdot, \alpha(k))$ equals to the random variable $\alpha(k+1)$. The randomness of $\alpha(k+1)$ stems from the uncertainties of locations of agents and also the observations in the next time period. While Equation (4.11) finds $\bar{U}_k^*(\alpha(k))$, the uniform randomization procedure performed on this set may put positive probability on undesirable future states. The principle of optimality is therefore possibly violated when the minimum is achievable by multiple $u(k)$'s in Equation (4.11).

4.3.3 Decentralized Parallel Reduction Heuristic

Before introducing our next decentralized planning heuristic, we make the following observation about the Decentralized Uniform Heuristic. While Equation (4.11) finds $\bar{U}_k^*(\alpha(k))$ the set that minimizes the future cost-to-go, uniform randomization procedure performed on this set may put positive probability on undesirable future states. At the same time, there may exist a different set of control vectors $\bar{U}_k'(\alpha(k))$ where 1) individually every control vector in $\bar{U}_k'(\alpha(k))$ leads to a higher cost-to-go than every control vector in $\bar{U}_k^*(\alpha(k))$; 2) applying the uniform randomization procedure to $\bar{U}_k^*(\alpha(k))$ leads to a higher expected cost-to-go than applying the same operation to $\bar{U}_k'(\alpha(k))$. A sensible question to ask then is whether $\bar{U}_k'(\alpha(k))$ in this case makes a more attractive candidate than $\bar{U}_k^*(\alpha(k))$ as the set of control vectors our dynamic programming recursion should focus on.

To take this reasoning one step further, suppose now we have two different sets of control vectors $\bar{U}_k(\alpha(k)$ and $\bar{U}_k'(\alpha(k))$ such that 1) any control vector from the first set has a different cost-to-go than any control vector from the second set; 2) the future cost-to-go of applying uniform randomization procedures to both sets are equal. That is

$$E_{\alpha(k+1)}[\bar{J}_{k+1}(\mathcal{G}(\alpha(k), \bar{\mu}_k(\cdot, \alpha(k)))] = E_{\alpha(k+1)}[\bar{J}_{k+1}(\mathcal{G}(\alpha(k), \bar{\mu}_k'(\cdot, \alpha(k)))]$$

where $\bar{\mu}_k$ and $\bar{\mu}_k'$ are the policy resulting from applying uniform randomization procedure to $\bar{U}_k(\alpha(k))$ and $\bar{U}_k'(\alpha(k))$ respectively. While one could make the argument to favor one over the other based on metrics such as cardinality of the sets, a simple choice here is to implement each policy with equal probability.

Indeed, we can take inspiration from the Parallel Reduction Algorithm introduced in the previous chapter and incorporate a similar parallel reduction procedure into our dynamic programming recursion. While at this point we are not formally defining symmetry and equivalence as we did for symmetric games that satisfy Assumption 1, in practice we can partition the set of all feasible control vectors $U(\alpha(k))$ into peer sets based on future cost-to-go associated with them. Uniform randomization procedure is applied to each partition to compute the expected cost of choosing any joint individual maneuvers associated with the partition with equal probability. We call this cost the *partition value* to differentiate it from the common cost of the control vectors in the same partition. When there exists a partition with a unique minimum partition value, the control vector resulting from applying $\mathbf{U}$ to it is included in the optimal policy. Otherwise, partitions with a common value are aggregated to form a coarser partition. In each subsequent iteration i, let P be a i-th iteration partition. Rather than applying $\mathbf{U}$ to partition P directly, the control vector applied to P would choose to implement the control vector associated with $i-1$ iteration partitions that form P with equal probability. The resulting control vector - when parallel reduction satisfies its termination condition - is taken to be the optimal control. This heuristic is presented in Algorithm 3.

4.3.4 Decentralized Policy Iteration Heuristic

The final decentralized heuristic is inspired by dynamic programming policy iteration. It is policy iteration in the sense that we start with a policy and then iteratively obtain a new policy with an equal or improved cost. However, unlike most policy iteration procedures, we start with a carefully chosen policy instead of a randomly generated one. We also apply the uniform randomization procedure (denoted by $\mathbf{U}$, as before) as part of the policy iteration procedure. As a result, the same complex interactions that may cause the decentralized uniform heuristic to fail to satisfy the

Algorithm 3 Parallel Reduction Heuristic

```
$\hat{J}_N(\alpha(N)) = R(\alpha(N))$.
For $k = N-1, \ldots, 0$,
Initialization:
Partition $U(\alpha(k))$ into $\{P_{01}(\alpha(k)), \ldots, P_{0N_0}(\alpha(k))\}$ such that $u(k)$ and $u'(k)$ belong to the same partition if and only if $E[\hat{J}_{k+1}(\mathcal{F}(\alpha(k), u(k)))] = E[\hat{J}_{k+1}(\mathcal{F}(\alpha(k), u'(k)))]$.

    $\mu_{0i}(x(k), \alpha(k)) = \mathbf{U}(x(k), P_{0i}(\alpha(k)))$
    $v_{0i}(\alpha(k)) = E_{(\alpha(k+1))}[\hat{J}_{k+1}(\mathcal{G}(\alpha(k), \mu_{0i}(\cdot, \alpha(k))))]$

In the $m$-th iteration:

if there exists unique minimum $v_{mj}(\alpha(k))$ then

    $j^* = argmin\, v_{mj}(\alpha(k))$
    $\hat{J}_k(\alpha(k)) = v_{mj^*}(\alpha(k))$
    $\hat{\mu}_k(x(k), \alpha(k)) = \mu_{mj^*}(x(k), \alpha(k))$

else
    aggregate $P'_{mj}s$ to obtain a coarser partition $\{P_{m+1,1}(\alpha(k)), \ldots, P_{m+1,N_{m+1}}(\alpha(k))\}$ such that $P_{m+1,i}$ is an exhaustive union of $m$ stage partitions with like values.

    $\mu_{m+1,i}(x(k), \alpha(k)) = \sum_{j \text{ s.t. } P_{mj} \in P_{m+1,i}} \frac{\mu_{mj}(x(k), \alpha(k))}{\text{number of m level partitions in } P_{m+1,i}}$
    $v_{m+1,i}(\alpha(k)) = E_{\alpha(k+1)}[\hat{J}_{k+1}(\mathcal{G}(\alpha(k), \mu_{m+1,i}(\cdot, \alpha(k))))]$

end if
```

principle of optimality are in play here as well, and a worsening of the cost is possible with a newer iteration of the policy. We mitigate that risk by allowing the agent to revert back to the previous policy for the state if cost increases, thereby achieving a non-increasing sequence of costs overall.

Our policy iteration begins with solving the centralized planning problem with dynamic programming value iteration recursion. As we have previously noted the centralized version of the problem can be formulated and the dynamic programming value iteration can be successfully carried out by all agents. Uniform randomization procedure is applied to the set of centralized optimal control vectors to obtain the initial policy for the polity iteration. Decentralized Policy Iteration Heuristic is presented in Algorithm 4.

Decentralized Policy Iteration Heuristic always produces a sequence of policies with non-increasing costs. Additionally, if we say that the policy iteration converges if $H_k^i(\alpha(k)) = H_k^{i+1}(\alpha(k))$ for every feasible $\alpha(k)$ for all k, then we can show that this policy iteration converges after no more than $N+1$ iterations where N is the number of decision periods for the search problem. These results are stated and proven below. To simplify our notations, we write $H_k^i = H_k^j$ to mean that $H_k^i(\alpha(k)) = H_k^j(\alpha(k))$ for every feasible $\alpha(k)$ and $H_k^i \leq H_k^j$ to mean that $H_k^i(\alpha(k)) \leq H_k^j(\alpha(k))$ for every feasible $\alpha(k)$.

Proposition 8. *For $k = 0, \ldots, N$ and $i = 1, 2, \ldots,$ $H_k^{i+1} \leq H_k^i$.*

Proof. We will prove this proposition using induction. The base case where $k = N$ is true since $H_N^{i+1} = H_N^i = R$. Now assume that $H_{k+1}^{i+1} \leq H_{k+1}^i$. There are two cases to consider given any feasible state $\alpha(k)$. The first case is when $\tilde{H}_k^{i+1}(\alpha(k)) \leq H_k^i(\alpha(k))$ during the iteration to find H_k^{i+1}. In this case we know that $H_k^{i+1}(\alpha(k)) \leq H_k^i(\alpha(k))$. The second case is when $\tilde{H}_k^{i+1}(\alpha(k)) > H_k^i(\alpha(k))$. If this is true, $\mu_k^{i+1}(x(k), \alpha(k)) =$

Algorithm 4 Decentralized Policy Iteration Heuristic

```
**Iteration 1**
$H_N^1(\alpha(N)) = R(\alpha(N))$.
For $k = N-1, \ldots, 0$,

    $U_k^*(\alpha(k)) = \arg\min_{u(k)} E[J_{k+1}(\mathcal{F}(\alpha(k), u(k))]$.    (4.14)
    $\mu_k^1(x(k), \alpha(k)) = \mathbf{U}(x(k), U_k^*(\alpha(k)))$    $\forall\, x(k) \in V$,    (4.15)
    $H_k^1(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^1(\mathcal{G}(\alpha(k), \mu_k^1(\cdot, \alpha(k))))]$.    (4.16)

**Subsequent Iterations**
$H_N^i(\alpha(N)) = R(\alpha(N))$.
For $k = N-1, \ldots, 0$,

    $U_k^i(\alpha(k)) = \arg\min_{u(k)} E[H_{k+1}^i(\mathcal{F}(\alpha(k), u(k))]$.    (4.17)
    $\tilde{\mu}_k^{i+1}(x(k), \alpha(k)) = \mathbf{U}(x(k), U_k^i(\alpha(k)))$    $\forall\, x(k) \in V$    (4.18)
    $\tilde{H}_k^{i+1}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))))]$    (4.19)

**if** $\tilde{H}_k^{i+1}(\alpha(k)) \leq E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^i(\cdot, \alpha(k))))]$ **then**

    $\mu_k^{i+1}(\cdot, \alpha(k)) = \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))$
    $H_k^{i+1}(\alpha(k)) = \tilde{H}_k^{i+1}(\alpha(k))$

**else**

    $\mu_k^{i+1}(\cdot, \alpha(k)) = \mu_k^i(\cdot, \alpha(k))$    (4.20)
    $H_k^{i+1}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \mu_k^i(\cdot, \alpha(k))))]$    (4.21)

**end if**
```

$\mu_k^i(x(k), \alpha(k))\ \forall\ x(k)$. Since $H_{k+1}^{i+1} \leq H_{k+1}^i$,

$$E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \mu_k^i(\cdot, \alpha(k))))] \leq E_{\alpha(k+1)}[H_{k+1}^{i}(\mathcal{G}(\alpha(k), \mu_k^i(\cdot, \alpha(k))))]$$

Note that the left hand side is just $H_k^{i+1}(\alpha(k))$ as indicated in Equation (4.21). The right hand computes the expected cost of using control $\mu_k^i(\cdot, \alpha(k))$ with the future cost H_{k+1}^i, which is by definition $H_k^i(\alpha(k))$. Therefore, $H_k^{i+1}(\alpha(k)) \leq H_k^i(\alpha(k))$. Since $\alpha(k))$ can be any feasible state in period k, $H_k^{i+1} \leq H_k^i$. □

Proposition 9. *If $H_{k+1}^i = H_{k+1}^{i+1} = H_{k+1}^{i+2}$, then $H_k^{i+1} = H_k^{i+2}$ for all $i \geq 1$.*

Proof. Consider any feasible state $\alpha(k)$ for period k. The first step in computing either $H_k^{i+1}(\alpha(k))$ and $H_k^{i+2}(\alpha(k))$ is to find the vector $\tilde{\mu}_k^{i+1}(x(k), \alpha(k))$ and $\tilde{\mu}_k^{i+2}(x(k), \alpha(k))$ respectively for all $x(k)$. For any given $\alpha(k)$, the set of feasible $u(k)$'s and the future state associated with each $u(k)$ remain the same regardless of which iteration Equation (4.17) is employed in. Together with the assumption that $H_{k+1}^i = H_{k+1}^{i+1}$, we can conclude that $U_k^i(\alpha(k)) = U_k^{i+1}(\alpha(k))$. We then apply uniform randomization procedure to $U_k^i(\alpha(k))$ and $U_k^{i+1}(\alpha(k))$ to obtain $\tilde{\mu}_k^{i+1}(x(k), \alpha(k))$ and $\tilde{\mu}_k^{i+2}(x(k), \alpha(k))$ respectively. It is clear that since we are applying the same procedure to two identical sets of vectors, the resulting control vectors should be identical as well, *i.e.*

$$\tilde{\mu}_k^{i+1}(\cdot, \alpha(k)) = \tilde{\mu}_k^{i+2}(\cdot, \alpha(k)).$$

Using Equation (4.19), we compute costs $H_k^{i+1}(\alpha(k))$ and $H_k^{i+2}(\alpha(k))$ as follows,

$$\tilde{H}_k^{i+1}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))))],$$

$$\tilde{H}_k^{i+2}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+2}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+2}(\cdot, \alpha(k))))].$$

It immediately follows that

$$\tilde{H}_k^{i+1}(\alpha(k)) = \tilde{H}_k^{i+2}(\alpha(k)).$$

There are two cases to consider: 1) $\tilde{H}_k^{i+2}(\alpha(k)) \leq E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^i(\cdot, \alpha(k))))]$ and 2) $\tilde{H}_k^{i+2}(\alpha(k)) > E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^i(\cdot, \alpha(k))))]$.

Case 1: During iteration $i+1$, we obtain $\mu_k^{i+1}(x(k), \alpha(k)) = \tilde{\mu}_k^{i+1}(x(k), \alpha(k))$ and $H_k^{i+1}(\alpha(k)) = \tilde{H}_k^{i+1}(\alpha(k))$. Now consider iteration $i+2$. Since

$$\begin{aligned}
\tilde{H}_k^{i+2}(\alpha(k)) &= \tilde{H}_k^{i+1}(\alpha(k)) \\
&= H_k^{i+1}(\alpha(k)) \\
&= E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))))] \\
&= E_{\alpha(k+1)}[H_{k+1}^{i+2}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))))],
\end{aligned}$$

by definition of the heuristic iteration, $\mu_k^{i+2}(\cdot, \alpha(k)) = \tilde{\mu}_k^{i+2}(\cdot, \alpha(k))$ and $H_k^{i+2}(\alpha(k)) = \tilde{H}_k^{i+2}(\alpha(k))$. Since we've established that $\tilde{H}_k^{i+1}(\alpha(k)) = \tilde{H}_k^{i+2}(\alpha(k))$, it follows that $H_k^{i+1}(\alpha(k)) = H_k^{i+2}(\alpha(k))$.

Case 2: During iteration $i+1$, we must follow the else clause and therefore $\mu_k^{i+1}(\cdot, \alpha(k)) = \mu_k^i(\cdot, \alpha(k))$ and

$$\begin{aligned}
H_k^{i+1}(\alpha(k)) &= E_{\alpha(k+1)}[H_{k+1}^{i+1}(\mathcal{G}(\alpha(k), \mu_k^i(\cdot, \alpha(k))))] \\
&= E_{\alpha(k+1)}[H_{k+1}^{i}(\mathcal{G}(\alpha(k), \mu_k^i(\cdot, \alpha(k))))] \\
&= H_k^i(\alpha(k)).
\end{aligned}$$

Since $\tilde{H}_k^{i+1}(\alpha(k) > H_k^i(\alpha(k))$ and $\tilde{H}_k^{i+2}(\alpha(k) = \tilde{H}_k^{i+1}(\alpha(k))$, it must be the case that $\tilde{H}_k^{i+2}(\alpha(k) > H_k^{i+1}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+2}(\mathcal{G}(\alpha(k), \tilde{\mu}_k^{i+1}(\cdot, \alpha(k))))]$ during iteration $i+2$. Therefore,

$$H_k^{i+2}(\alpha(k)) = E_{\alpha(k+1)}[H_{k+1}^{i+2}(\mathcal{G}(\alpha(k), \mu_k^{i+1}(\cdot, \alpha(k))))].$$

Since we have established during iteration $i+1$ that $\mu_k^{i+1}(\cdot, \alpha(k)) = \mu_k^i(\cdot, \alpha(k))$, together with the assumption that $H_{k+1}^{i+1} = H_{k+1}^{i+2}$ we conclude that $H_k^{i+1}(\alpha(k)) = H_k^{i+2}(\alpha(k))$.

□

For period N, the H_N^i's are the same for all $i = 1, 2, \ldots$. From this, we can conclude that $H_{N-1}^2 = H_{N-1}^i$ for all $i \geq 3$. Similarly, we can conclude that $H_{N-2}^3 = H_{N-2}^i$ for all $i \geq 4$ and so on. Indeed, it is easy to show by induction that

$$H_{N-k}^{k+1} = H_{N-k}^i \qquad\qquad \forall\, i \geq k+2.$$

If we set $k = N$, we have

$$H_0^{N+1} = H_0^i \qquad\qquad \forall\, i \geq N+2.$$

Therefore, for every k H_k^i converges or stabilizes after at most $N+1$ iterations. Additionally, $\{H_k^i\}$ is a non-increasing sequence of cost functions for $k = 0, \ldots, N$ by Proposition 8.

4.4 Numerical Evaluation

To gain a practical understanding of the absolute and relative behavior of our proposed heuristics, we present computational results from several instances of the search problem. In these computational experiments, the team consists of two homogeneous agents. In addition, other than the artificial NULL observation associated with nodes that are not visited, there are two feasible observations: 0 and 1. $f^i(1) = 0.9$. $g^i(1) = 0.1$. The misclassification cost is set to be 100 for each false alarm and miss.

Note that in general the optimal policy for the centralized planning problem will always outperform any decentralized heuristic. This numerical evaluation serves to reveal what kind of performance gaps can be expected among the heuristics relative

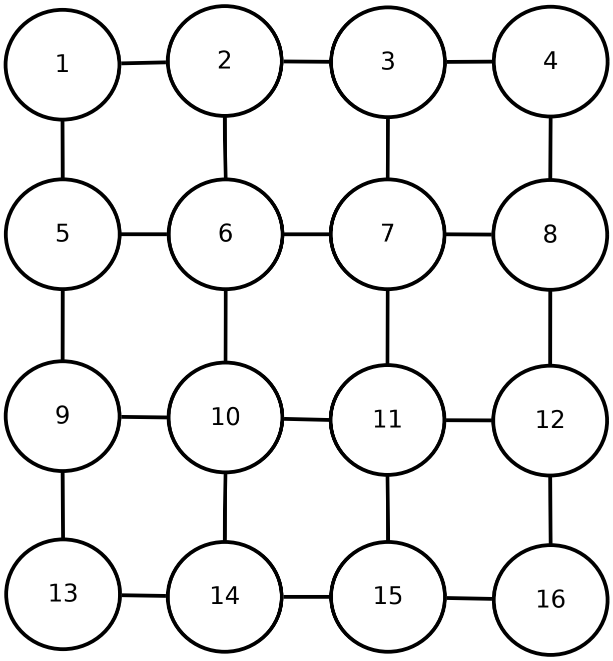


Figure 4.1: Network representing the search area

to the centralized optimal policy. Furthermore, even though we cannot say that one heuristic always dominates the others, we can assess their effectiveness in terms of achieving coordination with specially designed problems that exaggerate the cost of coordination failures.

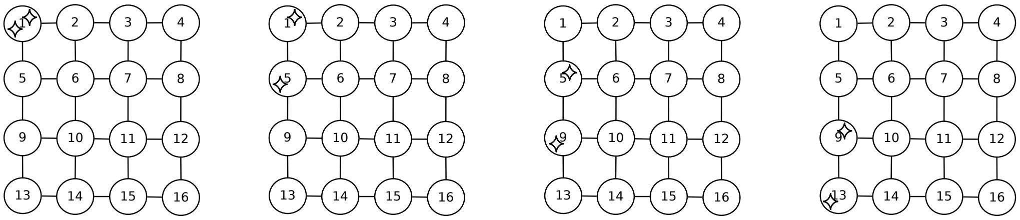
4.4.1 Case 1: Side by Side

Consider the case where the search area is represented by the network in Figure 4.1. Both agents are initially located at node 1. Node 13 and 14 have equal non-zero priors while all the other nodes have zero priors. N is set to be 4, *i.e.* there are 5 total time periods $t = 0, 1, \ldots, 4$. Note that when agents visit node 13 or 14 together only one meaningful observation is made. Because of the constraints given by the number of time periods, the most damaging coordination failures occur in the first time period.

The maximum number of observations at node 13 and 14 the two agents can manage within the time frame is 3. In a centralized planning problem, a policy that makes no observations at node 14 is always suboptimal. However, whether one or two observations are necessary at node 13 for centralized optimality depends on the first observation made at node 13. Regardless, it is generally desirable to have both agents reach nodes 13 and/or 14 as quickly as possible. Several types of optimal policies exist for the centralized planning problem. They are presented in Figures 4.2, 4.3, and 4.4. In the first type of optimal centralized policies, one agent stays at node 1 while the other agent visits node 5 in time period 1. They will travel down one in front of the other in the next two time periods. What's optimal next depends on the observation made at node 13. The second type of optimal centralized policies would have one agent visit node 2 and the other agent visit node 5 initially before both traveling downward. Again what is optimal depends on the first observation made at node 13. The final type of optimal centralized policies have both agents travel together to arrive at node 13 in time period 3. At that point, whether they try to make one or two additional observations depend on the observation made at node 13.

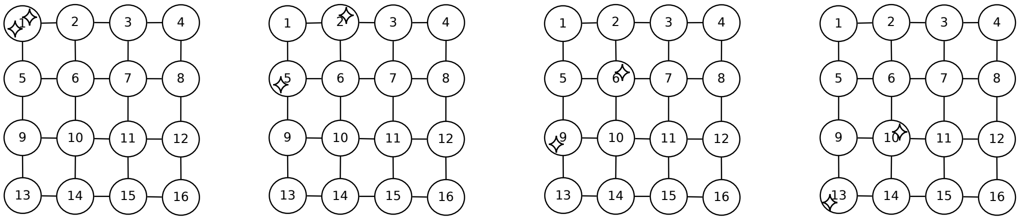
Expected total costs for the optimal policy for the centralized planning problem and policies resulting from the three decentralized heuristics are presented in Figure 4.5. Note that the expected cost functions are symmetric with respect to 0.5 prior probability, thus we only present the result up to 0.5 prior probability. While it is difficult to tell from Figure 4.5, with the chosen parameters Decentralized Parallel Reduction always produces a better policy than Decentralized Policy Iteration with these priors.

The policy produced by Decentralized Uniform Heuristic will generally try to implement the first type of optimal behavior above. However, there are two ways to achieve this initially. The policy resulting from the uniform randomization procedure has each agent stay at node 1 or visit node 5 with equal probability. The end result in time period 1 is as follows: both agents stay at node 1 with probability 1/4, one agent visits node 5 and another stays at node 1 with probability 1/2, and both agents



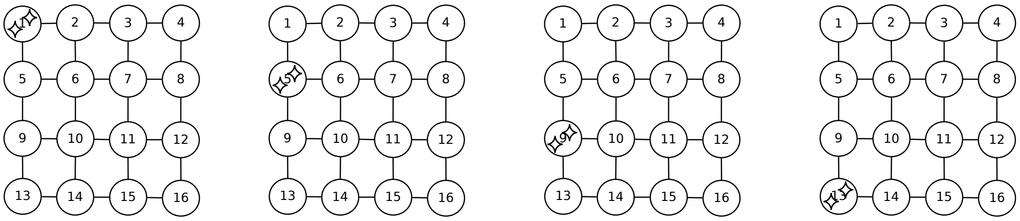
(a) t=0 (b) t=1 (c) t=2 (d) t=3

Figure 4.2: Type 1 Optimal Centralized Policy for the Side by Side Case



(a) t=0 (b) t=1 (c) t=2 (d) t=3

Figure 4.3: Type 2 Optimal Centralized Policy for the Side by Side Case



(a) t=0 (b) t=1 (c) t=2 (d) t=3

Figure 4.4: Type 3 Optimal Centralized Policy for the Side by Side Case

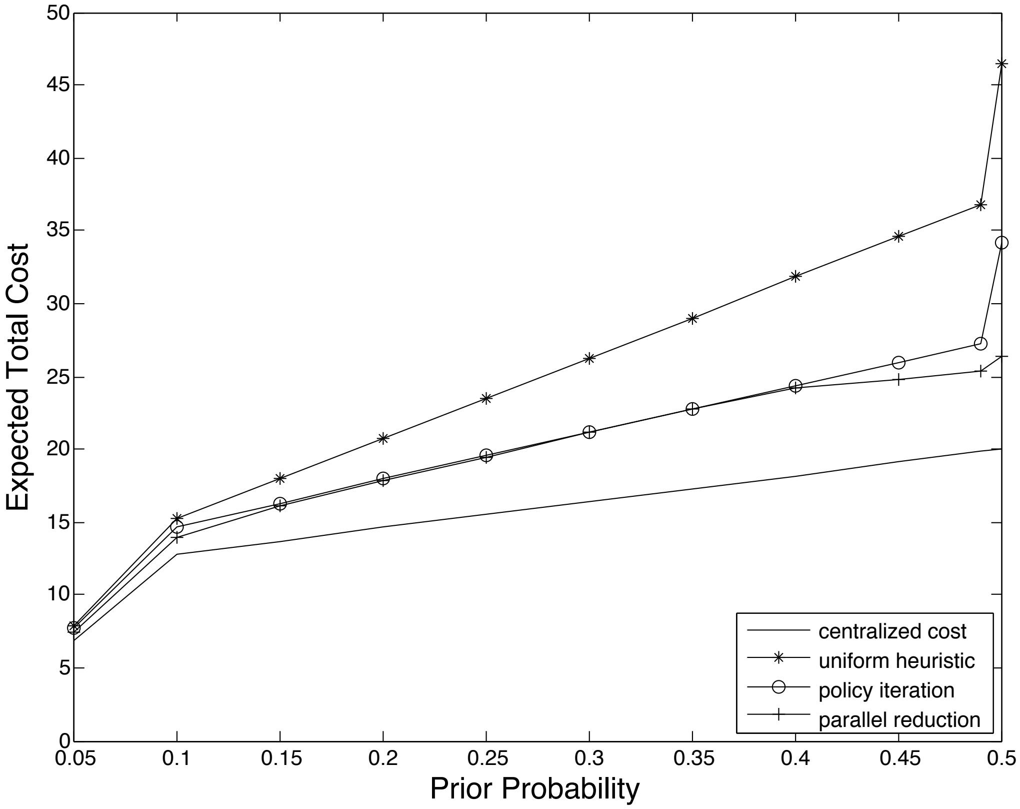


Figure 4.5: Expected total cost in the side by side case

reach node 5 with probability 1/4. When both agents stay at node 1 in time period 1, only a single observation at node 13 will be managed and no observation at node 14 is possible at that point.

The policy produced by Decentralized Policy Iteration directs each agent to stay with 1/5 probability, visit node 2 with probability 1/5 and visit node 5 with probability 3/5 in time period 1. We believe the effect is akin to applying uniform randomization procedure to all optimal control mentioned above for that initial period. However, the iterative procedure makes it difficult to state that with absolute certainty. Nonetheless, the probability distribution of agent locations in time period 1 is as follows: both agents at node 1 with probability 1/25, both agents at node 2 with probability 1/25, both agents at node 5 with probability 9/25, node 1 and node 2 with probability 2/25, node 1 and node 5 with probability 6/25, node 2 and 5 with probability 6/25. The least desirable outcomes are associated with co-locating at node 1 or 2 and probabilities of these events are much smaller compared to those resulting from Decentralized Uniform Heuristic.

Finally, the policy produced by Decentralized Parallel Reduction always sends both agents together to node 13 via nodes 5 and 9. While this policy avoids the initial coordination dilemma at the starting point, it does present a coordination dilemma when both agents arrive at node 13. However, failing to coordinate here has a far less detrimental effect on the final expected cost than failing to coordinate in the first time period.

4.4.2 Case 2: Diagonal

The second case we investigate is when node 4 and 13 have equal non-zero priors while all the other nodes have zero priors. As before the search area is represented by the network in Figure 4.1. Both agents are initially located at node 1. N is set to be 4, *i.e.* there are 5 total time periods $t = 0, 1, \ldots, 4$. The optimal policy to the centralized planning problem sends one agent to node 4 and one agent to node 13 in period 3. Depending on the observations made at node 4 and 13 at that time, it may be optimal to make additional observations at both of these nodes in the last time period. Or, it may be optimal to forgo the additional observation at one of these nodes. Regardless of what is optimal down the road, the optimal thing to do in the centralized setting is to always direct agents to split up and head toward non-zero prior nodes as soon as possible.

Policies resulting from Decentralized Uniform and Decentralized Policy Iteration both recognize the optimal policy in the centralized controlled scenario and attempt to send one agent to node 2 and the other agent to node 5 in time period 1. However, decentralized agents face a coordination dilemma because there are two distinct ways to achieve this. Both of these policies will uniformly randomize over the two ways to achieve centralized optimality and send each agent to node 2 and 5 with equal probability. The end result is with probability 1/2 agents will arrive at node 2 and 5 successfully in time period 1 but also with probability 1/2 both will end up at the same node in time period 1. When both of them arrive at node 2 in time period 1, no observation at node 13 is possible. Similarly, when both of them arrive at node 5 in time period 1, no observation at node 4 is possible. Both are always suboptimal.

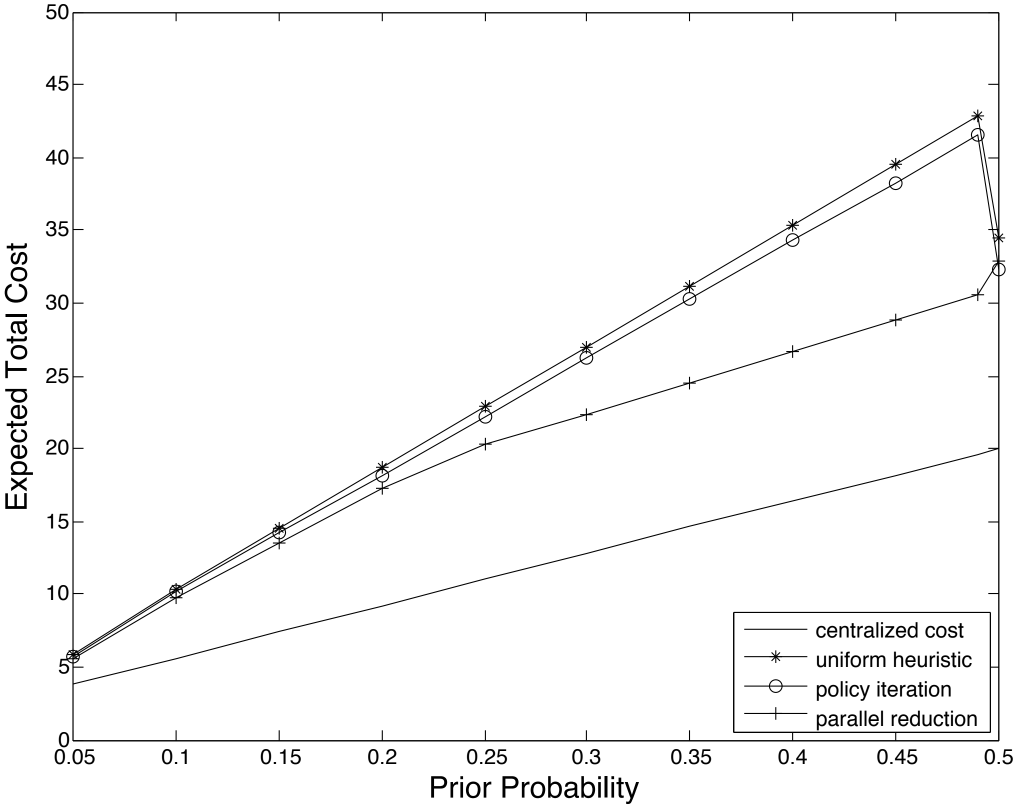


Figure 4.6: Expected total cost in the diagonal case

Decentralized Parallel Reduction for the most part produces a policy with very different behaviors. Each agent stays at node 1 with 1/2 probability and visits node 2 and 5 with 1/4 probability. The probability distribution for agent locations in period 1 is therefore: both agents stay at node 1 with probability 1/4, both agents at node 2 with probability 1/16, both agents at node 5 with probability 1/16, agents at node 1 and 5 with probability 1/4, agents at node 1 and 2 with probability 1/4, and finally agents at node 2 and 5 with probability 1/8. When agents are at node 1 and 2 in time period 1, the agent at node 1 will continue to node 4 and the agent at node 1 will visit node 13 according to the policy. Therefore, while they may not manage to make 4 observations when that is optimal, they are able to make 3 observations when that is optimal. Similar analysis holds true for when agents are at node 1 and 5 in time period 1. The least desirable (and also clearly always suboptimal) outcomes are when agents both reach node 2 or node 5. The probability of these events are much lower compared to those in the other decentralized policies.

Expected total costs for the optimal policy for the centralized planning problem and policies produced by three decentralized heuristics are presented in Figure 4.6. Policies from Decentralized Uniform and Decentralized Policy Iteration heuristics offer almost identical expected costs. This is not surprising as the policies produced by both are similar. Parallel Reduction produces a much better expected cost because coordination dilemmas are handled much more effectively in the initial decision period.

Chapter 5

Summary and Future Work

Decentralized control of a team of cooperative agents has received much attention in the past two decades. Much of the previous work in this area focuses on devising centrally computed policies that can be implemented in a distributed fashion to optimize team performance. In this thesis, we study decentralized decision problems each agent must formulate and solve in order to compute its own policy independently. Single agent sequential decision problems are typically formulated and solved as a MDP or a POMDP. Cooperative multiagent sequential decision problems can be formulated as a multiagent extension of MDP or POMDP, or as an identical interest stochastic game. Extending traditional solution techniques such as value iteration or policy iteration to these multiagent extensions is challenging because multiple optimal joint actions may exist and it is not clear how decentralized agents can coordinate on the same optimal joint action.

Solving cooperative decentralized decision problems often involves solving stage games that are identical interest strategic games. Coordination dilemmas arise when multiple pareto-dominant Nash equilibria exist. Solving these stage games thus reduces to an equilibrium selection problem. We propose the *natural solution*, a new solution concept, for a class of symmetric identical interest games in which a symmetric Nash equilibrium is uniquely determined by putting positive probability on a subset of actions and zero probability on all other actions. Therefore, selecting a Nash equilibrium in these games is reduced to selecting a subset of actions. We formally define equivalent actions and argue that the only rational thing to do is to treat all equivalent actions exactly the same. What this entails when we select a subset of actions is that we include all or none of the equivalent actions. We build the concept of atomic, proper and natural action groups upon this principle. There is no remaining ambiguity over how to achieve the expected payoff associated with the unique symmetric Nash equilibrium whose support is a natural action group. We define the natural solution to be the natural group whose uniquely determined symmetric Nash equilibrium offers the highest expected payoff. The natural solution's guaranteed existence and uniqueness means that it can be used as a equilibrium selection rule. Finally we show that static agreement games satisfy our assumptions, and therefore a natural solution is guaranteed to exist. We develop a linear time heuristic called Parallel Reduction Algorithm (PRA) for finding natural solutions. It is guaranteed to compute the natural solution when the heuristic terminates within 2 stages or with full support of the whole action set. While PRA is not guaranteed to always find the natural solution, empirical data shows that it does so with overwhelming likelihood in static agreement games. In the few instances where it terminates with an action group other than the natural solution of the game, the unique Nash equilibrium associated with the group is strictly better than the natural solution Nash equilibrium.

Static agreement games (pure coordination games) are an important class of games where existing equilibrium selection rules generally fail to produce a unique Nash equilibrium. Using the natural solution concept, a Nash equilibrium that assumes no arbitrary decisions among equivalent choices is guaranteed. While Nash equilibrium for strategic games is expensive to compute in general, finding a pure strategy Nash equilibrium for static agreement games (and indeed identical interest games in general) is much easier. Unfortunately, computing the natural solution for static agreement games is conjectured to be NEXP-complete. We suggest that the linear time heuristic PRA can be used as an equilibrium selection tool in practice since 1) it finds the natural solution fairly consistently; 2) when it fails to produce the natural solution it still produces a Nash equilibrium with better value than the natural solution Nash equilibrium. Therefore, if we were to use the concept of natural solution in a stochastic game whose stage games are static agreement games, we believe the result from PRA provides a reasonable approximation. In general, stage games are not going to be static agreement games. In order to apply the concept of natural solution to stochastic games in general, we may have to relax our assumptions while maintaining the qualities that ensure the existence and uniqueness of a natural solution.

The second part of the thesis is devoted to a decentralized planning problem for team Bayesian search. A team of agents is tasked with making observations in a search area where an unknown number of targets exist. Each agent must make its own individual decision about where its next observation will be. The common cost every agent tries to minimize is set to be the total final Bayes risk, given the observations made by all the agents. Each agent must formulate and solve a decentralized planning problem to compute its future actions. This planning problem is formulated as a POMDP whose objective function is evaluated based on the assumption that all agents will use the same mixed strategy policy. We propose three dynamic programming heuristics for this planning problem. Each of them can be used by agents in a decentralized fashion to compute an individual policy. The heuristics are designed such that all will arrive at the same policy so long as they use the same heuristics. We evaluated the performance of policies resulting from these heuristics using two instances of the planning problem where resolving coordination dilemmas is critical. The first two heuristics can be viewed as distributed dynamic programming value iteration while the third heuristic can be viewed as distributed dynamic programming policy iteration. Policies resulting from these heuristics exhibit distinct behaviors when coordination dilemmas arise. In both instances, the policy produced by Decentralized Uniform Heuristic fares the worst and the policy produced by Parallel Reduction Heuristic performs the best. Decentralized Policy Iteration heuristic performs similarly to the uniform heuristic in one instance and the parallel reduction heuristic in the other. Since each round of the policy iteration involves recalculating the cost-to-go function for each state, Parallel Reduction Heuristic is significantly more efficient in terms of computational complexity. As with most POMDPs, the curse of dimensionality means that we can only realistically solve problems of limited size. In the future, we plan to explore approximating methods such as limited lookahead for the decentralized planning problem.

The partially observed nature of the team search problem limits the size of problems our heuristics can be applied to in empirical evaluation. Furthermore, it does not allow the clearest differentiation among the resulting decentralized policies. We believe our heuristics can be easily extended to the general class of MDP or identical interest stochastic games. Our aim is to find problems that can serve as benchmark problems, not only for the proposed heuristics but also for future research in decentralized decision making in general.

Our heuristics also suggest methods of applying dynamic programming value iteration and policy iteration in the decentralized setting. We would like to extend this to decentralized learning algorithms that face similar coordination dilemmas due to multiple Nash equilibria. More specifically, we are interested in using uniform randomization procedure in the development of joint action reinforcement learning algorithms for MMDP or identical interest stochastic games.

Bibliography

[1] G. Weiss. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, 1999.

[2] J. Doran, S. Franklin, N. Jennings, and T. Norman. On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):309–314, 1997.

[3] M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand. Cooperative control for multiple autonomous UAV's searching for targets. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2823–2828, 2002.

[4] R. Beard and T. McLain. Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 25–30, 2003.

[5] L. Barri, P. Flocchini, P. Fraigniaud, and N. Santoro. Capture of an intruder by mobile agents. In *Proceedings of the Fourteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 200–209, 2002.

[6] L. Kaelbling, M. Littmann, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[7] P. Xuan, V. Lesser, and S. Zilberstein. Communication decisions in multi-agent cooperation: models and experiments. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 616–623, 2001.

[8] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000.

[9] Alessandra Russo Luke Dickens, Krysia Broda. Modelling MAS as finite analytic stochastic processes. In *Proceedings of the AISB Symposium on Behaviour Regulation in Multi-Agent Systems*, 2008.

[10] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195–210, 1996.

[11] David Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[12] Piotr J. Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:24–49, 2005.

[13] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27:819–840, 2002.

[14] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 133–139, 2005.

[15] Praveen Paruchuri, Milind Tambe, Fernando Ordez, and Sarit Kraus. Security in multiagent systems by policy randomization. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006.

[16] Frans Oliehoek, Matthijs Spaan, and Nikos Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[17] Byung Kon Kang and Kee-Eung Kim. Exploiting symmetries for single and multi-agent partially observable stochastic domains. *Artificial Intelligence*, 182:32–57, 2012.

[18] L. Shapley. Stochastic games. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 1095–1100, 1953.

[19] Frank Thusijsman. Optimality and equilibria in stochastic games. *Centrum voor Wiskunde en Informatica*, 1992.

[20] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[21] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 709–715, 2004.

[22] Akshat Kumar and Shlomo Zilberstein. Dynamic programming approximations for partially observable stochastic games. In *Proceedings of the Twenty-Second International FLAIRS Conference*, pages 547–552, 2009.

[23] Rosemary Emery-Montemerlo, Geoffrey Gordon, Jeff Schneider, and Sebastian Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 136–143, 2004.

[24] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175:487–511, 2011.

[25] C.J.C.H. Watkins and P. Dyan. Q-learning. *Machine Learning*, 8(3/4):279–292, 1992.

[26] Richard S. Sutton. Learning to predict by the methods of temporal differences. In *Machine Learning*, pages 9–44. Kluwer Academic Publishers, 1988.

[27] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38:156–172, 2008.

[28] Satinder Singh, Tommi Jaakkola, Michael Littman, and Csaba Szepesvri. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38:287–308, 2000.

[29] Junling Hu and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[30] Michael Littman. Friend-or-Foe Q-learning in general-sum games. In *Proceeding of the Eighteenth International Conference on Machine Learning*, pages 322–328, 2001.

[31] Amy Greenwald and Keith Hall. Correlated-Q learning. In *Proceedings of AAAI Spring Symposium*, pages 242–249, 2003.

[32] Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Advances in Neural Information Processing Systems*, pages 1571–1578, 2002.

[33] H. Peyton Young. The evolution of conventions. *Econometrica*, 61:57–84, 1993.

[34] Ronen Brafman and Moshe Tennenholtz. Learning to coordinate efficiently: A model-based approach. *Journal of Artificial Intelligence Research*, 19:11–23, 2003.

[35] Avraham Bab and Ronen I. Brafman. Multi-agent reinforcement learning in common interest and fixed sum stochastic games: An experimental study. *Journal of Machine Learning Research*, 9:2635–2675, 2008.

[36] Thomas Schelling. *The Strategy of Conflict*. Harvard University Press, Cambridge, 1960.

[37] R. Duncan Luce and Howard Raiffa. *Games and Decisions: Introduction and Critical Survey.* Wiley, New York, 1957.

[38] David Lewis. *Convention: A Philosophical Study.* Harvard University Press, Cambridge, 1969.

[39] D. Gauthier. Coordination. *Dialogue*, 14:195–221, 1975.

[40] M. Gilbert. Rationality and salience. *Philosophical Studies*, 57:61–77, 1989.

[41] Natalie Gold and Robert Sugden. Collective intentions and team agency. *The Journal of Philosophy*, 104:109–137, 2007.

[42] Michael Bacharach. *Beyond Individual Choices: Teams and Frames in Game Theory.* Princeton University Press, Princeton, 2006.

[43] John Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36:48–49, 1950.

[44] J. Harsanyi and R. Selton. *A General Theory of Equilibrium Selection in Games.* MIT Press, Cambridge, 1988.

[45] G. W. Brown. *Some notes on computation of games solutions.* The RAND Corporation, 1949.

[46] D. Foster and H. Young. On the nonconvergence of fictitious play in coordination games. *Games and Economic Behavior*, 25:79–96, 1998.

[47] A. Sela and D. Herreiner. Fictitious play in coordination games. *International Journal of Game Theory*, 28:189–197, 1999.

[48] Johannes Becker and Damian Damianov. On the existence of symmetric mixed strategy equilibria. *Economics Letters*, 90:84–87, January 2006.

[49] Shih-Fen Cheng, Daniel M. Reeves, Vorobeychik Yevgeniy, and Michael P. Wellman. Notes on equilibria in symmetric games. In *Proceedings of the Sixth Workshop on Game Theoretic and Decision Theoretic Agents at the Third Conference on Autonomous Agents and Multi-Agent Systems*, pages 23–28, 2004.

[50] Nicholas Bardsley, Judith Mehta, Chris Starmer, and Robert Sugden. Explaining focal points: Cognitive hierarchy theory versus team reasoning. *The Economic Journal*, 120:40–79, 2010.

[51] L. Stone. *Theory of Optimal Search.* Academic Press, 1975.

[52] B. O. Koopman. *Search and Screening: General Principles with Historical Applications.* Pergamon Press, 1980.

[53] D. Castanon. Optimal search strategies in dynamic hypothesis testing. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, pages 265–270, 1993.