

9-13-2004

Applications of Decision and Utility Theory in Multi-Agent Systems

Xin Li

University of Nebraska, xinli@cse.unl.edu

Leen-Kiat Soh

University of Nebraska, lsoh2@unl.edu

Follow this and additional works at: <http://digitalcommons.unl.edu/csetechreports>



Part of the [Computer Sciences Commons](#)

Li, Xin and Soh, Leen-Kiat, "Applications of Decision and Utility Theory in Multi-Agent Systems" (2004). *CSE Technical reports*. 85.
<http://digitalcommons.unl.edu/csetechreports/85>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Technical reports by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Applications of Decision and Utility Theory in Multi-Agent Systems

Xin Li, Leen-Kiat Soh

Department of Computer Science and Engineering

University of Nebraska–Lincoln

{xinli, lksoh}@cse.unl.edu

September 2004

Abstract

This report reviews the applications of decision-related theories (decision theory, utility theory, probability theory, and game theory) in various aspects of multi-agent systems. In recent years, multi-agent systems (MASs) have become a highly active research area as multi-agent systems have a wide range of applications. However, most of real-world environments are very complex and of uncertainty. An agent's knowledge about the world is rather incomplete and uncertain. The actions of the agent are non-deterministic with a range of possible outcomes. The agent may have many desires that conflict each other. The agent also needs to know about other agents and decide how to interact with others. These aspects may be handled by the application of techniques provided by decision-related theories. In this report, the mechanisms of decision-related theories are introduced especially a series of typical concepts and methodologies. The decision problems existing in multi-agent systems that can be handled by decision-related theories are discussed from different aspects. A variety of applications of decision-related theories in multi-agent systems are presented especially the application of the series of typical concepts and methodologies.

Keywords: multi-agent systems, decision theory, probability theory, utility theory, game theory

1. Introduction

In recent years, multi-agent systems (MASs) have become a highly active research area as multi-agent systems have a wide range of applications involving industrial manufacturing, traffic and transportation, electronic commerce, information management, exploration, entertainment, and others [Weiss 1999]. In a multi-agent system, there are multiple agents who operate in a specific environment and can interact with some others. An agent is a computational entity (or a computer system) that situates in some environment to pursue some set of goals or perform some set of tasks, and that is autonomous in that its behavior at least partially depends on its own experience rather than being merely the result of the intervention of humans or other entities [Wooldridge 1999]. Its function is to interact with its environment, perceive the state of the environment, and make decisions about how to respond to it [Parsons et al. 2002]. For the individual agents in a multi-agent system, their function is not only to interact with the environment but also to interact with other agents, perceive the states of other agents, and make decisions on how to respond to other agents' actions.

As a modern approach to distributed artificial intelligence (DAI), one of the long-term goals of multi-agent systems is to develop mechanisms and methods that enable agents to understand and interact with other entities in the system as well as humans (or even better). This goal is centered around agents' decision making about when and how to interact with whom for pursuing specific goals or performing specific tasks [Weiss 1999]. Thus when a multi-agent system is designed, an important objective is to ensure that agents make right and good decisions like humans, typically the best decision that they can do given what is known. Therefore, decision making, in some degree, is at the very heart of building multi-agent systems [Parsons et al. 2002].

In simple environments such as blocks-world scenarios studied in early work on artificial intelligence [Gupta and Nau 1992, Parsons et al. 2002], making right decisions is relatively easy. The status change of the environment is certain and an agent's knowledge (or beliefs) about the environment is complete and correct. An agent has a set of desires and a set of actions each of which has a single possible outcome that is deterministic. It tries to achieve a single goal and there are no other agents disrupting it as there is only one agent. As a result, all of what the agent needs to do is to figure out a plan (i.e., a sequence of actions) that will take it from the current

known position to the specified goal position. Simply executing those actions in sequence will transform the initial state to the goal state and definitely lead to the goal being achieved.

However, most of real-world environments are more complex than blocks-world scenarios. The real-world environment generally changes dynamically, uncertainly, and even is noisy. In such a complex environment, the initial states that prompt the agents' decision making process in the first place may dynamically change while the decision making process is still going on. An agent may not know the properties of the environment or other agents with certainty. The actions of an agent are non-deterministic with a range of possible outcomes. The outcome of an agent's performing an action might be influenced by other agents' behavior so different from the expected. In a noisy environment, an agent's knowledge about the world, which is acquired by the agent through sensors, may not be described accurately and is rather incomplete, uncertain, and even incorrect. An agent may hold many desires that conflict each other. In addition, in a multi-agent system, there are multiple agents operating in a same environment and they might have to interact with each other to exchange information and coordinate their behavior. These agents' goals may conflict and the outcome of an agent's action may be influenced by other agents. Thus an agent needs to know about other agents and decide how to work together with others, which makes the decision making process of an agent in a multi-agent system more complex than in a single agent environment.

Decision theory concerns the use of reason in human decision making and can be used to analyze which options should be taken when it is uncertain exactly what the outcome of taking the option will be [Lee 1971, Raiffa 1968]. Utility theory rests on decision making and it concerns the use of profit or cost as the reason upon which the decision is to be based [von Neumann and Morgenstern 1947]. Both theories provide the analytical method for decision making. Now it is widely believed that the crucial issue in designing autonomous agents is how to provide these agents with the ability to select the best action from a range of possible actions. To enable agents, the computational entities situated in complex environments, to work like humans, the techniques from decision and utility theory can be applied to handle the decision making issues in multi-agent systems to some degree. In addition, game theory [von Neumann and Morgenstern 1947], a close relative of decision theory, studies the interaction strategy between entities and can be applied to help autonomous agents make decisions during *interaction* since in multi-agent

systems, the issue of designing interaction strategies and mechanisms is very important. In fact, there are many concepts and tools in these theories used in multi-agent systems. These concepts and tools include probability distribution, multi-attribute utility functions, expected utility functions, decision trees, Bayesian networks, influence diagrams, Markov decision processes (MDPs), partially observable Markov decision processes (POMDPs), Nash equilibria, Pareto equilibria, and so forth (see Section 2.5).

Although there have been numerous applications of decision-related theories in multi-agent systems (e.g., [Gmytrasiewicz and Lisetti 2002], [Banerje and Sen 2002], [Bazzan et al. 2002], [Excelente-Toledo and Jennings 2003], [Stone and Veloso 1998], [Vassileva and Mudgal 2002], [Nair et al. 2004b]), no review work has yet been conducted to explore the theoretic background of the applications and build up a close relationship between theories and applications. This report summarizes the applications of decision theory, utility theory, and other decision-related theories in multi-agent systems. We take a thorough exploration into this area through presenting related issues in decision theories with multi-agent systems, discussing decision-theoretic requirement in multi-agent systems, and describing some typical applications.

The rest of this report is organized as follows. Section 2 introduces decision-related theories, and presents a series of typical theoretic concepts and methodologies that may be applied into multi-agent systems. Section 3 lists some decision problems existing in multi-agent systems that may be handled in decision-related theories. Section 4 presents a variety of applications of decision-related theories in multi-agent systems. Section 5 concludes the report.

2. Decision-Related Theories

A *decision* is an allocation of resources under control of the decision maker [Horvitz et al. 1988]. Decision theory applies mathematical and statistical methodologies to help provide information on which decisions can be made. It is based on the axioms of *probability* and *utility*. To some degree, it is a combination of probability theory and utility theory. Game theory is tightly relevant to decision theory and to a certain degree decision theory can be regarded as the study on the special case of game theory (see Section 2.4). Taking into account the roles of these theories in decision making, we call them together as decision-related theories. In this section we

present the basic mechanisms of these decision-related theories, and also describe some typical concepts and methodologies provided in these theories.

2.1. Decision Theory

Decision theory is a body of knowledge and related analytical techniques of different degrees of formality designed to help a decision maker choose among a set of alternatives in light of their possible outcomes [White 1969, Lee 1971]. It deals with the issues involved with decision making and concentrates on identifying the “best” decision option. A best decision option generally is one that maximizes the expected benefit to the decision maker.

2.1.1. History of Decision Theory

The history of decision making originated from the emergence of animals. But the formal decision theory can be said to start with the 1938 Battle of Britain during World War II (W.W.II). The English War Department banded together to a group of physicists, mathematicians, logic experts, crossword puzzle experts, and chess masters to solve the problem of locating positions for a new but limited technology called “radar”. This group of professionals were successful enough to be retained thereafter to solve an increasingly more diverse number of logistic and allocation problems during W.W.II. After the war, the British Government continued to use this group that they called the “Operational Research” group. Other governments and industry also saw the advantages of using this type of professionals in improving their operations. From then on, the decision theory and its applications developed very rapidly [White 1969].

2.1.2. Elements in Decision Problems

There are three primary elements in all decision theory problems: alternatives, states of nature, and payoffs [White 1969].

Alternatives (also called “choices”, “actions”, or “courses of actions”) are the independent decision variables. They represent the alternative action choices that decision makers choose from. A decision making problem is either a pure choice problem when only one alternative is allowed to be selected, or a mixed choice problem when portions of several alternatives can be selected at one time. For example, suppose a girl wants to spend a good weekend somewhere and

she has three choices: to the beach, to the zoo, and at home. If she can choose only one place from the three alternatives, it is a pure choice problem. But if she would like to spend different portions of time at two places, the problem transforms to a mixed choice problem because she can select two places at one time from all the three, for examples, to the beach in the morning and to the zoo in the afternoon. Obviously, the pure choice problem is a special case of the mixed choice problem.

States of nature (also called “states of the world”) are independent events that are assumed to occur in the future. The occurrence of these events is uncontrollable by the decision maker. In the weekend example, the weather states (sunny, cloudy, rainy, etc.) can be considered as states of nature. The girl can choose places from alternatives based on the states of nature but she cannot control which event will occur in the future.

Payoffs (also called “outcomes” as the results of “actions”) are dependent parameters that are assumed to occur given a particular alternative is selected and a particular state of nature occurs. Payoff values may be in terms of profit or cost. In the weekend example, payoff is the happy degree. There are many possible payoffs corresponding to different combinations of alternatives and states of nature. For example, generally, the payoff of going to beach is higher than staying at home in the sunny weather yet lower than staying at home in the rainy weather or lower than going to zoo in the cloudy weather.

In complex decision problems, how to select a specific action or a specific course of actions from multiple alternatives is complex since the payoff of the actions may be not obvious and even not foreseeable at the point of decision making. So in most cases the elements of a decision problem is not limited within the above three. The *strategies* that conduct the selection of alternatives are also an important element in decision problems [Raiffa 1968, White 1969], which can be regarded as the extension of the “alternatives” element.

2.1.3. Types of Decision Environments

Decision theory can apply to three primary types of environments: under certainty, under risk, and under uncertainty.

Decision under *certainty* means that each alternative leads to one and only one outcome, and a choice among alternatives is equivalent to a choice among outcomes. Under this environment the decision maker knows clearly what the alternatives are to choose from and knows clearly the payoffs that each choice will bring with certainty if the alternative is chosen. In the weekend example, if the girl definitely knows the weather status, she can decide which place to go in a certain manner to maximize her happiness.

In decision under *risk*, each alternative will have one of several possible outcomes, and the probability of occurrence for each outcome is known. Therefore, each alternative is associated with a probability distribution, and a choice among probability distributions. Under this environment some information on the states of nature and corresponding payoffs is available but is presented in a probabilistic fashion. In the weekend example, if the girl knows the occurrence probabilities of different weather states, she can make a decision under risk.

Decision under *uncertainty* occurs when the probability distributions are unknown. Under this environment no information about the likelihood of states of nature occurring is available. The decision maker can only assume that a particular payoff will occur if a given state of nature occurs. In the weekend example, if the girl has no any idea of the weather, the payoffs of alternatives might be estimated, but are only assumed to occur in an uncertain environment. They are not known with any degree of certainty.

The three environments are on a linear continuum ranging from complete knowledge (under certainty), to partial knowledge (under risk), and finally to no knowledge (under uncertainty). Indeed, the uncertainty cases can be reduced to the risk cases by assigning an equal probability to each state of nature, or using subjective probabilities based on expert assessments or on analysis of previous decisions made in similar circumstances. Hence, in the following sections, we use a unified term, uncertainty, to refer to both uncertainty cases and risk cases.

2.2. Probability Theory

The foundations of probability theory extend at least as far back as the seventeenth century in the works of Pascal, Bernoulli, and Fermat [Apostol 1969]. *Probability theory* concerns the analysis of random events. The outcome of a random event cannot be determined before it occurs, and it may be any one of several possible outcomes. The actual outcome is determined by a probability

[Wendt and Vlek 1975]. *Probability* is the numerical expression of the possibility of event occurrence, measurable in an uncertain situation. The notion for the probability of the occurrence of an event X conditioned on a state of information S may be specified as $P(X | S)$. A classical axiomatization of probability contains the following definitions where Y is also an event like X :

$$0 \leq P(X | S) \leq 1$$

$$P(X | S) + P(\text{not } X | S) = 1$$

$$P(X \text{ or } Y | S) = P(X | S) + P(Y | S) - P(X \text{ and } Y | S)$$

$$P(X \text{ and } Y | S) = P(X | Y, S) * P(Y | S)$$

Probability theory and the more encompassing decision theory provide principles for rational inference and decision making under uncertainty [Horvitz et al. 1988]. Probability provides a language for making statements about uncertainty and thus makes explicit the notion of incomplete information. Decision theory extends this language to allow people to make statements about what alternative actions are and how alternative outcomes are valued relative to one another.

Decision theory adds probability measures that indicate the likelihood of each possible outcome for each alternative into the belief (the knowledge about the state of nature) of the decision maker. Decision theory supposes that the decision maker does not know the actual situation, but does have beliefs or expectations about the consequences of a choice in different states. A probability of 100% corresponds to the absolute belief on a certain consequence of the choice, a probability of 0% to belief on the impossibility of the consequence of the choice, and intervening values to partial belief or knowledge on the consequence of the choice. From this perspective, in decision theory, probabilities are properties of the state of knowledge of the decision maker rather than properties of the event occurrence. Sets of belief assignments that are consistent with the axioms of probability theory are said to be *coherent*. A rational person would wish to make decisions based on coherent beliefs.

2.3. Utility Theory

In economics, *utility* means the real or expected ability of a good or service to satisfy a human want. In decision theory, *utility* is a measure of the desirability of outcomes of courses of actions

that applies to decision making under uncertainty with known probabilities [White 1969]. The utility of an action is usually some function of the cost, benefit, risk, and other properties of the action. *Utility theory* [von Neumann and Morgenstern 1947] is an analytical method for making a decision concerning an action to take, given a set of multiple criteria upon which the decision is to be based. Utility theory originated from the eighteenth century, and has significantly grown up from the beginning of the twentieth century [Fishburn 1970].

Decision making serves as the foundation on which utility theory rests [von Neumann and Morgenstern 1947]. Among a set of alternatives, a decision maker would rather implement a more preferred alternative than one that is less preferred. The preferences refer to the ordering relationship among alternatives in the opinion of the decision maker and may be represented in terms of *utilities* for outcomes and *probabilities* for states of nature.

2.3.1. Utility Functions

We may represent the set of preferences by means of a numerical *utility function* $U(x, d)$ [Horvitz et al. 1988], one of the central issues in utility theory, which assigns a number on a cardinal scale instead of an ordinal scale (on preferences) to each outcome x and decision alternative d , indicating the relative desirability, and ranks the alternatives in a linear order according to degrees of desirability, so that $U(x_A, A) < U(x_B, B)$ whenever $x_A \prec x_B$ and $U(x_A, A) = U(x_B, B)$ whenever $x_A \sim x_B$, where x_A and x_B are the outcome of decision A and the outcome of decision B respectively, $x_A \sim x_B$ means the same desirability degree of x_A and x_B , and $x_A \prec x_B$ means x_B is more preferred (desirable) than x_A . By working with utility functions instead of sets of preferences, the rational choice of a decision maker is to maximize utility. The same set of preferences may be presented by many different utility functions, as any strictly increasing transformation of a utility function will provide the same choices under maximization.

2.3.2. Expected Utilities

Amounts of cardinal utility can be added and subtracted to produce other amounts of utility. This makes it possible to combine the utilities foreseen in different possible outcomes into the *expected utility*, the utility of all possible outcomes weighted by their probability of occurrence.

Formally, the expected utility can be represented as $E[U(X, d) | S] = \sum_{x \in X} P(x | S)U(x, d)$, where X

is the set of all possible outcomes generated from the decision d and S is the state of information. When a decision maker has multiple decision alternatives and there is risk or uncertainty about their individual outcomes, the preferred decision d is the one that maximizes the expected utility $E[U(X, d) | S]$ over the probability distribution for X .

The concepts of utility function and expected utility can be integrated to get expected utility functions (see Section 2.5.2).

2.4. Game Theory

Game theory [von Neumann and Morgenstern 1947, Raiffa 1968, Binmore 1992], a theory of interdependent choice founded by von Neumann in 1928, is a close relative of decision theory. *Game theory* studies interactions between self-interested entities. In particular, it studies the problems of how interaction strategies can be designed that will maximize the welfare of an entity in an encounter, and how protocols or mechanisms can be designed that have certain desirable properties.

Decision theory is often claimed to enable an entity to make the most rational choice, so it provides a means of making rational decisions under uncertainty [Raiffa 1968]. Similarly, game theory provides a rational means of analyzing interactions between entities. Decision theory can be considered as the study of games against nature. The game against nature is the simplest type of games, where a single player makes a decision in the face of nature and the nature is an opponent that just acts randomly without the desire of gaining the best payoff or defeating the opponent [Binmore 1992].

Game theory concerns games of strategy [von Neumann and Morgenstern 1947, Burger 1959]. The elements of such a game include: (1) *players*—decision makers in the game, (2) *actions*—choices available to a player, (3) *information*—knowledge that a player has when making a decision, (4) *strategies*—rules that tell a player which action to take at each point of the game, (5) *outcomes*—results that unfold, (6) *payoffs*—utilities that each player realizes for a particular outcome, and (7) *equilibria*—stable results. Here stable results mean that each player behaves in the desired manner and will not change its decision.

In contrast to pure games of chance (e.g., the guess game in which one player guesses whether there is something in another player's hand), the outcome of the games of strategy does not depend on chance alone, but also on certain decisions that the players must take during the course of play. These decisions and certain random events determine the course of play. A player can be an individual, or a group of individuals functioning as a decision making unit. The strategies available to players to bring about particular outcomes can be decomposed into a sequence of decisions called choices. Players are assumed to be able to evaluate and compare the consequences associated with the set of possible outcomes and assign utilities to each outcome indicating a preference relationship among them.

The concept of equilibria constitutes a viable solution to games [Stirling et al. 2002]. An equilibrium for a game corresponds to a vector of options (one element for each player), or *joint option*, such that each player's individual option is acceptable to it according to some criterion. There are three most widely used equilibrium concepts: dominant equilibria, Nash equilibria, and Pareto equilibria [Sandholm 1999]. A joint option is a *dominant equilibrium* if each individual option is best for the corresponding player, no matter what options the other players choose. A joint option is a *Nash equilibrium* if, were any single agent to change its decision, it would reduce its level of satisfaction. A joint option is a *Pareto equilibrium* if no single agent, by changing its decision, can increase its level of satisfaction without lowering the satisfaction level of at least one other agent. We will discuss these three types of equilibria further in Section 2.5 where we present a series of typical theoretic tools.

Obviously, game theory is closely related with decision theory and utility theory. Compared to the three primary elements of a generic decision problem (alternatives, states of nature, and payoffs) described in section 2.1.2, the composition of a game is more complex than the composition of a generic decision problem. On one hand, a game has all the elements of a generic decision problem such as players—this element is hidden and relatively simple in a decision problem where the players are decision maker(s) and the nature (or the world in which decision maker(s) work), actions (the element of alternatives in a decision problem), information (the element of states of nature in a decision problem), outcomes and payoffs which are expressed as one single element “payoffs” in a decision problem. On the other hand, the basic element *strategies* of a game is just an advanced element appearing in complex decision

problems, and a decision problem does not concern the element *equilibria* which stabilizes the opponents' strategies.

2.5. Decision-Theoretic Concepts and Methodologies

Based on the previously described mechanisms, the presented decision-related theories provide a series of concepts and methodologies that can be applied into varieties of areas involved with decision making such as economics, sociology, military strategy, and so forth. The following are some typical concepts and methodologies.

2.5.1. Probability Distribution

Probability distribution is an important component in probability theory [Feller 1968]. It is also called *probability function* that describes all the values that the random variable can take and the probability associated with each. The term of probability distribution covers both discrete probability distribution (function) and continuous probability distribution (function).

Discrete probability functions are referred to as probability *mass* functions. The mathematical definition of a discrete probability function, $p(x)$, is a function that satisfies the following properties: (1) the probability that x can take a specific value is $p(x)$, that is, $P[X = x] = p(x) = p_x$, (2) $p(x)$ is non-negative for all real x , and (3) the sum of $p(x)$ over all possible values of x is 1, that is, $\sum_i p_i = 1$ where i represents all possible values that x can have and p_i is the probability at x_i . One consequence of properties (2) and (3) is that $0 \leq p(x) \leq 1$.

Continuous probability functions are referred to as probability *density* functions. The mathematical definition of a continuous probability function, $f(x)$, is a function that satisfies the following properties: (1) the probability that x is between two points a and b is $p[a \leq x \leq b] = \int_a^b f(x)dx$, (2) $f(x)$ is non-negative for all real x , and (3) the integral of the probability function is 1, that is, $\int_{-\infty}^{\infty} f(x)dx = 1$.

Probability distribution provides a quantitative way of estimating the occurrence chance of an event and the possible outcome of an action. It has been widely used in a variety of uncertain situations.

2.5.2. Multi-Attribute Utility Functions and Expected Utility Functions

Multi-attribute utility theory (MAUT) [Hill et al. 1982, Keeney and Raiffa 1976, Sycara 1988] provides a formal basis for describing or prescribing choices between alternatives whose consequences are characterized by multiple attributes. It is based on the fundamental axiom that a decision maker attempts to maximize some utility function $U = U(g_1, g_2, \dots)$ which aggregates all the different viewpoints currently taken into account. In the utility function, each parameter, g_i , represents an estimated value for a specific attribute. Such aggregation into a single numerical measure allows classical optimization algorithms to be applied to multi-criterion problems [Barber et al. 2000].

Multi-attribute utility theory has been widely used in situations where the decision making depends on multiple factors and the utility calculation of decision alternatives is based on multiple attributes. The *multi-attribute utility functions* are used more often than general *single attribute utility functions* in complex environments where a decision maker needs to evaluate the alternatives from different viewpoints (i.e., attributes).

Based on the description of the concept *expected utility* in Section 2.3.3, the *expected utility function* provides a formalized method to combine the utilities foreseen in different possible outcomes into the expected utility, the utility of all possible outcomes weighted by their probabilities of occurrence. Formally, the expected utility function of a decision d is represented as $EU(d) = \sum_{x \in X} P(x|S)U(x,d)$, where X is the set of all possible outcomes generated from the decision d and S is the state of information. When a decision maker has multiple decision alternatives and there is risk or uncertainty about their individual outcomes, the preferred decision d is the one that maximizes the expected utility over the probability distribution for X .

2.5.3. Decision Trees

Decision trees are the graphical representation of decisions involved in the choice of statistical procedures [Horvitz et al. 1988]. A decision tree is a map of the reasoning process. It can help

decision makers form an accurate, balanced picture of the risks and rewards that can result from a particular choice and help them to do *what-if* analysis or predict the unseen behavior in the future. Figure 2.1 shows the example structure of decision trees, where a small square indicates a decision and a circle represents an uncertain result of taking the decision. Starting from the leftmost square and going along any series of branches, the decision maker can estimate the results of the corresponding choices and take actions that can achieve the goal or maximize the benefit.

Decision trees are excellent tools for making number-based decisions where a lot of complex information needs to be taken into account. They provide a framework to quantify the values of outcomes of alternatives and the probabilities of achieving them.

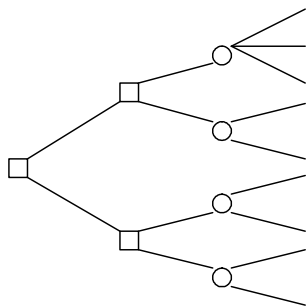


Figure 2.1. The structure of a decision tree.

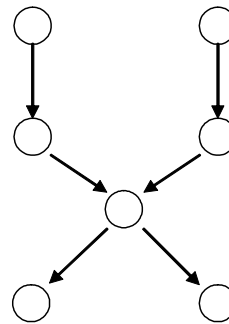


Figure 2.2. The structure of a Bayesian network.

2.5.4. Bayesian Networks

Bayesian networks (also called *belief networks*, *Bayesian belief networks*, *causal probabilistic networks*, or *causal networks*) [Pearl 1988, Neapolitan 1990] are directed acyclic graphs in which nodes represent random variables and arcs represent direct probabilistic dependences among them. They model the distribution of observations (prior knowledge) and represent probabilistic relations among uncertain variables describing the domain at hand. The structure of a Bayesian network follows the causal structure of the modeled domain. The causal structure gives a modular insight into the interactions among the variables and allows for prediction of effects of manipulation. Figure 2.2 shows the example structure of Bayesian networks.

Bayesian networks are built upon *Bayes' theorem*. Bayes' theorem allows people to reverse the direction of inference. Given a state of information S and the influence of hypothesis H on

observable evidence E , expressed as $P(E|H,S)$, the influence of E on H can be computed, expressed as $P(H|E,S)$. Bayes' theorem can be simply written as:

$$P(H|E,S) = \frac{P(E|H,S)P(H|S)}{P(E|S)}, \text{ or}$$

$$P(E|H,S) = \frac{P(H|E,S)P(E|S)}{P(H|S)}$$

This bi-directionality is a consequence of Bayes' theorem. The inferential symmetry of probability reasoning can be useful when probabilities are available in one direction but are required in the reverse direction.

Bayesian networks are an important methodology provided by decision theory and probability theory. They are very useful in showing the structure of the domain (i.e., the structure of the decision problem), probabilistic inference, and causal relationship learning. Based on the network, a decision maker can assess and refine probability distributions, and learn causal relationships between nodes. They have been widely used in applications where inference is needed. Compared to decision trees, Bayesian networks are more complex and more powerful in representing complex decision problems, especially when a decision problem exhibits many symmetries where bi-directional reasoning is very natural and easy to implement in Bayesian networks.

2.5.5. Influence Diagrams (IDs)

Influence diagrams (also called *probability influence diagrams*, *decision influence diagrams*, or *relevance diagrams*) [Howard and Matheson 1984, Neapolitan 1990, Shachter 1988] are Bayesian networks extended with utility functions and variables representing decisions. They are especially suited for modeling decision problems. The goal of influence diagram modeling is choosing such a decision option that will lead to the highest expected utility. An influence diagram is a directed acyclic graph which contains four types of nodes (decision, chance, deterministic, and value) and two types of arcs (influences and informational arcs). Figure 2.3 shows the example structure of influence diagrams, where rectangles represent decision nodes, ovals represent chance nodes, double ovals represent deterministic nodes, and diamonds represent value nodes.

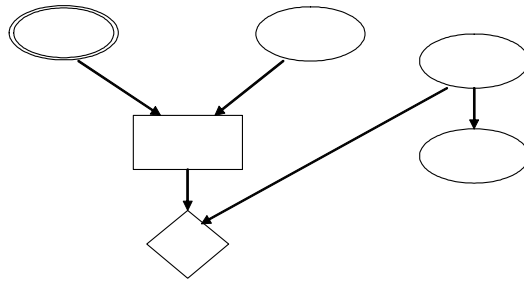


Figure 2.3. The structure of an influence diagram.

In Figure 2.3, *decision nodes* include a specification of the decision options available to the decision maker. *Chance nodes* are random variables and they represent uncertain quantities relevant to the decision problem. They are quantified by conditional probability distributions. *Deterministic nodes* represent either constant values or values determined from the states of their parent nodes. It means if the values of its parents are known, the value of a deterministic node is also known with certainty. Deterministic nodes are quantified similarly to chance nodes. The only difference is that their probability tables contain all zeros or ones as there is no uncertainty about the outcome of a deterministic node once all its parents are known. *Value nodes* represent utility, which is a measure of desirability of the outcomes of the decision process. They are quantified by the utility of each of the possible combinations of outcomes of the parent nodes. Normally, an arc in an influence diagram denotes an *influence*, which means that the node at the tail of the arc influences the value (or the probability distribution over the possible values) of the node at the head of the arc. So they have a causal meaning. However, the arcs coming into decision nodes have a different meaning. These arcs are *informational* ones representing temporal precedence (in the sense of information flow). The outcomes of all nodes at the tail of informational arcs should have been known before the decision is made.

The influence diagram is an important tool provided by decision theory, probability theory, and utility theory together. They can represent multiple objectives and allow tradeoffs in one area against costs in another. Similar to Bayesian networks, influence diagrams are very useful in showing the structure of the domain, i.e., the structure of the decision problem. Different from the qualitative illustration of the structure of the domain provided by Bayesian networks, influence diagrams allow accounting for uncertainty and are able to represent it in a quantitative way. Unlike decision trees, influence diagrams do not grow exponentially and they support

reverse inference very easily. They also suppress trivial details and hence are suitable for getting an overview of a complex problem.

2.5.6. Markov Decision Processes (MDPs)

Markov decision processes (MDPs) [Howard 1960, Boutilier et al. 1999] were developed within the context of operations research. In essence a Markov decision process is an iterative set of classical decision problems. At a conceptual level, most decision problems involved with sequential actions and states can be viewed as instances of Markov decision processes. A Markov decision process can be described with a graph. A state of the world (or an environmental state) can be represented as a node in a graph. Carrying out an action in that state will result in a transition to one of a number of states, each connected to the first state by an arc, with some probability, and will incur some cost. After a series of transitions a goal state may be reached. The sequence of actions carried out is called a *policy*, which is a mapping from environmental states to actions. Solving an MDP amounts to finding a minimal cost policy for moving from some initial state to a goal state.

Formally, an MDP is defined as a tuple $\langle S, A, T, R \rangle$ [Cassandra et al. 1994], where S is a finite set of environmental states that can be reliably identified by the decision maker; A is a finite set of actions; T is a state transition model of the environment, which is a function mapping elements of $S \times A$ into discrete probability distributions over S ; and R is a reward function mapping to the real numbers that specify the instantaneous reward that the agent derives from taking an action in a state. The state transition model T can be written as $T(s, a, s')$ for the probability that the environment will make a transition from the previous state s to the current state s' when action a was taken. The reward function can be written as $R(s, a)$ for the immediate reward to the decision maker for taking action a in state s . The policy π , mapping from S to A , specifies an action to be taken in each situation.

MDPs can capture many of the facets of real-world problems and are often used in decision making based on the history up to now. An environment is regarded as holding Markov property if the environment's response at time $t+1$ depends only on the state and action representations at time t . If an environment has the Markov property, then its one-step dynamics enable us to predict the next state and expected next reward given the current state and action. Iteratively one

can predict all future states and expected rewards from knowledge of the current state as well as the complete history up to the current time. It also follows that Markov states provide the best possible basis for choosing actions. That is, the best policy for choosing actions as a function of a Markov state is just as good as the best policy for choosing actions as a function of the complete history.

MDPs can be used to formalize the domains in which actions have probabilistic results and the decision maker has direct access to the state of the environment. An important aspect of the MDP model is that it provides the basis for algorithms that provably find optimal policies given a stochastic model of the environment and a goal [Howard 1960]. MDP models play an important role in research on planning and learning. But the assumption of complete observability to the states provides a significant obstacle to their application to real-world problems [Cassandra et al. 1994].

2.5.7. Partially Observable Markov Decision Processes (POMDPs)

As a generalization of MDPs, the *partially observable Markov decision processes* (POMDPs) also originated in the operation research literature. MDPs apply to the decision problems where the state information can be observed completely. In most real-world decision problems like machine maintenance and quality control, however, the problem settings are of state uncertainty where the state information is partially observable. A POMDP permits uncertainty regarding the state of an MDP and allows state information acquisition [Cassandra et al. 1994]. When the state is not completely observable, a model of observation must be added to represent the uncertainty of the state acquired. The model of observation includes a finite set of possible observations for the decision maker and an observation function representing the probability distributions over observations.

Formally, a POMDP is defined as a tuple $\langle S, A, T, \Omega, O, R \rangle$ [Cassandra et al. 1994], where S , A , T , and R are similar to those in the definition of an MDP; Ω is a finite set of possible observations for the decision maker; and O is an observation function mapping $A \times S$ into discrete probability distributions over Ω . The observation function can be written as $O(a, s, \omega)$ for the probability of making observation ω from the current state s after having taken action a . The policy π , mapping from Ω to A , specifies an action to be taken in each situation.

POMDPs are used for describing planning tasks in which the decision maker does not have complete information as to its current state. The POMDP model provides an elegant solution to the decision problem of acting in partially observable domains, treating actions that affect the environment and actions that only affect the decision maker's state of information uniformly. As a result, the POMDP model provides a convenient way of reasoning about tradeoffs between actions to gain reward and actions to gain information.

2.5.8. Nash Equilibria and Pareto Equilibria

Game theory assumes that one has opponents who are adjusting their strategies according to what they believe everybody else is doing. The exact level of sophistication of the opponents should be part of one's strategy. Sometimes it is possible for a player to take a dominant strategy to be best off no matter what strategies other players use. However, often a player's best strategy depends on what strategies other players choose. In such settings, dominant strategies do not exist, and other stability criteria are needed [Sandholm 1999]. If the players are disposed to cooperate, they may seek a Pareto equilibrium. A self-interest player in a game, however, would have no incentive to choose a Pareto equilibrium unless it would join a coalition. The concept of Nash equilibria [Nash 1950] is consistent with an attitude of exclusive self-interest and it is the most basic one of the stability criteria. If there is a set of strategies with the property that no player can benefit by changing her strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute the Nash equilibrium.

3. Decision-Theoretic Requirement in Multi-Agent Systems

Multi-agent systems are composed of a group of autonomous and distributed entities called agents, operating in an environment and interacting with one another to collectively achieve their goals [Weiss 1999]. In this section, we will discuss the decision-theoretic requirements in multi-agent systems from the perspective of BDI (Belief-Desire-Intention) architectures [Rao and Georgeff 1995, Wooldridge 1999]. Specifically, we discuss from the perspective of agents' attitudes such as the beliefs of agents to the states of the world, what the agents desire to do, and how the agents are intended to act in different situations, which consist of the BDI architecture.

The range of requirements discussed varies widely from an individual agent's knowledge representation to the coordination among multiple agents.

The research work on multi-agent systems involves a variety of aspects. We can discuss the decision-theoretic requirements in multi-agent systems from many perspectives like logic-based architectures and layered architectures [Wooldridge 1999]. Here we discuss the decision-theoretic requirements in multi-agent systems from the perspective of BDI architectures as the central issues of multi-agent system design include: how should an agent represent knowledge, and how should the agent operate on it to arrive at purposeful actions [Newell 1981]. The BDI architecture, very popular in the multi-agent system community, focuses on these issues. It evolves from a philosophical model of human practical reasoning (originally developed by Michael Bratman [Bratman 1987], see Section 3.2) and is intuitive—we all recognize the processes of deciding what to do and then how to do it, which are closely related with decision theory. In addition, the BDI architecture gives us a clear functional decomposition [Wooldridge 1999], which indicates the design requirement of building an autonomous agent in multi-agent systems. This enables us to discuss the decision-theoretic requirement clearly and systematically.

In this section, we will address the relationship between decision-related theories and agents' beliefs, desires, and intentions, and discuss the specific decision-theoretic requirements from the above aspects.

3.1. Agents and Multi-Agent Systems

Fundamentally, an agent is an active entity with the ability to perceive, reason, and act in order to satisfy its design objectives. An agent has the ability to communicate. This ability is part perception (the receiving of messages) and part action (the sending of messages) [Huhns and Stephens 1999]. An agent has a set of goals (or desires), certain capabilities to perform actions (conducted by intentions), and some knowledge (or beliefs) about its environment. Agents are assumed to have explicitly represented knowledge and mechanism for operating on or drawing inferences from their knowledge. As an autonomous entity, frequently, an agent needs to make decisions based on currently held beliefs (states of nature) to take specific actions (select alternatives) and achieve specified goals (maximize payoffs or utilities) even in the simplest

environment. In complex environments of such characteristics as dynamism and uncertainty, decision making is especially important and necessary.

The behavior of agents operating in multi-agent systems may be *reactive* or *rational* [Weiss 1999]. Being reactive means that the agent is capable of maintaining an ongoing interaction with the environment, and responding in a timely fashion to changes that occur in it; while being rational means that the agent behaves in a way that is suitable or even optimal for goal attainment. When there are multiple alternatives to select, decision making is necessary no matter whether in a reactive manner or in a rational manner the agent behaves. For the rational behavior, strategy-related long-term decision making is more important.

Multi-agent environments provide an infrastructure for communication and interaction among agents. Agents in multi-agent systems communicate and interact in order to achieve better the goals of themselves or of the society/system in which they operate [Huhns and Stephens 1999]. The environments are typically open and have no centralized designer. The designers of a multi-agent system may not know others' design objectives very well. Since the design of an agent's characteristics and behavior eventually depends on its designer, the non-centralized design makes the agents not know other agents in the system very well. Even if the system design is centralized, the autonomous behavior of agents may make others operate in an unpredictable, and hence dynamic and uncertain, environment. There can be multiple actions possibly to take at the moment and one same action can result in multiple possible outcomes. To perceive the environmental states, act and interact in uncertain situations, decision-theoretic tools are useful for agents.

3.2. The BDI (Belief-Desire-Intention) Architecture

The BDI architecture originated from the philosophical tradition of understanding *practical reasoning*—the process of deciding moment by moment which action to perform in the furtherance of the *goals* (i.e., a mutually consistent set of desires [Kraus et al. 1998, Singh et al. 1999]) [Bratman 1987, Wooldridge 1999]. Practical reasoning involves two important processes: deciding what goals to achieve, and how to achieve these goals. The former process is known as *deliberation*, the latter as *means-ends reasoning*. For a specific procedure of practical reasoning of an agent, the decision process typically begins by the agent's trying to understand what

options available given the current situation. After generating the set of alternatives, the decision maker must choose among alternatives, and commit to some. These chosen options will become intentions, which then determine the agent's actions. Intentions then feed back into the agent's future practical reasoning. We will further describe the practical reasoning and its two processes later in this section.

There are seven main components in a generic BDI architecture [Wooldridge 1999]: (1) a set of current *beliefs*, representing information the agent has about its current environment; (2) a *belief revision function*, which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs; (3) an *option generation function*, which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current intentions; (4) a set of current options (*desires*), representing possible courses of actions available to the agent; (5) a *filter function*, which represents the agent's main deliberation process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions. The desires originally generated may be inconsistent while the goal set is a consistent subset of desires. The agent forms intentions to make the goals true; (6) a set of current *intentions*, representing the agent's current foci—those states of affairs that it has committed to trying to bring about; and (7) an *action selection function*, which determines an action to perform on the basis of current intentions. Figure 3.1 illustrates the schema of a generic BDI architecture of the above main components.

From Figure 3.1, we see that the basic components of a Belief-Desire-Intention architecture are data structures representing the beliefs, desires, and intentions of an agent, and functions representing its deliberation (deciding what intentions to have—i.e., deciding what to do) and means-ends reasoning (deciding how to achieve—i.e., how to do) where the belief revision function is the basis of the agent's deliberation process. Beliefs represent what the agent knows about the states of the world, desires describe the specific states of the world the agent prefers to achieve, and intentions lead to the agent's actions. They are represented as sets (i.e., as unstructured collections) respectively. Let *Bel* be the set of all possible beliefs, *Des* be the set of all possible desires, and *Int* be the set of all possible intentions. Their values are acquired from the corresponding functions. Representing an agent's intentions as a set is generally too

simplistic in practice [Wooldridge 1999]. A more practical way is to associate a priority with each intention, indicating its relative importance.

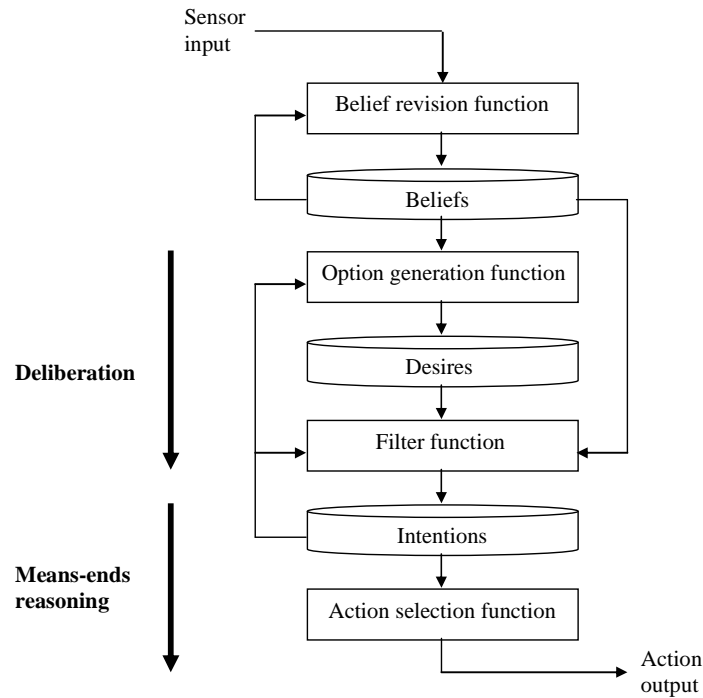


Figure 3.1. Schematic diagram of a generic BDI architecture [Wooldridge 1999].

An agent's *belief revision function* is a mapping from the current perception and a set of current beliefs to a new set of beliefs: $\wp(Bel) \times P \rightarrow \wp(Bel)$. The *option generation function* is a mapping from a set of beliefs and a set of intentions to a set of desires: $\wp(Bel) \times \wp(Int) \rightarrow \wp(Des)$. We can regard an agent's option generation process as one of recursively elaborating a hierarchical plan structure, considering and committing to progressively more specific intentions, until finally it reaches the intentions that correspond to immediately executable actions. The *filter function* is a mapping from the previously held intentions and the current beliefs and desires to the updated intentions: $\wp(Bel) \times \wp(Des) \times \wp(Int) \rightarrow \wp(Int)$. It represents the agent's deliberation process (deciding what to do). The *action selection function* is assumed to simply return any executable intention—one that corresponds to a directly executable action: $\wp(Int) \rightarrow A$. Combining these four functions together, we can get an outlined action function of an agent, which is a mapping from perceptions to actions: $P \rightarrow A$ [Wooldridge 1999].

Taking into consideration of the basic components of a generic BDI architecture, what need to be concerned in the design of BDI agents correspond to the two central issues of multi-agent system design mentioned before: how should an agent represent knowledge, and how should the agent operate on it to arrive at purposeful actions? Decision theory provides an answer by postulating that probability (to represent what an agent knows) and utility (to represent what an agent prefers) be combined to define the agent's behavior that maximizes its expected utility or achieves its goals.

3.3. Decision Theory and Agents' Beliefs

In this section, we will discuss the decision-theoretic requirement in multi-agent systems based on the issues related with agents' beliefs. Our discussion involves the first two components of the BDI architecture: the belief revision function, and the generated beliefs.

In a complex environment, which may be noisy and change dynamically and uncertainly, the perceived information by an agent may be inaccurate and an agent is inherently uncertain about the environment as well as other agents. Even if the environment itself is noiseless, static, and certain, other agents' behavior may make the agent operating in an unpredictable thus uncertain environment. Formally, the information the agent has about the state of the current world (both the environment and agents) is called *belief*. Each agent has its own beliefs about how the world is. These beliefs come from the agent's perception and cognition of the states of the world. In an uncertain environment, an agent cannot exactly discriminate among the states of the world. The fact that the actual state may be unknown to the agent can be formalized by specifying the set of all possible states of the world, S , together with a family of probability distributions, $P(S)$, over these states. Each of these distributions specifies which of these states are currently possible and how likely they are. Thus the probability distributions can be used to describe the information the agent has about the present state of the world.

As a result, decision theory, especially probability theory, can play a significant role in the definition of agents' belief revision functions and the formalization of agents' beliefs. Further, to model the distribution of knowledge and represent probabilistic relations among them, Bayesian networks and influence diagrams can be applied to create the possible relationship between beliefs for causal reasoning. Based on the beliefs it holds, an agent will generate a set of options

possible to achieve and decide what actions to do. To find an optimal sequence of actions from the present state to a goal state, Markov decision processes (MDPs) can be applied. MDPs assume that it is possible to measure some aspect of the world and from this measurement the state of the world can be known precisely. In realistic situations, however, from the measurement something can be uncertainly inferred about the world. In such a situation, the states of an MDP are replaced by beliefs about those states, resulting in the application of partially observable Markov decision processes (POMDPs).

3.4. Decision Theory and Agents' Deliberation

In this section, we will discuss the decision-theoretic requirement in multi-agent systems from the aspect of agents' deliberation processes. Our discussion involves the three components of the BDI architecture: the option generation function, desires, and the filter function, which compose an agent's deliberation process (deciding what to do). Compared to the *tactical* decision making on specific action selection possibly involved in the succeeding means-ends reasoning process, the decision making in this process is *strategic*. In this process, courses of actions (plans) will be decided.

In the agent's deliberation process, a set of desires are generated and the agent has the will to fulfill these desires. At any given time, the agent selects a consistent subset of its desires. This serves as its set of current goals. The set of goals motivates the agent's planning process which filters out its intentions [Kraus et al. 1998]. Our discussion will focus on the decision-theoretic requirement in agents' desires and goals generation, (individual) planning, and the coordination to (individual) planning, which is crucial in multi-agent systems [Huhns and Stephens 1999].

3.4.1. Agents' Desires and Goals Generation

An agent's *desires* refer to the states of affairs toward which the agent has a positive disposition [Wooldridge 1999]. A desire represents some desired end state of the world based on the agent's current beliefs. Each agent has its own desires about how it would like the world to be like. The concept of desire is closely related with another concept of *preference* as the desires of an agent to do something indicate its preferences over the possible outcomes or states. The preferences refer to the ordering relationship among alternatives in the agent's opinion. The preferences and desires come from the agent's user or owner. Since in multi-agent systems an agent's behavior

motivation is to maximize the expected benefit for itself (if it is self-interested) or for a group of agents (if it is cooperative) (see Section 3.4.2), the need to maximize payoffs of preferences essentially requires that there be a scalar representation for all the preferences of an agent. In other words, all of the preferences must be reduced to a single scalar with which they can be precisely compared. This requires identification, evaluation, and comparison of alternative solutions before the best solution is selected. Obviously, utility theory, especially utility function, can play a significant role in this task. When an agent needs to select some alternatives from a set of alternatives, it can evaluate the utility of each alternative following specific criterion. Sometimes this evaluation procedure is very simple. For example, if the agent only considers the cost to achieve a desire, it can just select the one with the minimal cost. So this is a single attribute utility function. If the agent needs to take into account multiple factors, the use of multi-attribute utility functions is necessary and the agent needs to assign corresponding weight values to those factors according to the different contributions of the factors to the utility computation.

The set of an agent's desires may not always be consistent. For example, an agent (or a person) may desire to get a doctoral degree, do lifelong research, enjoy parties everyday, interact with kinds of people as often as possible, and so on. However, the first two desires and the succeeding two desires lead to a contradiction generally and it is not possible to get all desires satisfied. The agent needs to select a consistent subset of its desires—goal set—to achieve. For all the goal sets in this example ({get a doctoral degree, do lifelong research}, {enjoy parties everyday, interact with kinds of people as often as possible}, etc.), the agent may ascribe different degrees of importance to them. Then he can select one with the highest importance degree from all the goal sets [Kraus et al. 1998]. Since each goal set may consist of more than one desire, it is not certain that the utility values of all desires in one goal set are higher than the utility values of all desires in another goal set. Otherwise, the agent can simply select the goal set in which each desire has a higher utility value than any desire in any other goal set. To compare the importance degrees of all goal sets, it is necessary to apply the utility function into the computation of the importance degrees (utilities) of goal sets, which is similar to the application of utility function in desires generation.

However, it is not enough for an agent to generate goals only depending on the importance degrees of different goal sets. Such an agent only tries to achieve the most important goals

irrespective of the possibility of achieving those goals. In the above example, even if the agent realizes that the first goal set is the most important one, he needs to give up those goals if his academic record is always very bad and he has been fifty years old. To build sensible agents, the probability of achieving goals needs to be taken into account together with the utility computation, which results in the application of the expected utility theory. The expected utility of a goal set can be calculated with the product of the probability of achieving the goals and its utility (importance degree), which can be formalized as $EU(GS_i) = \Pr(GS_i)U(GS_i)$ where GS_i is a goal set of all possible goal sets. The use of expected utility can avoid agents trying to achieve the goals with the greatest utility irrespective of the possibility of achieving them.

After a set of goals is selected, these goals will motivate the agent's planning process.

3.4.2. Agents' Planning

The design of a multi-agent system is to implement specified functions and achieve specified goals. Agents in such a system are assumed to be able to perceive the environment and carry out actions to implement the design objectives. The system's current state and the agents' choice of action jointly determine a probability distribution over the system's possible next states. An agent prefers to be in certain system states (e.g., goal states) to others. To achieve its goals, an agent must reason about its environment (as well as behavior of other agents) and determine a *strategy* (i.e., the agent's mapping from state history to action [Sandholm 1999]; also called a plan, a course of action, or a policy) that is likely to lead to the goals, possibly avoiding undesirable or inconsistent states along the way. The agent may not know the system's state exactly in making its decision on how to act, however, it may have to rely on the current beliefs and base its choice of action on a probabilistic estimate of the state.

The deliberation process hidden in the filter function of a BDI agent is indeed a planning procedure, resulting in a set of *paths* the agent having selected or preferred. Intentions can be regarded as the conditions that inevitably hold on each of the selected path [Singh et al. 1999]. To generate such a set of paths, the agent has to plan a series of strategies that provide long-term consideration for selecting actions towards specific goals. Each strategy "attacks" a solution space in a different manner. The agent has to select the appropriate strategy from the alternatives.

The capability of agents' strategy selection can enhance the flexibility and adaptability of a multi-agent system to dynamic and uncertain environments. To achieve this objective, there are several issues to be addressed, including: (1) an uniform representation of various strategies to assist the agent's comparison and evaluation process, (2) a meta-level reasoning mechanism for strategic decision making, (3) a set of characteristics that agents use to evaluate alternative strategies, and (4) adaptability or learning ability to improve the decision making required to select a strategy [Barber et al. 2000].

Basically, for the first and second issues addressed above, Bayesian networks, influence diagrams, decision trees, and expected utility functions can be applied to represent the structures of strategies, model their causal reasoning mechanism, and compare and evaluate various strategies. For the third issue, multi-attribute utility functions and expected utility functions can be applied and the characteristics to be considered include requirement imposed by the strategy, cost of strategy execution, solution quality, domain requirements, and so forth. For the fourth issue, decision trees, Bayesian networks, influence diagrams, MDPs, or POMDPs are applicable which provide continually learning ability. Specifically, for any strategy, given a probability distribution over the possible outcomes of an action in any state, and a reasonable preference function over outcomes, we can define an expected utility function on outcomes such that whenever the agent would prefer one strategy than another, the preferred strategy has higher expected utility. The task of the agent then seems straightforward—to find the strategy with the maximum expected utility. To calculate the expected utility of a strategy, all actions need to be concerned. For a course of action, the expected utility of the current action depends upon the expected utility of next action.

3.4.3. Coordination to Agents' Planning

In order to solve goals which require the action of multiple agents, coordination mechanisms are needed that can coordinate the agents' planning processes and integrate the resulting individual plans. *Coordination* is a choice of action that takes into account the anticipated actions of the other agents [Huhns and Stephens 1999, Gmytrasiewicz and Noh 2002]. Agents can coordinate their activities in a cooperative or a self-interested manner. Being *cooperative* means that the agents are non-antagonistic and they can cooperate to perform tasks or achieve desired goals. Being *self-interested* means that the agents are competitive and each of them tries to maximize

its own benefit. Note that an agent may be both cooperative in some cases and self-interested in other cases. No matter whether in a cooperative manner or in a self-interested manner the agent behaves, since they are in a shared environment, they need to coordinate their activities to achieve their goals.

Cooperative coordination can be implemented in the form of teamwork [Tambe 1997] or coalition formation [Luce and Raiffa 1957, Sandholm and Lesser 1997]. *Teamwork* (or *coalition formation*) in multi-agent systems is a process where agents form teams (or coalitions) and work together to solve a joint problem via coordinating their actions within each team (or coalition) [Sandholm 1999]. Teamwork takes place among cooperative agents and the agents' objective of forming a team is to maximize the system benefit while coalition formation takes place among self-interested agents and each agent's objective of joining a coalition is to maximize its own benefit although they show cooperative behavior as a coalition.

Self-interested coordination is generally implemented through negotiation (agents also can form a coalition through negotiation in the case of coalition formation). *Negotiation* is a process by which two or more agents make a joint decision to coordinate their activities, each trying to reach an individual goal or objective [Huhns and Stephens 1999, Raiffa 1982]. The negotiation protocol is a straightforward iterative process of agents making offers and counteroffers. From an individual agent's point of view, negotiation is a decision problem that requires a decision maker (agent) to weigh preferences and to choose an action that gives the maximum utility from the set of actions allowed by the negotiation protocol.

In the cooperation case, typically, to cooperate successfully, each agent must maintain a model of the other agents, and also develop a model of future interactions [Huhns and Stephens 1999]. From each agent's point of view, teamwork or coalition formation is also a decision problem that requires the agent to weigh preferences or benefits and to choose the way of joining and working in a team or a coalition.

To model the agent's decision making in the negotiation or cooperation process, Bayesian networks and influence diagrams can be used. In addition, to describe a series of state transition processes from the original states to the desired goal states, MDPs and POMDPs can be used.

In most multi-agent encounters, the overall outcome depends critically on the choices made by all agents in the scenario. This implies that in order for an agent to make the choice that optimizes its outcome, it must reason strategically. That is, it must take into account the decisions that other agents may make, and must assume that they will act so as to optimize their own outcome. Game theory gives a way of formalizing and analyzing such concerns. Game theory can be used to study what happens when rational and self-interested agents with different goals interact, each making its own decisions on the basis of what is best for itself, while taking into account that the others are doing the same [Parsons et al. 2002]. Nash equilibria and Pareto equilibria in the game theory can be used to motivate each self-interested agent to behave in the desired manner.

3.5. Decision Theory and Agents' Means-Ends Reasoning

In this section, we will discuss the decision-theoretic requirement in multi-agent systems from the aspect of agents' means-ends reasoning processes. Our discussion involves the last two components of the BDI architecture: intentions and the action selection function, which compose an agent's means-ends reasoning process (deciding how to do). Compared to the decision making on strategies in the agent's deliberation process, the decision making occurring in this process can be regarded as tactical decision making (on specific action selection).

Intentions play a central role in the Belief-Desire-Intention model: they provide stability for decision making, and act to focus the agent's practical reasoning. The obvious property of intentions is that they tend to lead to actions. To achieve an intention, an agent needs to carry out some course of actions that it believes would best satisfy the intention and this intention will constrain the agent's future practical reasoning. Since intentions are inevitably held conditions on each of strategies, the representation of intentions must be incorporated with the representation of strategies. So Bayesian networks, influence diagrams, or decision trees can be applied.

The strategies generated in an agent's deliberation process can help the agent to observe the environment, evaluate alternatives, and schedule actions. For any given problem, various strategies may be available. Although a strategy may help to achieve success through carrying out a course of action, it does not guarantee success. The failure of goal achievement may result

in iterative intention filtering and replanning in the strategic deliberation process. However, the agent also needs to do tactical decision making in the means-ends reasoning process.

Due to possible non-determinism, an action of the agent may lead to many possible states (resulting outcomes). Decision theory provides a means of handling the non-determinism of an agent's actions [Parsons et al. 2002]. The likelihood of the resulting states can be specified by probability distribution over the states of the world. The process of determining the probabilities of different outcomes (i.e., the probability distribution) has been called a *probabilistic temporal projection* [Boutilier et al. 1999, Gmytrasiewicz and Lisetti 2002]. The projection is a function from the current information about the state and the action to the resulting state: $P(S) \times A \rightarrow P(S)$ where $P(S)$ is the family of probability distributions over all possible states of the world, S , and A is the set of all actions currently possible.

In the uncertain environment, an agent may have a set of possible actions to select to take, each of which has a range of possible outcomes since the actions are not deterministic. The various possible outcomes of non-deterministic actions have degrees of value incurred, so an agent has preferences among the different outcomes. The value of taking a particular action will depend upon what the state of the world will be after taking this action. To choose an action to undertake, the agent will need to look at the utility value of the state it is in after the action. Doing this for each possible action, the agent can then choose the action that leads to the state it values most. The utility function in utility theory can be used as a numerical scalar on agents' preferences. Nearly in all places where evaluation and comparison of alternative solutions needs to be done before the best solution is selected, the utility function can be applied. However, only building a utility function to order the preferences is not enough. Such an agent only tries to achieve the most valuable state irrespective of the difficulty and the possibility of achieving it. To build more sensible agents, the probability of an outcome occurrence needs to be taken into account together with the utility computation [Parsons et al. 2002]. The expected utility theory can be used here. The expected utility of an action can be calculated with a weighted average of the utility of each possible outcome, where the weight is the probability of that outcome given the action being performed. Since each outcome is itself a state, we have

$$EU(A_i) = \sum_{S_j \in S} \Pr(S_j | A_i) U(S_j)$$

where S is the set of all states after executing the action. The

agent then selects action A^* where $A^* = \arg \max_{A_i \in A} EU(A_i) = \arg \max_{A_i \in A} \sum_{S_j \in S} \Pr(S_j | A_i) U(S_j)$ where A

is the set of all possible actions. The use of expected utility can avoid agents trying to achieve the state with the greatest utility irrespective of the possibility of achieving it, and on the other hand, can avoid agents trying to achieve the state, which has the greatest chance of being achieved irrespective of its value.

4. Decision-Theoretic Applications in Multi-Agent Systems

With regard to a variety of decision-theoretic requirements in multi-agent systems discussed in section 3, it is very natural to see numerous decision-theoretic application scenarios in multi-agent systems. In this section, we will present some typical and explicit decision-theoretic applications from the perspective of the series of concepts and methodologies described in section 2.5: probability distribution, multi-attribute utility functions, expected utility functions, decision trees, Bayesian networks, influence diagrams, MDPs, POMDPs, Nash equilibria, and Pareto equilibria.

4.1. Applications of Probability Distribution

In multi-agent systems, when the environmental change and the outcome of an action (or a set of actions) an agent carries out are certain, the agent can precisely estimate the states or outcomes, and does not need to consider any possibility. However, the open design paradigm of a realistic multi-agent system may result in uncertainty. The possible reasons include: there may be perception errors during the interaction between the agent and its world, there may be multiple designers in the design of a multi-agent system who do not know precisely others' objectives, and other agents' behavior may change the expected outcome of an agent's action. In such a situation, in order to accurately capture the likelihood of states or outcomes, it is very natural for the agents to estimate the occurrence probabilities of all the possible environmental states or all the possible outcomes of carrying out an action with probability distribution provided by probability theory. Correspondingly, the use of probability distribution in multi-agent systems can be classified into two fields: the computation on the occurrence probability of uncertain information in agents' beliefs, and the estimation on the non-determinism of actions' outcomes.

4.1.1. Probability Distribution and Agents' Beliefs

The probability distribution on the agents' beliefs is generally applied to estimate the accuracy degree of the agent perceiving the present environment and other agents. In an uncertain environment, the environmental status changes within a specific range. An agent may know all the possible occurrences but it does not know the current occurrence exactly. Even in a certain environment, the possible noise in the environment may make an agent's perception to the environment and recognition to other agents incomplete (or even inaccurate). In these cases, the agent needs to estimate the current environmental state transformation, the completeness degree and accuracy degree of its knowledge about the world.

In [Gmytrasiewicz and Noh 2002], Gmytrasiewicz and Noh present the implementation of knowledge bases of the agents that accommodate uncertainty and nested information agents may have about the world. Their design of the knowledge base is based on work on frame-based and object-oriented knowledge representation formalisms [Brachman and Levesque 1985]. A fundamental limitation of the frame formalisms is that they do not support uncertainty. Their design combines the frame formalisms with Bayesian networks. The basic idea is to treat the slots of frames (or attributes of objects) describing the properties of objects in the world that may not be known with certainty as nodes of a Bayesian network. Such probabilistic slots allow values in form of probability distributions. This knowledge base design will be described further in the later section about the applications of Bayesian networks in multi-agent systems.

In [Gmytrasiewicz and Lisetti 2002], Gmytrasiewicz and Lisetti study the role and usefulness of emotional states and personality in designing multi-agent systems. The emotional states of an agent are viewed as the agent's decision making modes, predisposing the agent to make its choices in a specific, yet rational, way. The personality of an agent consists of the agent's emotional states together with the specification of transitions taking place among the states. To enable an agent to model the personalities and emotional states of other agents that it interacts with, the authors provide a precise definition of a personality model of other agents. From the perspective of an agent Q , a personality model of another agent R is a probabilistic finite state machine $P_R = \langle D, IN, \Delta, N \rangle$, where D is a finite set of emotional states of agent R , IN is a set of environmental inputs, Δ is a probabilistic state transformation function, $\Delta : D \times IN \times D \rightarrow [0,1]$, and $N \in D$ is an initial (or neutral) emotional state of agent R . Agent Q , which has a personality

model of agent R , can use it to probabilistically predict R 's emotional state, given an initial state and an environmental input. The state transformation function is probabilistic to allow for uncertainty as to the next emotional state of the modeled agent R . Agent Q assigns a probability distribution to all the possible emotional state transformations due to an environmental input in IN . With such a model, an agent can build its belief on the personality of another agent. The main advantage of using this probability distribution based model is that a personality model can be learned, given limited amount of observations of the other agent's behavior. Then the probability distribution can be dynamically changed through learning.

4.1.2. Probability Distribution and Outcome of Agents' Actions

The probability distribution on the outcomes of agents' actions is generally applied together with the utilities of outcomes, and the application objective is to calculate expected utilities of actions. Generally, an agent cannot precisely predict the probability distribution of the outcomes of its action at the beginning but it can track the outcomes of the same action and get the probability distribution from the past experience.

In [Banerje and Sen 2002], Banerje and Sen develop a payoff-structure model for partner selection in coalition formation problem. They consider situations where a rational agent decides on which partnership to interact with given the number of interactions and possible payoffs. Each agent interaction is assumed to ultimately generate some utility for each of the interacting agents. An agent can get one of several payoffs or utilities for joining a particular coalition, and there is a static probability distribution that governs which of the payoffs is received at any particular interaction. So here the probability distribution indicates the possible payoff distribution of agents' coalition joining actions. The payoff-structure encoding in the form of a probability distribution over possible payoffs is used as the summary information on which the agent must base to make its decision.

In [Soh et al. 2003], Soh et al. use probability distribution to evaluate the outcome of negotiation between agents. This is a simple but typical case of probability distribution application in multi-agent systems. The possible outcomes of a negotiation action include success and failure. The negotiation success means the possible cooperation between agents. An agent profiles the negotiation outcome with each peer agent in the history and gets the dynamic probability

distribution on negotiation outcomes. Then based on this probability distribution and other factors, the agent can calculate the expected utility of selecting each peer agent in the future cooperation and then decide which agents to approach for negotiations.

4.2. Applications of Utility Functions

The use of utility functions (and expected utility functions) is very wide in multi-agent systems since each agent has its own preferences and desires about how the world is and the preferences and desires can be conveniently and formally captured by means of a utility function. Although in some multi-agent systems the utility functions are not given explicitly (e.g., [Bazzan et al. 2002]), the agent's evaluation on different alternatives is generally built upon a certain set of attributes and corresponding weights.

4.2.1. Multi-Attribute Utility Functions

Balogh et al. built a multi-agent system for negotiation and decision support [Balogh et al 2000]. The main entity of the system is a negotiation center which uses decision algorithms to rationally apportion goods and services into parts with equal utilities (Cut Cake algorithm) to ensure fair, fast, and efficient behavior. In order to enable the system to compute utilities, negotiation center needs to know the utility functions of particular goods and services. Individual utilities are functions of elements such as price, amount, time, etc. This is a simple application example of multi-attribute utility functions in multi-agent systems.

In [Barber et al. 2000], an application example of strategic decision making is presented. The planning process requires as input the current state of the world, the actions available, and a goal state to generate strategies. These three items are maintained by the agent and dynamically used. The world state is constantly changing through the actions of other agents, and the actions available to achieve any given goal change based upon the agents who are helping to achieve the goal. For the purpose of conflict resolution during strategy selection, the multi-attribute utility function is applied here to evaluate the strategies from different aspects. The attributes include the quantified states, and payoffs of actions.

An attribute in a utility function can be a domain factor (like time of executing a task) or a characteristic of the agent (like virtues an agent shows during interaction with others). In [Bazzan et al. 2002], one of the attributes contributed to the utility computation is the agents'

moral sentiment (i.e., emotions like generosity towards others and guilt for not having played fair with someone). The selfish rational agents act to maximize their gain in the short term while altruistic agents are led by moral sentiments and sacrifice rational decisions in some degree. Based on the Prisoner's Dilemma problem [Axelrod 1984], Bazzan et al. conduct a series of experiments and show that the selfish rational agents maximize their earnings in the short term but compromise their performance in the long run, while altruists with moral sentiments may not have the best performance at the beginning but normally end up much better than others. This result indicates that the agent's emotional stance is used as one attribute in the calculation of utility. The moral sentiments make an altruistic agent choose what is not best for its own goals but they are long-term utility maximizers.

In decision theories desires are usually formalized in terms of utility functions. In [Dastani et al. 2002], the authors study desires represented with utility functions in a dynamic environment. Desires of agents are assumed to reflect their utility functions that in turn reflect their preferences. They look for a formal model in which the utility functions are typically constant, desires are relatively stable, whereas goals change much more frequently. They model the agent's dynamic desires in the context of practical negotiations where agents can reach agreements by influencing other agents' desires. Rational agents in negotiation decide what action to take based on their desires that reflect their utility functions. Since negotiation is usually modeled by game theory and in game theory the utility function is assumed to remain constant during a game, they have to solve an apparent contradiction: on the one hand, it is reasonable to assume that the agent's desires can be changed during negotiation and on the other hand the utility function which is reflected by the rational agent's desire has to remain constant during negotiation. They solve this apparent contradiction by lifting the utility function to a desirability function and allowing the lifting condition to change on the basis of some context parameters. Then agents' utility functions can remain constant while their desires can dynamically change. This makes the agents' behavior more flexible. The application domain of the multi-agent system is washing clothes in one washing machine. There are two utilities in the utility function: washing time, and certainty of electricity delivery.

In [Kephart and Greenwald 2002], Kephart and Greenwald study markets consisting of shopbots and other agents representing buyers and sellers in which shopbots and agents are economically

motivated, strategically pricing their information services and selecting search strategy respectively so as to maximize their own profits. Whenever a rational buyer is fully informed by shopbots, it makes an optimal decision regarding which search strategy to employ to find the lowest-priced seller among a randomly chosen set of sellers, given the current state of the market. The optimal decision making is based on a multi-attribute utility function that specifies the expected profit per unit time of a seller. The attributes include the quantified search strategy, the seller's price, and the cost of production for seller.

4.2.2. Expected Utility Functions

In [Li and Soh 2004], Li and Soh create a multi-phase coalition formation model integrating case-based reasoning and reinforcement learning. Coalition formation is implemented through argumentation-based negotiations [Soh and Tsatsoulis 2001] between agents. For a negotiation-responding agent to decide whether to accept, reject, or counteroffer a request, it uses a utility function with the attributes corresponding to the domain information, agents' cooperation relationship, and so on. To compute the utility of a case, multiple multi-attribute utility functions are used and form a hierarchical structure, i.e., the outcome of one utility function is used as one attribute in another utility function. To select agents as coalition candidates, the authors build an expected utility function to compare the expected utilities of the actions of selecting candidates. They set different utility values for all possible coalition formation outcomes. To get the corresponding probability values of outcomes, they adopt the neighbor profiling technique. Each agent keeps track of its coalition formation history with others and records each neighbor agent's coalition execution success rate, coalition execution failure rate, negotiation success rate, and negotiation failure rate, to estimate the probabilities of different coalition formation outcomes.

In [Excelente-Toledo and Jennings 2003], agents can dynamically select coordination mechanisms based on expected utility functions. When deciding which of its coordination mechanisms to adopt, the agent computes the expected utility of each of them and selects the one that maximizes this value. The agent's aims are to maximize their reward, in particular their average reward per unit time. Each agent keeps track of its own average reward, and uses this reward to decide how much to charge for its own services and occasionally to approximate the expected average reward of other agents. Taking account of the reward and the success probability of a coordination mechanism, the agent can compute the expected utility of the

coordination mechanism. Here the alternative coordination mechanisms and their success probabilities are defined at the beginning and are constant.

In [Vane and Lehner 2000], Vane and Lehner propose an approach to the standard two-player zero-sum single-stage normal game that maximizes expected gain while quantifying possible loss. Agents use this formulation to select a plan based on its assessment of an opponent's intent, its assessment of an opponent's unpredictability, and its utility model of the situation. The plan selection problem is represented using an extended hypergame formulation and the plans are evaluated using hypergame expected utility. Each candidate plan is called a hyperstrategy and the hyperstrategy can determine a probability matrix. This probability matrix represents the expected probability (a weight) of each entry in the full game. The hypergame expected utility is then determined by performing a dot product of this matrix with the utility values in the full game. Such an expected utility function is more complex than a generic expected utility function. The authors conduct a series of experiments to conclude that hypergame expected utility is a robust, useful evaluation of the desirability of any hyperstrategy.

4.3. Applications of Decision Trees

As a decision making and decision-analysis tool, a decision tree can aid the decision maker to produce policies, and visualize the structuring and solving of decision situations. Decision trees can be used conveniently in multi-agents systems since there are mature decision tree algorithms available (e.g., C4.5 [Stone and Veloso 1998, Chiu and Webb 1988] and C5.0 [Nair et al. 2004a]).

However, although decision trees are widely used for classification tasks, they are typically not used for agent control. In [Stone and Veloso 1998], Stone and Veloso use decision trees for agent control in a complex multi-agent domain, Robotic Soccer, based on the confidence factors provided by the C4.5 decision tree algorithm. They incorporate a previously trained decision tree into a full multi-agent behavior that is capable of controlling agents throughout an entire game. Along with using decision trees for control, this behavior also makes use of the ability to reason about action-execution time to eliminate options that would not have adequate time to be executed successfully.

An agent may model others to predict their future actions. But the possible constraints of inadequate or contradictory relevant historical evidence can result in low prediction accuracy, or otherwise, low prediction rates, leaving a set of cases for which no predictions are made. In [Chiu and Webb 1988], Chiu and Webb use decision trees, specifically the C4.5 decision tree algorithm, for agents' modeling to others, and aim to improve prediction rates without affecting prediction accuracy. An agent-modeling system based on C4.5 is used to model agents' competencies with a set of decision trees, trained on all historical data. Each tree predicts one particular aspect of the agent's action. Predictions from multiple trees are compared for consensus. The agent-modeling system makes no prediction when predictions from different trees contradict one another. This strategy trades off reduced prediction rates for increased accuracy.

Decision trees are often used for agents' learning about own decisions (e.g., [Stone and Veloso 1998]) or for modeling others (e.g., [Chiu and Webb 1988]) in the presence of large amounts of data. Unlike these approaches that use decision trees as a model of prediction of agent behavior in unseen cases, Nair et al. [Nair et al. 2004a] use decision trees as a model to explain observed agent behavior, i.e., using decision trees as a decision-analysis tool. They develop an automated team analyst called ISSAC for post-hoc, off-line agent-team analysis on agents' behavior in teamwork. ISSAC employs multiple presentation techniques that can aid human understanding of the analyses. Decision trees can help to extract key features that discriminate between success and failure of critical actions, and extract rules for "what-if" analysis. The user submits logs of the team's behavior along with what are considered to be critical events, and also along with chosen features. The individual agent model uses the C5.0 decision tree algorithm to come up with rules that explain these examples, and when a user selects a particular rule, show the user all those cases of examples satisfying the selected rule.

In [Sridharan and Tesauro 2002], Sridharan and Tesauro study the use of single-agent and multi-agent Q-learning to learn seller-pricing strategies using a regression tree approximation scheme to represent the Q-functions. Q-learning is one of a variety of ways of endowing agents with the "foresight" ability to anticipate long-term consequences of actions for planning strategies to achieve desirable goals. As a special type of decision trees, *regression trees* [Breiman et al. 1984] are used here to represent the Q-functions. As with all regression techniques it is assumed

that there is a single response variable and one or more predicator variables. If the response variable is *categorical* then classification or decision trees are created. If the response variable is *continuous* then regression trees can be produced. Predicator variables can be a mixture of continuous and categorical variables. The final output is a tree where the decision maker decides which branch to follow after applying some test to one or more variables. Sridharan and Tesouro use axis-parallel splits, select splits that minimize variance, and approximate the function by constant values in the leaf nodes. The trees are constructed in a “batch” mode using a fixed set of training cases. Each training case has some input attribute values, and an associated function value which may be adjusted during training. Through the application of regression trees, stable seller pricing strategies can be learned out.

4.4. Applications of Bayesian Networks and Influence Diagrams

In recent years, the applications of Bayesian networks and their extensions called influence diagrams in multi-agent systems are becoming popular (e.g., [Gmytrasiewicz and Noh 2002], [Vassileva and Mudgal 2002]). It is very natural since Bayesian networks represent probabilistic relations among uncertain variables describing the domain at hand and there are varieties of relations between an agent and its environments and there are a great amount of uncertainty factors in these relations. Together with probability distribution and utility function, Bayesian networks and influence diagrams play a significant role in agents’ reasoning and planning based on Bayes’ theorem (see Section 2.5.4). Bayesian networks concern probabilistic relationship among uncertain variables but do not concern utility and decision variables. They are not appropriate for modeling complex decision making processes. Generally, an agent’s decision making processes can be modeled using an influence diagram, a Bayesian network extended with utility function and decision variables. Here we present their applications together just because their application scenarios are very similar. In some applications, the authors do not even clearly distinguish them and just use the two terms alternatively (e.g., in [Gmytrasiewicz and Noh 2002]).

In [Gmytrasiewicz and Noh 2002], Gmytrasiewicz and Noh present the implementation of knowledge bases of the agents that accommodate uncertainty and nested information agents may have about the world. As stated in Section 4.1.1, their design of the knowledge base combines the frame formalisms with Bayesian networks. The slots of frames (or attributes of objects)

describing the properties of objects in the world that may not be known with certainty are treated as nodes of a Bayesian network, or an influence diagram. Such probabilistic slots also contain information about the slots' parent nodes in the influence diagram, as well as the conditional probability tables that allow the probabilities to be updated in response to change in the parents' probabilities. In the implementation, as new objects are identified by the agent, they are automatically represented as objects belonging to appropriate classes in the frame-based knowledge base, and automatically become part of the influence diagram representation of the agent's decision making situation. As the authors proved, each influence diagram has a corresponding and unique payoff matrix representing the same decision making situation. Through combining the traditional knowledge representation form with Bayesian networks, the limitation of traditional form in representing uncertainty can be overcome. More importantly, decision theory provides a good theoretical support to the use of Bayesian networks and the Bayesian representation is more helpful to the decision making of agents in dynamic and uncertain environments than the traditional frame-based or object-oriented form as the application of Bayesian networks makes it possible to generate the Bayesian representation of decision making situation on-the-fly.

In [Vassileva and Mudgal 2002], the influence diagram technique is used in agent negotiation with incomplete and uncertain information, in the context of a distributed multi-agent peer help system supporting students in a university course. Personal agents bilaterally negotiate on their behalf to acquire help from other agents. The agent's decision making takes into account the preferences of the user, which depend on the domain of the negotiation. Ideally (as often is assumed in cooperative environments) negotiating parties have full knowledge about the opponents. When the agents are self-interested, however, it is unlikely that an agent is willing to share its private preferences with other agents. To cope with the uncertainty inherent in a dynamic environment with self-interested participants and negotiate more effectively, an agent models the preferences of the opponent using an influence diagram illustrated in Figure 4.1. The agents negotiate iteratively and create preference models of their opponents during negotiation, which help them predict their opponents' actions and make decisions better. In Figure 4.1, the only deterministic node represents the certainty and other chance nodes represent the uncertainty. The right side is a sub influence diagram for the opponent model. The outcomes of the opponent's action node are the probabilities that an opponent can decide to accept, reject, or

counteroffer. At every step the agents choose between these protocol actions by calculating the maximum expected utility for the actions. The domain-specific utility functions are created and incorporated into the probabilistic inference diagram. The utility of a negotiation decision depends on the role in which the agent is at the moment of decision making. The utilities of different roles at different states vary according to their risk behaviors.

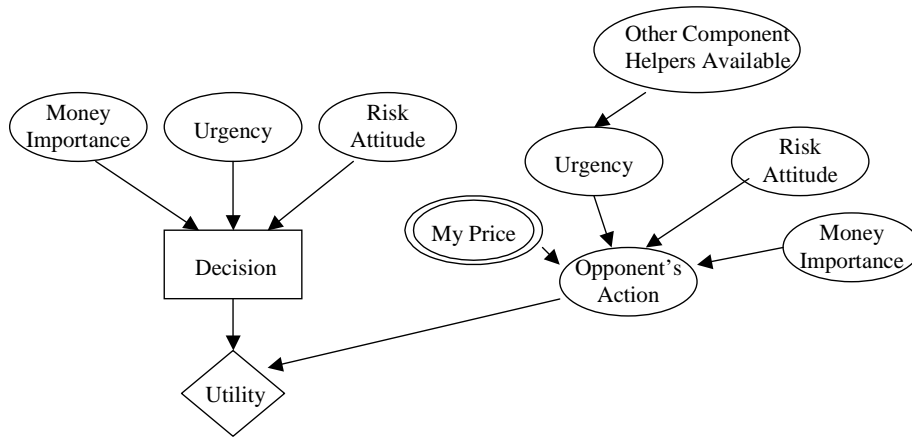


Figure 4.1. A practical influence diagram used in agents' decision making.

The application of influence diagrams is facilitated by its relatively unconstrained dependency structure at the level of relation. Since in Bayesian networks the inference based on Bayes' theorem have obtained wide applications in many areas, the application of influence diagrams has a solid theoretical background. To update the inferred probabilities to reflect the changing state of the world, Bayes' update rule can be used to recalculate the probabilities.

4.5. Applications of MDPs and POMDPs

Markov decision processes (MDPs) apply to the decision problems where the state information can be observed completely and have been used as the basis for much work in decision-theoretic planning. In most real-world decision problems, the problem settings are of state uncertainty where the state information is partially observable. Partially observable Markov decision processes (POMDPs) are more flexible as they permit uncertainty of observations and state information acquisition.

In multi-agent systems, there have been a variety of applications based on MDPs and POMDPs such as the multiagent Markov decision process (MMDP) model [Boutilier 1996], the identical payoff stochastic game (IPSG) and the partially observable identical payoff stochastic game

(POIPSG) [Peshkin et al. 2000], the decentralized Markov decision process (DEC-MDP) model and the decentralized partially observable Markov decision process (DEC-POMDP) model [Bernstein et al. 2002], the communicative multiagent team decision problem (COM-MTDP) model [Pynadath and Tambe 2002], the Dec-POMDP-Com model [Goldman and Zilberstein 2003], and the distributed POMDP model [Nair et al. 2004b]. MDPs and POMDPs are applied to model the state uncertainty in inter-agent coordination. In coordination, the agents may communicate to exchange information and synchronize behavior dynamically. Here, some applications integrate communication into the model while others not.

The MMDP model [Boutilier 1996] is a general model to coordinate the policies of individual agents in n -person cooperative games in which agents share the same utility function. Boutilier adopts MDPs as the underlying (single agent) decision model because the research interest is in planning under uncertainty with competing objectives and (potentially) indefinite or infinite horizon. An MMDP is formalized as a tuple of (1) a finite set of states, (2) a finite set of agents, (3) a series of finite sets of actions corresponding to each agent, (4) a probabilistic state transition function, and (5) a real-valued reward function. Each agent has prior beliefs about the policies of other agents and these beliefs are updated as the agents act and interact. The MMDP model is a multi-agent extension to the completely observable MDP model, so it assumes an individually fully observable environment. The MMDP model has no communication.

The IPSPG model [Peshkin et al. 2000] is a multi-agent MDP model and the POIPSG model is a multi-agent POMDP model. They are developed for multi-agent decision making in cooperative stochastic games, where the agents may have their own individual goals and preferences but share the same payoff structure. The POIPSG model is a tuple of (1) a set of states, (2) a probability distribution over the initial state, (3) a set of agents, where each agent is a 3-tuple of its discrete action space, discrete observation space, and observation function, (4) a probabilistic transition function, and (5) a reward function. This tuple is a generic one for multi-agent POMDPs. In the one-agent case, the model is essentially same as a generic POMDP model. When all agents have the identity observation function for all states, i.e., each state is uniquely determined by an observation, the game is completely observable. Then the model is an IPSPG model. The POIPSG model restricts the agents to share a single payoff function, appropriate for

modeling the single, global reward function of the team context. There is no communication in either model.

The DEC-POMDP [Bernstein et al. 2002] model is a general decentralized model. In this model, the decision process is controlled by multiple distributed agents, each with possibly different information about the state. A DEC-MDP is a DEC-POMDP with the restriction that at each time step the agents' observations together uniquely determine the state. The tuple of a DEC-POMDP is consistent with the generic tuple of multi-agent POMDPs described above. There is no communication in either model.

The COM-MTDP model [Pynadath and Tambe 2002] is a decentralized POMDP model and its application domain is the team coordination in teamwork. It is originated from STEAM [Tambe 1997] that is developed based on the BDI (Belief-Desire-Intention) model and extends joint intentions with decision-theoretic communication selectivity. The COM-MTDP model also has extension to explicitly represent communication. So the tuple of a COM-MTDP includes a new component representing communication. The communication is introduced to find locally optimal joint policies that allow agents to coordinate better through synchronization achieved via communication. Compared to the previously described models, the most significant difference of this model is there is communication.

The Dec-POMDP-Com model [Goldman and Zilberstein 2003] is a decentralized POMDP model for the decentralized control of cooperative multi-agent systems. There is communication in this model for dynamic information exchange between agents. Within the model, cooperative agents are represented by finite state controllers, whose actions control the process. The model treats both standard actions and communication as explicit choices that the decision maker must consider. The goal is to derive both action policies and communication policies that together optimize a global value function. In the model there are alternate communication and action phases.

The distributed POMDP model [Nair et al. 2004b] is evolved from the COM-MTDP model and is also for modeling multi-agent teamwork. Its tuple is very similar to the generic tuple of multi-agent POMDPs without communication as an explicit component of the tuple. But there is a communication action introduced into the tuple that can be initiated by any agent just like a

generic action initiated. Unlike the COM-MTDP model and the Dec-POMDP-Com model, where there are alternate communication and action phases, there is no separate communication phase in this model. In a particular epoch an agent can either choose to communicate or act. This setting models the missed opportunity cost that occurs when the agents communicate instead of acting.

4.6. Applications of Nash Equilibria and Pareto Equilibria

In multi-agent systems of self-interested agents, each agent tries to maximize its own benefits. In order to solve goals that require the action of multiple agents, coordination is needed and a joint option of agents may be arrived in which each agent's option is acceptable to it. Nash equilibria have been used widely in multi-agent systems to achieve a joint option or just used as an analysis tool for agents' self-interested behavior. Pareto equilibria have also been used in multi-agent systems to achieve a joint option when self-interested agents show a cooperative behavior.

In [Kephart and Greenwald 2002] (also see Section 4.2), Kephart and Greenwald build a multi-agent system of a set of self-interested agents: shopbots, buyers, and sellers, all of which are economically motivated to maximize their own profits. To get a joint option of sellers seeking to maximize profit, the authors derive a Nash equilibrium—a vector of prices at which sellers maximize their individual profits, and from which no seller has any incentive to deviate. If all buyers choose sellers at random, the unique Nash equilibrium is such that all sellers charge the monopoly price. Otherwise, there may exist multiple Nash equilibria. Specific to such issues as when there are multiple equilibria and how the shopbot can control which equilibrium is reached regardless of initial conditions, Kephart and Greenwald point out the trick is to use a time-dependent pricing strategy to strategically manipulate the equilibria and their basins of attraction so as to guide the market towards the desired equilibrium.

In [Markopoulos and Ungar 2002], Markopoulos and Ungar explore the role of shopbot and pricebot software agents in electronic service markets. They consider a stream of customers that arrive in a market and choose a seller from which they receive service based on their expected utility costs. The authors analyze the possibility of getting Nash equilibrium. They address that there exists no symmetric pure Nash equilibrium in a one-shot game that the sellers face, since the sellers are identical and will only set their price once, making such equilibria unrealistic.

Even pricing at zero is also not an equilibrium since instead of making zero profits a seller would raise his price and increase revenue from buyers that occasionally come and find all other sellers with non-zero expected queue waiting time. Consequently, the market cannot be in equilibrium.

In [Scully et al. 2004], Scully et al. present a solution to coalition calculation in a dynamic multi-agent environment. In order to obtain a true valuation of any coalition, they use the concept of Pareto equilibrium. They propose an algorithm called E-Pareto, which is based on a multi-objective optimization evolutionary algorithm combining multiple-objective decision making and evolutionary computation. The combination of Pareto equilibria and the evolution algorithm allows for the approximation of the Pareto optimal set of coalitions. A distance weighting algorithm is also incorporated to maintain diversity when searching for the Pareto optimal solution set, and to encourage search in areas of solution space that have been previously successful. The proposed technique is capable of eliciting metric importance and adapting to metric variation over time.

4.7. Summary

In this section, we have listed and described a variety of application examples of typical decision-theoretic concepts and methodologies in multi-agent systems. It can be seen that their applications are often interdependent. Different concepts and methodologies may be used in a same scenario at the same time for different purposes, and one concept or methodology can be a part of another one. For example, probability as well as probability distribution is a key concept in decision making and it appears in expected utility functions, decision trees, Bayesian networks, influence diagrams, MDPs, and POMDPs.

5. Conclusions

In this report, the basic mechanisms in decision theory, probability theory, utility theory, and game theory, the main applicable aspects of these theories in multi-agent systems, and their various applications have been presented.

In a multi-agent system, each agent has its own beliefs about how the world is, has desires about how it would like the world to be like, and has intentions about how it can make the world to be like. An autonomous agent needs to make decisions based on currently held beliefs to take

specific actions and achieve specified goals. These aspects correspond to the primary elements in a decision problem: states of nature, decision alternatives, and payoffs.

Due to the uncertainty in agents' viewpoint to the environment and other agents, and the uncertainty of outcomes of action in real-world environments, agents often need to make decisions for the estimation of states of the world, prediction and evaluation of outcomes of action, strategy planning, achieving the joint option of multiple agents, and so on. For effective decision making, special theoretic concepts and methodologies are needed. Up to date, some typical ones have been applied into multi-agent systems, for examples, probability distribution, (expected) utility functions, decision trees, Bayesian networks, influence diagrams, Markov decision processes (MDPs), partially observable Markov decision processes (POMDPs), Nash equilibria, and Pareto equilibria.

In this report, we have described these concepts or methodologies, and discussed their application areas in multi-agent systems from the perspective of BDI architectures. The BDI architecture is consistent with the human practical reasoning and provides the functional decomposition, which enable us to understand and discuss the decision-theoretic requirement in multi-agents more clearly and more systematically. We also describe a variety of example applications of these theoretic tools in multi-agent systems. Although these decision-related theories have been used in multi-agent systems very widely, some tools are not used sufficiently. For example, the use of Bayesian networks and influence diagrams is often limited within the representation of the specific causal relationship rather than the bi-directional inference.

What we also need to point out is that some applications of concepts and methodologies are too complicated in cases of large sets of data like the use of decision trees, Bayes networks, influence diagrams, MDPs, and POMDPs. How to model complex decision problems with appropriate decision-theoretic concepts and methodologies, how to effectively represent the complex structure of a diagram of a large set of decision information, and how to efficiently infer in diagrams are crucial problems.

References

[Apostol 1969] Apostol, T. M. *Calculus, Volume II*. 2nd edition, John Wiley & Sons, 1969.

[Axelrod 1984] Axelrod, R. *The Evolution of Cooperation*, New York: Basic Books, 1984.

[Balogh et al 2000] Balogh, Z., Laclavik, M., and Hluchy, L. *Multi Agent System for Negotiation and Decision Support*. Herlany, Slovakia, 2000.

[Banerje and Sen 2002] Banerje, B. and Sen, S. Selecting Partners. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 29-42, Kluwer Academic Publishers, 2002.

[Barber et al. 2000] Barber, K. S., Han, D. C., and Liu, T. H. Strategy Selection-Based Meta-Level Reasoning for Multi-Agent Problem Solving. *Agent Oriented Software Engineering 2000*, Limerick, England, 2000.

[Bazzan et al. 2002] Bazzan, A. L. C., Bordini, R. H., and Campbell, J. A. Evolution of Agents with Moral Sentiments in an Iterated Prisoner's Dilemma Exercise. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 43-64, Kluwer Academic Publishers, 2002.

[Bernstein et al. 2002] Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4): 819-840, 2002.

[Binmore 1992] Binmore, K. *Fun and Games: A Text on Game Theory*. Lexington, MA: D. C. Health and Company, 1992.

[Boutilier 1996] Boutilier, C. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195-210, Amsterdam, the Netherlands, 1996.

[Boutilier et al. 1999] Boutilier, C., Dean, T., and Hanks, S. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 11: 1-94, 1999.

[Brachman and Levesque 1985] Brachman R. and Levesque, H. *Readings in Knowledge Representation*. Morgan Kaufman, 1985.

- [Bratman 1987] Bratman, M. E. *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.
- [Breiman et al. 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees*. Monterey, CA: Wadsworth, 1984.
- [Burger 1959] Burger, E. *Introduction to the Theory of Games*. Englewood Cliffs, NJ: Prentice-Hall, Inc. Translated by Freund, J. E., 1963.
- [Cassandra et al. 1994] Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. Acting Optimally in Partially Observable Stochastic Domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 1994.
- [Chiu and Webb 1988] Chiu, B. and Webb, G. Using Decision Trees For Agent Modeling: Improving Prediction Performance. *User Modeling and User-Adapted Interaction*, 8(1-2): 131-152, 1988.
- [Dastani et al. 2002] Dastani, M., Huang, Z., and Torre, L. V. D. Dynamic Desires. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 65 - 80, Kluwer Academic Publishers, 2002.
- [Excelente-Toledo and Jennings 2003] Excelente-Toledo, C. B. and Jennings, N. R. The Dynamic Selection of Coordination Mechanisms. *Journal of Autonomous Agents and Multi-Agent Systems*, 9(1-2): 55-85, 2004.
- [Feller 1968] Feller, W. *An Introduction to Probability Theory, Volume I*. New York, NY: Wiley, 1968.
- [Fishburn 1970] Fishburn, P. C. *Utility Theory for Decision Making*. New York: John Wiley & Sons, Inc., 1970.
- [Gmytrasiewicz and Lisetti 2002] Gmytrasiewicz, P. J. and Lisetti, C. L. Emotions and Personality in Agent Design and Modeling. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 81-96, Kluwer Academic Publishers, 2002.

[Gmytrasiewicz and Noh 2002] Gmytrasiewicz, P. J. and Noh, S. Implementing a Decision-Theoretic Approach to Game Theory for Socially Competent Agents. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 97-118, Kluwer Academic Publishers, 2002.

[Goldman and Zilberstein 2003] Goldman, C. V. and Zilberstein, S. Optimizing Information Exchange in Cooperative Multi-Agent Systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-03)*, Melbourne, Australia, 2003.

[Gupta and Nau 1992] Gupta, N. and Nau, D. S. On the Complexity of Blocks-World Planning. *Artificial Intelligence*, 56(2-3): 223-254, 1992.

[Hill et al. 1982] Hill, P. H., Bedau, H. A., Chechile, R. A., Crochetiere, W. J., Kellerman, B. L., Ounjian, D., Pauker, S. G., Pauker, S. P., and Rubin, J. Z. *Making Decisions: A Multidisciplinary Introduction*. Reading, Massachusetts: Addison-Wesley Publishing Company (Advanced Book Program), 1982.

[Horvitz et al. 1986] Horvitz, E.J., Heckerman, D.E., and Langlotz, C.P. A Framework for Comparing Alternative Formalisms for Plausible Reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pages 210-214, Philadelphia, PA, San Mateo, CA: Morgan Kaufmann, 1986.

[Horvitz et al. 1988] Horvitz, E., Breese, J., and Henrion, M. Decision Theory in Expert Systems and Artificial Intelligence. *Journal of Approximate Reasoning, Special Issue on Uncertainty in Artificial Intelligence*, 2: 247-302, 1988.

[Howard 1960] Howard, R. *Dynamic Programming and Markov Processes*. Cambridge, MA: the MIT Press, 1960.

[Howard and Matheson 1984] Howard, R. A. and Matheson, J. E. Influence Diagrams. In Howard, R. A. and Matheson, J. E. (eds.) *Readings on the Principles and Applications of Decision Analysis*, pages 721-762, Strategic Decisions Group, Menlo Park, CA, 1984.

[Huhns and Stephens 1999] Huhns, M. N. and Stephens, L. M. Multiagent Systems and Societies of Agents. In Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 79-120, Cambridge, MA: the MIT Press, 1999.

[Keeney and Raiffa 1976] Keeney, R. and Raiffa, H. *Decision with Multiple Objectives: Preferences and Value Trade-offs*. New York: John Wiley & Sons, Inc., 1976.

[Kephart and Greenwald 2002] Kephart, J. O. and Greenwald, A. R. Shopbot Economics. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 119-158, Kluwer Academic Publishers, 2002.

[Kraus et al. 1998] Kraus, S., Sycara, K., and Evenchik A. Reaching Agreements through Argumentation: A Logical Model and Implementation. *Artificial Intelligence Journal*, 104(1-2):1-69, 1998.

[Lee 1971] Lee, W. *Decision Theory and Human Behavior*. New York: John Wiley & Sons, Inc., 1971.

[Li and Soh 2004] Li, X. and Soh, L.-K. Learning-Based Multi-Phase Coalition Formation. In *Proceedings of the Workshop on Coalitions and Teams at AAMAS-04*, pages 9-16, New York City, NY, 2004.

[Luce and Raiffa 1957] Luce, R. D. and Raiffa, H. *Games and Decisions: Introduction and Critical Survey*. New York, NY: Dover, 1957.

[Markopoulos and Ungar 2002] Markopoulos, P. M. and Ungar, L. H. Shopbots and Pricebots in Electronic Service Markets. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 177-195, Kluwer Academic Publishers, 2002.

[Nair et al. 2004a] Nair, R., Tambe, M., Marsella, S., and Raines, T. Automated Assistants for Analyzing Team Behaviors. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(1): 69-111, 2004.

- [Nair et al. 2004b] Nair, R., Tambe, M., Roth, M., and Yokoo, M. Communication for Improving Policy Computation in Distributed POMDPs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04)*, pages 1098-1105, New York City, NY, 2004.
- [Nash 1950] Nash, J. Equilibrium Points in N-Person Games. *Proceedings of the National Academy of Sciences*, 36: 48-49, 1950.
- [Neapolitan 1990] Neapolitan, R. E. *Probabilistic Reasoning in Expert Systems*. New York: John Wiley & Sons, Inc., 1990.
- [Newell 1981] Newell, A. The Knowledge Level. *AI Magazine*, 2(2): 1-20, 1981.
- [Parsons et al. 2002] Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, Kluwer Academic Publishers, 2002.
- [Pearl 1988] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, 1988.
- [Peshkin et al. 2000] Peshkin, L., Kim, K.-E., Meuleau, N., and Kaelbling, L. P. Learning to Cooperate via Policy Search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 489-496, Stanford, CA, 2000.
- [Pynadath and Tambe 2002] Pynadath, D. V. and Tambe, M. The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models. *Journal of Artificial Intelligence Research*, 16: 389-423, 2002.
- [Raiffa 1968] Raiffa, H. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. MA: Addison Wesley, Reading, 1968.
- [Raiffa 1982] Raiffa, H. *The Art and Science of Negotiation*. Cambridge, MA: Harvard University Press, 1982.

[Rao and Georgeff 1995] Rao, A. S. and Georgeff, M. P. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 312-319, San Francisco, CA, 1995.

[Sandholm 1999] Sandholm, T. W. Distributed Rational Decision Making. In Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201-258, Cambridge, MA: the MIT Press, 1999.

[Sandholm and Lesser 1997] Sandholm, T. and Lesser, V. Coalitions among Computationally Bounded Agents. *Artificial Intelligence*, 94(1): 99-137, 1997.

[Scully et al. 2004] Scully, T., Madden, M. G., and Lyons, G. Coalition Calculation in a Dynamic Agent Environment. In *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Canada, 2004.

[Shachter 1988] Shachter, R. Probabilities Inference Diagrams. *Operations Research*, 34(36): 871-882, 1986.

[Singh et al. 1999] Singh, M. P., Rao, A. S., and Georgeff, M. P. Formal Methods in DAI: Logic-Based Representation and Reasoning. In Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: the MIT Press, 1999.

[Soh and Tsatsoulis 2001] Soh, L.-K. and Tsatsoulis, C. Agent-Based Argumentative Negotiations with Case-Based Reasoning. In *Working Notes of the AAAI Fall Symposium Series on Negotiation Methods for Autonomous Cooperative Systems*, pages 16-25, North Falmouth, MA, 2001.

[Soh et al. 2003] Soh, L.-K., Tsatsoulis, C., and Sevay, H. A Satisficing, Negotiated, and Learning Coalition Formation Architecture. In Lesser, V., Ortiz, C. L., and Tambe, M. (eds.) *Distributed Sensor Networks: A Multiagent Perspective*, pages 109-138, Kluwer Academic Publishers, 2003.

[Sridharan and Tesauro 2002] Sridharan, M. and Tesauro, G. Multi-Agent Q-Learning and Regression Trees for Automated Pricing Decisions. In Parsons, S., Gmytrasiewicz, P., and

Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 217-234, Kluwer Academic Publishers, 2002.

[Stirling et al. 2002] Stirling, W. C., Goodrich, M. A., and Packard, D. J. Satisficing Equilibria. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.). *Game Theory and Decision Theory in Agent-Based Systems*, pages 235-265, Kluwer Academic Publishers, 2002.

[Stone and Veloso 1998] Stone, P. and Veloso, M. Using Decision Tree Confidence Factors for Multiagent Control. In Kitano, H. (ed.) *RoboCup-97: Robot Soccer World Cup I*, pages 99-111, Berlin: Springer Verlag, 1998.

[Sycara 1988] Sycara, K. P. Utility Theory in Conflict Resolution. *Annals of Operation Research*, 12: 65-84, 1988.

[Tambe 1997] Tambe, M. Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, 7: 83-124, 1997.

[Vane and Lehner 2000] Vane, R. and Lehner P. Using Hypergames to Increase Planned Payoff and Reduce Risk. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 315-336, Kluwer Academic Publishers, 2002.

[Vassileva and Mudgal 2002] Vassileva, J. and Mudgal, C. Bilateral Negotiation with Incomplete and Uncertain Information. In Parsons, S., Gmytrasiewicz, P., and Wooldridge, M. (eds.) *Game Theory and Decision Theory in Agent-Based Systems*, pages 337-354, Kluwer Academic Publishers, 2002.

[von Neumann and Morgenstern 1947] von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[Weiss 1999] Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: the MIT Press, 1999.

[Wendt and Vlek 1975] Wendt, D. and Vlek, C. (eds.) *Utility, Probability, and Human Decision Making*. D. Reidel Publishing Company, 1975.

[White 1969] White, D. J. *Decision Theory*. Chicago: Aldine Pub. Co. 1969.

[Wooldridge 1999] Wooldridge, M. Intelligent Agents. In Weiss, G. (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: the MIT Press, 1999.