Exploiting Focal Points Among Alternative Solutions: Two Approaches

Sarit Kraus
Department of Mathematics and Computer Science
Bar-Ilan University
Ramat-Gan, 52900 Israel and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
sarit@umiacs.umd.edu

Jeffrey S. Rosenschein
Department of Computer Science
Hebrew University
Givat Ram, Jerusalem, Israel
jeff@cs.huji.ac.il

Maier Fenster
Department of Mathematics and Computer Science
Bar-Ilan University
Ramat-Gan, 52900 Israel
fenster@cs.biu.ac.il

Abstract

Focal points refer to prominent solutions of an interaction, solutions to which agents are drawn. This paper considers how automated agents could use focal points for coordination in communication-impoverished situations. Coordination is a central theme of Distributed Artificial Intelligence. Much work in this field can be seen as a search for mechanisms that allow agents with differing knowledge and goals to coordinate their actions for mutual benefit. Additionally, one of the main assumptions of the field is that communication is expensive relative to computation. Thus, coordination techniques that minimize communication are of particular importance.

Our purpose in this paper is to consider how to model the process of finding focal points from domain-independent criteria, under the assumption that agents cannot communicate with one another. We consider two alternative approaches for finding focal points, one based on decision theory, the second on step-logic. The first provides for a more natural integration of agent utilities, while the second more successfully models the difficulty of finding solutions. For both cases, we present simulations over randomly generated domains that suggest that focal points can act as an effective heuristic for coordination.

Key words: Coordination, distributed AI, decision theory, step-logic.

1 Introduction

Coordination is a central theme of Distributed Artificial Intelligence (DAI). Much of the work in this field can be seen as a search for mechanisms that will allow agents with differing views of the world, and possibly with different goals, to coordinate their actions for mutual benefit. In this paper, we present techniques based on the concept of *focal points* [68] for coordination in communication-impoverished situations. Focal points refer to prominent solutions of an interaction, solutions to which agents are drawn.

1.1 Communication-Impoverished Interaction

Work in DAI has actively explored coordination techniques that require little or no communication. Researchers in this area may allow some limited communication in their models, especially insofar as it is required to establish problem constraints. So, for example, in [31], agents are assumed to perceive jointly an interaction (the joint perception could conceivably involve communication), and then proceed without further communication. Similarly, there have been attempts to get multiple agents to interact effectively with little communication, while allowing the sensing of other agents' external actions or conditions (e.g., location) [53].

Another motivation for studying communication-impoverished interactions, other than the expense of communication, has been that communication is sometimes impossible (agents may speak different languages) or inconsistent with the environment (communication has been cut off or is inadvisable in the presence of hostile forces). There has also been a deep-seated intuition that humans are sometimes capable of sophisticated interaction with little explicit communication and that it ought to be possible for automated agents to emulate this.

1.2 Communication-Rich Interaction

Other researchers have freely incorporated communication of all kinds into their models of interaction. Again, the analogy to humans and human organizations is clear: communication is an indispensable aid to interaction, in setting up problem parameters, exchanging information, and coordinating action. Within DAI, this focus on communication has become embodied by a single over-burdened word, *negotiation.* As used by different researchers, the term has come to mean all kinds of communication that further coordination, ranging from the exchange of Partial Global Plans [20], to communication of information intended to alter other agents' goals [74], to incremental offers and counter-offers leading to joint plans of action [79, 50, 51].

As Gasser has pointed out [29], "'negotiation' [is] a term that has been used in literally dozens of different ways in the DAI literature." Even though the word has been over-used, its introduction into the DAI community has served a useful purpose: it has focused researchers' attention on the multiple uses of explicit communication to achieve coordination.

There is also a powerful intuitive appeal to the idea of negotiation, precisely because it is clear that various *kinds* of negotiation play an essential role in real-world interactions. Exploiting similar techniques in automated agents has become a central concern of DAI.

In this paper, we present another technique, namely *focal points*. Although the term is less well-known than "negotiation," it plays an important and ubiquitous role in both communication-impoverished and communication-rich human interactions. Just as the incorporation of negotiation techniques may make agents better able to coordinate themselves, the exploitation of focal points in automated agents holds similar promise. Just as DAI benefits from studying automated negotiation techniques, it can benefit from exploring automated focal point techniques.

This paper presents two approaches to finding focal points, one based on decision theory and the other on a logic approach, which can be used in different domains and settings. We compare the advantages and disadvantages of each approach, along with various simulations (over randomly generated worlds) that demonstrate the basic power of each approach. This paper does not present a complete blueprint about how one would put focal point discovery into an automated agent. Instead, it provides two general frameworks for building such discovery into an agent, with certain elements left to be determined by the implementor from domain-specific considerations.

2 Focal Points

Originally introduced by Schelling [68, 66], focal points refer to prominent solutions of an interaction, solutions to which agents are drawn. Schelling's work on this subject explored a number of simple games where, despite similarity among many solutions, human players were predictably drawn to a particular solution by using contextual information.

2.1 Simplified Focal Point Examples

Before discussing real-world examples of focal points, it is useful to consider a "toy" example that illustrates the concept clearly.

Consider two people who have each been asked to divide 100 identical objects into two arbitrarily-sized piles. Their only concern in deciding how much goes into each pile is to match the other person's behavior. If the two agents match one another, they each win \$40,000; otherwise they get nothing. Schelling found that most people, presented with this scenario, choose an even division of 50 objects per pile. They reason that, since at one level of analysis all choices are equivalent, they must focus on any uniqueness that distinguishes a particular option (such as symmetry) and rely on the other person's doing likewise. A similar problem has each person asked to choose any positive number, with their only concern being to match the other person's choice. Most people seem to choose the number 1, it being the only positive number without a predecessor.

What is interesting about these scenarios is that traditional formal representation techniques, like those of the theory of rational choice where players choose their strategies on the basis of perceived differences in payoffs, cannot capture why people are consistently drawn to a particular solution among many equivalent ones. The notion of equilibrium points (stable strategies) is insufficient—there are 101 such points in the first example above. In addition, they are all identical in terms of payoffs to the agents, and decision theoretic techniques that only consider optimal strategies will be unable to distinguish which action should be chosen. The exploitation of focal points is in large part a knowledge representation issue; we discuss the problems of standard representation techniques at greater length in Section 2.4 below.

Nevertheless, human beings are able to coordinate effectively in these scenarios, using information about the context of the interaction. The symmetry of a 50–50 split has been abstracted away in the traditional game theory representation, but is still available to humans who are considering what action to take.

2.2 Automated Agents' Use of Focal Points

For automated agents, the possibility of exploiting focal points might enable more effective coordination in communication-impoverished scenarios. The key is to allow automated agents to consider the context of their interaction in choosing a coordinated action, as humans do. Since we assume that the agents are rational, the choice should be among action profiles which are equilibria.

Consider a group of automated agents who have been parachuted into enemy territory. The agents are unwilling to communicate for fear of being discovered, but need to meet up at some location. The unpredictability of where each would land, and lack of knowledge about the area prior to the fly-over, have made prior choice of a meeting place impossible. Even choosing some ad hoc rule, such as "meet at the highest point," may be meaningless if they land in a level area or if the highest point is unreachable. Now they each perceive the features of the area, and we want them to coordinate the choice of a meeting place. One

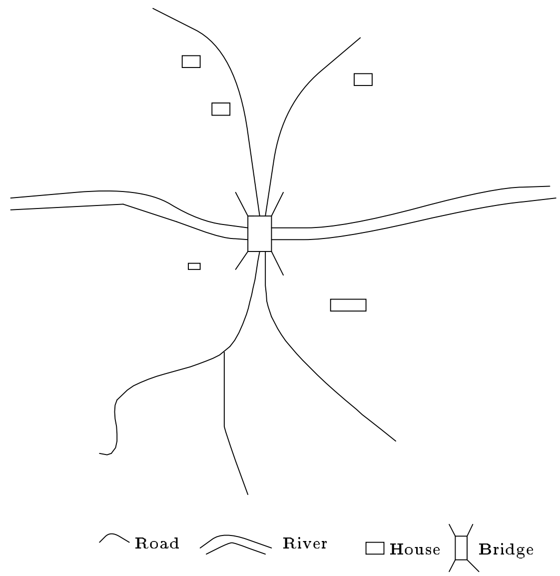

Figure 1: The Parachute Problem.

possibility is for them to choose a prominent meeting place (e.g., if there are many buildings but only one bridge, they might choose the bridge, as in Figure 1).

Consider, as another example, automated agents working together on Mars, who have lost communication with one another (e.g., their radio has developed interference). They would like to meet again so as to reestablish their lines of communication, but need to decide independently where the meeting will take place. The agents could not establish an *a priori* protocol for how to get back together, because they did not have sufficient information about what the terrain would be like. The search for a focal point meeting place would be a natural mechanism for solving this problem.

The Pursuit Problem has generated considerable interest in DAI [5, 47, 53, 71, 72]. In this abstract domain, four blue agents are attempting to coordinate their actions so as to pursue and capture a single red agent. Various protocols have been suggested, with and without explicit communication. The solution of pursuit problems in real-world domains, however, could make use of context that is absent from the abstract Pursuit Problem model. If four allied agents are attempting to capture an enemy agent, and communication is unwise (for fear that their prey would intercept it), then agents might use the concept of focal points to help them decide on a coordinated plan of attack. So, for example, if there were a number of dead ends towards which the agents might drive the prey to, but only one dead end with some particular property (i.e., it is the lowest of the dead ends), the agents might, without communication, coordinate their pursuit towards that point.

In some situations there are competing focal points. If, in such situations, communication is possible but expensive, the focal point techniques can limit the number of options considered in the communication phase. For example, agents searching for a joint plan, where there are a large number of possible joint plans, may limit the number of plans under consideration by discussing only those that are focal points.

Another motivation for having automated agents use focal point techniques is to make their interaction with humans more natural. For example, one of the Mars workers above might be human, and the automated search for a focal point meeting place mirrors his own thought processes. In another example, a robot might be cleaning up an auditorium, come across a forgotten item, and have to reason about where to put it so that it will be found by its owner on the following day.

Our purpose in this paper is to consider how to automate the process of finding focal points from domain-independent criteria, under the assumption that agents cannot communicate with one another.[1] If the agent designer has detailed information about the domain in which his agents will operate, he could build in domain-specific techniques for coordination (e.g., meet at the highest location). If the designer has *perfect* information about the domain, he could build in exact instructions for coordination (e.g., meet at the top of Pike's Peak). We are, in contrast, interested in the case where the designer has considerably less information about the domain and is interested in his agents independently discovering focal points.

Following Schelling, most of the researchers believe that people use their own point of view to identify prominent solutions. As Schelling says: "A prime characteristic of most of these "solutions" to the problems..., is some kind of prominence or conspicuousness." ([68] pp. 57). In particular, Schelling's understanding of focal points is that they are established by some "conventional priority" which is commonly known [2]. Experimental work that has focused on the processes by which people attempt to coordinate their actions supports Schelling's and the other researchers' assumption [56] (see also sections 2.4.4,2.4.5 below.)

However, it is agreed by most of the researchers that identifying a focal point involves two parts:"(a) some formal structure that represents the players' apprehension of the game situation," [8] i.e., an agent tries to guess how the other agent represents the game situation; and "(b) a mechanism to derive a salient option from this structure" [8].

In this paper we present methods for representing the world and mechanisms to derive focal points and assume that the representation and the mechanism are common knowledge. That is, they serve as Schelling's "conventional priority." Following Schelling and the re-

[1] Communication among agents can also affect focal points, but that is beyond the scope of the current paper.

searchers mentioned above, we assume that an agent uses its own point of view of the world when applying our focal point mechanism.

2.3 Intuitive Properties

There are a number of intuitive properties that seem to qualify a given agreement as a focal point. We make no claims for completeness here. These properties provide good coverage of the focal point examples in [68], but additional properties may be appropriate in other cases.

Uniqueness: An object may be a focal point if it is the only object with a given property. For example, in the Parachute scenario, the bridge has the uniqueness property since it is the only object of type "bridge."

Uniqueness Complement: Lack of information can also cause a solution to be prominent. An object may be a focal point if it is the only object without a given property, for example, if there are four houses in the area where the parachutists have landed, $h1, h2, h3, h4$, and the parachuters believe that all the houses but $h3$ are white and have no information about the color of $h3$. This lack of knowledge makes this house prominent.

Centrality: Centrality is the intuitive property of a central point around which a domain (or sub-domain) is symmetric. An object may be a focal point if it is a central object within a given domain. For example, a church that is located in the center of a village has the centrality property.

Extreme: An object can sometimes be prominent because it is the highest object, or the tallest, or the smallest, among the elements of the domain. An object may be a focal point if it is an extreme object in a totally-ordered domain. For example, the highest hill in the area where the parachutists have landed has the extreme property.

Our overall intent is to consider the formal representation techniques that would allow an agent to perceive focal points, so that they can be exploited for communication-free coordination. Even when we consider these special properties, more must be done to identify focal points. *There are bound to be competing potential focal points, since there is* something *unique about any solution.* Another fairly strong contender for a solution in the original game presented above is the choice of 0 objects in A, and 100 objects in B (or vice-versa). Of course, it is precisely the "vice versa" aspect of this solution that makes it appear less appealing in comparison with the 50–50 split.

Any solution, though, will have something to recommend it—but the less obvious that something is, the less attractive the alternative becomes, precisely because it becomes less obvious that the other agent will duplicate our line of reasoning. For example, the choice of 10–90 recommends itself, since it is the only choice where the number of tens in both piles is a perfect square (1 squared and 3 squared), and where, at the same time, the first pile is smaller than the second. And of course, we might choose 16–84 as our split, reasoning that our partner will realize, as we did, that these are the only years in the 20th century (whose last two digits add up to 100) that have seen the election of United States presidents with the same number of letters in their last names (Wilson in 1916 and Reagan in 1984).

This is a farfetched example, but the point should be clear: a focal point is produced not only because it satisfies one of the intuitive principles mentioned above, but because it seems computationally more accessible—it seems more likely that the other agent will also recognize the point than that he will recognize competing points.

2.4 Related Work

2.4.1 The Traditional Game Theory Approach

In game theory, an interaction might be represented as a game in normal form, where agent strategies are condensed into single choices. The possible outcomes of the game comprise pairs of such choices. Typically, the game is represented as a matrix, where each column represents a particular strategy for one agent and each row represents a particular strategy for a second agent. Each element of the matrix represents a particular state and contains values which are the expected payoff as a result of a particular choice of strategies. In a zero sum game, only one value is necessary for each matrix element. In the most basic types of games (e.g., games with complete information), it is assumed that the agents have common knowledge of those final payoffs, that they have unlimited computational power, and that, in particular, they are able to generate the complete game matrix and can find so-called "equilibrium points." Two strategies S and S' are said to be in Nash equilibrium [61] if, assuming that one agent is using strategy S', the best the other agent can do is to use S. An equilibrium point is an outcome resulting from two agents' use of equilibrium strategies. For example, in the game matrix in Figure 2, the strategies whereby agent J chooses move b and agent K chooses move d are in Nash equilibrium.

2.4.2 Multiple Equilibrium Points

In the matrix in Figure 2 there is a second equilibrium point, where agent J chooses move a and agent K chooses move c. This simple game has an inherent symmetry, since both

		K: c	K: d
J	a	2, 1	-1, -1
J	b	-1, -1	1, 2

Figure 2: Two Equilibrium Points.

equilibria are attractive to the agents. Schelling cites this precise game as an instance where contextual clues might help the agents resolve the game's inherent symmetry.

Consider again the original problem given above, with two contestants dividing 100 objects into two piles. The problem can be represented very easily, using a payoff matrix, with the elements of the diagonal being \$40,000 and all the other elements being zero. The shortcoming of this representation is that it does not allow the agents to reason about anything other than the relationships among the payoffs, and these relationships are unenlightening. There are 101 payoffs of \$40,000, but there is no way, within the framework of the payoff matrix, of reasoning about why one action is better than any other (i.e., there are 101 equilibrium points).

Such games are called by game theoreticians *matching games*. A matching game is a pure coordination game[2] in which there are two players with the same strategies; both get a payoff if and only if both choose the same act; and the payoff is the same, whatever this act may be [4]. The symmetries of a matching game have the consequence that, for all standard game theory solution concepts, if the concept recommends any action it recommends all [4], i.e, there are always multiple (equivalent) equilibria.

For games in which the payoff matrix is asymmetric, equilibrium selection theories have been developed which can discriminate between Nash equilibria (e.g., [34, 38, 11]). A significant amount of work has been performed on the evolution of equilibria in coordination games that are played repeatedly within a population (e.g. [78, 77, 42, 6]). Crawford and Haller [12] and later Kramarz [48] investigated how the players of an iterated coordination game can converge on a pattern of coordinated play. Another approach is using cheap-talk which may be roughly defined as non-binding, non-payoff relevant pre-play communication [26, 39] for selecting an equilibrium. However, these papers do not address the problem of choosing between multiple equilibria in one shot, symmetric cooperative games, without communication. Our methods can be considered as a method for choosing among equilibrium points in such situations.

[2] A coordination game is a game in which there is at least one outcome which both agents prefer over other outcomes.

2.4.3 Limited Rationality

The standard game theory assumption that agents have unbounded computational power is problematic and unsuited to the real world. In most "real-world" situations, there is a limit on the amount of computation ability which may be devoted to solving the interaction problem. Moreover, there is a limit on the amount of time available. Thus, even in cases where there might be a unique equilibrium point, it cannot be taken for granted that two agents, with differing computational capabilities, will both discover it. Ideally, the search for focal points will need to take limited computational power and/or time constraints into account.

2.4.4 Labeling Theories

Gauthier [30] initiated a line of research based on the assumption that when a person chooses an option, she chooses it under some "description." A choice problem exists only if the player conceives of a set of distinct alternative options; the problem is defined by the player's description of the world.[3] Gauthier claims that in order to understand focal points, we need to consider the players' own description of their options. Gauthier assumes that players are able to choose among alternative ways of describing their options. In making this choice, and then choosing between the options themselves, rational players follow a *principle of coordination*[4], which roughly corresponds to Harsanyi and Selten's [34] principle of *payoff dominance*: this ensures that Pareto-dominant equilibria are selected.

Following Gauthier, several theories [3, 73, 4, 40] which take information about "description" into consideration were developed. Sugden [73] uses the term *label* for the description by which players refer to strategies. He presented a general theory of how labels can influence decisions in games; he examined its applications to pure coordination games. Sugden focuses on the question: given the labeling schemes that players use, what choice is it rational for them to make? He then proposes the notion of *collective rationale*, which is similar to Gauthier's principle of coordination. In this paper, we assume that the labels are given and propose rules to select focal points that we show, via simulations, to be successful.

Bacharach and Bernasconi [3, 4] present a variable frame theory (VFT). They claim that players choose strategies in a way that is rational in a perfectly familiar game-theoretic sense; however, the *game* that gets played is determined by non-rational features of the players. They refer to these features as *frames*. A player's frame is the set of variables she uses

[3]Note that in traditional game theory, the question of how individuals conceive their option is normally avoided: the analysis uses the theorist's descriptions.

[4]The principle of coordination specifies that in a pure coordination game with multiple equilibria, it is rational to play one's part in the unique Pareto-optimal equilibrium, if there is one.

to conceptualize the game. Frames may vary both across players and within players from occasion to occasion. The VFT consists of three main parts: the idea that to choose among alternatives a player must first have "described them to herself," a model of what determines players' frames, and an equilibrium notion that allows the variability of frames. The theory implies that it is rational to play focal points in coordination games with multiple equilibria. Their reported experiment confirms most of the theory's claims for such games. Janssen's work [40] is closely related to that of Bacharach [3]. He extends Bacharach's work mainly by considering general classes of dimensions, instead of examples, and proves the optionality of the procedure for this general class. The approach suggested in our paper is directed at studying mechanisms to find focal points by automated agents that are given their frames by their designers.

2.4.5 Experimental Work

In the last few years several experiments, which replicate Schelling's "informal experiments" with pure coordination games, were performed. These experiments also test several theories concerning focal points.

Mehta et al.'s results [58] confirm that players of pure coordination games are more successful at coordination than if they would have chosen a strategy (or an item) at random. In addition to this confirmation, Mehta et al. also added a control group of subjects who faced the same set of questions as the main group. However, the members of the control group were instructed merely to give "some" response without being given any incentive to choose any particular response (rather than to try to match their partner's choice, as in the main group). The success rate of the main group was significantly higher than that of the control group, which had no incentive for coordination.

Mehta et al. [58] also define the notion of *primary salience* as the strategy (or item) which "comes" to the player's mind.[5] They presented two alternative hypotheses which might explain the success of humans in coordination games: "secondary salience"— each player will choose the strategies whose label she believes most likely to have a primary salience; "Schelling salience"— each player will look for a *rule of selection* which, if followed by both players, would tend to produce successful coordination. Their experiment was not designed to allow the discrimination between these two hypotheses by formal statistical tests. However, their results suggest that Schelling's salience may play a significant role in human success in coordination games.

[5]The concept "salient" itself has received no formal treatment in the literature [4], but there is a wide acceptance of Lewis' characterization [54] of "being salient," as *being the only one which has some conspicuous attribute.*

In our experiments we focus on Schelling's salience, since we believe that it is more dependable and also simpler to calculate. That is, we provided our agents with a rule of selection.

In another set of experiments [57], Metha et al. focused on a particular class of pure coordination games which they call *assignment games*. In an assignment game, two players are presented with a set of objects of some type A, and with two objects B_1 and B_2 of a different type. The players are instructed to assign each A-object to one or other of the B-objects; each is rewarded if they choose the same assignment. Their purpose was to isolate a small number of rules of selection of focal points. They presented three rules: (i) the rule of closeness; (ii) a two-step variation on the rule of closeness; and (iii) the rule of equality. In their experiments they found strong support for the hypothesis that the subjects used these rules. In our experiments, we studied matching games, which are more general than assignment games. In addition, the rules of selection which we propose are more general and also more mathematically defined than in [57]. The particular rules which we tested are shown to be appropriate for the development of automated agents which can successfully cooperate without communication.

Mehta [56] developed a complementary experimental methodology that focuses on the processes by which people attempt to coordinate their actions, rather than on the outcomes of those actions. In her experiments, subjects were interviewed, alone and in teams, immediately after completing a series of bargaining and pure coordination games. She found that the comments of some subjects support Schelling's hypothesis. For example, several subjects spoke of choosing their responses because they were "obvious" or "logical," where the "logic" of the response, or its "obviousness," derived what they perceived to be a mutually recognized attribute of the solution.

2.4.6 DAI Approaches to Reduce Communication

As mentioned above, most of the research in DAI assumes that the agents can communicate with one another. However, since communication consumes time and resources, there is usually an attempt to reduce it. Furthermore, it is clear that agents in a realistic system cannot have complete knowledge of the goals, actions, and interactions of the other agents in the community. They must make some decisions without communicating with each other.

Georgeff [32] suggests using a set of finite and simple fixed signals to synchronize multiagent plans, thus reducing the overhead of coordination. In the case that he considers, agents must exchange messages in order to know that an action has been completed. We do not address such problems, but focal point mechanisms can be applied in choosing among different plans without communication.

In the Partial Global Planning (PGP) approach [22, 20, 21], each node builds *partial global plans* that represent its partial views of the joint problem-solving activity. In deciding what action to take next, an agent refers to its local plans, and in deciding among its possible plans, it refers to collective plans for all agents. Thus, in this approach, an agent communicates its partial global plans to other agents in order to reach coherent behavior in the overall system. Unfortunately, as we mentioned above, even if the nodes exchange information, a node will still be uncertain about the role being played by each of the other nodes. In [19], Durfee et al. developed a sophisticated local control that allows the nodes to make rapid, intelligent local decisions based on changing problem characteristics, without the overhead of conferring with each other to coordinate these decisions. Instead, coordination is based on an organizational view of individual node activity. This is similar to the approach taken by agents searching for focal points. However, the methods applied by the PGP nodes are domain-dependent and lead to more sophisticated cooperation. Our methods are general and do not depend on the specific problem that is being considered. However, our methods allow for only relatively limited cooperation.

Others have also suggested using organizational structures to reduce communication. That is, the relationship specified by the organizational structures gives general long-term information about the agents and the community as a whole [41]. For example, Werner uses "roles" for describing expectations about individual behavior [76]. Social laws [70, 60] are another way to achieve better coordination by pre-defined organizational regulations. The focal point approach can be used to reach joint decisions without communication or to limit the domain of consideration and thus reduce communication.

Ronald Arkin [1] demonstrates the efficiency of multi-agent schema-based navigation for object retrieval without communication. In this model, cooperation is achieved when several agents are attracted to the same object and, together, transfer the object to its destination. In this application, domain-specific knowledge is used to decide on objects that serve as focal points, while, in contrast, we look for general methods for selecting focal points.

2.4.7 Related Work on Plan Recognition

The field of plan recognition [69] bears an interesting, complementary relationship to the work on Focal Points described in this paper. Plan recognition assumes the existence of some group of base-level actions carried out by an agent; an observer extrapolates from that group of actions to a higher level goal, and by implication, to future actions (or unobserved actions).

Much of the work in plan recognition has focused on specific domains, such as story understanding, psychological modeling, natural language pragmatics, and computer system interfaces. The benefits of successful plan recognition are clear. Based on fragmentary evidence (the actions of an observed agent), the observer could understand the agent's underlying intentions, and use that understanding to guide its own activities. In a cooperative scenario, the observer may discover that additional information should be communicated, or that there exists some collaborative opportunity. In a competitive scenario, the observer may learn what is necessary to effectively counteract his opponent.

Plan recognition can thus be used, and indeed, has been used [37] as a direct mechanism for multiagent coordination. The rationales for using plan recognition techniques, as opposed to (or in addition to) conventional communication (even between cooperating agents) are similar to the rationales for using our Focal Point techniques, such as unreliable communication channels, lack of a common language for communication, and the risk of communication being intercepted by hostile agents.

This, then, is the common ground between our Focal Point techniques and plan recognition: both can be used as mechanisms for communication-free coordination. The primary difference between the two approaches is that plan recognition uses as its starting point the actions of an agent, while Focal Point analysis is fundamentally an analysis of the domain. Focal Point analysis sometimes views the domain in light of *potential* agent actions, but not generally with reference to past agent actions.

There have been several axes for categorizing research in plan recognition. "Intended" recognition is carried out when the observed agent's actions have been intentionally structured so as to aid in the plan recognition process; "keyhole" recognition assumes no such helpful structuring. Two other important distinctions in the literature are whether the observer has full knowledge of the domain, and whether there might be "errors" on the part of the observed agent (i.e., actions carried out by the agent that are inconsistent with its actual plan).

The fundamental problem in effective plan recognition is that a sequence of actions may be consistent with many high-level plans. Assume, for example, that the observed agent walks down the street and enters a supermarket. The actions are consistent with a plan to buy groceries, but also with a plan to rob the supermarket. In the absence of more information, we might want the plan recognizer to prefer the former interpretation. On the other hand, the overall process must be defeasible (altering conclusions based on additional knowledge), since if we later discover that the agent picked up a gun before walking down the street to the supermarket, we might prefer the robbery explanation of his plan.

Ideally, the plan recognition process should therefore, a) work from fragmentary information to reach conclusions, b) be capable of entertaining multiple explanations, but perhaps focusing on one or more as "most likely", c) be capable of coming to new conclusions based on new information.

This paper, in analyzing Focal Point discovery, presents both a decision-theoretic and a step logic approach to the problem. There have similarly been several distinct approaches to plan recognition. The major approaches have been:

- An argumentation approach, using truth-maintenance-like systems to support or deny particular plan recognition conclusions;
- A circumscription approach, that seeks to minimize the set of plans that could plausibly be implied by a set of actions;
- A probabilistic approach, that uses Bayesian nets to derive the most likely plan consistent with the observed actions.

The decision-theoretic and step logic approaches to Focal Points that we present in this paper are weakly analogous to the probabilistic approach and argumentation approaches to plan recognition, respectively.[6]

The Argumentation Approach to Plan Ascription The work of Konolige and Pollack [46] frames the plan recognition problem in the traditional artificial intelligence context of "belief and intention ascription", that is, ascribing beliefs or intentions to the observed agent. The mechanism for carrying out this process is Konolige's argumentation system ARGH, similar in many respects to the justification-based Truth Maintenance System of Doyle [18]. The actions of the observed agent, which are plan fragments, serve as arguments in favor of certain high-level plans (and as arguments against other high-level plans). The system considers the "support" that a given plan has; if the plan is supported by an argument whose premises are accepted, and the plan is uncontested (there is no conflict of propositions within the given domain), then the conclusion is accepted.

The plan recognition process takes in local cues, actions of the observed agent, and attempts to fit this local information into the global coherence of a high-level plan. Since the argumentation system itself is defeasible, i.e., it will come to new conclusions when presented with new information, the plan recognition system itself has this property. The system also has the advantages that it does not rely on the observer having complete knowledge of the

[6]For a non-declarative approach to plan representation and recognition, which does not fit neatly into the above taxonomy, see [52].

domain, nor does it depend on the observed agent having a correct plan, from the perspective of the observer. Finally, the argumentation system provides a precise, comprehensible rationale for why one conclusion should be preferred over others.

The Circumscriptive Model of Plan Recognition Kautz [43, 44, 45] has proposed an entirely different way of approaching the plan recognition problem, one based on McCarthy's circumscription scheme [55] which minimizes the set of inferences that can be derived from an initial group of assumptions. Every observed action is assumed to be part of some high-level plan; Kautz uses circumscription as the formal technique to minimize the number of high-level plans deduced by the system.

Say, as a simple example, that an agent is observed carrying out actions p and q, where p could be part of plans A or B, and q could be part of plans B or C. Having observed p and q, the system proposed by Kautz would deductively conclude that B is the agent's plan. There is no need for circumscription in this case, but it illustrates in a simple way the overall approach.

There are advantages to Kautz' approach, not least of which is the formal rigor that he brings to the plan recognition problem (others have continued to pursue this deductive theory of plan recognition, for example [75]). Some of the drawbacks of this approach, however, are that the observed agent is assumed to be carrying out correct plans, and that the observer has complete knowledge of the domain. Perhaps even more importantly, the technique has no way of distinguishing between the *a priori* likelihood of various plausible plans. As pointed out in [10], upon observing an agent packing a bag and going to the airport, the system would not be able to prefer the plan where the agent is taking a plane trip (a likely scenario), or the plan where the agent is going to carry out a terrorist bombing (a less likely scenario).

The Bayesian Model of Plan Recognition Handling this shortcoming of the circumscriptive approach, Charniak and Goldman [9, 10] propose a probabilistic model for carrying out plan recognition. The plan recognition problem is converted into a Bayesian network, and the resulting network provides a "most likely hypothesis" regarding the observed agent's plan. One advantage of this scheme is that the lowered computational overhead of using a Bayesian network makes the probabilistic assumptions and calculations more tenable. In addition, the likelihood of a given interpretation plays a direct role in the conclusions of the system (as opposed to the set minimization approach, above). In related work, Carberry [7] similarly approached plan recognition using probabilistic reasoning, though instead of a Bayesian model the system used Dempster-Shafer belief functions.

3 A Decision Theoretic Model for Focal Points

Using focal points to choose among competing solutions requires that contextual information regarding the solution be modeled and also requires a method for selecting a particular solution based on the contextual information. Decision theory is a convenient paradigm for both modeling contextual information and selecting a focal point.

In the decision theoretic model, the agents are assumed to be able to assign utilities to various outcomes, similar to the assumption in game theory. An agent attempting to decide on an action using a decision theoretic framework constructs a decision tree, leading to different outcomes, with probabilities associated with each branch of the tree. The agent's expected payoff is the probability on the branch times the value at the leaf.

The decision theoretic focal point algorithm exploits focal point intuitive properties (such as uniqueness) to establish more accurately the probability values on the decision tree's branches. This altering of the probabilities in turn affects the agent's calculation as to the most beneficial action to take; when the agents have found a focal point that sufficiently alters their probabilities, they will coordinate.

Consider the following primitive example. Two agents must decide on a meeting point, and the choice is between two houses (A and B) and a bridge (C). The utilities for agent J of A and B are 5, and the utility for C is 10 (he must go further to reach C). For agent K, the utilities for A and B are 10, while the utility for C is 5. In game theoretic terms, there are three equilibrium points, with no way of distinguishing among them. Discovery that the bridge is a focal point, however, increases the probability that the other agent will choose it (even though, for K, it is a less preferred solution). If the focal point raises the probability of the other agent choosing the bridge from .33 to above .5, then even agent K will choose the bridge, and the agents will meet. This technique provides for a natural integration of payoffs into the decision-making process. In this paper, we focus on techniques for identifying focal points and only briefly consider the incorporation of the agents' utilities into the decision process. However, in all the cases, we assume that the agents consider only action profiles that are equilibria.

3.1 The Agent Model

The database DB of an agent is a set of consistent sentences over a language $\mathcal{L}$. For simplicity, we assume that in $\mathcal{L}$ there is a set of predicates $\mathcal{P}red$ and a set of terms $\mathcal{T}erm$ over which the focal point computation is going to be done.[7] Each of the predicates $P \in \mathcal{P}red$ has

[7] $\mathcal{P}red$ may be given to the agent by its designer and the agent may assume that the other agents are given the same set. Another possibility is that the agent may try to estimate the predicates of the other

two arguments: an element of $\mathcal{T}erm$ and a constant from the set $\mathcal{V}alue_P$. For any predicate P and a term T, the database may include at most one fact of the form $P(T,v)$ for some $v \in Value_P$. The values in $\mathcal{V}alue_P$ may be ordered, and there may be other predicates as well as functions in $\mathcal{L}$. A focal point should be chosen from the objects described by $\mathcal{T}erm$.[8]

For example, Color(h1,red) might mean that object h1 is red, Height(h1,4) might mean that the height of object h1 is 4 meters, and Type(h1,House) may mean that $h1$ is a house. This database does not change during the search for a focal point. Of course, the database does change over time as the agent operates in the world and draws new inferences. We propose that each fact in the database be tagged with a "measure of difficulty," the effort that went into derivation of the fact (e.g., number of supporting arguments, depth of derivation, number of conjuncts in the compound predicate). Thus, although the focal point search is carried out over a static database, the dynamic aspect of that database is partially captured by the "measure of difficulty." The way to compute the "measure of difficulty" depends on the methods the agent uses to obtain information. We demonstrate the intuition behind the "measure of difficulty" using a deductive database example, but we do not require that the agent use logic for obtaining new facts.

Consider the following domain. There are three houses, labeled h1, h2, and h3. Agent A's database, at some point, included the following facts and rules (by design and through observation):

$\forall x\, Type(x, House) \rightarrow Color(x, White) \vee Color(x, Black)$

$\forall x\, Type(x, House) \wedge Less\text{-}Than(Age(x), 25) \rightarrow Color(x, Black)$

$\forall x\, Type(x, House) \wedge Made\text{-}of(x, Bricks) \rightarrow Color(x, Black)$

$\forall x\, Type(x, House) \wedge \text{Architect-Of}(x, Smith) \rightarrow Color(x, Black)$

Less-Than(Age(h2), 25), *Less-Than*(Age(h3), 25)

Architect-Of(h1, Smith), *Made-of*(h1,Bricks)

Type(h1,House), Type(h2,House), Type(h3,House).

In this case $\mathcal{T}erm = \{h1, h2, h3\}$, and $\mathcal{P}red = \{Type, Color, Made\text{-}of, \text{Architect-Of}\}$. We will consider how new facts are derived and adjust their measure of difficulty accordingly. For example, the fact that Color(h2,White) and Color(h3,White) can be derived is a deduction of depth 2. Color(h1,Black) can be derived in two separate deductions, each of depth 1. The measure of difficulty assigned to the first two facts would be higher than the measure of difficulty assigned to the last fact. Since multiple derivations lower the measure of difficulty, it is necessary to keep a record of derivations over time (as in a Truth Maintenance System [15]).

agents (e.g., based on knowing their sensors) and may include these predicates in $\mathcal{P}red$.

[8] In the rest of the paper, we will abuse notation and we will use object and its term interchangeably. Similarly, we will use value and their constants interchangeably.

The measure of difficulty is intended to be a heuristic indication of how likely it is that the other agent knows the fact. The deeper the derivation, the less likely that the other agent has discovered it; the more alternative derivations there are, the more likely it is that the other agent has found one.

The $\mathcal{P}red$ set contains a collection of primitive predicates and also contains "conjunctive predicates" that are the conjunction of two or more primitive predicates. The $\mathcal{P}red$ set ideally includes all possible conjunctions of predicates, though in an actual system only some subset of these would actually be considered.

We want agents, in their search for focal points, to consider both explicit knowledge and "obvious" knowledge that is easily computed from their databases. For example, if "less than" is a predicate that the agent is considering, and both 5 and 6 are terms of which he is aware, then we want the agent to use the knowledge that 5 is less than 6, even though this fact is not explicitly represented in his database. We therefore use a special notation to signify that a fact is "known" to the agent. We write $\in^*$ to mean that the fact is either explicitly listed in DB, or that it can be simply computed over the constant terms or values that are in the database.[9]

The decision theory framework provides a quantitative technique for evaluation that can take a number of factors into account. For example, the search for a focal point can consider, in a weighted fashion, the complexity of a term relative to a predicate (as a heuristic indication of whether the other agent has it in his database), as well as other factors, such as the utility a successful matching on that term would have for the agent(s), its rareness, and centrality.

3.2 Recasting Focal Point Criteria

We now consider how the four focal point intuitive criteria (see Section 2.3) — uniqueness, uniqueness complement, centrality, and extremeness — can be formalized in a decision theory framework. The characterizations are flexible, in that they may consider a term to be a focal point even when it is not *actually* unique (but almost so), or not *actually* central (but almost so). Because the definitions here differ from their related counterparts above, we will sometimes change their names (for example, the uniqueness described above is related to the rareness definition below).

[9]The question of what can be simply computed is domain-dependent, as well as agent-dependent. There is an analogy here with the idea of being "operational" in the Explanation Based Learning literature [59]. Checking "less than" might be operational in some machines; in other machines, deciding in a game of chess whether a given board position is reachable from the current state might be operational because of specialized hardware.

As mentioned above, finding a focal point ends up altering the probabilities associated with a given choice. Once probabilities have been assigned, a standard decision theory analysis is made (probability times outcome utility) to decide what move to make. An evaluation is made of each Predicate/Term pair in the agent's database. The focal point criteria below give an agent "credit points" for each Predicate/Term pair, and these credit points can accumulate from different sources (rareness, centrality, etc.). These points are then combined and normalized to give us the probabilities (at this stage, as well, a meta-analysis can be done that will lower the probabilities assigned to multiple points that compete in the same category, such as extremeness).

The way in which the contributions of different sources are combined remains a matter open for future research. It is our belief that it will require experimental evaluation to determine how sources should be weighted when they are combined.[10] Our intent below is to show how to exploit the general relationships between predicate/term pairs in the database and to show (grossly) how they contribute to a particular pair being a focal point.

3.2.1 Rarity

The probability of an agent's partner making a certain choice is increased if that choice has a property not shared by other choices. As mentioned above, the decision theory approach allows us to use a more sensitive concept than that of "uniqueness." For example, if there are 1000 objects, of which all but three are black, and the three non-black objects, h1, h2, and h3, are all white, then we would like to consider the non-black objects as focal points, even though their color is not unique, but only rare. We may use other properties to choose among the white objects. In particular, if it is easier to conclude that h1 is white than to conclude that h2 and h3 are white, we may choose h1. Thus, given an item and a property, we consider the number of objects with the same property and combine this information with the measure of difficulty of the information.

That is, to capture the notion of "rareness" of an object T with respect to a predicate P in a decision theoretic model, we want to weight positively the appearance of $P(T, v)$, and this weight will increase with the decrease of $P(T, v)$'s difficulty. We negatively weight the appearance of other $P(t, v)$, where $t \neq T$, elements in the database in evaluating the rareness of T with respect to P, though the greater $P(t, v)$'s difficulty, the less negatively it is weighted. Intuitively, if $P(t, v)$ is associated with great difficulty, there is more of a chance that the other agent will not have it in his database, and since in such situations we would like T's rareness measurement to be higher, we decrease its rareness measurement less than

[10] We demonstrate the choice of such weights in Section 3.4.

we would if $P(t, v)$ were to have a lower difficulty measurement.

Example 1 Consider the objects h1 and h2 mentioned above, and assume that the agent's database includes the facts Color(h1,White), Color(h2,White) – both with difficulty 1 – and Color(h3,white) – with difficulty .5. For all the other objects, t, such that t is not equal to h1, h2 or h3, the database includes Color(t,Black). We will consider h1 and h2 to be more rare than the 997 black objects, with respect to the predicate Color, since there are only three objects which are white. h1 and h2 rarity is the same with respect to color since they have the same color and the same difficulty. We consider h3 to be more rare than all the other objects (including h1 and h2), since it is white as h1 and h2, but its measure of difficulty is lower.

If there is an additional object, h4, which is blue, and its measure of difficulty is .5, it will be considered more rare than h1, h2, h3, and the black objects, since it has both a unique color and a low measure of difficulty.

We will use the following notation below. T is a given constant term in $\mathcal{T}erm$; P is an element in $\mathcal{P}red$; and $v \in \mathcal{V}alue_P$. $D[P(T, v)]$ for any P, T, and v, such that $P(T, v)$ is in the database, is the "measure of difficulty" for that term/predicate pair (the measure of difficulty for a term/predicate pair that does not appear in the database is taken to be infinite).

For all $P \in \mathcal{P}red$, $T \in \mathcal{T}erm$, such that $P(T, v)$ is in the agent's database,

$$R_T^P = f_r(D[P(T, v)]) + g_r(\{D[P(t, v)] | t \neq T, P(t, v) \in^* \text{DB}\}).$$

The functions f_r and g_r allow for more flexibility by providing different scaling of the difficulty measurement. f_r decreases with the difficulty measurement of $P(T, v)$, indicating that the prominence of an object decreases as its measure of difficulty increases. g_r's domain is the set of difficulty measurements. g_r decreases as the number of elements of its set increases and as their difficulty measures decrease. That is, if there are more objects with the same property as T, its rareness decreases; however, if the difficulty of these objects is high, and there is a chance that the other agent does not have this information in its database, then T's rareness is affected less. Using two different functions captures the intuition that different weights should be given to the fact that $P(T, v)$ is in the database, and that for other terms, denoted by t, $P(t, v)$ is also in the database. Since the function D (i.e., the measure of difficulty) may be used for computing the rareness of different objects simultaneously, it is not possible to change D to reflect this intuition. This is because the same object may play a different role in the computation of its own rareness and in the computation of the rareness of another item.

Example 2 This rule would be applicable in the case where we know about two Bridges, C125 and C412. The measure of difficulty associated with Type(C125,Bridge) is low (e.g., 2), while the measure of difficulty associated with Type(C412,Bridge) is high (e.g., 5). The fact that Type(C125,Bridge) is in the database contributes significantly to the R^{Type}_{C125} utility, while Type(C412,Bridge) does not decrease that utility by very much. Type(C125,Bridge) is rare, even though it is not unique. In particular, if we choose $f_r(x) = \frac{1}{x}$ and $g_r(X) = -\sum_{x \in X} \frac{1}{x^2}$, then $R^{Type}_{C125} = 0.46$ and $R^{Type}_{C412} = -0.05$.

Similarly, if we consider the objects of example 1 and apply the above functions, i.e., $f_r(x) = \frac{1}{x}$ and $g_r(X) = -\sum_{x \in X} \frac{1}{x^2}$, we obtain $R^{Color}_{h1} = 1$, $R^{Color}_{h2} = -4$, $R^{Color}_{h3} = -4$, and $R^{Color}_{h4} = 2$.

3.2.2 Centrality

The probability of an agent's partner making a certain choice is increased if that choice is somehow "central" to the group of terms in the domain, relative to some predicate. For example, given a group of houses in a line, the central one would have a greater likelihood of being a focal point.

We introduce the following function to capture, in our decision theory framework, the notion of "centrality" (or "symmetry"):

For all $P \in \mathcal{Pred}$, $T \in \mathcal{Term}$, such that $P(T, v)$ is in the agent's database,

$$
\begin{aligned}
C^P_T \;=\; & f_c(D[P(T,v)])+ \\
& g_c(\{max(D[P(t_1,v_1)], D[P(t_2,v_2)])\} \mid \forall t_1 \neq T, P(t_1,v_1) \in^* \text{DB} \wedge \\
& \qquad \exists t_2 \neq t, t_2 \neq T, P(t_2,v_2) \in^* \text{DB} \wedge \mathit{Diff}(v_1,v) - \mathit{Diff}(v_2,v) < \epsilon\}) - \\
& g_c(\{D[P(t_1,v_1)] \mid \forall t_1 \neq T, P(t_1,v_1) \in^* \text{DB} \wedge \\
& \qquad \nexists t_2 \neq t, t_2 \neq T, P(t_2,v_2) \in^* \text{DB} \wedge \mathit{Diff}(v_1,v) - \mathit{Diff}(v_2,v) < \epsilon\}).
\end{aligned}
$$

Centrality for T is increased for each pair of terms that lie roughly equidistant from T, using the *Diff* domain-specific metric. The size of the increase depends on the maximal measure of difficulty of the pair. As the measure of difficulty of the pair decreases, T becomes more central. Centrality for T is decreased for each term in the domain that has no matching term that is roughly equidistant from T. As the measure of difficulty of the term increases, T becomes more central. It is important that *Diff* properly capture the notion of centrality in the given domain. For example, if terms are arranged along a single dimension (e.g., the integers), *Diff* might return the difference between its arguments' locations. In practice, when we want the above definition of centrality to be combined with our definitions of rareness, the weighting functions f_c and g_c must be chosen carefully.

Example 3 Consider five houses in a row, labeled A, B, C, D, and E (each contiguous pair being equidistant). House C is considered central, with House A being matched (within ϵ) by House E, and House B being matched (within ϵ) by House D.

3.2.3 Extreme

The probability of one's partner making a certain choice is increased if that choice is somehow "extreme" to the group of terms in the domain, relative to some predicate. For example, given a group of houses in a line, the end ones would have a greater likelihood of being focal points. Also, the tallest house in town would be a candidate for being a focal point.

For all $P \in \mathcal{P}red$, $T \in \mathcal{T}erm$, such that $P(T,v)$ is in the agent's database, and a partial order Q on $\mathcal{V}alue_P$,

$$\begin{aligned} E_T^{P,Q} &= f_e(D[P(T,v)]) - \\ & \quad g_e(\{D[P(t',v')] \mid t' \neq T, P(t',v') \in^* \text{DB} \wedge Q(v',v) \in^* \text{DB} \wedge Q(v,v') \notin^* \text{DB}\}). \end{aligned}$$

If T is an extreme point candidate, we subtract points from it whenever it is "exceeded" by some other term. On the other hand, we do not add points to it when it is preceded by other terms. If an object is extreme, this quality is not strengthened by more terms below it (e.g., the tallest building does not become more pronounced because additional small buildings have been built). Extremeness is weakened, however, by exceptions and depends on the exception measure of difficulty.

Example 4 In the set $\{1,4,5,13,29\}$, the number 1 is extreme, using "less than" as the Q predicate above. The number 4 is relatively less extreme than 1, using "less than," but is relatively more extreme than the number 5. The number 29 is extreme, using "greater than" as the Q predicate, and 13 is relatively more extreme than 5 and 4 with respect to "greater than."

Similarly, consider a database that has Height(h1,10) (with difficulty 8), Height(h2,11) (with difficulty 4), Height(h3,15) (with difficulty 10), Height(h4,6) (with difficulty 3). Assume also that $f_e(x) = \frac{1}{x}$ and $g_e(X) = \sum_{x \in X} \frac{1}{x}$ and uses "greater than" ($>$) as the Q predicate. We obtain: $E_{h1}^{Height,>} = -0.225$, $E_{h2}^{Height,>} = 0.15$, $E_{h3}^{Height,>} = 0.1$, $E_{h4}^{Height,>} = -0.14$. It is interesting to observe that the tallest object, h3, is considered less extreme than h2, since its measure of difficulty is much higher.

On the other hand, if we use "less than" ($<$) as the Q predicate, we obtain: $E_{h4}^{Height,<} = 0.333$, $E_{h1}^{Height,<} = -0.208$, etc., and the lowest object is considered as the most extreme one.

3.3 Success and Implementation Issues

There are many details in the above formalization that have been left unspecified; for example, the functions that relate difficulty of derivation to points were simply labeled f, g, etc., without giving real values for them. The combination of these points is another major unresolved issue for the implementation of the decision theory procedure. These issues need to be resolved relative to any specific domain. We demonstrate the development of such a procedure for the robot rendezvous domain in Section 3.4.

When interacting human agents search for focal points, there is generally no guarantee that their choices will be identical. When interacting automated agents search for focal points, they are following a set of algorithms. Depending on their own knowledge, and their knowledge of each other and of the domain, they may be able to reach a guaranteed solution. In other cases there is no guaranteed agreement, but the focal point algorithm can be thought of as a heuristic to "prune" the search for a focal point. That is, as mentioned above, the focal point algorithm can be used to limit the number of possible solutions that will be explicitly discussed by the agents.

As with various forms of communication, the agents can benefit from having some common background when they use a focal point algorithm. For example, agents that negotiate the allocation of a common resource should have some common language and some protocol for negotiations.

3.3.1 Guaranteed Joint Selection

Consider the case where agents have identical knowledge about everything and in addition, the difficulty of derivation they attach to various propositions is the same. It is clear that in this case, if there is a set of focal points, the set will be generated identically by both. If, furthermore, the identified set of focal points includes only one object, then it is clear that the agents will agree on the same object.

3.3.2 Incomplete Information

The focal point decision theory approach is trying to exploit information an agent has about the domain, but it really does not relate to the other agent's information, except implicitly when taking into account the "measure of difficulty." An agent gives lower weight to facts with higher measures of difficulty. The intuition behind this approach is that as the measure of difficulty increases, the likelihood that the other agent knows the fact decreases.

That is, while searching for focal points, an agent does not try to model the beliefs of its specific partner, but rather examines the possible solutions and tries to choose prominent solutions. However, in situations of incomplete information, when each agent has a different set of possible solutions or attached different properties to some of the solutions, combining the focal point approach with other approaches that do model partners may be beneficial.

For example, in some situations agents do have explicit information about one another's knowledge or beliefs. In the best case, they might have common knowledge [33] of the domain. It probably makes sense, when common knowledge is sufficient, to use it and only it in the focal point derivation. In most situations, however, common knowledge will not be available to the agents.

If an agent A knows that B does not know the fact P, it is a reasonable heuristic for A to avoid using that fact when searching for a focal point. However, there is no guarantee that avoiding P will increase the chances of finding a focal point. For example, B may use some other fact, Q, of which A is unaware, and coincidentally arrive at the conclusion that A would have found if A had used P.

When A does not know whether B knows or does not know P, then a reasonable heuristic is for A to continue using P. There is simply too much likelihood of not knowing very much about the agent's partner's information, and the default should be to assume that he knows what the agent does; otherwise the search will be too restricted. Again, however, there can be no guarantees that using or not using P is the most efficacious strategy.

Even when A has specific information about things that B does know, A cannot simply use only that information: A does not necessarily even know that B knows that A knows that B knows the information, without which its use in the focal point search will not be useful. If the two agents have more than one level of cross-knowledge, they could reason back and forth, eventually reaching a minimal group of facts known to all levels greater than i (perhaps empty). Nevertheless, it is far from clear that carrying out this search for a base group of facts constitutes a desirable strategy. Consideration of these issues of inter-agent knowledge and its effects on focal point derivations is left for future work. In this paper we follow Schelling and other researchers and assume that the agents use their own point of view to identify a focal point.

3.3.3 Effects of Representation on Joint Selection

One important characteristic of the focal point algorithms is how they are affected by agents' representations. One agent might have a predicate "Size" with the possible value "Big," and the second agent have a predicate "Dimension," with a possible value "Huge." As long as the extensions of the predicates are the same, the different words that the two agents use will not matter. Thus, even when agents do not make the same lexical choices, they may still reach the same focal point. However, when the agents have differing ontologies, their focal point searches may not reach the same conclusion.[11]

[11] In Distributed Artificial Intelligence, there is often a distinction made between "distributed problem solving" (DPS) and "multi-agent systems" (MA). The former often consists of centrally-designed groups of

Some focal points that are discovered represent a "syntactic" aspect of the domain, while others that are discovered represent a semantic aspect of the domain:

Example 5 Different ontologies, the same focal point Agent A uses miles, while agent B uses kilometers. If the agents choose "the middle point on the road" as a focal point, they choose the same location, even though their representations are very different. Similarly, if agent A uses Celsius, and agent B uses Fahrenheit, the freezing (boiling) point of water (or the normal human body temperature) are the "same" for both, and might be chosen as focal points. Another example might be agents choosing π, or the square root of 2, or $e \approx 2.71828$, regardless of what base number system they are using.

Here, the focal point is associated with a natural phenomenon, and the different methods agents use for measuring won't disturb their agreement on a focal point. This, of course, assumes that the action which the agents are coordinating is itself independent of the *representation* of the focal point.

Example 6 Different ontologies, different focal points Agent A uses miles; agent B uses kilometers; both use integers for measuring; A chooses a focal point of 10 miles; B chooses a focal point of 10 kilometers. Similarly, if A uses U.S. dollars and B uses pounds sterling, they might have different focal points (using the concept of round numbers[12]). Similarly, if agent A uses the base 10 number system, and agent B uses the base 8 number system, they may choose different focal points, with different concepts of what constitutes a round number.

Here, the focal point is associated intrinsically with the method of measurement used by agents. Therefore, when agents use different measurement systems, they will reach different focal points.

In the focal point algorithms we present below, we use methods that are not sensitive to the specific ontologies used by the agents (as in Example 5). For example, our "uniqueness" criterion is used to evaluate how many objects have a specific property, but it does not depend on the name of the property.

---

agents who cooperate to solve a global task. The latter consists of independently-designed agents who, while interacting, pursue their local goals without consideration of global group utility. The ability to rely on one's interaction partner's having an identical ontology is one aspect that truly differentiates DPS from MA systems.

[12] "Round numbers" seem to be good focal points because they are the numbers that uniquely satisfy the predicate Evenly-divisible-by-base. Although this may seem like an unusual predicate, it is actually strongly reinforced by cultures, playing a role in how people learn about numbers and the words that name those numbers (twenty, thirty, etc.).

3.4 The Application of the Decision Theory Focal Point Method to Robot Rendezvous

We have, up to this point in the paper, demonstrated the concept of focal points primarily through examples found in [68]. Most of these examples, originally presented by Schelling, are quite simple. Even though they bear a relationship to real world examples, they were specifically designed to demonstrate the focal point concept.

Validation of the power of focal points as a communication-free coordination heuristic would seem to be difficult and certainly dependent on the specific domain in which it is used. A real-world test of a focal point algorithm, however, runs the risk of being overly specific to the domain over which it is held.

In order to overcome this problem, we apply the focal point decision theory model to an abstraction of a real world situation. We assume that there are objects in the world with various properties, and we want two agents to choose one of the objects (i.e., the same one) without communicating. If the two agents choose the same object, we have a "meeting," and a success. Our approach here is to check a focal point decision theory algorithm in simulations of various randomly generated worlds and to see just how useful the focal point technique is.

We make no assumptions about the properties of the objects, the way they are ordered, or any other special characteristics of the domain, beyond the very general description given above. The algorithm we explore in this section is intended to be useful in any domain that satisfies the outlines of the description above.

The power of focal points in coordinating meetings among agents was highly evident in our simulations. We found that in most situations there is more than a 90% probability that the agents will meet, and in many situations the probability rises to 100%.

3.4.1 Situation Description

Two agents are trying to choose the same object out of a set of objects. The following examples might occur in an environment where communication is difficult (radio frequency disturbance, or secrecy demands during a battle, or the simple inability to communicate because a specific frequency has been jammed), and therefore an attempt must be made to come to an agreement without any communication. There are various scenarios that might require this kind of communication-poor interaction. For example, two agents that are out of touch with one another must agree on the same plan of action, out of a set of several equally reasonable plans. Another example is of agents who are unable to communicate, but who must choose one of several "safe houses" where they will meet and communicate. Another possibility is when agents may actually need to reestablish communication by choosing a radio frequency to use for future messages.

The worlds we examine have the following characteristics:

- There is a group of objects (denoted by $\mathcal{T}erm$) out of which the agents must choose one (ideally the same one).

- As described above, there is a set of predicates $\mathcal{P}red$. Each of the predicates $P \in \mathcal{P}red$ has two arguments: an object in $\mathcal{T}erm$ and a value from the set $\mathcal{V}alue_P$, so that each object has a value for each predicate.

- Any characteristic of an object can be encoded in the values that the predicates can take. We assume, without loss of generality, that the predicates in $\mathcal{P}red$ are ordered and numbered by 1, 2, 3, and so on.[13]

We make the following additional assumptions:

1. The agents observe the same objects and properties in the world. They have the sets $\mathcal{T}erm$, $\mathcal{P}red$, and $\mathcal{V}alue_P$'s, as described above.

2. The agents have great flexibility regarding their internal representations of the world, and these internal representations of the different agents need not match one another. For example, they may have different predicates and may represent the value of the predicates differently.

3. Utility is only attached to success in choosing the same object, not to the selection of any specific object (i.e., the agents are indifferent about the specific objects). In game theory, the above problem can be described using a game matrix. One agent needs to choose a row, and the other agent needs to choose a column. The payoff for each agent is written in the cell specified by their respective choices.

The algorithm described below is useful in situations where a single designer builds both agents[14] and sends them to an environment about which s/he does not have advance information. If the designer suspects that the agents may lose communication and need to

[13] The predicates' numbers are used only for the presentation of the paper. As is explained below, we allow flexibility regarding the agents' internal representation. In particular, we do not assume that the agents assign the same names to the predicates. Furthermore, we do not assume that the agents assign the same values to the same object, with respect to a given predicate. In some situations we even do not require that their ordering of the values be the same. However, we do require that two values be different in the presentation of one agent, if and only if they are different according to the presentation of the other agent.

[14] The algorithm is also useful if the agents were designed by *different* designers and if they agreed by prior communication to use this algorithm.

get back in touch, s/he might choose to provide them with a mechanism as described below. Since s/he doesn't know the exact details of their environment, the coordination policy can't make use of instructions such as "go to the highest building," since there may not be a unique building that satisfies the criterion.

The important point here is that the designer wants to use as little prior information as possible to aid the agents' coordination, but some prior information (e.g., the existence of certain predicates) might still be required by a focal point algorithm. This is not an unreasonable demand. For example, the fact that the agents have certain sensors to which they have access mirrors the prior existence of predicates that can be used in a focal point algorithm.

3.4.2 Decision Theory (DT) Focal Point Algorithms

In our simulations we have made use of the properties of rareness (Section 3.2.1) and extremeness (Section 3.2.3). As we explained above, the premise of the work done was that in any random world some objects will have a predicate-value vector that is different from those of other objects, and so the object itself will be marked as special. Several agents examining the same world will see the same "special" objects. Focal point algorithms provide a technique to choose one of these "special objects" uniquely.

It is important to emphasize that there is no unique "focal point algorithm"; rather, there are many algorithms that might make use of the basic focal point idea, as presented above (the identification and use of "prominent" objects to aid coordination heuristically). The simulations we carried out below make use of a specific DT focal point algorithm. Our intent was to test how effective this algorithm was over a series of randomly generated worlds. Other focal point algorithms, making use of different formulas, might have performed better or worse. The ones which we tried did worse than the algorithm we presented [27]. The relative success of our own focal point formula shows the basic power of the technique.

The specific formula we used in our simulations is rather simplified compared to the full discussion that appeared above; for example, we do not make use of relative utility attached to different objects that might be chosen, nor use of the "measure of difficulty" discussed above. These simplifications can be viewed as a further removal of context from the world in which the algorithm operates. The simulations thus use a technique of greater domain-independence, since the formula they make use of requires less information about the domain than some alternative formulas.

An important consideration in choosing a method for finding focal points is the technique's generality. Optimally, the method chosen should succeed in finding focal points in all possible worlds. Of course, since there are worlds in which all objects are alike and no focal points can be defined, there is no perfect method. However, we would like a method that, in the majority of possible random worlds, will find a single focal point.

The method used in our simulations has three steps, based on the algorithm presented in Section 3.

Algorithm 1 Joint selection of an object using DT focal points

1. *Calculate the focal point value for all objects $i \in \mathcal{T}erm$ using the following equation:*

$$F(i) = \sum_{P \in \mathcal{P}red} R_i^P + 0.5 * E_i^{P,\leq,>} \tag{1}$$

where R_i^P is the rarity of i with respect to predicate P — i.e., how rare is the value of i relative to the other objects — and which E_i^P is the extremeness of i with respect to predicate P, i.e., how close (relative to the other objects) is the value of i to one of the extreme values that predicate P takes in this particular world.[15] *Formally, assume $P(i,x)$; then,*

$$R_i^P = \frac{100}{|\{i'|P(i',x) \text{ is true in this world}\}|} \tag{2}$$

Suppose we have $P(i,x)$, the order on $\mathcal{V}alue_P$ denoted by $\leq$ and $>$, and let $MAX(i,P)$ be the largest of the following numbers: (1) number of objects that have the value x or less for predicate P; (2) number of objects that have a value greater than x for predicate P. Then we have

$$E_i^{P,\leq,>} = \frac{100MAX(i,P)}{|\mathcal{T}erm|} \tag{3}$$

2. *Choose the object c with the largest value that is unique in having that value. Formally, let $UFP = \{i | i \in \mathcal{T}erm, \forall i' \in \mathcal{T}erm;\ if\ i' \neq i,\ then\ F(i) \neq F(i')\}$. If $UFP \neq \emptyset$, then $c = argmax_{i \in UFP} F(i)$.*

There are several aspects of the algorithm that were chosen arbitrarily. To normalize the values calculated, an arbitrary factor of 100 was chosen. The extremeness property was given a lower weight (0.5 in Equation 1), because it seemed to be intuitively weaker than the rarity property. Most importantly, the definitions of the rarity and extremeness properties are arbitrary.

Another problem we faced in creating this algorithm was the relative weight of the different predicates. Since we chose to assume the maximum possible flexibility in the internal representation of the agents, we could not assume that agents would identify the predicates

[15] E_i^P is only calculated if there is an order on the values of the predicate P.

object	type	size	color
1	1 (=bridge)	1 (=small)	1 (=red)
2	1 (=bridge)	2 (=big)	2 (=blue)
3	2 (=house)	1 (=small)	1 (=red)
4	2 (=house)	2 (=big)	3 (=brown)
5	2 (=house)	3 (=huge)	3 (=brown)

Figure 3: A world description.

in a similar manner. The solution, as mirrored in the formula, was to give equal weight to all the predicates.[16]

Note that if the focal point algorithm succeeds (i.e., $UFP \neq \emptyset$), the agents will definitely meet. That is, both agents, when choosing the object according to the algorithm, know that the other agent will also choose the same object. That is the simplest case. Sometimes, however, the focal point algorithm fails to provide the agents with a solution. Even then, the algorithm can help agents coordinate.[17] We discuss below, in Section 3.4.9, the way the agents can use the information provided by the DT focal point algorithm, even when it fails, to increase the probability of meeting.

3.4.3 Example

Suppose there are five objects in the world and three predicates: Type, Size, and Color. The values of the predicates with respect to the objects are given in Figure 3.

Some examples of how one would calculate the extremeness and rarity values are:

E_1^{type} = Not relevant $\qquad R_1^{type} = \frac{100}{2} = 50$

$E_1^{size,\leq,>} = 100 * \frac{3}{5} = 60 \qquad R_1^{size} = \frac{100}{2} = 50$

The general formula for calculating the focal point value in this example is: $F(i) = R_i^{type} + .5 * E_i^{type,\leq,>} + R_i^{size} + .5 * E_i^{size,\leq,>} + R_i^{color} + .5 * E_i^{color,\leq,>}$. See Figure 4.

Thus, the agents choose the big blue bridge, i.e., Object 2, which has the largest unique focal point number.[18]

[16]In order to reformulate R_i^P in 2 of Algorithm 1, using the functions presented in sections 3.2, it is first important to note that the measure of difficulty of all the facts is the same. For example, we can set them all to one, e.g., $D[P(i,x)] = 1$. In our algorithm, for any predicate P, term T, and value v, $f_r(D[P(T,v)]) = 0$. If S is a set, then $g_r(S) = \frac{100}{|S|}$. Similarly, concerning $E_T^{\leq,>}$ in equation 3 above, using the notions of Section 3.2.3, we choose $f_e = f_r$ and $g_e(S) = \frac{100|S|}{|Term|}$ and $E_T^{P,\leq,>} = max\{E_T^{P,\leq}, E_T^{P,>}\}$.

[17]The analogy here is to a heuristic function in chess. Although such a heuristic function does not guarantee victory, it can help guide the agent in the right direction. Similarly, the focal point heuristic can help guide the agents to a meeting.

[18]Note that in this case all the objects belong to the set *UFP* that is defined in Algorithm 1; however,

Object	R_i^{type}	E_i^{type}	R_i^{size}	$E_i^{size,\leq,>}$	R_i^{color}	$E_i^{color,\leq,>}$	F(i)
1	50	0	50	60	50	0	180
2	50	0	50	80	100	0	240
3	33	0	50	60	50	0	163
4	33	0	50	80	50	0	173
5	33	0	100	100	50	0	233

Figure 4: The extremeness, rareness, and the focal point values of the objects of the example.

3.4.4 Properties of the Algorithm

The DT-focal point algorithm described above has the following properties:

Success Rate: The high success rate of the algorithm is demonstrated in the "Results" section below.

Front End: If the DT-focal point algorithm succeeds (i.e., $UFP \neq \emptyset$), the agents will definitely meet. That is, when choosing the object according to the algorithm, both agents *know* that the other agent will also choose the same object. That is the simplest case. In the rare cases that the DT-focal point algorithm fails, both agents know that it has failed (also common knowledge), and so this algorithm can be used as a front end for any other coordination algorithm.

Simplicity and Intuitiveness: The algorithm is simple to describe, which is important in case it needs to be transmitted in a noisy environment (e.g., just before communication cut-off). In addition, the algorithm resembles human thought processes, which can help in communication between man and machine. It seems that such simple computations can be easily done by people (see also Section 5.)

Complexity of the Algorithm: One of the advantages of the DT focal point algorithm is its low complexity:

Lemma 1 *Given a set of objects $\mathcal{T}erm$ and a set of predicates $\mathcal{P}red$, the complexity of Algorithm 1 is $O(|\mathcal{T}erm| * Max(|\mathcal{P}red|, Log(|\mathcal{T}erm|)))$.*

Domain Independence: The algorithm is applicable in any domain where there are objects, predicates, and the need to choose one of the objects.

Object 2 is the one that has the largest value.

Independence of Agents' Internal Representations: All agents must have sets of objects, predicates, and values for the predicates. However, the agents may have different names for objects, predicates and values. For example, one agent might see a big house and a little house, i.e., $Size(1, big)$ and $Size(2, little)$, while the other agent sees a medium sized house $Size(1, medium)$ and a small house $Size(2, small)$, respectively. Furthermore, agents may have different names for the houses; i.e., the first agent may denote the big house by 1 and the small house by 2, and the second agent may call them 2 and 1, respectively. They may also use different terminology internally; the first agent may use the concept of "house" and the second the concept of "building." Furthermore, the algorithm is applicable even if the agents order the values of a predicate differently.

3.4.5 Description of the Simulations

A *configuration* of the world included the number of objects and the number of predicates in the world. For example, a possible configuration is two predicates and four objects. In each run of the simulation with a given configuration, a new random world was created by giving values to the predicates with respect to the objects. First, the number of values that a predicate could take was randomly chosen from the range 2–20.[19] Second, the values were generated for each predicate/object pair. For example, in the configuration of two predicates and four objects, it could be determined that the first predicate would have 3 values, while the second predicate would have 2 values. A randomly generated world may specify that the value of the first predicate with respect to the first two objects is 1, that it is 3 for the third object, and 2 for the fourth object. In addition the value of the second predicate with respect to the first object is 2, and it is 1 for the other three objects. After randomly generating a world, the third step was to calculate the focal point value for each object of the world, as described by Algorithm 1. Finally, if there was an object with a unique focal point value, the run was considered a success; otherwise, it was deemed a failure.

To make the simulations computationally tractable, we assumed that the world contained up to 19 objects, that there were up to 9 orthogonal predicates, and that each predicate had up to 19 different possible values.

For each configuration, 500 runs over random worlds were simulated, giving a calculated accuracy of 5% (with 95% probability [14] [Section 7]). The final output of 500 runs for each configuration was a number between 0 and 100 that represented the probability of finding an object with a unique focal point value in a random world with the parameters described

[19] We also consider situations where, the number of values of the predicates, was the same for all the predicates and was fixed in advance.

above. That is, the number represented the probability that the agents would be in a world such that they would *definitely* meet when they use the focal point mechanism.[20]

We conducted four sets of simulations. In each set of simulations, we varied some aspect of the world (such as the distribution of the values of predicates and the homogeneity of the world) in order to cover a variety of situations in which agents might find themselves.

A. Same number of values, even distribution: In this set of simulations we consider configurations where:

1. The number of possible values for all predicates are the same.
2. There is an even distribution of values.

Thus, first we ran simulations when there was only one predicate, and then when there were two predicates, etc., up to 9 predicates. For each case, we tested the algorithm when all of these predicates had 2 possible values (e.g., true and false), then when all the predicates had 3 values etc., until the case where all the predicates had 19 possible values. For each such setting (e.g., 3 predicates, each with 4 values), we considered situations of 2 objects, 3, objects etc., until we reached situations of 19 objects. For any given configuration (i.e., number of predicates, number of values for all the predicates and number of objects), we ran the simulations as described above 500 times, with the specific value with respect to each predicate which is assigned to a given object chosen randomly each time among the possible value of the predicate. For example, when the configuration of 5 objects, 6 predicates, and 4 values for each predicate was considered, for each object and predicate, one of the 4 possible values was chosen with an even distribution.

That is, we considered $18 \times 18 \times 9 = 2916$ configurations, and for each of them we randomly generated 500 worlds and for each world calculated the focal point value. As mentioned above, running the simulations 500 times gives an accuracy of 5% (with 95% probability [14] [Section 7]). Note that the type of worlds that were considered in this set of experiments is extremely homogeneous.

B. Different number of values, even distribution: In this set of simulations, the world had the following details (in addition to the general structure described above):

1. The number of values for each predicate was randomly chosen among all the possible ones.

[20]This in no way depends on the number of agents in the world.

2. There was an even distribution of values.

In this case, given a possible configuration, for each predicate the number of possible values was randomly drawn from the range of the number of values. For example, when the configuration of 5 objects, 6 predicates, and $2-19$ values for each predicate was considered, first (for each predicate) the number of possible values was randomly chosen from the range $2-19$. Then, for each object and predicate, one of the possible values was chosen with an even distribution.

For each configuration, we again ran the three-step simulation 500 times.

This set of experiments was motivated by the observation that the size of possible values for different predicates may vary. For example, in a given world there may be two types of objects, but they may have ten possible colors.

C. Different number of values, even and dependency distribution, ordered values: In this set of simulations, the world had the following details (in addition to the general structure described above):

1. The possible values in the world were distributed using an even distribution.
2. Some of the predicates were statistically dependent on other predicates. For example, suppose there were two predicates, as in the example of section 3.1. Suppose that Type can have two values: House and Tree and Color can be either Green or White. It is clear that, for a given object, its color depends on its type; e.g., there are more green trees than green houses.
3. The predicates' values were ordered.

The dependency among predicates was defined as follows: we randomly chose[21] $\frac{1}{3}$ of the predicates to be dependent on the predicates before them, in the following manner. Assume $P_j(x, v_j)$ is true in the generated world. Suppose that P_{j+1} was selected to depend on P_j. Then, we added $P_{j+1}(x, v_{j+1})$ to the world, where $v_{j+1} = \lfloor \frac{v_j}{3} + r \rfloor$, and where r was randomly chosen between 0 and 1.

Again, for each configuration, we ran the three-step simulation 500 times. This set of experiments was motivated by the observation that there may be dependencies among the different predicates.

[21] In each run, different predicates were chosen.

D. Different number of values, binomial and dependency distribution, ordered values: This case is similar to the previous one, but the values of the independent predicates were chosen with a binomial distribution, rather than an even distribution. We considered this case also so as to show that the type of distribution (which is not realistic) does not really affect our results.

The consideration of the above four different cases demonstrates the applicability of our algorithm in a wide range of situations.

3.4.6 Results and Explanations

The results of cases B–D are presented in Figures 5–7, respectively[22]. The rows correspond to the given number of objects, and the columns correspond to the given number of predicates. In general, the results of the simulations show that the success rate of the algorithm was very high.

For example, in case B (Figure 5), if there are at least 6 objects and more than one predicate, then in at least 99% of the possible worlds the agent will definitely choose the same object. If there are more than 2 predicates and more than 2 objects, then in at least 97% of the worlds the agents will definitely choose the same object. The only cases where, only in a very low percentage of the worlds, the agents will definitely choose the same object are when there is only one predicate or two objects. When there is only one predicate, there is not enough information about the objects to allow for identifying one of them as a focal point. When there are only two objects, in situations which are symmetric, e.g., if one of the objects is green and the other blue, the measure of their rareness is the same. As we explained above, in these cases the agents may still use a variation of the algorithm to increase the probability of their choosing the same object (see Section 3.4.9 below for details).

A similar success rate was obtained in Cases C and D. In Case C, if there are at least 5 objects and more than 2 predicates, then in at least 99% of the possible worlds the agents will definitely choose the same object. If there are more than 2 objects and at least 3 predicates, the success rate is at least 97%. We suspect that the need for at least 3 predicates to obtain the high rate, compared with Case B, where 2 predicates were enough, is due to the dependency among the predicates in Case C. Nevertheless, the results demonstrate that even in this case of dependency among the predicates, a low success rate is obtained only in cases where there are only one predicate and two objects.

[22] We discuss Case A below.

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
2	85	61	83	73	85	82	88	83	87
3	97	100	100	99	99	99	99	99	99
4	91	90	97	98	97	98	99	99	99
5	93	98	99	99	100	99	100	100	100
6	91	99	100	100	100	100	100	100	100
7	89	99	100	100	100	100	100	100	100
8	87	99	100	100	100	100	100	100	100
9	85	99	100	100	100	100	100	100	100
10	87	99	100	100	100	100	100	100	100
11	83	99	100	100	100	100	100	100	100
12	84	100	100	100	100	100	100	100	100
13	82	99	100	100	100	100	100	100	100
14	81	100	100	100	100	100	100	100	100
15	80	99	100	100	100	100	100	100	100
16	77	99	100	100	100	100	100	100	100
17	79	99	100	100	100	100	100	100	100
18	73	99	100	100	100	100	100	100	100
19	72	99	100	100	100	100	100	100	100

Figure 5: Probability of definitely choosing the same object in case of different number of values, even distribution (B).

In Case D, we obtained slightly better results than in Cases B and C. In Case D, if there are at least 5 objects (as in Case C) and more than 1 predicate (as in Case B), then in at least 99% of the possible worlds the agents will definitely choose the same object. If there are more than 1 object and at least 1 predicate (as in Case B), the success rate is at least 97%. Thus, changing the way in which the values of the independent predicates were chosen, from even to binomial, further improved the results slightly; but in all cases, regardless of the exact configuration, we obtained excellent results.

Case A, discussed above, consists of nine tables, not all of which are presented here. Instead, we present the two extreme situations from among these nine tables. In Figure 8 the results of the simulations when there is only one predicate are presented, and in Figure 10 the results in the case where there are nine predicates are presented. In both cases the rows correspond to the given number of objects (as in Figures 5–7), but the columns correspond to a given number of values of the predicates (which was chosen, in Cases B–D, randomly.)

In Cases B–D, when there was only one predicate, and the number of values for each predicate was chosen randomly between 2–19, the focal point algorithm did not have a success

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
2	85	67	86	76	87	85	83	88	89
3	97	99	99	99	99	99	99	100	99
4	91	93	97	97	99	99	99	100	100
5	93	98	99	100	100	99	100	100	100
6	91	98	100	100	100	100	100	100	100
7	89	99	100	100	100	100	100	100	100
8	87	98	99	100	100	100	100	100	100
9	85	98	99	100	100	100	100	100	100
10	87	97	100	100	100	100	100	100	100
11	83	98	100	100	100	100	100	100	100
12	84	98	100	100	100	100	100	100	100
13	82	97	100	100	100	100	100	100	100
14	81	98	99	100	100	100	100	100	100
15	80	98	100	100	100	100	100	100	100
16	77	97	100	100	100	100	100	100	100
17	79	99	100	100	100	100	100	100	100
18	73	96	99	100	100	100	100	100	100
19	72	97	99	100	100	100	100	100	100

Figure 6: Probability of definitely choosing the same object in case C.

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
2	81	72	82	79	86	85	90	89	92
3	95	99	99	99	99	99	100	99	99
4	90	97	98	98	99	100	99	99	100
5	93	99	100	100	100	100	100	100	100
6	91	99	99	100	100	100	100	100	100
7	89	100	100	100	100	100	100	100	100
8	87	99	100	100	100	100	100	100	100
9	87	99	100	100	100	100	100	100	100
10	89	99	100	100	100	100	100	100	100
11	87	100	100	100	100	100	100	100	100
12	84	100	100	100	100	100	100	100	100
13	85	100	100	100	100	100	100	100	100
14	82	100	100	100	100	100	100	100	100
15	83	99	100	100	100	100	100	100	100
16	80	99	100	100	100	100	100	100	100
17	84	100	100	100	100	100	100	100	100
18	78	99	100	100	100	100	100	100	100
19	78	99	100	100	100	100	100	100	100

Figure 7: Probability of definitely choosing the same object in case D.

No.	No. of Values for each Predicate																
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	7	18	19
2	53	65	77	79	83	88	87	89	89	93	92	93	95	93	93	93	96	94
3	78	92	93	95	96	97	97	99	99	99	99	99	99	99	100	99	99	99
4	46	74	83	87	91	95	95	97	97	97	97	99	98	98	99	99	99	99
5	28	72	88	94	96	98	97	99	99	99	99	99	99	99	99	100	100	99
6	20	64	84	92	95	96	97	97	98	98	99	99	99	99	99	99	99	99
7	9	58	77	87	91	95	96	98	98	98	99	99	99	99	99	99	99	99
8	7	41	77	89	94	97	98	97	99	99	99	99	100	99	99	100	99	100
9	4	30	72	80	91	91	94	97	98	98	98	98	99	99	99	100	99	99
10	2	28	63	83	90	95	97	98	99	99	100	99	99	99	99	99	99	100
11	2	20	51	73	89	93	94	97	98	97	99	98	99	99	99	98	99	99
12	0	12	44	77	89	94	95	97	99	99	99	99	100	99	100	100	99	100
13	0	6	34	65	84	91	93	95	97	98	97	98	99	98	99	99	99	99
14	0	9	31	63	81	92	93	97	99	98	99	99	100	99	100	100	99	100
15	0	6	26	55	73	90	92	95	98	98	99	99	99	98	99	99	100	99
16	0	3	18	48	75	86	91	95	97	99	100	99	99	100	99	99	100	99
17	0	2	17	43	71	83	93	97	96	98	99	99	99	99	99	99	100	100
18	0	0	13	39	58	83	92	96	98	98	98	99	99	99	100	100	99	99
19	0	0	11	33	59	75	89	94	96	96	98	99	99	99	99	99	99	99

Figure 8: Probability of definitely choosing the same object in case A, with one predicate.

rate as high as in the other configurations. The results in Figure 8 allow us to conclude that, even in the case where there is only one predicate in the world (e.g., height), if the number of values for each predicate is large enough, the chances that in any given world the agents will definitely meet are very good. For example, if there are more than seven predicate values, more than two objects, and fewer than 19 objects, in at least 91% of the worlds the agents will definitely meet. The only cases where there is a low chance that the agents will definitely be able to meet are when there are two or three values to the predicate. When there are 9 predicates (Figure 10), and in all cases where there are more than two objects, in at least 98% of the possible worlds the agents will definitely meet.

It can be observed that the results presented in Figure 10 are better than the results presented in Figure 9, which are better than those presented in Figure 8. In general, as the number of predicates and/or values increases in the world, the data describing an object become more complex. If the data are more complex (i.e., chosen out of a wider range of possibilities), there is a higher chance of finding a special object, i.e., an object which is rare or extreme. Thus, in these cases the probability that both agents will find a unique focal point is higher than in the cases where the agents' data is simpler, and thus the objects have similar properties.

There are also some specific configurations which lead to better results. For example, in most of the cases (see for example Figures 5, 6) when there are 3 objects with a small number of predicates, the probability is higher than when there are 2 or 4 objects. A similar effect is found for higher numbered odd and even objects as well. However, the effect weakens. Although the selection of an object as a focal point is, in our algorithm, driven by a combination of factors, it should be noted that some of the factors may have greater weight in some situations. A typical reason for this differential effect of factors is that some factors have an overall relatively low contribution to the focal point score and/or a low variability between the objects. However, as the number of relevant factors goes down, the individual behavior of the factors begin to take central stage. Thus, a possible explanation for the supremacy of low-count odd numbers of objects is that the probability of an object having a "unique" score may be higher if there is an odd number of objects, possibly since, if there is an even number of objects, the scores may be paired. When the number of objects increases, the probability of pairing goes down, so there is less of an effect of the parity. Generally, an overall low focal point score is indicative of situations where only a small number of factors are relevant.

No.	No. of Values																
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	80	83	80	85	82	86	85	88	88	91	88	88	90	89	92	89	92	92
3	99	99	99	100	100	99	99	99	100	99	100	100	100	100	100	100	100	99
4	98	100	99	100	99	99	99	98	98	99	98	98	98	97	97	99	99	99
5	99	100	100	100	100	100	99	100	100	100	99	99	100	100	99	99	99	99
6	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
7	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
8	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
11	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
12	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
13	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
14	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
15	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
16	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
17	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
18	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
19	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Figure 9: Probability of definitely choosing the same object in case A, with seven predicates.

No.	No. of Values																
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	82	82	86	84	87	86	87	89	88	88	92	91	90	89	91	87	92	91
3	100	99	100	100	99	99	100	99	99	99	99	99	100	99	98	98	98	98
4	98	99	100	100	99	99	99	99	99	99	99	98	98	99	98	99	98	98
5	100	100	100	100	100	100	99	100	100	100	100	100	99	99	100	99	100	100
6	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
7	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
8	99	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
11	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
12	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
13	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
14	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
15	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
16	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
17	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
18	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
19	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Figure 10: Probability of definitely choosing the same object in case A, with nine predicates.

3.4.7 Modification of the experiment

Several simple modifications of the simulated worlds were tested. The results presented above covered cases where both agents had the exact values of predicates in the world. We ran similar sets of experiments where the agents have only the ordinal values. For example, suppose there are three houses in the environment. The first is 3 meters tall, the second is 5 meters tall, and the third is 9 meters tall. Then, in this modified simulation, one agent can have the information $Height(1,2)$, $Height(2,3)$, and $Height(3,5)$, while the second agent may have $Height(1,4)$, $Height(2,7)$, and $Height(3,9)$. The probability of the possible worlds where the agents definitely meet, using the focal point algorithm, was very similar to the world in which both have the same exact values.

With the same focal point function, we tested the case in which not all the predicates had an ordering and only some (i.e., half) had values. For example, colors and fruit types cannot be ordered. The results obtained seemed to show that the algorithm disregarded the predicates that did not have an ordering and gave similar results in the case where the number of predicates was half the number of predicates tested in this case. This is reasonable, since the predicates where both extremeness and rareness were computed dominated the ones in which only rareness was considered.[23]

To take care of the case where no predicate had an ordering, we slightly modified the focal point function and used only the rareness property. The results in Case B are presented in Figure 11. It is clear that when there are only two objects, there is no way to choose one of them based on rareness; hence the zeros in row 2. Also, when there is only one predicate, the probability that the agents will definitely meet is very low. However, if there are at least 7 objects and at least 4 predicates, then, in more than 94% of the worlds, the algorithm will succeed. In general, as the number of predicates and objects increases, the probability of definitely meeting also increases, even in this case, where the values are not ordered. This seems to be the case in all of our experiments and indicates the convergence of our algorithm. As mentioned above, as the number of predicates and values increases, the information describing each object becomes more complex, which in turn increases the probability of finding a special object which is chosen as a focal point.

The next case tested was one where both agents see the same world, but each agent has its sensors calibrated differently, so that each "sees" in a different dynamic range (one has $\frac{3}{4}$

[23]This is essentially a feature of our specific algorithm. The focal point value for each object was a combination of the rareness (R_i^P) and the extremeness ($E_i^{P,\leq,>}$) with respect to all predicates. If the values of a predicate P are not ordered and $E_i^{P,\leq,>} = 0$, it does not contribute to the focal point value, so the value that this predicate contributes to the sum is, on the average, only half of what is contributed by each of the other predicates, and thus P has less effect on the focal point value than do the ordered predicates.

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
2	0	0	0	0	0	0	0	0	0
3	31	54	68	77	80	87	88	91	91
4	8	33	42	55	62	71	75	82	82
5	17	55	76	86	90	93	94	97	97
6	9	49	68	81	88	93	95	96	98
7	11	65	88	94	97	98	98	99	100
8	9	61	82	96	97	99	99	99	100
9	10	80	92	99	99	99	99	100	100
10	10	75	92	98	99	99	100	100	100
11	11	81	97	99	99	100	100	100	100
12	12	80	97	99	100	100	100	100	100
13	11	86	99	99	100	100	100	100	100
14	11	83	97	99	100	100	100	100	100
15	11	89	99	99	100	100	100	100	100
16	12	87	99	100	100	100	100	100	100
17	12	92	100	100	100	100	100	100	100
18	10	90	99	100	100	100	100	100	100
19	10	92	100	100	100	100	100	100	100

Figure 11: Probability of definitely choosing the same object in case of a different number of values, even distribution (B), non-ordered values.

.

of the range of the other). This algorithm yields a probability of meeting which is lower than 50%, and decreases as the number of objects and values rises. If the results are compared to the random case (where objects are chosen at random, for lack of a better method), the results are still improved, using the focal point algorithm, but not as dramatically improved. We believe that this effect is due to some specific technical features of our algorithm. Our algorithm strongly depends on two predicate values being differentiated by one agent iff they are also differentiated by the other one. Thus, for example, if one agent observes that objects 1 to 5 have values 1, 2, 2, 4, and 4, respectively, with respect to a given predicate, then it will choose the first object as a focal point. If the sensors of the other agent are calibrated differently and it cannot "see" the value 1, observing only the values 2, 2, 2, 4, and 4 for the same predicate, for objects 1 to 5 respectively, then the second agent will not select the first object, but will randomly decide between the fourth and fifth objects.

The above results were obtained when we used both the uniqueness and the extremeness criteria. We suspect that if we had used only extremeness and assumed that, given values a and b of a given predicate, $a \neq b$ for agent one iff $a \neq b$ for the second agent, then the results obtained would be closer to 100%, as in the other cases.

3.4.8 Environments with Noise

In our model we assume that an agent uses its own view of the world to select a focal point. However, there are situations where the agents' information about the world is not exactly the same. We tested the performance of our algorithm in such situations by introducing noise to the environment. As can be expected, the algorithm does not perform as well in these cases as in cases with full information; however, it still performs much better than random choice and other methods that we checked.

In this set of simulations we considered configurations as in Case A: i.e., in each run, the number of possible values for all predicates are the same. In each run of the simulation, one agent received a randomly generated world as in the previous cases. The other agent received a modified world, in which with probability X, the value associated with a given predicate and a given object was different from the one in the original world and was chosen randomly from the rest of the values. For example, suppose that, in a configuration of a specific run, the predicate Color has three possible values: white, black, and green. If in the original world (given to the first agent), the color of object 1 is white, then, with probability X, the color will *not* be white, but either black or green. We consider situations where X was either 5% or 10% and present the results when there is only one predicate (Figures 12 and 13), and when there are seven predicates (Figures 14 and 15.) We also considered other situations and obtained similar results.

It is clear that as the amount of noise in the environment increases, the performance of the algorithm decreases, but it is still much better than choosing randomly. In particular, as the number of objects increases the performance of the algorithm decreases (but so does the probability of success when choosing randomly). In addition, as the number of predicates increases, the performance decreases. We suspect that this is because the number of errors increases when the number of objects and the number of predicates increase. For example, the probability that the agents will have exactly the same world when there are 9 objects, 7 predicates with only 2 values, and 5% or 10% noise is only 0.039 and 0.0013, respectively. On the other hand, as the number of values of the predicates increases, the performance of the algorithm improves. This is because the rareness and the extremeness properties do not depend on the exact values of the predicates, and as the number of values increases, the damage that can be caused by the errors is less severe.

We also developed other algorithms which did not depend on the exact properties of the objects, but rather on their "distance" from each other in the "properties space." We found out that the performance of our original algorithm was the best ([27].)

3.4.9 Extension of the Algorithm to Probabilistic Choice

In previous sections, the focal point algorithm provided the agents with a mechanism for guaranteeing their meeting. That is, the algorithm provided a technique to identify if there was a unique object. If the algorithm succeeded, the meeting was guaranteed. If the algorithm failed, the agents were aware of the failure. As we explained before, the data presented in the tables above measured the probability of definitely meeting in a random world with different sets of configurations. However, even when the focal point algorithm fails, the information that it provides can be used to increase the probability of a meeting.

A focal point algorithm cannot guarantee a definite meeting in a given world when $UFP = \emptyset$. In such situations, the agents should look for the smallest set of objects whose focal point value is the same and choose one of them randomly. This is described in the following algorithm.

Algorithm 2 Extended Focal Point Algorithm for Probabilistic Meeting

1. *Perform steps* $(1-3)$ *of Algorithm 1.*

2. *If* $UFP = \emptyset$*, then divide the objects in* $\mathcal{T}erm$ *to sets,* $S_1,\ldots,S_k$*, where* $1 \leq k \leq |\mathcal{T}erm|$[24] *such that the objects of each* S_j *have the same focal point value, i.e., for all* $1 \leq j \leq k$

[24]Note that since $UFP = \emptyset$, $k \leq \frac{|\mathcal{T}erm|}{2}$.

No.	No. of Values for each Predicate																	
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	7	18	19
2	51	61	72	75	79	83	82	85	86	88	88	89	91	91	90	89	91	93
3	70	81	86	91	91	90	93	91	92	94	95	93	93	94	94	95	95	92
4	45	68	78	81	87	88	90	89	90	90	89	91	92	91	93	94	91	93
5	30	67	81	82	86	86	89	89	90	89	91	92	90	92	92	91	92	91
6	19	58	75	83	82	86	86	86	88	90	89	90	90	90	90	90	88	92
7	8	48	67	77	79	81	88	85	88	89	87	90	92	91	91	89	91	88
8	4	34	66	71	81	82	88	88	87	88	85	88	86	87	85	87	91	90
9	4	33	57	72	80	81	82	83	86	86	88	87	88	87	88	89	88	91
10	0	21	51	69	76	79	83	86	83	82	87	85	86	87	85	88	88	89
11	0	13	46	62	73	76	80	83	82	82	86	83	81	86	87	87	85	87
12	0	8	41	64	72	77	76	82	84	83	85	83	84	88	87	86	88	88
13	0	7	32	57	70	73	79	78	78	82	83	84	82	87	86	83	87	87
14	0	5	30	48	67	76	80	83	83	79	82	83	84	81	82	85	87	86
15	0	2	20	45	60	69	74	80	79	80	80	85	81	84	83	83	85	85
16	0	3	15	37	60	68	73	75	78	79	80	85	82	82	82	83	84	86
17	0	2	10	33	50	64	75	80	77	80	79	82	81	81	85	85	87	83
18	0	1	12	27	52	62	69	74	78	77	81	82	81	81	81	83	85	87
19	0	1	8	22	49	58	70	72	73	75	80	81	80	81	83	84	84	84

Figure 12: Probability of definitely choosing the same object in case A, with one predicate and 5% of noise.

No.	No. of Values for each Predicate																
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	7	18	19
2	48	62	68	72	78	81	78	86	80	86	87	87	87	91	87	88	87	89
3	67	74	82	84	83	85	88	87	89	87	89	87	90	89	87	89	89	89
4	41	66	72	76	79	80	82	83	83	83	83	85	87	83	86	85	85	86
5	30	61	71	78	79	81	84	79	82	84	82	82	83	80	84	85	86	82
6	15	58	62	74	76	77	78	80	83	76	81	86	83	83	85	83	82	85
7	6	47	61	69	71	76	74	77	79	78	81	78	81	84	85	83	82	84
8	3	35	55	60	70	69	76	79	77	77	79	77	77	79	80	81	82	79
9	2	25	48	59	62	66	72	74	73	76	78	77	76	76	77	75	78	78
10	0	18	47	60	65	68	69	74	70	76	78	77	81	76	76	77	79	74
11	1	15	35	55	59	66	70	70	69	71	73	75	76	74	74	76	77	73
12	0	7	33	50	62	64	64	68	69	72	72	77	73	76	74	77	78	76
13	0	8	28	49	55	60	63	65	64	68	71	72	72	75	74	73	76	74
14	0	3	20	37	52	58	64	64	68	67	72	72	69	74	71	74	75	76
15	0	2	18	36	48	56	63	66	66	65	72	73	67	71	70	71	72	72
16	0	2	10	32	45	54	62	64	65	69	70	65	68	70	74	70	70	73
17	0	1	11	29	45	55	61	64	65	66	67	69	66	70	66	70	71	74
18	0	1	9	24	41	49	54	58	59	64	63	66	67	68	73	73	73	73
19	0	1	6	22	36	49	57	57	64	61	63	63	69	67	66	68	70	68

Figure 13: Probability of definitely choosing the same object in case A, with one predicate and 10% of noise.

No.	No. of Values for each Predicate																	
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	7	18	19
2	74	79	79	78	77	79	80	84	82	83	83	84	86	85	89	88	88	85
3	83	85	86	86	84	84	85	82	86	87	87	87	86	85	85	83	87	86
4	80	78	77	81	79	76	79	76	79	77	77	74	77	79	78	76	75	77
5	84	76	75	78	75	76	74	76	77	76	72	76	75	73	77	74	74	74
6	78	73	76	73	69	75	73	74	73	71	77	76	73	75	75	75	75	73
7	73	76	71	72	67	73	76	74	71	73	75	76	72	71	76	73	71	74
8	75	72	72	70	68	71	67	69	73	72	70	72	70	69	70	72	67	67
9	75	72	70	66	68	66	65	70	67	64	68	68	72	67	68	67	69	70
10	74	70	65	67	68	66	65	68	66	68	69	67	69	69	70	72	69	71
11	75	70	67	68	68	67	65	66	64	63	65	67	62	63	69	67	64	62
12	75	68	69	69	64	67	64	68	67	63	64	66	63	64	65	64	64	66
13	72	65	67	61	62	65	64	65	68	63	65	66	64	67	63	68	61	68
14	71	67	63	63	62	69	65	64	63	66	67	65	62	69	65	66	61	62
15	76	67	64	68	61	59	62	59	62	67	68	63	64	60	61	62	67	66
16	69	64	67	62	62	63	62	64	60	59	59	63	65	62	65	59	62	61
17	73	70	66	62	57	62	63	65	58	62	61	65	64	60	62	65	60	62
18	68	67	66	64	64	62	62	63	63	58	62	61	60	61	62	60	62	61
19	70	65	63	62	64	57	62	63	63	61	57	61	60	59	59	59	64	63

Figure 14: Probability of definitely choosing the same object in case A, with seven predicates and 5% of noise.

No.	No. of Values for each Predicate																
Obj	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	7	18	19
2	71	76	73	74	72	77	78	77	80	76	76	81	78	81	80	84	79	79
3	73	73	74	72	73	73	72	73	73	78	76	79	75	77	74	76	76	73
4	67	69	67	66	67	67	69	69	65	62	67	64	64	64	62	63	65	62
5	64	67	66	60	63	61	62	60	59	63	63	64	61	58	59	61	60	59
6	65	63	62	56	60	62	63	55	57	63	60	63	63	58	59	57	59	62
7	61	58	58	58	61	62	58	56	58	60	59	56	57	61	59	61	61	61
8	61	55	57	53	55	55	55	53	57	53	57	53	55	54	55	56	53	55
9	56	60	53	55	53	57	54	53	55	54	55	49	55	56	51	51	57	54
10	59	56	56	57	50	53	48	53	56	53	56	50	50	50	50	55	53	52
11	61	55	56	58	52	52	52	51	50	55	53	50	51	47	52	55	52	51
12	58	50	49	44	49	51	48	48	49	50	49	50	49	51	52	46	49	51
13	60	52	54	52	52	50	49	51	47	51	53	45	52	48	50	50	53	49
14	58	50	49	52	53	54	47	46	47	49	49	50	47	53	51	52	48	50
15	59	53	51	50	48	49	45	50	51	47	50	52	44	48	43	47	47	50
16	54	51	51	47	46	45	48	49	49	49	46	48	47	48	47	47	45	45
17	61	54	53	45	46	45	51	48	45	49	44	44	48	45	47	49	48	45
18	58	54	52	49	46	47	47	44	50	45	43	43	51	44	47	45	45	45
19	53	53	48	47	48	43	45	49	51	47	45	49	46	48	47	48	44	45

Figure 15: Probability of definitely choosing the same object in case A, with one predicate and 10% of noise.

and for all $i, i' \in S_j$, $F(i) = F(i')$. Denote by f_j the focal point value of the elements in S_j.

3. *Let SFP $= \{S_j \mid 1 \leq j \leq k,$ and $\forall 1 \leq l \leq k, |S_j| \leq |S_l|\}$. That is, SFP includes all the sets whose number of elements are minimal. Let $\bar{S} = argmax_{S_j \in SFP} f_j$.*

4. *Choose randomly one of the objects of $\bar{S}$ and denote it by c.*

In this case, the algorithm does not guarantee a meeting, but increases the probability of meeting, as compared to a random choice. The probability that the agents meet in cases where $UFP = \emptyset$ and the agents use Algorithm 2 is $\frac{1}{|\bar{S}|}$, while the probability that they meet when they don't use the focal point algorithm and just choose one object randomly is $\frac{1}{|Term|}$.

Let p denote the probability of definitely meeting in a random world in one of the configurations that we considered in the previous section. The overall probability of meeting in that case, using Algorithm 2 (i.e., in both the cases where $UFP \neq \emptyset$ and $UFP = \emptyset$), is as follows: $P_a = p + \frac{1-p}{|\bar{S}|}$.

In Figure 16, we present the results for P_a after running the three-step simulation 500 times for the configuration in case B described above. In this case, if there are at least 2 objects and 2 predicates, the probability of choosing the same object is at least 98%. In the other cases, the probability of success is at least 80%.

Similarly, when we consider the non-ordered case using the technique above, we get the results for P_a that appear in Figure 17. In this case, the probability of choosing the same object is at least 94% when the number of predicates is at least 4 and the number of objects is at least 7.

The results which are reported in Figure 16 should be compared with those of Figure 5, and Figure 17 should be compared with Figure 11. For example, it is easy to see that for any configuration, the probability in Figure 16 is greater than the corresponding one in Figure 5. This follows immediately from the above formula of P_a, which is used in Figure 16 and which is always greater than p, which is used in Figure 5. For example, when the number of objects in the world is only two, and the original focal point algorithm does not prefer one object over the other, i.e., $UFP = \emptyset$, $\bar{S}$ includes both objects, and the probability of choosing one of them randomly is .5. In particular, if there are five predicates in the world, the probability of definitely meeting, in the original algorithm (Figure 5), is 85%, and the probability of meeting in the extended algorithm (Figure 16), is 92%. However, when there are more objects, the extended algorithm provides better results than does choosing randomly in case of a failure (i.e., $\bar{S}$ does not include all the objects). For example, when there are 10 objects and one predicate, the probability of definitely meeting (Figure 5) is

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
2	92	80	91	86	92	91	94	91	93
3	98	100	100	99	99	99	99	99	99
4	95	94	98	99	98	99	99	99	99
5	95	99	99	99	100	99	100	100	100
6	93	99	100	100	100	100	100	100	100
7	92	99	100	100	100	100	100	100	100
8	90	99	100	100	100	100	100	100	100
9	89	99	100	100	100	100	100	100	100
10	91	99	100	100	100	100	100	100	100
11	88	99	100	100	100	100	100	100	100
12	89	100	100	100	100	100	100	100	100
13	87	99	100	100	100	100	100	100	100
14	86	100	100	100	100	100	100	100	100
15	85	99	100	100	100	100	100	100	100
16	83	99	100	100	100	100	100	100	100
17	85	99	100	100	100	100	100	100	100
18	80	99	100	100	100	100	100	100	100
19	80	99	100	100	100	100	100	100	100

Figure 16: Probability of choosing the same object using algorithm 2 in case of different number of values, even distribution (B).

No. of Objects	No. of Predicates								
	1	2	3	4	5	6	7	8	9
3	53	69	79	85	86	91	92	94	94
4	41	58	66	74	78	84	86	90	90
5	47	73	85	91	94	95	96	98	98
6	39	68	80	88	93	95	97	98	99
7	42	78	93	96	98	98	99	99	100
8	38	76	89	97	98	99	99	99	100
9	39	87	95	99	99	99	99	100	100
10	37	84	95	99	99	99	100	100	100
11	38	89	98	99	99	100	100	100	100
12	38	88	98	99	100	100	100	100	100
13	36	91	99	99	100	100	100	100	100
14	35	90	98	99	100	100	100	100	100
15	34	93	99	99	100	100	100	100	100
16	34	92	99	100	100	100	100	100	100
17	35	95	100	100	100	100	100	100	100
18	32	94	99	100	100	100	100	100	100
19	33	95	100	100	100	100	100	100	100

Figure 17: Probability of choosing the same object using algorithm 2 in case of different number of values, even distribution (B), non-ordered values.

87%. Using the naive algorithm of choosing randomly among the 10 objects will increase the probability to 88%. However, using our extended algorithm, we obtained a probability of meeting of 91% (Figure 16).

3.4.10 Discussion of the experimental results

The series of simulations which we described above and which were run over various randomly generated worlds demonstrates the usefulness of the focal point algorithm as a heuristic for multi-agent coordination.

The algorithm is shown to be successful in a wide variety of cases, including cases where the only information that the agents can use for their coordination is the non-ordered values of a few properties (predicates). In all the cases, the algorithm converges when the number of predicates and their values is large enough.

It is difficult to find experiments with settings similar to ours to make comparisons. Related experiments were performed by Stephens and Merx [72], measuring the performance of three different control strategies – local-control, distributed-control, and central-control – in solving the Pursuit Problem. In their experiments, the local-control system had the lowest communication overhead: 4 broadcasts for an entire pursuit. However, it had a very low capture ration: 0.1. The distributed-control system required 4 broadcasts for each unit move and was the most efficient strategy. However, in term of capture ratio (0.833), it did worse than did the central-control (1.000), which required only 1 broadcast for each unit move. Currently, our algorithm assumes that all the objects have the same utility; thus more work is required in order to apply it to the Pursuit Problem. However, based on our high success ratio when all the objects have the same utility, we expect it to do better than the distributed-control strategy.

4 A Logic Approach to Focal Points

In the previous section, we assumed that the knowledge bases of the agents do not change while they search for a focal point. This assumption greatly simplified the algorithm, but may not be realistic. Agents acting in dynamic worlds obtain information continuously. This information may be useful in finding a focal point. Furthermore, computational complexity seems central to identifying focal points. This condition has been captured in the decision theory model by assuming that each fact in the database is tagged with a "measure of difficulty." However, taking the time to reason when looking for a focal point seems essential in situations in which not all the information is provided to the agents in advance, and when they need to collect and process information while searching for focal points.

The logic-focal point approach considers these issues. The intuition behind it is that the agent will continually look for candidates in the domain that have certain properties (such as uniqueness). If something in the domain has that property, it is a focal point at the time it is identified. As time goes on, new beliefs are derived (e.g., through modus ponens), and the domain over which the search is being conducted also expands (through observations or consideration of new conjunctive properties). Then the search for candidate focal points is repeated — and an old focal point, given the new information, may no longer remain as one. The search for focal points is cut off at some depth of computation, depending on time constraints, at which point the agent attempts to resolve competing focal points. The logic approach does not assume that the agents have utilities to assign to different outcomes or measures of difficulty that are associated with the facts of the database. The search for focal points proceeds without that information.

We assume that as time passes while the agent is searching for focal points, the agent may discover new properties of objects via three main mechanisms:

1. Deduction of new facts from old ones.

2. Observation.

3. Increase in the set of predicates.

Their role in searching for focal points is as follows. If the agent has a deductive database and can reach new conclusions based on its explicit database, then it may deduce new facts about the candidates for focal points. Using these new facts, it may be realized that some objects satisfy some criteria, e.g., uniqueness, to qualify them as focal points. We demonstrate this mechanism in the following example.

Example 7 Suppose the original database of an agent includes the following facts: Type(h1,House), Type(h2,House), Type(h3, house), Type(b1,Bridge),Type(b2,Bridge), *Architect-Of*(h2, Smith), *Architect-Of*(b2, Smith), Color(h1,White) and Color(b1,White). In addition, it has the following axioms:
$\forall x\, Type(x, House) \wedge \text{Architect-Of}(x, Smith) \rightarrow Color(x, Black)$
$\forall x\, Type(x, Bridge) \wedge Color(x, White) \rightarrow Size(x, Large)$
$\forall x\, Type(x, House) \wedge Color(x, Black) \rightarrow Age(x, 25)$.

The agent is not able to identify any focal point from the initial database: there is no object which is unique, extreme, central, or even a unique complement. The agent may start to deduce new facts about the objects, using its axioms and modus ponens. It first realizes that Color(h2,Black) and Size(b1,Large). Now it has two candidates for focal points: h2, because it is the only black object, and b1, because it is the only object which is large. The agent may continue its reasoning and realize that Age(h2,25). In that step, h2 may be chosen as a focal point since it is unique with respect to two properties, color and age, and there is no other object which is unique with respect to two properties.

In other situations, an agent may observe new properties of objects that it is familiar with, or may observe that there are new objects in the environments. Consider the following example.

Example 8 Suppose an agent knows that there are three houses in the environment, but does not know any facts about them. In particular, its database includes exactly three facts: Type(h1,House), Type(h2,House), and Type(h3, house). Thus, it cannot choose any of the objects as a focal point. However, it may look around and observe that houses h1 and h2 are white, while h3 is black. After doing so, it can choose h3 as a focal point, since its color is unique. This agent may observe the area some more and find out that there is also a black bridge in the area. This new information may disqualify h3 from being a focal point, since its color is no longer unique, but may lead the agent to choose the bridge as a focal point since its type is unique.

There are situations in which the agent can find focal points by using combinations of predicates specified in its initial database, as in the following example.

Example 9 Suppose the initial database of the agent consists of the following facts: Type(h1,House), Type(h2,House), Type(h3, House), Type(b1,Bridge), Type(b2,Bridge), Type (b3,Bridge), Color(h1, White), Color(h2, White), Color(h3, White), Color(b1,White), Color(b2, Black), and Color(b3, Black). In this initial database there is no object that may qualify as a focal point. However, the agent may consider a combination of properties, i.e., Type and Color, and realize that b1 is the only bridge which is white.

Often, using the above mechanisms will produce competing focal points. Furthermore, objects that were identified as focal points, at one point, may be disqualified later. For example, in the scenario of Example 7, after one iteration of deductions, both h2 and b1 are considered as focal points, and only in the second round is it possible to choose one of them, i.e., h2 as a focal point. In example 8 above, first the black house qualified as a focal point, but further observation of a black bridge disqualified it as a focal point. In such situations, different heuristics, relying on the time at which these focal points were identified, may be used to resolve conflicts among the competing focal points. For example, one may give priority to objects that were identified as focal points earlier, rather than later, since there is a higher probability that the other agent will be able to find them too. For the same reasons, it may choose focal points that have been focal points for more steps than have the others.

4.1 Discussion of Step-Logic

In order to implement the mechanisms discussed above, there is a need for a logical framework which is able to model the computational process itself in the reasoning procedure as the agents search for focal points. For example, if the agent finds a focal point after making some deductions, then we would like to identify the ones that were easier to find. In addition, there is need for a framework which allows for observations and the incremental increase of the set of predicates. Classical first order logic does not model the computational process.

Consider the deduction database of Example 7 described above, with the additional facts Near(h2,b2), Near(h2,h3), and the following additional axiom: $\forall x, y, z(Age(x, 25) \wedge Color(x, z) \wedge Near(x, y)) \rightarrow Color(y, z) \wedge Age(y, 25) \wedge Size(y, Large)$. The first and second iterations proceed as in Example 7; i.e., in the first iteration, Color(h2,Black) and Size(b1,Large) are deduced, and in the second iteration, Age(h2,25) is deduced. In the third iteration of the deduction process, the agent may conclude that Color(b2, Black), Color(h3,Black), Size(b2, large), Size(h3,Large), Age(b2,25), and Age(h3,25). At this stage, no object may be qualified as a focal point; however, h2 may still be considered as a focal point, based on its properties in the previous iteration.

The identification of the above focal points is not possible in first order logic, where there is no formal way to model the deduction process and its time frame. An agent, using FOL, may close the initial data base under inferences and obtain all the facts that were deduced in the three iterations. Using this set, no object may qualify as a focal point. Thus, we need some framework in which each iteration will be formally defined and the agent will be able to maintain information about the time at which a specific fact was proven and will be able to take this into consideration in its search for focal points.

Since classical first order logic is not sufficient for our needs, we turn, instead, to a modification of first order logic, called *step-logic*, that deals explicitly with the passage of time as an agent reasons and allows us, formally, to introduce observations.[25]

In the formalism of step-logics, introduced by Elgot-Drapkin, Miller, and Perlis ([64, 24, 23]), inferences are parameterized by the time taken for their inference, and these time parameters themselves can play a role in the specification of the inference rules and axioms.[26] Step-logics offer a natural representation of the evolving process of reasoning itself. A *step* is a fundamental unit roughly characterized by the time it takes the agent to draw a single inference.

Observations, which are inputs from the external world, may arise at the beginning of a discrete time-step and may be denoted by by the meta-predicate Observe. When an observation appears, it is considered to be a belief in the same time-step. Apart from his observations at the beginning of step i, the only information available to the agent is a snapshot of his deduction process completed up to and including step $i-1$. During step i, the agent applies all available inference rules in parallel, but only to beliefs at step $i-1$ (denoted by $\mathcal{F}acts_{i-1}$).[27]

New beliefs thus generated through applications of inference rules are not available for use in further inference until step $i+1$. For example, consider an agent which has the

[25] There is some related work in Artificial Intelligence that addresses the issues of the passage of time during the reasoning process. In [36], [35], and [67], decision theoretic approaches are used to optimize the value of computation under uncertain and varying resource limitations. In these works, deadlines and the passage of time while reasoning are taken into consideration in computing the expected computational utility. Dean and Boddy [13] formulated an algorithmic approach to the solution of time-dependent planning problems by introducing "anytime algorithms" that capture the notion that utility is a monotonic function of deliberation time. Etzioni [25] considered using a decision theoretic architecture, with learning capabilities, to control problem solving searches.

[26] Step-logics have also been used for planning in deadline situations [63] and for handling some interesting real-time variants of the Yale shooting problem [62].

[27] In [63], an improved version of step-logic that considers space and computation bounds in each step is described. For simplicity, we use the original schema of step-logic here.

following inference rules:

Modus ponens: $\frac{i:a,a\rightarrow b}{i+1:b}$, indicating that if in step i the agent believes that a and that $a \rightarrow b$, then in the next step it believes b.

Inheritance rules: $\frac{i:Type(x,z)}{i+1:Type(x,z)}$, $\frac{i:Color(x,z)}{i+1:Color(x,z)}$, indicating that beliefs with respect to color and type are inherited from one step to the other. In addition, all axioms are inherited from one step to the other.

Time: $\frac{i:\mathrm{Now}(i)}{i+1:\mathrm{Now}(i+1)}$, specifying the way the predicate Now changes over time.

The following describe the reasoning from step i to step $i+1$ and then to step $i+2$, given that the original beliefs of the agent in step i are as specified below.

i**:** Color(c2,White); Type(c2,House); Color(e9,White); Type(e9,House); Type(e3,House); Now(i); $Color(x, White) \wedge Type(x, House) \rightarrow$ Size(x, Big); Size(x, Big) $\wedge$ Size(y, Big) $\rightarrow$ Near(x, y) ...

$i+1$**: Size(c2,Big); Size(e9,Big);** Type(c2,House); Color(e9,White); Type(e9,House); Type(e3,House); **Now (i+1)**; $Color(x, White) \wedge Type(x, House) \rightarrow$ Size(x, Big); Size(x, Big) $\wedge$ Size(y, Big) $\rightarrow$ Near(x, y) ...

$i+2$**: Near(c2,e9); Near(e9,c2)**; White(c2); House(c2); White(e9); House(e9); House(e3); Now($i+2$); White(x) $\wedge$ House(x) $\rightarrow$ Big(x); Size(x, Big) $\wedge$ Size(y, Big) $\rightarrow$ Near(x, y) ...

The new facts in step $i+1$, which are emphasized, are derived by the application of modus ponens to the axiom $Color(x, White) \wedge Type(x, House) \rightarrow$ Size(x, Big) and the relevant facts and application of the time inference rule, which yields Now(i+1). The other facts and axioms are inherited, using the inheritance inference rules. Note that in step $i+1$, the agent does not conclude that c2 is near e9, since it only applies the inference rules to the agent's beliefs at step i. However, at step $i+2$ it reaches the conclusions, Near(c2,e9) and Near(e9,c2), by applying modus ponens to its beliefs in step $i+1$, Size(c2,Big), and Size(e9,Big), and the appropriate axiom. Size(c2,Big) and Size(e9,Big) are not inherited, since there is no relevant rule.[28]

Step-logics are inherently nonmonotonic, in that further reasoning always leads to retraction of some prior beliefs. The most obvious case is $Now(i)$, which is believed at step i, but not at $i+1$. In the following section we will show how the mechanisms discussed in the

[28]The agent may adopt a general default rule that all beliefs are inherited, apart from Now or other special predicates.

previous section can be formalized by using step-logic. We first consider the way in which the agent models the (changing) domain, and then the rules that qualify a candidate as a focal point. Finally, we consider the ways in which an agent resolves competing focal points.

4.2 Domain of Consideration

As in the decision theory case, before the process starts, the agent is given two finite sets enumerating the domain constants (one, *Pred*, is a set of predicates, and the second, $\mathcal{T}erm$, is a set of term constants) over which the focal point computation is going to be done initially. While in the decision theory case this set does not change over time, in the step-logic approach both lists can grow as the computation progresses.

For example, in the Schelling's scenario of dividing 100 objects into two piles (Section 2.4.2), the vectors that sum to 100, with no element less than 0, can be given as an initial finite domain over which properties will be discovered. That is, $\mathcal{T}erm = \{[x, y] \mid x + y = 100,\ x, y \geq 0,\ x, y \in I\!N\}$.

It should be noted that these finite sets represent the *explicit* knowledge of the agent, not its implicit knowledge. For example, an agent may be implicitly aware of the infinite set of positive integers, but for the moment only be considering the finite set of integers from 1 to 500. As time goes on, numbers above 500 may come under the explicit scope of consideration.

4.2.1 Addition of Term Constants

There are two mechanisms for adding new explicit terms. The first is observation, where new term constants are observed over time (e.g., a new bridge is observed, as in Example 8 above). The second mechanism is the use of inductive rules, such as a successor rule that generates new integers.

Example 10 In step i, the domain includes $\mathit{Type}(Bridge, C125)$. In step $i + 1$, the agent has Observe$\{\mathit{Type}(Bridge, C237)\}$. In step $i + 2$, it then has C237 in *Term*.

Example 11 If $Int(x) \rightarrow Int(x + 1)$ is an axiom in step i, and $Int(5)$ is known at step i, then in step $i + 1$ the agent will know $Int(5 + 1)$. Assuming that the agent has the requisite procedure attached to the symbol $+$, it will (in step $i + 2$) add the term 6 to *Term*.

4.2.2 Addition of Predicate Constants

Here we consider the formalization of situations such as the one described in Example 9. When the agent starts, it considers predicates of the original set given to it in choosing a focal point. For example, there might be only one object whose color is white. However, as in Example 9, such a unique object may not exist, and the agent may then consider conjunctions of predicates. For example, there might be only one bridge which is white.

Thus, when the process starts, $\mathcal{P}red$ is equal to the finite set of predicates provided to the agent. In the second step, the agent considers binary conjunctions of predicates from the original list. In step three, it considers ternary conjunctions of predicates from the original list, and so on. Since we assume that each predicate in $\mathcal{P}red$ has two arguments — a $\mathcal{T}erm$ and a value — the combination should be of the same form. In particular, the first argument will be a term, and the second will be a vector of the values of the combined predicates. For example, the values of the combination of $P, Q \in \mathcal{P}red$, V_{P_Q}, will be $\{(v_1, v_2) \mid v_1 \in V_P,\ v_2 \in V_Q\}$. Furthermore, if $P(t, v_1), Q(t, v_2) \in \mathcal{F}acts_i$, then $P\_Q(t, (v_1, v_2))$ will be in $\mathcal{F}acts_{i+1}$. The following lines describe the evolution of $\mathcal{P}red$ through successive steps. Note that in the decision theory approach we assumed that all these combinations appear in the original $\mathcal{P}red$ set.

Step 1: $Pred_1 = \{$domain constant predicates and their negations$\} =$
$\{P_1, \neg P_1, P_2, \neg P_2 \ldots\}$.

Step 2: $Pred_2 = \{$binary combinations of predicates of $Pred_1\} =$
$\{P_1\_P_2, P_1\_P_3, P_2\_P_3, \ldots, \neg P_1\_P_2, \neg P_1\_\neg P_2, \ldots\} \cup Pred_1$.

Step 3: $Pred_3 = \{$ternary combinations of predicates of $Pred_1\} =$
$\{P_1\_P_2\_P_3, P_2\_P_3\_P_4, \ldots, \neg P_1\_\neg P_2\_\neg P_3, \ldots\} \cup Pred_2$.

Of course, this may lead to combinatorial explosion and some heuristics should be used to limit the growth of $Pred_i$. Consider the scenario of Example 9. In step 1, $\mathcal{P}red_1 = \{Type, Color\}$, and $\mathcal{F}acts_1$ consist of the original facts. In step 2, the agent may consider a new predicate which is the combination of Color and Type, i.e., $Type\_Color, Color\_Type \in \mathcal{P}red_2$, and new facts are added to $\mathcal{F}acts_2$, e.g., Color_Type(h1,(White,House)) and Color_Type(b2,(Black,Bridge)).

4.3 Focal Point Rules

In this section we present the actual rules by which an agent identifies candidates for focal points in the step-logic approach. Identification of focal points in the step-logic approach is a two-stage process. First the agent identifies candidates by looking for meta-characteristics of objects, such as uniqueness. Second, the agent resolves competing candidates to the best of his ability (using other rules) and decides on one or more focal points. As in the decision theory case, we will use the notation $\in^*$ and write $P(x, v) \in^* \mathcal{F}acts_i$ to denote that $P(x, v)$

is either explicitly listed in the agent's facts at level i, or that it can be simply computed over the constant terms or values that exist in the facts of level i.

4.3.1 Uniqueness

As was discussed in Section 2.3, an object may be a focal point if it is the only object with a given property. Formally, if in the $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, $v \in \mathcal{V}alue_P$[29], and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$P(x,v) \in^* \mathcal{F}acts_{i-1} \wedge \forall y \in \mathcal{T}erm_{i-1}, y \neq x[P(y,v) \notin^* \mathcal{F}acts_{i-1}],$$

then in step i the agent will believe

$$\text{Unique}(x, P, i).$$

Note that the term x is considered unique with respect to the predicate P; this will be important later, when competing focal points must be resolved.

Example 12 This rule would be applicable in the case in which we know about only one Bridge, namely b1. For example, if
$\mathcal{F}acts_1$= { Type(b1,Bridge), Type(h1,House), Type(h2,House)}, then in step 2 the agent will add Unique($b1, Type, 2$) to $\mathcal{F}acts_2$.

In the logic approach, an object is either unique or not. The agent does not measure an object's "rareness" in a given time step, as in the decision theory approach. However, as we demonstrated above, an object may be unique at time t, but may turn not to be unique at $t+1$. The accumulated information about its uniqueness over time may be used for more "fuzzy" evaluation of its rareness.

4.3.2 Uniqueness Complement

As we discussed in Section 2.3, an object may be a focal point if it is the only object without a given property. Formally, if in the $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$(\forall v \in Value_P, P(x,v), \neg P(x,v) \notin^* \mathcal{F}acts_{i-1}) \wedge (\exists v' \in \mathcal{V}alue_P \forall y \in \mathcal{T}erm, y \neq x[P(y,v') \in^* \mathcal{F}acts_{i-1}]),$$

then in step i the agent will believe

$$\text{Unique-Comp}(x, P, i).$$

[29]For simplicity, we assume that the value set of a predicate does not change over time. However, it will be easy to extend our formalization for handling this option, too.

Example 13 This rule would be applicable in the case in which we know that everybody in the domain is a member of the Democratic Party, except that we have no information one way or the other about John. Although we do not know whether or not John is also a member, this lack of knowledge causes him to be prominent. Formally, if $\mathcal{T}erm(1) = \{John, Dan, Dalia, David\}$,
$\mathcal{F}acts_1 = \{\text{Member}(Dan, D), \text{Member}(Dalia, D), \text{Member}(David, D)\}$, then in step 2 the agent will conclude that $\text{Unique-Comp}(John, \text{Member}, 2)$.

4.3.3 Centrality

Another intuitive criterion is the concept of centrality. Recall, that an object may be a focal point if it is a central object within a given domain. Formally, if in the $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, $v \in Value_P$, and there exists an $x \in \mathcal{T}erm_{i-1}$ such that

$$\begin{aligned}
&P(x,v) \in^* \mathcal{F}acts_{i-1} \wedge \\
&\forall y \in \mathcal{T}erm, y \neq x, \exists v_1 \in \mathcal{V}alue_P \wedge P(y, v_1) \in^* \mathcal{F}acts_{i-1} \rightarrow \\
&\qquad (\exists z \in \mathcal{T}erm, v_2 \in \mathcal{V}alue_P, z \neq y \wedge P(z, v_2) \in^* \mathcal{F}acts_{i-1} \wedge \mathit{Diff}(v_1, v) = \mathit{Diff}(v, v_2)),
\end{aligned}$$

where Diff is a difference function defined on the values in $\mathcal{V}alue_P$ (e.g., "$-$" if the values are numbers), then in step i the agent will have

$$\text{Central}(x, P, i).$$

Example 14 In the range between 0 and 10, the number 5 is Central, where P is the predicate Integer, and Diff is defined as the minus function.

4.3.4 Extreme

An object may be a focal point if its value with respect to some predicate is extreme. Formally, if in the $i-1$th step we have $P \in \mathcal{P}red_{i-1}$, and Q is a total order on $\mathcal{V}alue_P$, and there exists an $x \in \mathcal{T}erm_{i-1}$, $v \in Value_P$ such that

$$\begin{aligned}
&P(x,v) \in^* \mathcal{F}acts_{i-1} \wedge \\
&\forall y \in \mathcal{T}erm_{i-1}, y \neq x \wedge (\exists v' \in Value_P) P(y, v') \in \mathcal{F}acts(i-1) \rightarrow \\
&\qquad\qquad Q(v, v') \in^* \mathcal{F}acts_{i-1} \wedge Q(v', v) \notin^* \mathcal{F}acts_{i-1})),
\end{aligned}$$

then in step i the agent will believe

$$\text{Extreme}(x, P, Q, i).$$

Example 15 In the range between 1 and 10000, the number 1 is Extreme (with the predicate Q being "less than" and P being an integer).

Every object that is unique is also central and extreme, trivially.

4.4 Computing Focal Points—The Resolution Rules

The rules above specify when an object is unique or extreme, etc. They do not relate directly to the question of when the object is actually a focal point. Thus we need a rule to use in tying together these attributes with the notion of a focal point. The most straightforward approach is to relate each of the meta-predicates above to a focal point attribute:

$$\frac{i: \quad \text{Unique}(x, P, i)}{i+1: \quad \text{FocalPoint}(x, i)}$$

$$\frac{i: \quad \text{Unique-Comp}(x, P, i)}{i+1: \quad \text{FocalPoint}(x, i)}$$

$$\frac{i: \quad \text{Central}(x, P, i)}{i+1: \quad \text{FocalPoint}(x, i)}$$

$$\frac{i: \quad \text{Extreme}(x, P, i)}{i+1: \quad \text{FocalPoint}(x, i)}$$

These rules, of course, may not supply us with a unique focal point, since there could be one term that satisfies Unique and another that satisfies Extreme, etc. There could even be two separate terms that are Unique with respect to different predicates. Moreover, two separate terms that are (for example) extreme might receive less attention than would a single term that is central, precisely because the two extremes are competing with one another. There is still utility for the agent in discovering the set of focal points, since even if the choice is made among them probabilistically, there is an increased chance for coordination among the agents.

A more sophisticated rule will associate some measure for the item being a focal point, e.g., by counting the number of times the object appears in the database as Unique, Extreme, etc. Formally, let us denote the set of meta-predicates by *MPred*, i.e.,

$$\mathit{MPred} = \{\text{Unique}, \text{Unique-Comp}, \text{Central}, \text{Extreme}\}$$

.

If in the ith step there is at least one meta-predicated MP $\in \mathit{MPred}$, such that $\text{MP}(x, P, i) \in \mathcal{F}acts_i$, then in step $i+1$ the agent will conclude that

$$\text{FocalPoint}(x, i, d), \ \ where$$

$$d = |\{\text{MP}(x, P', i) \mid P' \in \mathcal{P}red_{i-1}, x \in \mathcal{T}erm_{i-1}, \text{MP}(x, P', i) \in \mathcal{F}acts_i\}|.$$

Example 16 Consider Example 7, where in the third step the agent has the following facts: Type(h1,House), Type(h2,House), Type(h3, house), Type(b1,Bridge),Type(b2,Bridge), *Architect-Of*(h2, Smith), *Architect-Of*(b2, Smith), Color(h1,White), Color(b1,White), Color(h2,Black), Size(b1,Large), and Age(h2,25).

In Step 4, it will conclude, Unique(h2,Age,4), Unique(h2,Color,4), Unique(b1,Size,4), Extreme(b1,Size,4), Extreme(h2,Age,4). In Step 5, it will conclude FocalPoint$(h2, 4, 3)$ and FocalPoint$(b1, 4, 2)$.

Other techniques can be used to identify focal points. In particular, the decision theory approach can be incorporated into the step-logic mechanism. That is, in each step, an algorithm such as the one described in Section 3.4 can be applied to the agent's database during the previous step in order to identify focal points.[30]

Regardless of the method used for identifying focal points, there is no guarantee that a unique one will be found. Furthermore, in different steps, different objects can be considered as focal points. It is critical to resolve which focal points to choose so that the ones that are discovered more easily have higher priority. Step-logic provides us with a natural tool for dealing with this. Using step-logic, there are several mechanisms for relating priority to complexity. We do not attempt here to provide additional rules that guarantee a single focal point. Instead, we illustrate that one could introduce additional rules so as to reduce the size of the focal point set.

As mentioned above, a focal point might be generated (given the above rules) at a given level, and then not be a focal point at a subsequent level.[31] An agent looks for focal points only up to a certain level k. At this level, there might be several competing focal points that are still valid (e.g., arising from different rules or from different predicates). One possible initial winnowing mechanism can be to keep the focal points that were generated earliest and to discard the others. The intuition is that, since the other agent may not go as deeply into the deduction as we have in looking for a focal point, we are more likely to match the other agent by taking the earliest focal point. This will provide the solution most likely to have been reached by the other agent.[32]

Example 17 In the range between 1 and 10000, the number 1 is Extreme (with the predicate P being "less than"), and 10000 is Extreme (with the predicate P being "greater than"), after the first step. If the domain of considered integers grows at each step, 1 will still be extreme, while 10000 will no longer be extreme. Thus, at the end of the process, 1 will be

[30]The measure of difficulty of a given fact can be chosen as the step in which the fact was concluded.

[31]Note that in step-logic, beliefs at step i are not automatically beliefs at step $i + 1$.

[32]Other approaches present themselves, such as considering the *coverage* of a focal point; e.g., if a term is a focal point for much of the deduction, though it is not in the final step, we would still consider it a likely solution. We could also then probabilistically weight the steps of the deduction, so that, for example, earlier steps receive more weight than later steps. These methods are left for future work.

chosen, since it has been "extreme" for the longest period. This disambiguates between the two extreme ends of a finite domain that grows in only one direction.

The procedure only considers "term-property" pairs; if a term was a focal point because of some property at level i, then it was no longer a focal point because of that property at level i+1. It then again became a focal point because of a different property at level i+2 (and remains a focal point until the end), and is then considered to have been generated at level i+2.[33] We may also choose to introduce rules that assign a priority to the meta-predicates (such as Unique), so that, for example, a unique object gets priority as a focal point over an extreme object.

We emphasize that the step-logic framework does not determine which inference rules we choose for focal point identification and the resolution between competing focal points. However, it allows us to develop general rules which can take into consideration the time of the reasoning process and the time frame in which an object is considered as a focal point.

4.5 Conditions for Joint Selection

The conditions for joint selection of the same object by agents using the logic approach is similar to the case in which the agents use the decision theory approach. However, there are additional aspects that influence the joint selection in this case.

Consider the case in which agents have identical knowledge about everything, including the original axioms and inference rules, run-time observations, the domain of predicates, terms, and functions, and, in addition, the agents' computational "power" (i.e., how deep the search for focal points will go) is the same. It is clear that, in this case, where search depth is identical, if there is a set of focal points, the set will be generated identically by both; i.e., the procedure guarantees joint selection.

When the search depth is not identical, but is known to both agents, they need only consider the derivation to the depth that can be reached by the weaker of the two agents. In this case, the focal point set will also be generated identically by both agents.

When the agents have identical knowledge about everything other than the power of computation (i.e., how deep the focal point search will go), there is another consideration that affects whether or not the agents will select the same object: the monotonicity of the focal point derivation.

[33]The idea behind looking at term-property pairs to establish the first appearance of a focal point is that once a focal point has disappeared because of other terms with the same property, its prominence because of that original property is completely negated.

Definition 1 *A focal point derivation is said to be* monotonic *if a focal point generated at step i continues to be re-derived at all steps greater than i.*

The monotonicity of the focal point meta-predicates (such as unique, extreme, central) is, in general, not related to whether or not the base level predicates are monotonic. For example, one might have a monotonic domain (such as theorem proving) where a focal point a is found related to some property P with the value v (by uniqueness). Later we may derive $P(b, v)$, causing $P(a, v)$ to be no longer unique.[34]

When agents are involved in a monotonic focal point derivation, and when they have reached a level at which their focal point candidate set is non-empty, they can cease computation of focal points. How can agents (or their designers) know whether or not they are participating in a monotonic focal point derivation? The following is one example of such a monotonic focal point derivation.

Assume that we are dealing with Horn clause databases, and the derivations are monotonic (e.g., no observations). If a focal point a is found to be related to some property, P, then if P does not appear as a consequence of *any* rule, a is a monotonic focal point. These kinds of focal points are based only on facts that were known at the beginning of the derivation. This is true even when their focal point status is only discovered by using conjunctions of properties.

Example 18 A database contains $P(a, v)$, $P(b, v)$, $Q(a, v')$, $Q(c, v')$, plus some rules that do not have P or Q as consequences. In the first step, there is no "unique" focal point. In the second step, we add the combination P_Q to the *Pred* set; in the third step we discover the "unique" property of a relative to P_Q; and in the fourth step we conclude that a is a focal point. In the next step we see that neither P nor Q appears as a consequent of any rule in the database, so there is no chance of a losing its status as a focal point.

Definition 2 *a is a* literal *focal point with respect to predicate P, when it was derived solely from the set of literals in the database, not from rules. If, in addition, the predicate P does not appear as a consequence of any rule in the database, we call the focal point* strictly literal.

If P appears as a consequence of any rule in the database, there may be some (lengthy) derivation of k steps that will exclude a as a focal point. For example, if a was chosen as a focal point since it was unique with respect to P with the value v, after k steps, $P(b, v)$ can be deduced using the rule in the database where P appears as a consequence, and thus a

[34]In general, any combination of monotonicity and nonmonotonicity is possible between the base level and the meta-level.

will no longer be unique. There is no way of discovering whether a will remain a focal point except by actually carrying out the derivations over k time steps. Note, that in the case of Datalog databases (i.e., Horn-clause rules with no function symbols), the depth, in general, of this derivation is polynomial. It is also clear that almost any focal point can be excluded at step i by an observation at step $i-1$, most often by observing another term or predicate in the domain, or by observing a new property of a term in the domain. The above property of focal point monotonicity can only hold, in general, when observations are excluded from consideration.

If the set of focal points is non-empty, and the search goes on long enough and is monotonic, agents will reach agreement using the above procedure (this is true regardless of the finite axiomatization of their theory). However, it is difficult to characterize, for an arbitrary database, just how long the agents need to search (the special case of Datalog was mentioned above).

We now propose a minor extension to the step-logic focal point procedure presented above. The extension only affects when an agent can safely conclude his search.

Extension to the focal point procedure:

Case 1: Strictly literal focal point a is discovered at step i, and no other focal point has been discovered at step $h \leq i$. The search can stop and use a as the focal point. If there are no observations, and the agents have identical knowledge, then the joint selection of this single focal point is guaranteed.

Case 2: Strictly literal focal point a is discovered at step i, no other focal point has been discovered at step $h < i$, and a non-literal focal point has also been derived at i. The search should then stop and use a as the focal point (as above). In this case, joint selection is only guaranteed if the other agent happens to be following this same heuristic; otherwise, the other agent might actually choose the other non-strictly literal focal point. Nevertheless, this heuristic is reasonable, as the non-strictly literal focal point may later vanish, while the strictly literal focal point will not.

Case 3: In other cases, follow the original procedure.

To summarize, as in the decision theory approach, when the agents truly have common knowledge about their databases, then any arbitrary algorithm will satisfy their need to coordinate, including the algorithm we have presented above. The focal point algorithms are really designed for cases in which the agents do not have common knowledge, and where coordination is not guaranteed, but where the design of the algorithms is such as to raise the probability of coordination. If designers of automated agents could coordinate search strategies among those agents ahead of time, then (as long as the strategy is general enough), those designers could settle on an arbitrary joint strategy (alphabetic, for example). One motivation for using a focal point search strategy is its relationship to the way people work and its generality. Barring specific information about the other agent, these algorithms are reasonable candidates for communication-limited coordination.

4.6 The Application of the Step-Logic Focal Point Method to Robot Rendezvous

In the beginning of Section 4, we discussed three mechanisms specifying how the agent's beliefs may change over time and affect the objects it may consider as focal points. We demonstrated how these mechanisms can be modeled in step-logic. In this section, we present an algorithm which is based only on the third mechanism: an increase in the set of predicates over time. We apply this algorithm to the robot rendezvous problem described in Section 3.4.

We concentrate only on the mechanism related to the increase in the set of predicates over time since "observation" is too domain-dependent, and we try to develop a general algorithm. Deduction of new facts from old ones is difficult to implement, and a lot of open questions which are related to it still exist. It also seem to us that mechanisms that are related to "deduction" are mainly beneficial for very large databases. We leave the study of these issues for future work.

The situation description of the robot rendezvous problem and the assumptions which we make are exactly as described in Section 3.4.1. We first describe the data-structures that are created when applying the step-logic focal point algorithm to such situations. Given a set of predicates, there are two ways in which the agent can create new predicates: by adding Unique or Extreme in front of the given predicates[35], or by combining several predicates of the given set.[36] This is done repeatedly in several iterations. To illustrate the sets which are created, we use a tree, as in the example given in Figure 18. The algorithm may consider these sets using different orders.[37]

A node in the tree which is used to illustrate the above sets is labeled with a set of predicates that may be created while searching for a focal point. Each node has two children, reflecting the two ways new predicate sets are created: the label of the left child is created

[35] The new predicates which are created by adding unique or extreme in front of old ones have two arguments: an object and a truth value, where its possible value is either 1, indicating true, or 0, indicating false.

[36] A new predicate which is a combination of l old predicates has two arguments. The first argument is an object, and the second argument is an l-tuple of the original values, and they are ordered using a lexicographic order.

[37] The specific algorithm described below uses a depth-first approach.

by adding Unique or Extreme in front of the predicates in the node; the label of the right child is created by combining l predicates from the set of predicates of the node, where l is the height of the right child.

In the process of choosing a focal point, a set of relevant facts is associated with each node. The agent considers only the set of facts which are associated with left nodes when trying to identify a focal point. After each time that such a set is created for any $i \in \mathcal{T}erm$, a measure $F(i)$ is calculated. This calculation is done similar to the way in which d is computed in Section 4.4. After F(i) is calculated, the step-logic algorithm proceeds as in the decision theory algorithm. This process is demonstrated in the following example.

Example 19 Consider the case in which the original set of predicates includes three predicates: P, Q, and R. Thus, the root of the tree in Figure 18, which illustrates the sets of predicates that are created in this case, is labeled by the original predicate set, i.e., P, Q and R. The left child of the root is labeled by UP, UQ, UR, EP, EQ, and ER, with U standing for Unique and E for Extreme (e.g., Unique_P is denoted by UP.) The right child of the root is labeled by PQ, PR, and QR, which are the possible combinations of length two of the original sets of predicates specified in the root. We refer to the predicates which appear in left nodes (the oval nodes) as the "left" predicates, and the predicates that appear in right nodes (the rectangle nodes) as the "right" predicates.

As mentioned above, all the predicates have two arguments, as in the original set. The first argument is an object. The second argument of a "left" predicate is a truth value, whose possible value is either 1, indicating true, or 0, indicating false. The second argument of a "right" predicate, which is a combination of l predicates (of the father's node), is an l-tuple of the original values, and they are ordered using a lexicographic order. Since the values of predicates starting with U or E are always extreme, and thus do not influence the choice of a focal point, we have omitted from the example predicates of the form EUQ or EEQ, etc. This omission does not influence the focal points that are found.

After a set of predicates is created, a new set of facts with respect to the new predicates is also created. In our example, suppose that there are three houses in the environment, with the properties P, specifying its price, Q, specifying the size of the house, and R, specifying the street where the house is located. In Figure 19, the facts that may be created when trying to identify a focal point are specified. The original set of facts associated with the root of the tree is specified in the first table in Figure 19, e.g., the price (P) of house h1 is 150K. The facts associated with the left child of the root (Table 2 in Figure 19) indicates the uniqueness and extremeness of the houses with respect to the original set of facts; e.g.,

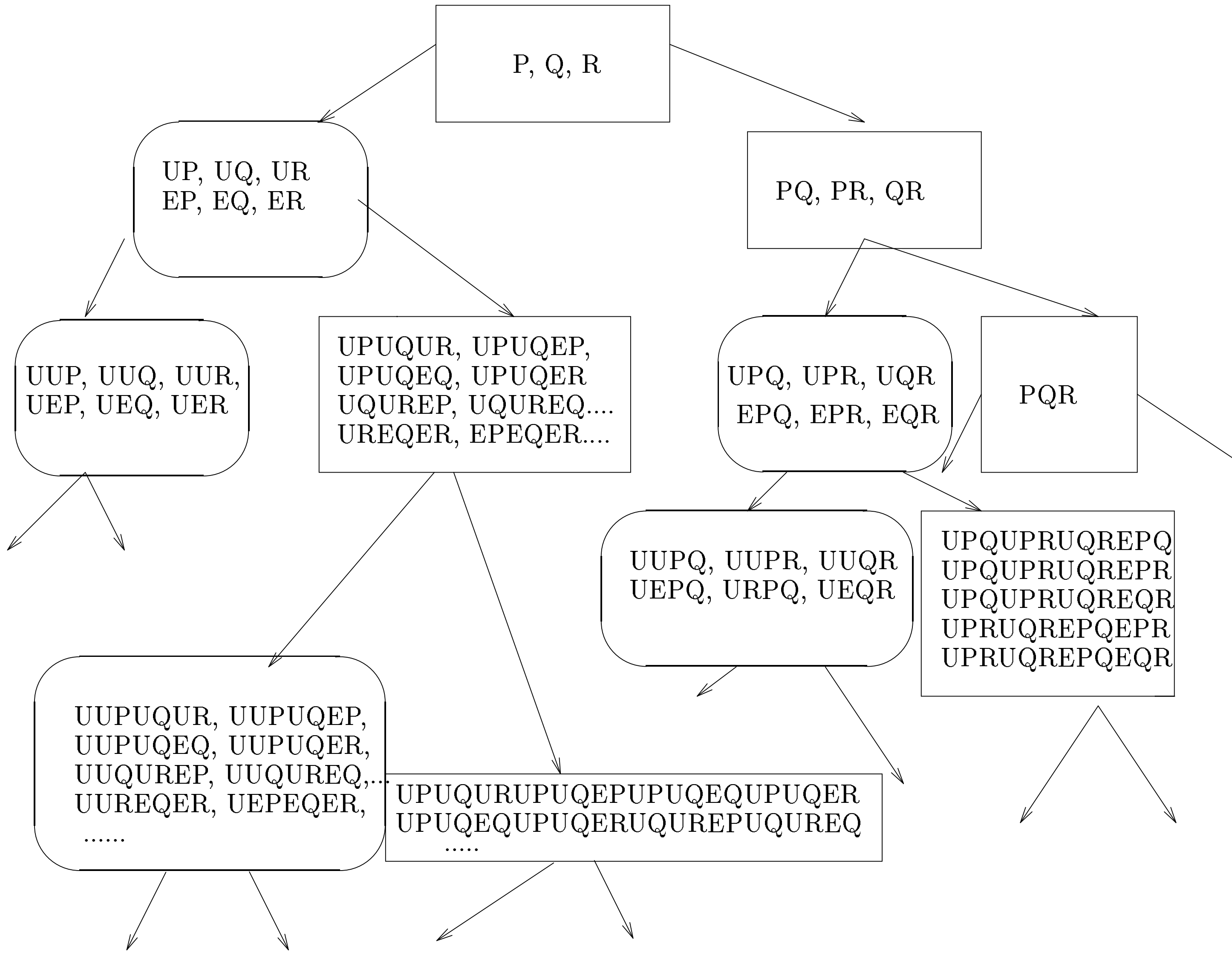

Figure 18: An example of the creation of new predicates in Algorithm 3. P and Q are predicates. "U" is added to indicate Unique, and "E" to indicate Extreme. The arguments of the predicates in the oval nodes are an object and a truth value. The arguments of the rectangle nodes are an object and a tuple.

Table 1: ROOT			
Obj	P	Q	R
h1	150K	Small	5
h2	150K	Big	3
h3	100K	Small	9
h4	150K	Med	5

Table 2: LEFT-ROOT							
Obj	UP	UQ	UR	EP	EQ	ER	F
h1	0	0	0	1	1	0	2
h2	0	1	1	1	1	1	5
h3	1	0	1	1	1	1	5
h4	0	1	0	1	0	0	2

Table 3: RIGHT-ROOT			
Obj	PQ	PR	QR
h1	(150K,Small)	(150K,5)	(Small,5)
h2	(150K,Big)	(150K,3)	(Big,3)
h3	(100K,Small)	(100K,9)	(Small,9)
h4	(150K,Med)	(150K,5)	(Med,5)

Table 4: LEFT-LEFT-ROOT							
Obj	UUP	UUQ	UUR	UEP	UEQ	UER	F
h1	0	0	0	0	0	0	0
h2	0	0	0	0	0	0	0
h3	1	0	0	0	0	0	1
h4	0	0	0	0	1	0	1

Table 5: LEFT-RIGHT-ROOT							
Obj	UPQ	UPR	UQR	EPQ	EPR	EQR	F
h1	1	0	1	0	1	1	4
h2	1	1	1	1	0	1	5
h3	1	1	1	1	1	0	5
h4	1	0	1	0	1	0	3

Table 6: RIGHT-LEFT-ROOT												
Obj	UPUQUR	UPUQEP	UPUQEQ	UPUQER	UPUREP	UPUREQ	UPURER	UPEPEQ	UPEPER	UPEQER	UQUREP	UQUREQ
h1	(0,0,0)	(0,0,1)	(0,0,1)	(0,0,0)	(0,0,1)	(0,0,1)	(0,0,0)	(0,1,1)	(0,1,0)	(0,1,0)	(0,0,1)	(0,0,1)
h2	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(0,1,1)	(1,1,1)	(1,1,1)
h3	(1,0,1)	(1,0,1)	(1,0,1)	(1,0,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(0,1,1)	(0,1,1)
h4	(0,1,0)	(0,1,1)	(0,1,0)	(0,1,0)	(0,0,1)	(0,0,0)	(0,0,0)	(0,1,0)	(0,1,0)	(0,0,0)	(1,0,1)	(1,0,0)

Table 6 (Continue): RIGHT-LEFT-ROOT							
Obj	UQURER	UQEPEQ	UQEQER	UREPEQ	UREPER	UREQER	EPEQER
h1	(0,0,0)	(0,1,1)	(0,1,0)	(0,1,1)	(0,1,0)	(0,1,0)	(1,1,0)
h2	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
h3	(0,1,1)	(0,1,1)	(0,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
h4	(1,0,0)	(1,1,0)	(1,0,0)	(0,1,0)	(0,1,0)	(0.0.0)	(1,0,0)

Table 7:RIGHT-RIGHT-ROOT	
Obj	PQR
h1	(150K,Small,5)
h2	(150K,Big,3)
h3	(100K,Small,9)
h4	(150K,Med,5)

Table 8: LEFT-RIGHT-RIGHT-ROOT			
Obj	UPQR	EPQR	F
h1	1	0	1
h2	1	1	2
h3	1	1	2
h4	1	0	1

Figure 19: Some of the sets which may be created with respect to the tree in the previous figure. “Root” refers to the root node; “Left-Root” indicates the left child of the root, “Left-Left-Root” indicates the left child of the left child of the root, etc.

h1's value with respect to predicate P is not unique; thus there is 0 in the box of the second column and the second row. The last column of the table specifies the focal point value (F value) that is computed for each house according to the algorithm presented in the next section. For example, F(h1)=2 since h1 has an extreme value with respect to predicates P and Q. F(h2)=5 since h2 has a unique value with respect to Q and R, and an extreme value with respect to all the predicates. In this step of the example, there is no house that could be chosen as a focal point, since there is no house with a unique focal-point. However, h1 could be chosen as a focal point when the set of the left child of the right child of the root is created (Table 5 of Figure 19) since it has the highest unique F value.

4.6.1 Step-Logic (SL) Focal Point Algorithms

In order to simplify the description of the algorithm, we present it as a recursive function rather than presenting the appropriate inference rules of step-logic. We first specify two functions that we use in the algorithm. The first function, Focal, creates new predicate lists and new fact lists and uses the second function, FP, to check if there is a unique focal point in the new sets.

The function Focal receives a set of predicates and an associated set of facts. The first time Focal is called the predicate set is equal to the original set of predicates (e.g., P, Q) and the associated original set of facts (e.g. P(h1,150K), Q(h2,Big)). In the other cases, it consists of predicates which are either combinations of the original set of predicates (e.g., P_Q), or which are obtained by applying Unique or Extreme to some predicates (e.g., Unique_P, Unique_Extreme_Q), or the combination of such predicates (e.g., Unique_P_Unique_Q_Unique_R). Given these sets, it creates a new set of predicates (in step (2a)) by adding Unique and Extreme and an associated set of facts. Then it calls the function FP to check whether the new set consists of a focal point. If it does, then it returns this object. Otherwise, it tries to find a focal point, either by calling itself recursively with the new set or by creating a different set of predicates and facts by combining the predicates of its input, and then calling itself recursively with the "combination" of new sets.

Function Focal

Input:

lev: the current level of the search.

$\mathcal{P}red$: the current set of predicates.

$\mathcal{F}acts$: the current set of facts.

Output: If there is an object which is a focal point, it is returned; otherwise False.

Temporal variables:

TMPPRED: stores the new predicates created in the current iteration (by adding Unique or Extreme).

TMPFACTS: stores the new facts which refer to the new predicates of TMPPRED.

maxlev: is a global variable which specifies the maximal search level.

1. If $lev \geq maxlev$, then return False.

2. Else,

   (a) Create new predicates by adding Unique and Extreme in front of the predicates of Pred ("left" predicates):
   TMPPRED= $\{\text{Unique\_}P, \text{Extreme\_}P \mid P \in \mathcal{P}red\}$.

   (b) Create a new set of facts using TMPPRED and the old facts $\mathcal{F}acts$:
   For any $P \in \mathcal{P}red$,

      i. Compute the truth value of the new "unique" predicates:
      if $P(x,v) \in \mathcal{F}acts$ and $\forall y \in \mathcal{T}erm, y \neq x[P(y,v) \notin \mathcal{F}acts]$, then add Unique_P$(x,1)$ to TMPFACTS.
      Else, add Unique_P$(x,0)$ to TMPFACTS.

      ii. Compute the truth value of the new "extreme" predicates:
      Suppose $\leq$ is the total order on $\mathcal{V}alue_P$. If $P(x,v) \in \mathcal{F}acts$ and $\forall y \in \mathcal{T}erm, y \neq x$, such that $P(y,v') \in \mathcal{F}acts$ $v' \leq v$ or $\forall y \in \mathcal{T}erm, y \neq x$, such that $P(y,v') \in \mathcal{F}acts$ $v \leq v'$, then add Extreme_P$(x,1)$ to TMPFACTS
      Else, add Extreme_P$(x,0)$ to TMPFACTS.

   (c) Search for a focal point in the new set of facts:
   If (res=FP(TMPFACTS)), then return res.

   (d) If failed, then try again by repeating the same process on the new set of predicates and facts:
   Else, if (res=FOCAL(lev+1,TMPPRED,TMPFACTS)), then return res.

   (e) Else,

      i. If failed, create new predicates which are tuples of length lev of predicates from $\mathcal{P}red$ ("right" predicates):
      $TMPPRED = \{P_0\_P_1\_...\_P_{lev} \mid P_i \in \mathcal{P}red\}$.

      ii. Create a new set of facts using TMPPRED and the old facts $\mathcal{F}acts$:
      For any $P_0, ..., P_{lev} \in \mathcal{P}red$,

if $P_0(x,v_0), P_1(x,v_1), ..., P_{lev}(x,v_{lev}) \in \mathcal{F}acts$, add $P_0\_P_1\_..\_P_{lev}(x,(v_1,...,v_{lev}))$ to $TMPFACTS$.

iii. Repeat the process using the new sets of predicates and facts: return(FOCAL(lev+1,TMPPRED,TMPFACTS)).

The next function receives a set of facts of the form Unique_P$(i,1)$, indicating that i is unique with respect to P, Unique_P$(j,0)$, specifying that j is not unique with respect to P, Extreme_P$(i,1)$, specifying that i is extreme with respect to P and Extreme_P$(j,0)$, indicating that j is not extreme with respect to P. It calculates the focal point value (F) of each object by counting the number of unique and extreme facts associated with it. Then it looks for all the objects with a unique focal point value. If there are such objects, it chooses from among them the one with the largest focal point value and returns it as a focal point. Otherwise, (i.e., if no unique focal point is found) it stores the set with the smallest number of objects with the same focal point value and returns False for future consideration.

Function FP

Input: $\mathcal{F}acts$

Output: If there is a focal point, based on $\mathcal{F}acts$, then it is returned; else, the set $\hat{S}$ is updated and False is returned.

Temporal variables:

UFP: the set of objects with a unique focal point value.

SFPT: the set of sets of objects; each set of objects consists of objects with the same focal point number; the size of the sets is the same and minimal with respect to other such sets.

1. Calculate the focal point value for all objects $i \in \mathcal{T}erm$, using the following equation: $F(i) = |\{P(i,1) \mid P(i,1) \in \mathcal{F}acts\}|$.

2. Choose the object c with the largest F value that is unique in having that value. Formally, let $UFP = \{i | i \in \mathcal{T}erm; \forall i' \in \mathcal{T}erm,\ if\ i' \neq i,\ then\ F(i) \neq F(i')\}$.
If $UFP \neq \emptyset$, then $c = argmax_{i \in UFP} F(i)$; return(c).

3. Else, divide the objects in $\mathcal{T}erm$ to sets, S_1,...,S_k, where $1 \leq k \leq |\mathcal{T}erm|$, such that the objects of each S_j have the same focal point value, i.e., for all $1 \leq j \leq k$ and for all $i, i' \in S_j$, $F(i) = F(i')$. Denote by f_j the focal point value of the elements in S_j.

4. Let SFPT $= \{S_j \mid 1 \leq j \leq k,$ and $\forall 1 \leq l \leq k, |S_j| \leq |S_l|\}$. That is, SFPT includes all the sets whose number of elements are minimal. Add $argmax_{S_j \in SFPT} f_j$ to $\bar{S}$; return(False).

The following algorithm uses the function Focal to identify focal points. It fixes the maximal level of the search (step (1)) and then calls Focal with the original set of predicates and facts (step (2)). If Focal succeeds, then c includes the chosen focal point, and the agents will definitely choose the same object; i.e., the algorithm succeeds. If Focal fails (step (3)), then, like Algorithm 2 of the decision theory approach, the algorithm will try to increase the probability of choosing the same object by randomly choosing among the objects of the smallest set with objects with the same focal value.

Algorithm 3 Joint selection of an object using SL focal points

1. *$\bar{S} = \emptyset$; determine maxlev.*

2. *Try to find a definite focal point using the recursive function, Focal. Formally, $c = Focal(1, \mathcal{P}red, \mathcal{F}acts)$.*

3. *If $c = False$, let SFP $= \{S \mid S \in \bar{S}$ and $\forall S' \in \bar{S}, |S| \leq |S'|\}$. That is, SFP includes all the sets whose number of elements are minimal. Let S^* be the set in SFP which was added first to $\bar{S}$. Choose randomly one of the objects of S^* as the chosen object c.*

4.6.2 Properties of the Algorithm

The logical-focal point algorithm described above has the following properties:

Success Rate: The high success rate of the algorithm is demonstrated in the "Results" section below.

Front End: As in the DT-focal point algorithm, if the SL-focal point algorithm succeeds (i.e., Focal at step 2 returns a value), the agents will definitely choose the same object. In the rare cases in which the SL-focal point algorithm fails to find a unique object, both agents know that it has failed (also common knowledge), and so the results collected in the iterations of the function Focal and saved in $\bar{S}$ are used in the third step of the algorithm to increase the probability of "meeting." In the algorithm above, we choose the set from which an object is chosen randomly by using the criterion: the smallest set that was found first. However, there are other criteria that can be used, such as the set that was most persistent.

No. of Objects	No. of Predicates			
	2	3	4	5
3	100	100	100	100
4	92	99	99	100
5	92	100	100	100
6	89	98	100	100
7	86	98	99	100
8	87	97	100	100
9	88	97	99	100
10	83	97	99	100

Figure 20: Probability of definitely choosing the same object in case of a different number of values, even distribution, using the step-logic, focal point algorithm.

Any Time Algorithm: While the DT-focal algorithm provides an answer only when it finishes, the step-logic focal point algorithm may be terminated after each iteration, and the sets that are accumulated in $\bar{S}$ may be used to increase the option of choosing the same object, i.e., choosing an object randomly out of one of the sets of $\bar{S}$.

Domain-Independence: The algorithm is applicable in any domain in which there are objects, predicates, and the need to choose one of the objects, as in the DT-focal point algorithm.

Independence of Agents' Internal Representations: All agents must have sets of objects, predicates, and values for the predicates. However, the agents may have different names for objects, predicates, and values, as in the DT-focal point algorithm.

4.6.3 Results and explanations

We conducted simulations, as in the decision theory case. However, while the decision theory algorithm implements most of the aspects of our ideas, the focal point algorithm is partial and implements only the mechanism for creating new predicates. Therefore, we conducted only a limited number of simulations. We concentrated on Case A of Section 3.4.5. The number of possible values was 10, and even distribution was used. We considered configurations in which the number of objects was between 3 and 10, and the number of predicates was between 2 and 5. For any given configuration (i.e., number of objects, number of predicates), we ran the three steps of the simulations 1000 times.

Figure 20 specifies the probability of definitely choosing the same object, i.e., the probability that the function Focal in the second step of Algorithm 3 will succeed. It is easy to

No. of Objects	No. of Predicates			
	2	3	4	5
3	100	100	100	100
4	96	99	99	100
5	96	100	100	100
6	94	99	100	100
7	93	99	100	100
8	94	98	100	100
9	94	98	100	100
10	91	98	100	100

Figure 21: Probability of choosing the same object in case of a different number of values, even distribution, using the step-logic, focal point algorithm.

see that our algorithm did very well. In all the configurations, the probability of definitely choosing the same object was at least 83%. In configurations with more than 2 predicates, the probability was at least 97%, and in configurations with 5 predicates, regardless of the number of objects, there was a probability of 100%.

Comparing these results with the ones obtained in similar configurations for the DT-focal point algorithm (Figure 5), we observe that the DT algorithm performed a little better when there were only two predicates, but the SL algorithm performed a little better when there were 5 predicates. However, the differences are very small, and it is difficult to speculate about the reasons for this.

Figure 21 presents the general probability of choosing, i.e., the success rate for all steps of Algorithm 3. This increases the chances to at least 91% for any configuration that we considered. These results can be compared with the ones obtained for similar configurations by the DT algorithm (Figure 16.) The SL algorithm did a little worse when the number of predicates was 2 or 3, but both algorithms provide a high probability for choosing the same object. Since, in this case, as well, the differences are small, it is difficult to explain them. However, the SL algorithm may be a little weaker, since it uses less available information than does the DT algorithm. For example, the SL algorithm only uses the fact whether an object has a unique value with respect to a given predicate or not, but the DT algorithm, in the case in which an object is not unique, uses the information about how many objects have the same value with respect to that predicate.

Further work is needed to incooperate the two other mechanisms described at the beginning of the section, i.e., the inference mechanism and the observation mechanism, into the SL algorithm. For example, the agents can make observations and some deductions after each level of the algorithm. The exact details of such implementation are left for future work. However, since the simple SL algorithm has already performed so well, we also expect the extensions to do well.

5 Decision Theory/Step-Logic/People: Comparisons

Focal points provide a test case for representation and reasoning in multi-agent domains. By studying focal points, we gain insights into representing and reasoning about multi-agent encounters. We have presented two alternative formalisms for enabling an automated agent to discover focal points, one based on step-logic and the other on decision theory. Each has certain advantages and disadvantages.

As Doyle points out [17, 16], logic and decision theory "are not competing theories, but instead are two complementary parts of the solution." It thus makes sense to consider how each might be used to treat a difficult new problem in knowledge representation, to consider their strengths and weaknesses, and ultimately, perhaps, to combine them into a unified solution.

Decision theory allows for a natural integration of payoffs into the decision-making process. In the step-logic-focal point algorithm, in contrast, payoffs were considered only indirectly, when they affected uniqueness, etc. Another difference between the two approaches is that with decision theory, the changes brought about by searching "one level deeper" will be continuous and can even be analyzed ahead of time (i.e., the agent can ask the question, "What is the maximum utility that I can derive by searching one level deeper?"). None of this is true of the step-logic approach: searching one level deeper can lead to a non-continuous change in the choice of focal point.

Another difference between the two approaches is that, with decision theory, the measure of difficulty is intended to take into account, among other factors, multiple derivations—if a fact is derived from multiple sources, its measure of difficulty is lower. In the step-logic approach, it doesn't matter if a fact was derived at level n in several ways from the $n-1$ level.

At times, of course, the deductive approach of step-logic will succeed, while the decision theoretic approach will fail to cause coordination. If the probability in the example of the beginning of Section 3 was raised to only .45, then agent K might not choose to go to C, because his payoff values steer him away from that solution—even though it is a unique focal point. The step logic approach would cause the agents to coordinate.

At other times, the step-logic approach will fail, while the decision theory approach will succeed. For example, there can be interactions in which there are multiple equilibrium points and multiple focal points, as well. Even when the utility to the agents is greater from one choice, step-logic would not help the agents coordinate. The decision theoretic approach, on the other hand, causes them both to choose the high utility option.[38] Another problem with decision theory is that the agent designer must somehow supply the necessary numeric information, which, in general, is difficult to generate. In certain ways, the step-logic approach requires less information from the agent designer and is thus suitable even when utility information is difficult to come by.

In order to compare the decision theory and step-logic approaches, we conducted a preliminary experiment which also involved people. Based on this preliminary experiment, our hypothesis is that the current version of the DT algorithm, when acting in a static environment such as the one described in Figure 3, is able to match the actual choices of people. The SL algorithm did worse than did the average person in this experiment. Further work is needed to check our hypothesis.

In order to compare the decision theory and step-logic approaches, we conducted an experiment which also involved people. We constructed a questionnaire consisting of twelve questions. In each question there was a description of a number of objects (between 4 and 12) and their properties. The situation described in Figure 3, which was used to demonstrate the DT algorithm, and the situation of Example 9, which was used to demonstrate the SL algorithm, were included in the questionnaire.

We asked 20 senior computer science students to choose one object for each question and to try to match their friends' choices. That is, we explained to them that a correct answer would be one in which the chosen object would be the one which was selected by most of the other students. We compared the students' choices with the objects chosen by the step logic-focal point algorithm and the decision theory-focal point algorithm. A summary of the results is presented in Figure 22. In all cases, the DT algorithm and the SL algorithm were able to find a focal point.

In 10 of the questions, at least half of the students chose the same object (i.e., all questions but 6 and 11). In 9 of them, the DT algorithm also chose the same object which they selected. The SL algorithm was able to choose only 5 of these objects. The average student score was 6. Only in 4 questions did the DT and the SL algorithms choose the same object.

If we consider as a "correct answer" for any question the object which was chosen by the largest number of students, even if fewer than half of the students selected it, then the DT algorithm was able to choose 9 objects correctly; i.e., 2 of its misses were in cases in which fewer than half of the students chose the same object. The SL algorithm scored 6; the

[38]Game theory might resolve this interaction by noting that one equilibrium point yields maximal utility for all players. This solution is sometimes mandated axiomatically [65].

Ques. No.	Object No. 1	2	3	4	5	Algo. DT	SL	additional objects
1	3	**10**	3	2	2	2	2	
2		1	**13**	5	1	3	4	
3	2	**10**		3	5	2	5	
4	**12**	3		2	3	1	4	
5	**13**	3	1	2	1	1	1	
6	2	6	**9**	3		2	3	
7	1	4	**15**			3	3	
8	3	**11**	1		5	2	5	
9	**17**		1	1	1	1	4	
10	1	3	**16**			2	3	
11	3	1	1	1	1	8	8	6-**8**; 8-4; 10-1
12	1	1		1		7	7	6-3; 7-**12**; 10-2

Figure 22: Results of experiments with people. Each row corresponds to one of the questions. Columns 2-6 correspond to the number of students who chose the specified object. Columns 7 and 8 specify the objects that were chosen by the DT and SL algorithms, respectively. The situations in questions 11 and 12 consisted of ten objects, and the number of students who chose objects 6–10 are specified in the last column. For example, for question 11, 8 students chose object 6, 4 students chose object 8, and one student chose object 10.

average student score was 7.05.

This is a preliminary experiment, and more experiments are needed to reach significant results. However, our hypothesis is that the current version of the DT algorithm, when acting in a static environment such as the one described in Figure 3, is able to match the actual choices of people. The SL algorithm did worse than did the average student in this experiment. From an observation of the computations of the SL algorithm, we realized that it wasn't "sensitive" enough. For example, in some questions it gave the "correct" object the highest focal point value, but gave the same value to an additional object or objects. Since it selects an object with a unique focal point value, it chose a third object that received the second highest focal point value. Further work is needed to improve the SL algorithm and, in particular, to develop an inference system that will use the special properties of step-logic. We suspect that such an extended algorithm will do better than the DT algorithm does in a dynamic environment in which not all information is given to agents in advance.

In addition, we made several interesting observations concerning the behavior of people. In most of the questions in the questionnaire, the properties of the objects were meaningful, i.e., type, size, color, etc. Also, their values were meaningful, e.g., house, big, white (as in Figure 3). However, in three questions we specified the properties of the objects with abstract names, e.g., P1, P2, and the values were also abstract, e.g., lines, shapes, letters, and numbers. In particular, question 3 considers the same situation as does question 8, but, while in question 3 we used meaningful properties and values to describe the objects, in question 8 we used an abstract specification. Similarly, questions 4 and 9 refer to the same situation as do questions 6 and 10. In all three cases, the same object was chosen by most of the students, regardless of the representation. For example, in question 4, 12 students chose object 1, and in question 9, 17 students chose object 1. However, in all three cases, more students were able to choose "correctly" when the descriptions were abstract, than in the case of meaningful representations. Of course, both the SL and DT algorithms chose the same object in the similar questions, regardless of the representation. From interviewing the students, we realized that in the case of meaningful presentations, the specific value of the property of the object influenced their decisions. For example, they were leaning toward choosing a yellow house rather than a white house, since yellow is a "stronger" color than white. Further experiments are needed to study such influences and the way to incooperate them into automated agents which cooperate with humans.

6 Summary

We have presented the concept of focal point solutions to interaction problems and discussed why conventional representation techniques are insufficient for focal point discovery. An algorithm was presented for discovering focal points, using a decision theoretic framework; this algorithm provides a natural way to incorporate theories of utility into the focal point calculation. A second algorithm was developed that allows for the uncovering of focal points through the use of step-logic, special inference rules, and sets of predicates, functions, and terms that change over time. The technique is particularly well-suited for modeling the time-dependent nature of a focal point search.

A series of simulations were run over various randomly generated worlds that demonstrated the usefulness of the focal point algorithms as a heuristic for multi-agent coordination. We also conducted an experiment which compared the behavior of people and our two approaches. The results indicate that, in case of a static database, the decision theory algorithm matched human behavior better. The question of how people and the algorithms will behave in a dynamic environment is left for future work.

Acknowledgments

A preliminary description of the step-logic approach appears in [49]. A preliminary version of the decision theory algorithm is described in [28]. This research has been partially supported by the National Science Foundation under Grant No. IIS-9820657 and by the Israeli Ministry of Science and Technology under Grant 032-8284. The authors wish to thank Dr. S. Hochberg for editorial assistance.

References

[1] R. C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotics Systems*, 9(3):351–364, 1992.

[2] L. Aspin. Focal points, preferences and the generation of meaning. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[3] M. Bacharach. Variable universe games. In *Frontiers of Game Theory*, pages 255–275. The MIT Press, Cambridge, Massachusetts, 1993.

[4] M. Bacharach and M. Bernasconi. The variable frame theory of focal points: an experimental study. *Games and Economic Behavior*, 19:1–45, 1997.

[5] M. Benda, V. Jagannathan, and R. Dodhiawalla. On optimal cooperation of knowledge sources. Technical Report BCS-G2010-28, Boeing AI Center, Boeing Computer Services, Bellevue, Washington, August 1985.

[6] V. Bhaska. Breaking the symmetry: Optimal conventions in repeated symmetric games. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[7] S. Carberry. Incorporating default inferences into plan recognition. In *Proceedings of AAAI-90*, pages 471–478, Boston, Massachusetts, July 1990.

[8] A. Casajus. Focal points, preferences and the generation of meaning. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[9] E. Charniak and R. P. Goldman. A probabilistic model of plan recognition. In *Proceedings of AAAI-91*, pages 160–165, Anaheim, California, July 1991. American Association for Artificial Intelligence.

[10] Eugene Charniak and Robert P. Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, November 1993.

[11] R. W. Cooper, D. V. DeJong, R. Forsythe, and T. W. Ross. Selection criteria in coordination games: Some experimental results. *The American Economic Review*, 80(1):218–233, 1990.

[12] V. P. Crawford and H. Haller. Learning how to cooperate: Optimal play in repeated coordination games. *Econometrica*, 58:571–595, 1990.

[13] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of AAAI-88*, pages 49–54, July 1988.

[14] J. L. Devore. *Probability and Statistics for Engineering and Sciences.* Brooks/Cole Publishing Company, Pacific Grove, California, 1991.

[15] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[16] J. Doyle. Rational belief revision. In *Proc. of KR-91*, pages 163–174, 1991.

[17] J. Doyle. Rationality and its role in reasoning. *Computational Intelligence*, 8(2):376–409, 1992.

[18] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3), 1979.

[19] E. Durfee, V. Lesser, and D. Corkill. Using partial global plans to coordinate distributed problem solvers. *IEEE Trans. on Computers*, 36:1275–1291, 1987.

[20] E. H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston, 1988.

[21] E. H. Durfee and V. R. Lesser. Partial global planning: a coordination framework for distributed hypothesis formation. *IEEE Trans. on Systems Man and Cybernetics*, 21(5):1167–1183, 1991.

[22] E. H. Durfee and V.R. Lesser. Global plans to coordinate distributed problem solvers. In *Proc. of IJCAI-87*, pages 875–883, 1987.

[23] J. Elgot-Drapkin. *Step-logic: Reasoning Situated in Time*. PhD thesis, Department of Computer Science, University of Maryland, College Park, Maryland, 1988.

[24] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

[25] Oren Etzioni. Embedding decision-analytic control in a learning architecture. *Artificial Intelligence*, 49:129–159, 1991.

[26] J. Farrell. Meaning and credibility in cheap-talk games. In M. Dempster, editor, *Mathematical Models in Economics*. Oxford University Press, 1988.

[27] M. Fenster. Reaching consensus without communication through the usage of focal points. Master's thesis, Bar-Ilan University, Ramat-Gan, Israel, 1994.

[28] M. Fenster, S. Kraus, and J. Rosenschein. Coordination without communication: Experimental validation of focal point techniques. In *Proc. of the First International Conference on Multiagent Systems*, pages 102–116, California, USA, 1995.

[29] Les Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1–3):107–138, 1991.

[30] D. Gauthier. Coordination. *Dialogue*, pages 195–221, 1975.

[31] M.R. Genesereth, M.L. Ginsberg, and J. S. Rosenschein. Cooperation without communication. In *Proc. of AAAI-86*, pages 51–57, Philadelphia, Pennsylvania, 1986.

[32] M. Georgeff. Communication and interaction in multi-agent planning. In *Proc. of the National Conference on Artificial Intelligence*, pages 125–129, Washington, D.C., 1983.

[33] J. Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the Association for Computing Machinery*, 37(3):549–587, 1990.

[34] J. C. Harsanyi and R. Selten. *General theory of equilibrium selection in games.* MIT Press, Cambridge, Mass, 1988.

[35] E. Horvitz, G. Cooper, and D. Heckerman. Reflection and action under scare resources: Theoretical principles and empirical study. In *Proceedings of IJCAI-89*, pages 1121–1127, Detroit, Michigan, 1989.

[36] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceeding, AAAI-88*, pages 111–116, 1988.

[37] Marcus J. Huber and Edmund H. Durfee. Deciding when to commit to action during observation-based coordination. In *ICMAS 95, Proceedings of the First International Conference on Multi-Agent Systems*, pages 163–170, San Francisco, California, June 1995.

[38] John B. Van Huyck, Raymond, C. Battalio, and Richard O. Beil. Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review*, 80(1):234–248, 1990.

[39] J. C. Jamison. Valuable cheap-talk and equilibrium selection. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[40] M. Janssen. Rationalizing focal points. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[41] N. R. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–201. John Wiley & Sons, Inc., 1996.

[42] M. Kandori, G. J. Mailath, and R. Rob. Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):29–56, 1993.

[43] H. A. Kautz and J. F. Allen. Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 32–37, Philadelphia, Pennsylvania, August 1986. American Association for Artificial Intelligence.

[44] Henry Kautz. *A Formal Theory of Plan Recognition.* PhD thesis, Department of Computer Science, University of Rochester, 1987. Technical Report 215.

[45] Henry Kautz. A formal theory of plan recognition and its implementation. In *Reasoning About Plans*, chapter 2, pages 69–126. Morgan Kaufmann Publishers, 1991.

[46] K. Konolige and M. E. Pollack. Ascribing plans to agents. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 924–930, Detroit, Michigan, August 1989.

[47] Richard Korf. Representing and using organizational knowledge in DAI systems. In *Proceedings of the Eleventh International Workshop on Distributed Artificial Intelligence*, Glen Arbor, Michigan, February 1992.

[48] F. Kramarz. Dynamic focal points in N-person coordination games. *Theory and Decision*, 40(3):277–313, 1996.

[49] S. Kraus and J. S. Rosenschein. The role of representation in interaction: Discovering focal points among alternative solutions. In E. Werner and Y. Demazeau, editors, *Decentralized Artificial Intelligence, Volume 3*, pages 147–165, Germany, 1992. Elsevier Science Publishers.

[50] S. Kraus and J. Wilkenfeld. Negotiations over time in a multiagent environment: Preliminary report. In *Proc. of IJCAI-91*, pages 56–61, Australia, 1991.

[51] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

[52] Jaeho Lee. Procedural agent model for plan recognition. In *Proceedings of the Workshop on the Next Generation of Plan Recognition Systems: Challenges for and Insight from Related Areas of AI*, pages 72–77, Montreal, Canada, August 1995. International Joint Conference on Artificial Intelligence.

[53] Ran Levy and Jeffrey S. Rosenschein. A game theoretic approach to distributed artificial intelligence and the pursuit problem. In *Decentralized Artificial Intelligence III*, pages 129–146. Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1992.

[54] D. K. Lewis. *Convention: A philosophy study.* Harvard University Press, Cambridge, Massachusetts, 1969.

[55] John McCarthy. Circumscription — a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[56] J. Mehta. Telling tales: Actors' accounts of their behaviour in coordination games. In *The 17th Arne Ryde Symposium on Focal Points: Coordination, Complexity and Communication in Strategic Contexts*, Sweden, 1997.

[57] J. Mehta, C. Starmer, and R. Sugden. Focal points in pure coordination games: An experimental investigation. *Theory and Decision*, 36:163–185, 1994.

[58] J. Mehta, C. Starmer, and R. Sugden. The nature of salience: An experimental investigation of pure coordination games. *The American Economic Review*, 84(3):658–673, 1994.

[59] T. M. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, January 1986.

[60] Y. Moses and M. Tennenholtz. Off-line reasoning for on-line efficiency. In *IJCAI-93*, pages 490–495, French, 1993.

[61] J. F. Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

[62] M. Nirkhe and S. Kraus. Formal real-time imagination. *Fundamenta Informaticae, special issue on Formal Imagination*, 23(2,3,4):371–390, 1995.

[63] M. Nirkhe, S. Kraus, D. Perlis, and M. Miller. How to (plan to) meet a deadline between *now* and *then*. *Journal of Logic and Computation*, 7(1):109–156, 1997.

[64] D. Perlis, J. Elgot-Drapkin, and M. Miller. Stop the world! – I want to think! Invited paper, *International J. of Intelligent Systems*, special issue on Temporal Reasoning, K. Ford and F. Anger (eds.), vol. 6, 1991, pp. 443–456.

[65] Anatol Rapoport and M. Guyer. A taxonomy of 2×2 games. *Yearbook of the Society for General Systems Research*, XI:203–214, 1966.

[66] E. Rasmusen. *Games and Information.* Basil Blackwell Ltd., Cambridge, MA, 1989.

[67] S. Russell and E. Wefald. Principles of metareasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 400–411. Morgan-Kaufman, 1989.

[68] Thomas C. Schelling. *The Strategy of Conflict.* Oxford University Press, New York, 1963.

[69] C. Schmidt, N. Sridharan, and J. Goodson. The plan recognition problem: An intersection of artificial intelligence and psychology. *Artificial Intelligence*, 11(1):45–83, 1978.

[70] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proc. of AAAI-92*, pages 276–281, San Jose, California, 1992.

[71] L. M. Stephens and M. B. Merx. Agent organization as an effector of DAI system performance. In Miroslav Benda, editor, *Proceedings of the 9th Workshop on Distributed Artificial Intelligence*, pages 263–292, Bellevue, Washington, September 1989.

[72] L. M. Stephens and M. B. Merx. The effect of agent control strategy on the performance of a DAI pursuit problem. In *Proceedings of the 10th International Workshop on Distributed Artificial Intelligence*, Bandera, Texas, October 1990. Chapter 14.

[73] R. Sugden. A theory of focal points. *The Economic Journal*, 105:533–550, 1995.

[74] K. P. Sycara. Argumentation: Planning other agents' plans. In *Proc. of IJCAI-89*, pages 517–523, Michigan, 1989.

[75] Robert Weida. Knowledge representation for plan recognition. In *Proceedings of the Workshop on the Next Generation of Plan Recognition Systems: Challenges for and Insight from Related Areas of AI*, pages 119–123, Montreal, Canada, August 1995. International Joint Conference on Artificial Intelligence.

[76] E. Werner. Cooperating agents: A unified theory of communication and social structure. In *Distributed Artificial Intelligence: Volume II*, pages 3–36. Morgan Kaufmann, 1990.

[77] P. Young. The evolution model of bargaining. *Journal of Economic Theory*, 59:145–168, 1993.

[78] P. Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.

[79] G. Zlotkin and J. S. Rosenschein. Negotiation and conflict resolution in non-cooperative domains. In *Proceedings of AAAI-90*, pages 100–105, Boston, MA, 1990.