Compromise in Negotiation: Exploiting Worth Functions over States*

Gilad Zlotkin
Center for Coordination Science
Sloan School of Management, MIT
1 Amherst Street, E40-179, Cambridge, MA 02139, USA
gilad@mit.edu

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

Abstract

Previous work [32, 33, 35, 37] discussed inter-agent negotiation protocols. One of the main assumptions there was that the agents' goals remain fixed—the agents cannot relax their initial goals, which can be achieved only as a whole and cannot be *partially* achieved. A goal there was considered a formula that is either satisfied or not satisfied by a given state.

We here present a more general approach to the negotiation problem in non-cooperative domains where agents' goals are *not* expressed as formulas, but rather as *worth functions*. An agent associates a particular value with each possible final state; this value reflects the degree of satisfaction the agent derives from being in that state.

With this new definition of goal as worth function, an agreement may lead to a situation in which one or both goals are only partially achieved (i.e., agents may not reach their most desired state). We present a negotiation protocol that can be used in a general non-cooperative domain when worth functions are available. This *multi-plan deal type* allows agents to compromise over their degree of satisfaction, and (in parallel) to negotiate over the joint plan that will be implemented to reach the compromise final

*Some material in this paper appeared in preliminary form in [36].

state. The ability to compromise often results in a better deal, enabling agents to increase their overall utility.

Finally, we present more detailed examples of specific worth functions in various domains, and show how they are used in the negotiation process.

1 Introduction

1.1 Distributed Problem Solving vs. Multiagent Systems

One of the primary research issues in Distributed Artificial Intelligence (DAI) is how to have agents interact coherently. There are two main points of view in DAI research, namely that of Distributed Problem Solving (DPS) and Multiagent (MA) systems; each considers the problems of coherent interaction from a different perspective [8].

In DPS, agents are centrally designed, and the system has been built to be distributed so as to improve system performance, scalability, modularity, and/or reliability. In addition, certain problems, through geographical or functional distribution of data and knowledge, are appropriately dealt with through a distribution of control among distinct agents. The central research issue in DPS is how to have these independent problem solvers work together coherently to solve the global problem, while maintaining acceptably low levels of inter-agent communication: agents should not duplicate work unnecessarily, should not distract one another unnecessarily, and should share tasks and partial results in a productive way. Work in this field includes [27, 2, 3, 18, 6, 7, 4, 5].

In multiagent systems, agents are not assumed to be centrally designed, and no assumptions can be made about agents working together cooperatively. On the contrary: agents are assumed to be autonomous utility maximizers, who will cooperate only when they can benefit from that cooperation (perhaps in the broad sense; what benefits an individual agent, of course, is dependent on how it is designed). MA researchers are also concerned with the coherence of interaction, but must build agents without having the luxury of designing their interaction opponents. The central research issue in MA is how to have these autonomous agents identify common ground for cooperation, and choose and perform coherent actions. When no common ground can be found, the agents must resolve conflict to the best of their ability. Work in this field includes [25, 28, 16, 14, 15, 9].

DPS and MA research share many concerns, since even centrally-designed agents in DPS can be in fundamental conflict (having different perspectives, different knowledge, and different local goals); thus, DPS can benefit from MA research on how to resolve inter-agent conflict. Similarly, MA research concerns itself with finding effective protocols and interaction frameworks with certain desirable properties—and this is exactly the main issue that concerns DPS (although there does not seem to be a complete overlap between the properties DPS requires from protocols and the properties required in MA systems).

In effect, DPS and MA systems occupy the extreme poles of what is the full spectrum of DAI research. There also exists work that directly combines the two points of view. For example, work described in [20] falls within the general DPS category: agents are solving a global problem cooperatively. The technique used by those agents, however, stresses the autonomy of each one. Global problem solving behavior is achieved by its suitable instantiation as local, independent goals for each of the autonomous agents. Similarly, some of Durfee's research [6] assigns great autonomy to DPS agents, as does the work of Ferber on Eco-Problem Solving systems [13].

1.2 Negotiation in Distributed Artificial Intelligence

Since coherent action is central to both DPS and MA systems, the concept of "negotiation," in different guises, has appeared throughout the DAI literature. DPS researchers [6, 17] see negotiation as an important mechanism for assigning tasks to agents, for resource allocation, and even for deciding which problem-solving tasks to undertake. MA researchers [32, 34] have agents use negotiation to share the work associated with carrying out a joint plan (for the agents' mutual benefit), or to resolve outright conflict arising from limited resources (again, to the agents' mutual benefit).

In this paper, we are concerned with negotiation among autonomous agents. The agents are autonomous in the sense that they have their own utility functions, and no global notion of utility (not even an implicit one) plays a role in their design. Thus, this research falls squarely in the multiagent systems camp.

In our previous work [32, 33, 35, 37, 30, 31] we discussed several domains for multiagent negotiation protocols. One of our main assumptions has been that agents' goals remain fixed—agents cannot "relax" their initial goals to reach agreement. Goals can be achieved only as a whole and cannot be *partially* achieved.

To cope with essential conflict between two agents' goals, we presented in [33] a negotiation protocol that allows the agents to reach an agreement in which they cooperate until a certain point, and then flip a coin to decide which agent is going to continue on alone to achieve its goal. In conflict situations, only one agent will achieve its goal (with some probability).

This paper presents a more general approach to the negotiation problem in non-cooperative domains where agents can relax their initial goals. An agreement may lead to a situation in which one or both goals are only partially achieved. We also present several approaches to the goal relaxation problem and give some examples of domains in which each of the different approaches can be used.

When goals can be relaxed, agents are negotiating both over what *parts* of their goals will be satisfied, and (in parallel) over the joint plan that will be implemented to satisfy those parts. A key ingredient of our approach is the notion of worth, which we will use to specify the value, to an agent, of total or partial achievement of its goal. Although agents can relax their goals, however, they do not fundamentally alter their computation of worth during the negotiation process (a process that would be similar, for example, to what happens in the PERSUADER system [28, 29]).

2 Rules of the Game: The Overall Negotiation Framework

Because two agents co-existing within the same environment might interfere with the actions of the other, there needs to be coordination of activity. Depending on the particular domain and goals involved, the possibility may exist that the agents will be able to help each other and achieve both goals with a lower overall cost.

A *deal* between agents is a joint plan, in which agents share the work of transforming the world from the initial state to some final state. The plan is "joint" in the sense that the agents probabilistically share the load, compromise over which agent does which actions, or even compromise over which agent gets its goal satisfied.[1] In the broad sense, the utility for an agent of a deal is the difference between the worth of the agent's goal achieved through that deal, and the cost of that agent's part of the deal.

We assume the interaction between agents occurs in two consecutive stages. First the agents negotiate, then they execute the entire joint plan that has been agreed upon. No divergence from the negotiated deal is allowed. The sharp separation of stages has consequences, in that it rules out certain negotiation tactics that might be used in an interleaved process.

2.1 The Concept of a Solution

When we say that we are looking for a "solution" to the negotiation problem, we mean we wish to establish three things:

1. A precise definition of what a goal is.

2. A precise definition of a deal type and utility; these might include probabilistic sharing of actions, probabilities associated with achieving final states, partial achievement of goals, and many domain dependent attributes, including the nature of the conflict deal (the domain dependent default deal that the agents live with if *no* agreement is reached). Our previous work has discussed the *pure deal type*, *mixed deal type* [32, 30], and *semi-cooperative deal type* [33].

3. A specification of how one should design an agent to negotiate in a particular well-defined negotiation environment.

How should one evaluate a solution? There are several ways of doing this, primarily focusing on how we evaluate the properties of deal types. Deal types may have a variety of

[1] Our use of the term *joint plan* differs somewhat from other uses in the artificial intelligence (AI) literature (for example, [19]). There, the term joint plan implies a joint *goal*, and mutual commitment by the agents to full implementation of the plan (e.g., if one agent dropped out suddenly, the other would still continue). In our use of the term, the agents are only committed to their own goal and their *part* of the combined plan. Each may do its part of the plan for different reasons, because each has a different goal to achieve. Were one agent to drop out, the other agent may or may not continue, depending on whether it suited its own goal.

attributes that are considered desirable. For example, certain kinds of deal types provide solutions to more general situations. The mixed deal type (discussed in [32, 30]), where agents probabilistically divide the work associated with bringing the world to a mutually acceptable state, can only be used in cooperative situations where such a mutually acceptable state exists. When such a state does not exist (i.e., there exists real conflict), the agents can still each benefit by negotiating using the semi-cooperative deal type (discussed in [34, 31]). This deal type allows agents to agree to cooperate as far as they can, then flip a coin to see who will continue and accomplish his goal alone. The semi-cooperative deal type thus increases the size of the *negotiation set*, the set of possible (rational) agreements.

Deal types may have other positive attributes, such as requiring less initial information, and may also change the availability of negotiation strategies that are pair-wise optimal and/or stable (in equilibrium). It is also desirable to have negotiating agents converge to a *pareto optimal* deal (a deal that could only be improved for one agent in ways that make it worse for the other agent).

As mentioned above, it is highly desirable that an agent's negotiation strategy be in *equilibrium*. A strategy S is said to be in symmetric Nash equilibrium [22] if assuming that your opponent's agent is using S, the best you can do is to also have your agent use S. Thus, no other agent will be able to take advantage of yours by using a different negotiation strategy. Moreover, there is no need to exercise secrecy regarding the design of your agent—on the contrary, it is actually beneficial to broadcast its negotiation strategy, so that the other agent doesn't blunder and potentially cause both harm.

There is, in a sense, a meta-game going on between the designers of autonomous agents [24, 26]. Each one wants to design an agent that maximizes its designer's utility. There are strong motivations to design an agent so that it uses a negotiation strategy in equilibrium, in that it results in "best performance" in pair-wise competitions between your agent and any other given agent. Conflicts will be avoided whenever possible, and deals that are reached will be pareto optimal.[2]

3 Goal Relaxation

The main distinction between the approach of this paper and our previous work is the definition of what constitutes an agent's goal. In Task Oriented Domains (TODs) [32, 30, 37], a goal specifies a set of tasks that the agent is required to carry out. In State Oriented Domains (SODs) [33, 34, 31], a goal specifies a state or set of states that an agent wishes to reach. In contrast, in Worth Oriented Domains (WODs) the goal definition is subsumed by a worth function over all possible final states. Those states with the highest value of worth

[2]However, there may be ecological motivations for designing agents that don't use equilibrium strategies, since multiple non-equilibrium agents might all benefit from their deviant strategies. For example, ecologically a group of agents that all play Cooperate in the Prisoners' Dilemma will do better than a group using the equilibrium strategy of Defect, even though a Defecting agent will win in a head-to-head competition with a Cooperating agent [1]. Nevertheless, we here concentrate on pair-wise equilibrium, and search for symmetric equilibrium negotiation strategies.

might be thought of as those that satisfy the full goal, while others, with lower worth values, only partially satisfy the goal.

We assume that the worth of the final state, and the cost associated with the plan, are comparable and can be cast into equivalent units of utility. There is the potential for a tradeoff in cost and worth; for example, an agent might accept an outcome with lower worth if he ends up doing less work to bring it about.

The ability to relax one's goal opens up new opportunities for cooperative agreement. In SODs, when both agents' goals are achievable, agents can agree on a joint plan. The division of labor in a joint plan serves as a kind of implicit utility transfer between agents. When one agent agrees to do more work in a joint plan that achieves both agents' goals, he increases the utility of the second agent. The ability to transfer utility in this fashion is the basis for cooperative activity in SODs; since we do not consider explicit utility transfer, these implicit transfers serve a critical role. In Worth Oriented Domains, the flexibility of this implicit utility transfer is increased. There is now another dimension along which agents can compromise—not only can they tune the amount of work they do, they can also adjust which final state they reach and how much worth they extract from the plan.

3.1 Meeting Scheduling—An Example

As an example, consider two agents that are trying to set up a meeting. The first wishes to meet later in the day, and the second wishes to meet earlier in the day. Both agents prefer meeting today (regardless of the time) rather than tomorrow. In addition, each wishes to have the meeting in his own office (so that the costs associated with travel and travel time will be low). While the first agent assigns highest worth to a meeting at 4pm, he also assigns (progressively smaller) worths to a meeting at 3pm, 2pm, and so on. Similarly, the second agent assigns highest worth to a meeting at 9am, but also assigns (progressively smaller) worths to a meeting at 10am, 11am, and so on. Having the meeting on another day is assigned a very low worth by both agents.

Agents can reach several different agreements, by showing flexibility both about the time and the place of the meeting. By accepting a sub-optimal time, an agent accepts a lower worth payoff, but may be able to offset this depending on where the meeting is held, and how much it will cost him to attend. For example, the first agent might suggest having the meeting at the second agent's office, late in the day. The second agent might counter-offer by agreeing to the location but asking for a slightly earlier time.

A deal in this encounter is a place and a time for the meeting. Let's flesh out the details of this example and see how agents might evaluate the situation. The worth function for agent A_1 can be seen in Figure 1. The y-axis denotes the worth the agent associates with meeting A_2 at the time given on the x-axis, regardless of the place the meeting is held. The function shown in the figure is logarithmic.

Figure 1

Of course, some meeting places are better for A_1 than others, because he will spend less to get there. The cost of meeting in his own office is 0, while if he drives across town to A_2's office, it will cost him roughly 70 units. It will cost him 40 to go halfway, and 22 to meet A_2 a quarter of the way (at University Avenue). The cost is non-linear, as just getting into his car will cost a disproportionate amount of time; the cost function is logarithmic. The graph in Figure 2 shows the utility agent A_1 assigns to each possible deal. The height at each point of the plane shows the utility (i.e., worth minus cost) that A_1 assigns to the deal.

Figure 2

Agent A_2 is in the symmetric situation. He most wants to meet at 9am in his own office. The graph showing his worth, and the one showing his utility, assume shapes symmetric to Figures 1 and 2. The sum of the agents' utilities can be seen in the graph in Figure 3. Note that the deals with the highest utility sums are at the four corners of the plane. Those deals give to one or both of the agents a best time or place. The meeting might be held at 4pm in A_1's office, at 9am in A_2's office, at 4pm in A_2's office, or at 9am in A_1's office.

Figure 3

Although there are four efficient deals that maximize the sum of the agents' utilities, there are only two that maximize the product of the agents' utilities (see Figure 4). Either the agents can meet at A_1's preferred time at A_2's preferred place, or vice versa (9am at A_1's office, or 4pm at A_2's office). Any product maximizing mechanism will choose one of those two deals.

Figure 4

The shapes of the sum and product graphs are sensitive to the worth and cost functions of the agents. For example, if the worth and cost functions were linear (instead of logarithmic), the sum of the agents' utilities would be a constant sum—all deals would have the same utility sum (as can be seen on the left side of Figure 5). The product of agent utilities creates a maximal plateau stretching between the previous two maximal deals (A_1's time, A_2's place, and vice versa). Any deal on the plateau might be reached (including the previous two deals). If, however, the worth and cost functions were exponential, the sum and the product of the agents' utilities assume the shapes seen in Figure 6, a kind of mound with a maximal point in the center. A product maximizing mechanism would then have the agents meet in the middle at noon.

Figure 5

Figure 6

3.2 Mechanisms that Maximize the Product of Utilities

We considered above deals that maximize the sum of agents' utilities, and those that maximize the product of their utilities. In general, we focus on the latter criterion, and pay particular attention to *Product Maximizing Mechanisms*, or PMMs. Our emphasis on this class of deal-making mechanisms has its roots in game theory research.

There are a number of existing approaches to the bargaining problem in game theory. One of the earliest and most popular was Nash's axiomatic approach [22, 21]. Nash was trying to axiomatically define a "fair" solution to a bargaining situation. He listed the following criteria as ones that a fair solution would satisfy:

1. Individual rationality (it would not be fair for a participant to get less than he would anyway without an agreement);

2. Pareto Optimality (a fair solution will not specify an agreement that could be improved for one participant without harming the other);

3. Symmetry (if the situation is symmetric, that is, both agents would get the same utility without an agreement, and for every possible deal, the symmetric deal is also possible, then a fair solution should also be symmetric, i.e., give both participants the same utility);

4. Invariance with respect to linear utility transformations. As an example, imagine two agents negotiating over how to divide \$100. If one agent measures his utility in dollars while the other measures his in cents, it should not influence the fair solution. Similarly, if one agent already has \$10 in the bank, and evaluates the deal that gives him x dollars as having utility $10 + x$ while the other evaluates such a deal as having utility x, it should not influence the fair solution (i.e., change of origin doesn't affect the solution);

5. Independence of irrelevant alternatives. Imagine two agents negotiating about how to divide $10,000$ cents. The Nash solution will be 5,000 cents for each, due to the symmetry assumption above. Now imagine that the same agents are negotiating over \$100. Even though there are now some deals that they can't reach (for example, the one where one agent gets \$49.99, and the other gets \$50.01), the solution should be the same, because the original solution of 5,000 cents can still be found in the new deal space.

Nash showed that the product maximizing solution not only satisfies the above criteria, but it is the only solution that satisfies them. We use the Nash solution, in general, as a reasonable bargaining outcome, when it is applicable. Nash, however, had some assumptions about the space of deals that we do not have. For example, the Nash bargaining problem assumes a bounded, convex, continuous, and closed region of negotiation. In our agent negotiations we do not assume that the space of deals is convex, nor that it is continuous.

4 Domain Definition

We here present the formal definitions of Worth Oriented Domains and their associated encounters, which incorporate the notion of worth functions that assign to every state some worth value.

A Worth Oriented Domain is a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{J}, c >$ where:

1. $\mathcal{S}$ is the set of all possible world states;
2. $\mathcal{A} = \{A_1, A_2, \ldots A_n\}$ is an ordered list of agents;
3. $\mathcal{J}$ is the set of all possible joint (i.e., n-agent) plans. A joint plan $J \in \mathcal{J}$ moves the world from one state in $\mathcal{S}$ to another (i.e., $J: \mathcal{S} \rightarrow \mathcal{S}$). If $s \in \mathcal{S}$ is the initial state of the world, then $J(s) \in \mathcal{S}$ is the state of the world after the joint plan J has been executed. The actions taken by agent k are called k's *role* in J, and will be written as J_k. We can also write J as $(J_1, J_2, \ldots, J_n)$;
4. c is a function $c: \mathcal{J} \rightarrow (\mathbb{R}^+)^n$. For each joint plan J in $\mathcal{J}$, $c(J)$ is a vector of n positive real numbers, the cost of each agent's role in the joint plan. $c(J)_i$ denotes the cost of agent i's role in J. If agent i plays no role in J, his cost $c(J)_i$ is 0.

Definition 1 *An* encounter *within a WOD* $< \mathcal{S}, \mathcal{A}, \mathcal{J}, c >$ *is a tuple* $< s, (W_1, W_2, \ldots, W_n) >$ *such that* $s \in \mathcal{S}$ *is the initial state of the world, and for all* $k \in \{1 \ldots n\}, W_k: \mathcal{S} \rightarrow \mathbb{R}$ *is the* worth *function of agent* k*.* W_k *assigns some value to each possible final state of the world.* W_k *will also be called* A_k*'s* goal*.*

We will assume that each worth function W_i is defined for all states. The worth function stands in place of and encapsulates the classical artificial intelligence notion of a goal as a set of acceptable final states. $W_i(f)$ is an indication of how much of agent A_i's goal has been achieved, or how "close" state f is to the achievement of A_i's whole goal. Note that the worth function makes no reference, for example, to the other agent's goals; it is a (fixed) function over states. A more general formulation might attempt to allow an agent to have as a goal blocking the other agent's goals (whatever they might be), but we do not consider that case.

The definition of worth (value) of a goal in State Oriented Domains [33, 34, 31], which assumed *no* goal relaxation, was actually a subcase of the current definition. The worth value of a goal in an SOD can be thought of as a two-valued worth function, which assigned some constant value w_i to states that achieved A_i's goal g_i.

$$W_i(f) = \begin{cases} w_i & \text{if } f \models g_i \\ 0 & \text{otherwise.} \end{cases}$$

We write $f \models g_i^k$ to mean that subgoal g_i^k has been achieved in state f. This notation relates to the classic approach in artificial intelligence of viewing states and goals as collections of formulas.

The above (SOD-like) function W_i corresponds to a situation in which there does not exist a state that partially satisfies agent A_i's goal. In the general WOD case, W_i can be any function whose range is $\mathbb{R}$.

Definition 2 *Given a WOD* $< \mathcal{S}, \mathcal{A}, \mathcal{J}, c >$*, we define:*

- $\mathcal{P} \subset \mathcal{J}$ *to be the set of all* one-agent *plans, i.e., all joint plans in which only one agent has an active role.*
- *The cost* $c(P)$ *of a one-agent plan in which agent* k *has the active role,* $P \in \mathcal{P}$*, is a vector that has at most one non-zero element, in position* k*. When there is no confusion, we will use* $c(P)$ *to stand for the* k*-th element (i.e.,* $c(P)_k$*) rather than for the entire vector.*

The exact (domain-specific) definition of the cost of a plan is not critical to the subsequent discussion (for example, whether or not the cost of a plan depends on the plan's initial state). The cost function may, in fact, have parameters other than the plan itself, such as the initial state, day of the week, and other domain dependent variables. What is important is the ability of an agent to measure the cost of a one-agent plan, and his ability to measure the cost of one agent's part of a joint multiagent plan.

Definition 3 Best Plans

- $s \xrightarrow{k} f$ *is the minimal cost one-agent plan in* $\mathcal{P}$ *in which agent* k *plays the active role and moves the world from state* s *to state* f *in* $\mathcal{S}$*.*
- *If a plan like this does not exist then* $s \xrightarrow{k} f$ *will stand for some constant plan* $\bowtie_k$ *that costs infinity to agent* k *and* 0 *to all other agents.*
- *If* $s = f$ *then* $s \xrightarrow{k} f$ *will stand for the empty plan* Λ *that costs* 0 *for all agents.*
- $s \xrightarrow{k} F$ *(where* s *is a world state and* F *is a* set *of world states) is the minimal cost one-agent plan in* $\mathcal{P}$ *in which agent* k *plays the active role and moves the world from state* s *to one of the states in* F*:*

$$c(s \xrightarrow{k} F) = \min_{f \in F} c(s \xrightarrow{k} f).$$

5 Underlying Assumptions

In [32, 30], we introduced several simplifying assumptions, some of which are in force for our discussion here as well:

1. **Utility Maximizer:** *Each agent wants to maximize his expected utility.*

2. **Complete Knowledge:** *Each agent knows all relevant information.*

3. **Isolated Negotiation:** *An agent cannot commit himself as part of the current negotiation to some behavior in a future negotiation, nor can he expect that his current behavior will in any way affect a future negotiation. Similarly, an agent cannot expect others to behave in a particular way based on their previous interaction history. Each negotiation stands alone.*

4. **Bilateral Negotiation:** *In a multiagent encounter, negotiation is done between a pair of agents at a time.*

5. **Symmetric Abilities:** *All agents are able to perform the same set of operations in the world, and the cost of each operation is independent of the agent carrying it out.*

6 One Agent Best Plan

Utility for an agent in general is the difference between the worth of a final state and the cost that an agent has to pay to bring the world to this final state. If agent i were alone in the world, it would bring the world to its "stand-alone optimal" state f_i that satisfies the following condition:

$$W_i(f_i) - c(s \to f_i) = \max_{f \in \mathcal{S}} \left(W_i(f) - c(s \to f) \right).$$

We define U_i to be i's stand-alone maximal utility, e.g., $W_i(f_i) - c(s \to f_i)$, which is the maximum utility that agent i could achieve if it were alone in the world.

Consider the following example of how a *single* agent might use a worth function to decide on its actions.

6.1 Subgoal Set

One category of multivalued worth functions would assign some value to each of several distinct subgoals that an agent has to achieve. States in which only some of the subgoals are achieved would be rated lower than states in which they are all achieved (we explore other types of worth functions below).

An agent i may have a set of distinct subgoals or tasks $\{g_i^k \mid k = 1 \ldots n_i\}$ that he has to achieve. Each subgoal g_i^k's worth to him is w_i^k. In this case, his overall goal is a conjunction of subgoals, $g_i = \bigwedge_{k=1}^{n_i} (g_i^k)$. The worth function in this case might be defined as

$$W_i(f) = \sum_{f \models g_i^k} (w_i^k).$$

6.1.1 Example: The Blocks World Domain

Let the initial state of the world be as shown in Figure 7.

Figure 7

Agent A_1 has two subgoals: g_1^1 is "The Black box is clear at slot 2, but not on the table" and g_1^2 is "The White box is alone at slot 3". The worth of the two subgoals is $w_1^1 = 4$ and $w_1^2 = 6$. If A_1 were alone in the world, he could bring the world to one of the three stand-alone optimal states, as seen in Figure 8.

Figure 8

Agent A_1 is indifferent among f_1^1, f_1^2 and f_1^3, because in all cases his utility is 2. They are all stand-alone optimal states, as can be seen below.

- $f_1^1 \models g_1^1$, therefore $W_1(f_1^1) = 4$. $c(s \rightarrow f_1^1) = 2$, therefore the utility of agent 1 from f_1^1 is $2 = (4 - 2)$.
- $f_1^2 \models g_1^2$, therefore $W_1(f_1^2) = 6$. $c(s \rightarrow f_1^2) = 4$, therefore the utility of agent 1 from f_1^2 is $2 = (6 - 4)$.
- $f_1^3 \models g_1^1 \cap g_1^2$ therefore $W_1(f_1^3) = 10 = (6 + 4)$. $c(s \rightarrow f_1^3) = 8$, therefore the utility of agent 1 from f_1^3 is $2 = (10 - 8)$.

Agent 1 is indifferent among f_1^1, f_1^2 and f_1^3 even though f_1^3 fully satisfies his goal, and f_1^1 and f_1^2 only partially satisfy his goal. This is due to the definition of W_1—an agent gets positive utility from achieving a subgoal, and zero utility from unachieved subgoals.

There can be domains where an agent gets some penalty (negative utility) from unachieved subgoals. Let p_i^k be the penalty that agent i gets when his subgoal g_i^k is not achieved (it is convenient to think of penalties as negative numbers). We could then redefine the worth function to be

$$W_i(f) = \sum_{f \models g_i^k} w_i^k + \sum_{f \not\models g_i^k} p_i^k.$$

In the previous example, if $p_1^1, p_1^2 < 0$ then agent 1 prefers f_1^3 over f_1^1 and f_1^2.

7 Negotiation over Sub-optimal States

Assume now that two agents, each with their own worth function, find themselves in a shared environment. It may be impossible for them to each perform their single-agent best plan (because of resource constraints, interfering actions, or conflicting goals). On the other hand, they may actually be able to benefit from the presence of the other agent, if their actions are chosen and coordinated correctly. Designers of such agents would likely be interested in having them come to an agreement that resolves their conflicts or exploits cooperative opportunities.

7.1 Using Mixed Joint Plans with Multivalue Worth Functions

Agents can negotiate over joint plans J that move the world from the initial state s to a final state. To overcome the problem of indivisible operations, they may negotiate over *mixed joint plans.*

Definition 4 Mixed Joint Plan *Given a joint plan $J \in \mathcal{J}$, a* mixed joint plan *is* $J{:}\,p;\ 0 \leq p \leq 1\ \in \mathbb{R}$.

The semantics of a mixed joint plan is that the agents will perform the joint plan $J = (J_1, J_2)$ with probability p, or the symmetric joint plan (J_2, J_1) with probability $1 - p$ (i.e., where the agents have switched roles in J). Under the Symmetric Abilities Assumption (5) from Section 5, both agents are able to execute both parts of the joint plan, and the cost of each role is independent of which agent executes it.

In situations where goals cannot be relaxed (e.g., no worth functions exist), agents negotiate over mixed joint plans. In such cases, the agents consider only plans that bring the world to a state satisfying both agents' goals (fully). The utility of a deal to an agent is then the stand-alone cost to reach that final state, minus the (expected) cost of his role in the plan. In Worth Oriented Domains, the basic approach is the same, but now agents are able to agree on a plan that carries them to any final state, not necessarily one that fully satisfies both agents' goals. Therefore, although the agents are explicitly negotiating over mixed joint plans, they are implicitly negotiating over two separate items:

1. What will be the final state of the world?

2. How will they share the work of bringing the world to that final state?

There is a tradeoff between these two items. As mentioned above, an agent might accept an outcome with lower worth if he ends up doing less work to bring it about. This was exactly the situation that we saw above in the scheduling example (in Section 3.1), where time of the meeting (final state) was traded off with location of the meeting (cost) (the symmetric definition of final state and cost would also have been possible). Several deals with different time and location choices were equally valued.

The WOD utility for an agent of a mixed joint plan is thus simply the difference between the final state's worth to him, and the expected work to which the deal commits him.

Definition 5

- *If* δ *is a mixed joint plan, then* Utility$_i(\delta)$ *is defined to be* $W_i(\delta(s)) - c_i(\delta)$ *(where* $c_i(\delta)$ *is the expected cost for agent* i *of the mixed joint plan* δ*, i.e.,* $c_i(\delta) = pc(J)_i + (1-p)c(J)_k$*, where* k *is* i*'s opponent).*

- *Failure to reach a deal will result in the two agents doing nothing, leaving the world in the initial state. The* conflict deal θ *is thus defined to be the empty joint plan* $(\Lambda, \Lambda){:}\,p$ *for any* p*.*

Note that, given the above definitions, if the initial state has non-negative worth to both agents, the negotiation set (i.e., the set of all individual rational and pareto optimal deals) will never be empty. In the worst case, it will include the conflict deal.

7.1.1 A Mixed Joint Plan Negotiation

Consider the following example of two-agent negotiation over mixed joint plans, using a worth function of the same type as was presented above (i.e., a worth function that assigns value to distinct subgoals).

Let the initial state of the world be as in Figure 9.

Figure 9

- Agent A_1 has two subgoals: g_1^1 is "The Black box is on the White box at slot 1" and g_1^2 is "The Gray box is clear at slot 3". The worths of the two subgoals are $w_1^1 = 10$ and $w_1^2 = 4$. The penalties for unachieved goals are $p_1^1 = p_1^2 = -2$.

- Agent A_2 has two subgoals: g_2^1 is "The Black box is on the White box at slot 1" and g_2^2 is "The Gray box is clear at slot 4". The worths of the two subgoals are $w_2^1 = 10$ and $w_2^2 = 4$. The penalties for unachieved goals are $p_2^1 = p_2^2 = -2$.

The agents share one subgoal ($g_1^1 = g_2^1$) and have a conflict over the other subgoals. If each agent were alone in the world, to fully achieve his goal he would have a cost of at least $10 = (8+2)$, which would give him a utility of $U_i = 4 = (10+4) - 10$. There is no final state that fully achieves both goals, but there are final states that partially achieve them. The best joint plan T that achieves the swap in slot 1 (g_1^1 and g_2^1) costs 2 for each agent.

There are two deals that maximize the product of the agents' utility, one preferred by A_1 and one preferred by A_2. A_1 prefers the mixed joint plan in which they both cooperate in performing the swap, and then A_1 brings the gray block to slot 3 (deal δ_1, whose final state is shown in Figure 10). A_2 prefers the mixed joint plan in which they both cooperate in performing the swap, and then A_2 brings the gray block to slot 4 (deal δ_2). This is a symmetric situation, and any product maximizing mechanism will choose one or the other arbitrarily (see Section 3.2).

Figure 10

$$
\begin{aligned}
\text{Utility}_1(\delta_1) &= \text{Utility}_2(\delta_2) = (10+4) - (2+2) = 10 \\
\text{Utility}_1(\delta_2) &= \text{Utility}_2(\delta_1) = (10-2) - 2 = 6
\end{aligned}
$$

The expected utility for both agents, who have flipped a coin to choose between the above two alternatives, is 8—which is greater than U_i, which is 4. In other words, even though the two goals can be only partially achieved, there exists a deal that is better for both agents than full achievement of their own goal by themselves.

Although negotiating over mixed joint plans can be used with worth functions, it is possible for agents to "do better," in the sense that they will have more utility to divide between themselves. A more general negotiation protocol would be to negotiate over deals that are *pairs* of mixed joint plans, i.e., *multi-plan deals.*

Definition 6

- *A Multi-Plan* Deal *is* (δ_1, δ_2, q) *where* δ_i *are mixed joint plans and* $0 \le q \le 1 \in \mathbb{R}$ *is the probability that the agents will perform* δ_1 *(they will perform* δ_2 *with probability* $1-q$*).*

- $\text{Utility}_i(\delta_1, \delta_2, q) = q(W_i(\delta_i(s)) - c_i(\delta_i)) + (1-q)(W_i(\delta_j(s)) - c_i(\delta_j))$.

Agents are thus able to use a multi-plan deal based Unified Negotiation Protocol, such as the one introduced in [34, 31], using worth functions as the base of the utility evaluation. Note that, unlike the SOD analysis there where we had several types of discrete interactions (cooperative, compromise, conflict), we here have a continuous situation. In an SOD cooperative or compromise interaction, both agents' goals are fully achieved, and in the conflict situation, only one is. In a WOD, both agents' goals may be fully achieved, or one may be fully achieved, or both may be partially achieved, and so on. The range of possibilities increases.

Consider the following example, which shows the increased utility available for the agents to share when they negotiate over multi-plan deals instead of over mixed joint plans.

Figure 11

The initial state s of the world can be seen in Figure 11. A's goal is to swap the position of the blocks in slot 1, but to leave the blocks in slot 2 in their initial position (there is only one state that satisfies this goal; call it f_A). B's goal is to swap the position of the blocks in slot 2, but to leave the blocks in slot 1 in their initial position (f_B).

To achieve his goal alone, each agent needs to do at least 8 PickUp/PutDown operations (each with a cost of 1). Assume that A's Worth function assigns 10 to f_A and 0 to all other states, and that B's Worth function assigns 10 to f_B and 0 to all other states. In this case, the negotiation set over mixed joint plans includes the deals $(s \to f_A, \Lambda)\colon 1$ and $(\Lambda, s \to f_B)\colon 0$. Using some product maximizing mechanism, the agents will break the symmetry of this situation by flipping a coin. The utility of each agent from this deal is $1 = \frac{1}{2}(10-8)$.

Negotiation over the multi-plan deal type will cause the agents to agree on $(\delta_A, \delta_B)\colon \frac{1}{2}$, where δ_i is the mixed joint plan in which both agents cooperatively achieve i's goal. The best joint plan for doing the swap in one of the slots costs 2 PickUp/PutDown operations for each agent. The utility for each agent from this deal is $3 = (\frac{1}{2}(10-2) + \frac{1}{2}(-2))$. By negotiating using the multi-plan deal type instead of mixed joint plans, there is more utility for the agents to share, 6 instead of 2.

8 Examples of Worth Functions

Now that we've seen the definition of a Worth Oriented Domain, and considered several ways that agents might reach agreements in a WOD, we have another question to address.

What should an agent's worth function look like? There is no single best answer, since it depends to a large degree on the domain in which the agent is operating. An agent's designer might have one reasonable way of assigning worth in one environment, and use a completely different method in a different environment. Below, we examine a few of the possibilities that might be used by that designer.

8.1 Subgoal Set

Above, we mentioned one type of worth function, that which assigns some value to each of several distinct subgoals that an agent wants to achieve. States in which only some of the subgoals are achieved are rated lower than states in which they are all achieved.

Examples of domains in which this type of worth function is appropriate include those in which an agent has a collection of independent subgoals to achieve. The agent's degree of satisfaction is monotonically related to the set of subgoals satisfied—the more subgoals achieved, the better off he is. For example, an electronic mail server has a set of messages to forward, and might assign varying value to different messages. The total worth of any state is the sum of the values of messages that have been successfully delivered. The individual subgoals are decoupled. An agent is happy with each separate subgoal's completion, independent of other subgoals' completion.

8.1.1 The Multiagent Tileworld

Let's examine another domain, the multiagent version of the Tileworld [23] (see Figure 12).[3]

Figure 12

A single hole in the grid is represented by a set of contiguous grid squares, marked by repetitions of the same number. The number represents the value of filling in the entire set

[3]The use of simplified domains like the Tileworld or the Blocks World has sometimes been criticized because these domains do not deal with the genuine issues involved in building robots in the real world. However, we are *not* using these domains in order to understand robot action. Instead, we use these domains with their original intent—to explore abstract issues of goal interaction. These abstract issues will arise, for example, when software agents [12], sharing limited resources (such as memory, communication lines, printers), need to coordinate. The management of these resources among software agents maps well to artificial domains such as the Tileworld, Blocks World, Postmen Domain, and so on. The problems of perception and action, which make (for example) the Blocks World inappropriate for planning research in robotics, are not directly relevant to our research agenda.

of squares, i.e., the hole's worth. In principle, different agents might assign different worth to filling the same hole. If the two agents do associate different worth to the same hole, two numbers appear in the grid square (e.g., the 3 and 4 in the sample figure; the lower left number denotes lower left agent's worth, so 3 is A_1's worth). Tiles are represented by black squares (■) inside the grid squares. Obstacles are represented by thick black lines (▌).

Agents can move from one grid square to another horizontally or vertically (unless the square is occupied by a hole or an obstacle—multiple agents can be in the same grid square at the same time). While moving, the agent can push a tile or a chain of tiles in the same direction, unless they would then run into an agent or an obstacle. When a tile is pushed into any grid square that is part of a hole, the square is filled and becomes navigable as if it were a regular grid square. The domain is static except for changes brought about by the agents. When there is a potential collision between two objects that are attempting to move into the same grid square at the same time (agent and block, block and block), neither motion can be performed.

In the original Tileworld, there was only one agent. We are interested in the case where two or more agents are pursuing their goals concurrently within the same grid. For the sake of simplicity, we maintain symmetry and assume that the two agents A_1 and A_2 start at the same location (e.g., the upper left-hand grid square). Thus both agents are able to carry out a given plan at equivalent cost. The cost of moving from one square to another is 1, whether or not the agent is pushing a tile or chain of tiles.

In this domain, an agent's goal is a subset of holes that he wants to see filled. Each hole has an associated worth for an agent, which is the value that agent associates with having the hole filled (regardless of who did the work). The worth of a world state for an agent is the sum of the worths for that agent of the holes that are filled in that state. This is an example of a subgoal set worth function. The utility for an agent of bringing the world to some specific state is, as above, the worth of that state for him minus the cost of the work he had to perform to reach that state. Each additional hole that gets filled, from among the holes he wants to have filled, independently contributes to the agent's worth.

Consider a Tileworld encounter whose initial state is shown in Figure 13. We will use the notation (x,y) to refer to specific grid squares, with the bottom left square having coordinates (1,1), as shown in the figure.

Figure 13

Note that, since there are three holes and only two tiles, not all goals can be achieved.

The agents appear to be in a conflict situation. One will have both of his goals achieved, while the other will have only one of his goals achieved. Note, also, that our agents are not assumed to be "reactive" (in the artificial intelligence sense), the agent architecture normally associated with the Tileworld domain.

If A_1 were alone in the world, he could fill the (5,8) hole with cost 10;[4] since it has worth 15, he gets a utility of 5. Filling just the (8,9) hole would cost him 12, with a negative utility of -3. If A_1 were alone in the world, he could fill *both* the (5,8) and the (8,9) holes at a cost of 16;[5] since together the goals' worth is 24, he would get a utility of 8. Thus A_1 (alone) would choose to fill both his holes ($U_1 = 8$).

If A_2 were alone in the world, he could also fill the (5,8) hole with cost 10 (using the same plan as A), and also getting a utility of 5 ($U_2 = 5$). If A_2 were alone in the world, he could fill his (8,7) hole at a cost of 6;[6] with worth of 9, his utility is 3. Alone in the world, A_2 can achieve his two subgoals at a cost of 22;[7] their combined worth is 24, leaving him with utility of 2. Thus, A_2 (alone) would prefer to fill only the (5,8) hole, and get a utility of 5; he has no motivation to fill the other hole.

The best two-agent plan to achieve A_1's goals has a cost of 13 (agent A_1 does 8 while agent A_2 does 5).[8] The accomplishment of these two goals has a worth of 24 for A_1, and a worth of 15 for A_2. A_1 would get 16 points of utility while A_2 would get 10 points of utility—both agents would benefit from this deal.

The best two-agent plan to achieve A_2's goals also has a cost of 13 (agent A_1 does 4 while agent A_2 does 9).[9] The accomplishment of these two goals has a worth of 24 for A_2, and a worth of 15 for A_1. A_2 would get 15 points of utility, while A_1 would get 11 points of utility—both agents would also benefit from this deal.

Negotiating over mixed joint plans they would agree on the latter of the above joint plans (since the product of the agents' utilities is larger, $165 = 11 * 15 > 16 * 10 = 160$). However, negotiating over multi-plan deals actually allows them to "split" the expected utility equally, getting 13 each. The multi-plan deal that they will agree on assigns a probability of $\frac{2}{5}$ that A_1's preferred two-agent plan will be carried out, and a probability of $\frac{3}{5}$ that A_2's preferred two-agent plan will be carried out. This results in an expected utility of 13 for each of the agents $(((\frac{2}{5}) * 10) + ((\frac{3}{5}) * 15) = ((\frac{2}{5}) * 16) + ((\frac{3}{5}) * 11) = 13)$.

Note that the situation, where the agents appeared to be competing over resources, actually ended up being highly cooperative. Both agents benefit by having the other agent around, even if they lose the coin toss. In the worst case, A_1 would get 11 (instead of $U_1 = 8$), and A_2 would get 10 (instead of $U_2 = 5$).

[4](7,6);(7,7);(6,7);(5,7);(5,6);(4,6);(4,7);(3,7);(3,8);(4,8)

[5](7,6); (7,7); (7,8); (7,7); (6,7); (5,7); (5,6); (4,6); (4,7); (3,7); (3,8); (4,8); (4,9); (5,9); (6,9); (7,9)

[6](6,5); (5,5); (5,6); (5,7); (6,7); (7,7)

[7](7,6); (7,7); (6,7); (5,7); (5,6); (4,6); (4,7); (3,7); (3,8); (4,8); (5,8); (5,9); (6,9); (7,9); (7,8); (7,9); (6,9); (5,9); (5,8); (5,7); (6,7); (7,7)

[8]A_2's role=(7,6); (7,7); (6,7); (7,7); (7,8); A_1's role=(6,5); (5,5); (5,6); (5,7); (5,8); (5,9); (6,9); (7,9)

[9]A_1's initial role=(7,6); (7,7); (6,7); A_2's role=(6,5); (5,5); (5,6); (5,7); (5,8); (5,9); (6,9); (7,9); (7,8); A_1's final role=(7,7)

8.2 Distance Between States

It is not always the case that an agent has independent subgoals that he is trying to achieve. Sometimes, it is more reasonable to model the situation differently. An agent might want to reach a certain ideal target state, but be partially satisfied the closer he can get to that state. The question is, closer in what sense? There may exist a reasonable distance measure over world states, and a correspondence between this measure and the degree of satisfaction that an agent assigns to an arbitrary state. In other words, the "closer" that a state is to the ideal state, using this measure, the greater the amount of the goal that has been achieved. If a state is farther from the ideal state, less of the goal has been achieved.

Imagine that you are to attend a concert in Central Park. Since you want to really feel like you're up there on stage with the performers, you would ideally like to be in the first row of the audience, in the center. The further your seat is from that location, the less your seat is worth to you—and you will be willing to pay less to get it. The distance of your seat from front row, center, might be an appropriate base for establishing your worth function over all possible final states. This distance is a natural way of comparing final states in this situation.

As another example, imagine that you are assembling a new PC to be used in your job. Your ideal machine might have a very fast processor with a large amount of RAM, a large and fast hard disk, a cache, and a floating point co-processor. Other configurations will be inferior, but how do you compare a machine without a floating point co-processor to one with a co-processor but without a cache? And how do either of these compare to your ideal? The way to solve this evaluation problem is to impose some distance measure over the possible configurations. For example, we could use some standard benchmark to quantify the performance of any given PC. The ideal configuration produces a particular performance benchmark. Other, less ideal, configurations might be measured by their deviation from that performance benchmark. The closer the benchmark is to the ideal, the higher the machine's worth to you. The further the benchmark, the less the machine is worth.

The above examples give an intuitive sense of what we mean by evaluating worth using a distance measure between states. Below, we formalize the intuitions a bit and present two examples showing different approaches to the measurement of inter-state distance.

8.2.1 The One-Agent Plan Distance Measure

One useful prototypical distance measure between states is the cost of the one-agent plan that moves the world from one to the other [10].

Definition 7 *The distance between two world states $s, f \in \mathcal{S}$ will be marked as $d(s, f)$, and will be defined as $c(s \to f)$.*

This definition can be thought of (informally) as a *metric* over $\mathcal{S}$; it is reflexive and satisfies the triangle inequality.

- *Reflexivity* — $\forall t \in \mathcal{S}\colon d(t, t) = 0$.

- *Triangle inequality* — $\forall s, t, f \in \mathcal{S}$: $d(s,t) + d(t,f) \geq d(s,f)$.

This distance measure, however, is not necessarily symmetric:

- *Symmetric* — $\forall t, f \in \mathcal{S}$ $d(t,f) = d(f,t)$. This property of symmetry should strictly be true when talking about metrics, but in our domains it is not always true—nor do we need it to be true.

Using this distance measure, we can measure the worth of any particular state relative to the worth of some state that fully satisfies some particular goal g. Assuming that the full achievement of g has an associated worth of w, we can define the worth function over a state t to be

$$W(t) = w - \min_{f \models g}(d(t,f)),$$

i.e., it is the difference between the worth of the goal and the distance from t to the closest state that achieves it. For further discussion about alternative worth functions of this type, see [11].

8.2.2 The Delivery Problem

To illustrate the use of the above definition of worth, we define the Delivery Domain with Bounded Storage Space. Delivery agents are using trucks to move containers between warehouses. To do the deliveries, agents can rent trucks, an unlimited supply of which are available for rental at every node. A truck can carry a limited number of containers. Each warehouse also has a limited capacity. The operations that can be done in this domain are Load and Unload (each costs 1), and Drive (which costs 1 per each distance unit). There is another activity that can occur: consumers come from time to time to take containers from the warehouses.

Agents can cooperate by using the same truck for delivery of both agents' containers. The only conflict that can occur is when the two agents need to deliver containers to the same warehouse, and there is not enough space for all the containers. In this case, one or both agents would not be able to fully achieve his goal. The goal may be completed later when some containers have been "consumed," i.e., removed from the warehouse by a consumer.

The one-agent plan metric is a reasonable heuristic measure to use when an agent whose goal has *not* been fully achieved can later achieve his whole goal. The one-agent distance as defined in this domain, for example, is an upper bound on how much the agent will need to spend to achieve his goal, when (or if) that achievement becomes possible.

A Simple Delivery Problem:

Consider the graph in Figure 14. The weight (length) of each edge is written beside it. There is place for only 2 more containers in warehouses y and z. Truck capacity is equal to 4.

Figure 14

A_1's goal is to deliver two containers C_1 and C_2 from x to z. A_2's goal is to deliver container C_3 from x to y, and C_4 from x to z. Each agent assigns a worth of 50 to the full achievement of his goal.

If A_1 were alone in the world, he could achieve his goal by renting a truck at warehouse x to take his two containers to z. The best one-agent plan to do this is to load the two containers (with cost of 2), drive them to z (with cost of 40), then unload them both at z (with cost of 2, and worth of 50). There is exactly enough room at z to accommodate the two containers. This plan gives A_1 a utility of $U_1 = 6$ (i.e., $50 - (1 + 1 + 40 + 1 + 1)$).

If A_2 were alone in the world he could achieve his goal by renting a truck at warehouse x to take his two containers to y and z. The best one-agent plan to do this gives him a utility of $U_2 = 6$ also (i.e., $50 - (1 + 1 + 30 + 1 + 10 + 1)$).

The two goals cannot both be fully satisfied, because there isn't room for 3 additional containers in warehouse z. However, the agents can cooperate by using only one truck to make a partial delivery of the 4 containers. From one point of view, the agents are in conflict. However, from another point of view, they have a clear opportunity to benefit from cooperation. In this way, it is similar to the Tileworld example above in Section 8.1.1.

The agents will agree to rent only one truck at x which will be loaded with the 4 containers—each will rent the truck with probability 0.5. The truck will unload container C_3 at y. C_4 or C_1 (each with probability 0.5) will also be unloaded at y. The truck will continue to z where the rest of the containers will be unloaded. Each agent has a 0.5 chance of fully achieving his goal. In both cases each agent is expected to pay half of the cost of the whole journey. If the agent's goal is only partially achieved, he will be at a "distance" of 12 from the achievement of his whole goal (one Load, a drive of length 10, and one Unload). The Utility for both agents from this deal is: $\text{Utility}_i = 20$ (i.e., $0.5 \times (50 - 12) + 0.5 \times 50 - 0.5 \times 40 - 4$). Both agents have benefited from cooperating, even though one agent has not fully achieved his goal.

8.3 Probabilistic Distance

Consider two chip manufacturers who are interested in high-yield production of integrated circuits. They know that if they purify the air in their factories, accurately measure kiln temperatures, etc., they will increase the likelihood of their producing working chips. Since there are a limited number of air filters, accurate thermometers, and so on, they must negotiate over who gets how many of each.

Having these resources is not the agents' goal; production of working chips is the goal. The agents, however, cannot directly negotiate over the final states where each has a certain number of working chips (reaching such a given final state cannot, in any case, be guaranteed due to uncertainty in the production process). They are constrained to negotiate over intermediate states that can be deterministically generated. Nevertheless, there is a probabilistic relationship between the resource allocation in this intermediate state and the final (non-deterministic) state. Given a resource allocation configuration, an agent can estimate the distribution of possible final states. Using this distribution, he can evaluate the worth of a given resource configuration, and it is this resource configuration that is negotiable.

This bears a resemblance to the PC assembly example above. We impose a distance measure over possible configurations, and evaluate how good a state is by how far it is from the ideal state.

Some negotiation domains, as above, consist of such intermediate states that can be reached deterministically, and final states that are then reached nondeterministically. Agents will be negotiating over the intermediate states in the domain. They can evaluate the worth of these intermediate states by considering the "probabilistic distance" between intermediate and final states, and those final states' worth values.

Definition 8 *Given a set of final states F, and a worth function that is defined for all members of F, we define the worth of an intermediate state $t \notin F$ to be*

$$W(t) = \sum_{f \in F} p(f|t)W(f),$$

where $p(f|t)$ is the probability of reaching final state f given that we are in intermediate state t. That is, the worth of t is the weighted average of the worths of the final states likely to result.

The above definition does not take into account the cost to the agent of the non-deterministic process, which could be subtracted from this worth.

9 Conclusion

We have presented an approach to the negotiation problem in non-cooperative domains where agents' desires are encapsulated in worth functions. These worth functions serve in place of the more traditional notion of goal. Agreements may lead to a situation in which agents compromise, and do not reach their most desired state. The negotiation protocol that was presented (using multi-plan deals) has agents compromise over their degree of satisfaction, and (in parallel) negotiate over the joint plan that will be implemented to reach the compromise final state.

These two aspects of negotiation, what final state will be reached and how it will be reached, are both captured in the multi-plan deal formalism. Moreover, the multi-plan deal makes these two aspects comparable, in that an agent can trade off one for the other. An agent, for example, could offer to do more work if a state more to his liking will result.

Domain-dependent approaches can be used in defining the worth function. We here presented several different definitions of the worth function, including one for a domain in which a goal can be a set of independent subgoals, one in which a conflict can be later resolved (and distance between world states is thus an appropriate measure), and one in which a nondeterministic process follows a deterministic setup phase (and the negotiation is over the intermediate states at the end of the setup phase).

Acknowledgments

This research has been partially supported by the Leibniz Center for Research in Computer Science at the Hebrew University of Jerusalem, by the Israeli Ministry of Science and Technology (Grant 032-8284) and by the Israel Science Foundation (Grant 032-7517). We would like to thank Barbara Grosz for her comments on an earlier version of this paper, and the anonymous reviewers who contributed to its improvement.

References

[1] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, Inc., New York, 1984.

[2] Daniel D. Corkill. *A Framework for Organizational Self-Design in Distributed Problem-Solving Networks.* PhD thesis, University of Massachusetts, Amherst, MA, 1982.

[3] Daniel D. Corkill and Victor R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 748–756, Karlsruhe, West Germany, August 1983.

[4] Keith S. Decker and Victor R. Lesser. An approach to analyzing the need for meta-level communication. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 360–366, Chambery, France, August 1993.

[5] Keith S. Decker and Victor R. Lesser. A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 210–216, Washington, DC, July 1993.

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

[7] Edmund H. Durfee and Thomas A. Montgomery. A hierarchical protocol for coordinating multiagent behaviors. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 86–93, Boston, Massachusetts, July 1990.

[8] Edmund H. Durfee and Jeffrey S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence*, pages 94–104, Seattle, Washington, July 1994.

[9] Eithan Ephrati and Jeffrey S. Rosenschein. The Clarke Tax as a consensus mechanism among automated agents. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 173–178, Anaheim, California, July 1991.

[10] Eithan Ephrati and Jeffrey S. Rosenschein. Constrained intelligent action: Planning under the influence of a master agent. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 263–268, San Jose, California, July 1992.

[11] Eithan Ephrati and Jeffrey S. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 423–429, Chambery, France, August 1993.

[12] Oren Etzioni, Neal Lesh, and Richard Segal. Building softbots for UNIX (preliminary report). Technical Report 93-09-01, University of Washington, 1993.

[13] Jacques Ferber and Eric Jacopin. The framework of eco-problem solving. In *Decentralized A.I. 2*, pages 181–193. Elsevier/North-Holland, Amsterdam, 1991.

[14] Piotr J. Gmytrasiewicz, Edmund H. Durfee, and David K. Wehe. The utility of communication in coordinating intelligent agents. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 166–172, Anaheim, California, July 1991.

[15] Sarit Kraus. Agents contracting tasks in non-collaborative environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 243–248, 1993.

[16] Sarit Kraus and Jonathan Wilkenfeld. The function of time in cooperative negotiations. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 179–184, Anaheim, California, July 1991.

[17] Brigitte Lâasri, Hassan Lâasri, and Victor R. Lesser. Negotiation and its role in cooperative distributed problem problem solving. In *Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence*, Bandera, Texas, October 1990. Chapter 9.

[18] Victor R. Lesser and Daniel D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, Fall 1983.

[19] Hector J. Levesque, Philip R. Cohen, and José H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 94–99, Boston, July 1990.

[20] Ran Levy and Jeffrey S. Rosenschein. A game theoretic approach to the pursuit problem. In *Decentralized A.I. 3*, pages 129–146. Elsevier/North-Holland, Amsterdam, 1992.

[21] R. Duncan Luce and Howard Raiffa. *Games and Decisions*. John Wiley & Sons, Inc., New York, 1957.

[22] John F. Nash. The bargaining problem. *Econometrica*, 28:155–162, 1950.

[23] Martha E. Pollack and Marc Ringuette. Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings of The National Conference on Artificial Intelligence*, pages 183–189, Boston, Massachusetts, August 1990.

[24] Jeffrey S. Rosenschein. Consenting agents: Negotiation mechanisms for multi-agent systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 792–799, Chambery, France, August 1993.

[25] Jeffrey S. Rosenschein and Michael R. Genesereth. Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 91–99, Los Angeles, California, August 1985.

[26] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, Massachusetts, 1994.

[27] Reid G. Smith. *A Framework for Problem Solving in a Distributed Processing Environment*. PhD thesis, Stanford University, 1978.

[28] Katia P. Sycara. Resolving goal conflicts via negotiation. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 245–250, St. Paul, Minnesota, August 1988.

[29] Katia P. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 517–523, Detroit, Michigan, August 1989.

[30] Gilad Zlotkin and Jeffrey S. Rosenschein. Mechanism design for automated negotiation, and its application to task oriented domains. *Artificial Intelligence*. To appear.

[31] Gilad Zlotkin and Jeffrey S. Rosenschein. Mechanisms for automated negotiation in state oriented domains. *Journal of Artificial Intelligence Research*. To appear.

[32] Gilad Zlotkin and Jeffrey S. Rosenschein. Negotiation and task sharing among autonomous agents in cooperative domains. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 912–917, Detroit, Michigan, August 1989.

[33] Gilad Zlotkin and Jeffrey S. Rosenschein. Negotiation and conflict resolution in non-cooperative domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 100–105, Boston, Massachusetts, August 1990.

[34] Gilad Zlotkin and Jeffrey S. Rosenschein. Cooperation and conflict resolution via negotiation among autonomous agents in non-cooperative domains. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1317–1324, December 1991.

[35] Gilad Zlotkin and Jeffrey S. Rosenschein. Incomplete information and deception in multi-agent negotiation. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 225–231, Sydney, Australia, August 1991.

[36] Gilad Zlotkin and Jeffrey S. Rosenschein. Negotiation and goal relaxation. In Y. Demazeau and J. P. Müller, editors, *Decentralized A. I. 2*, pages 273–286. Elsevier/North-Holland, Amsterdam, 1991.

[37] Gilad Zlotkin and Jeffrey S. Rosenschein. A domain theory for task oriented negotiation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 416–422, Chambery, France, August 1993.

List of Figures

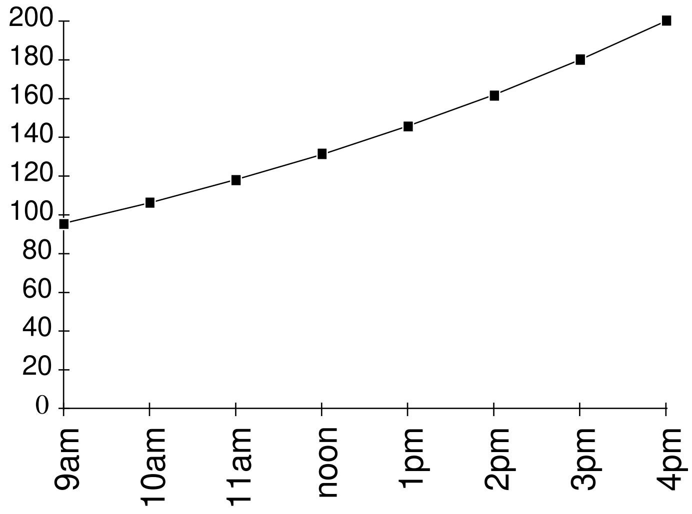

Figure 1: Agent A_1's Worth Function for a Meeting

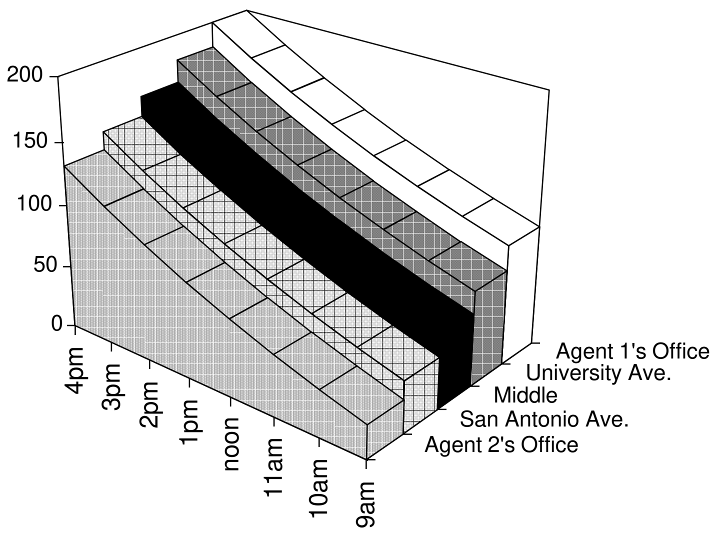

Figure 2: Agent A_1's Utility Function for All Possible Deals

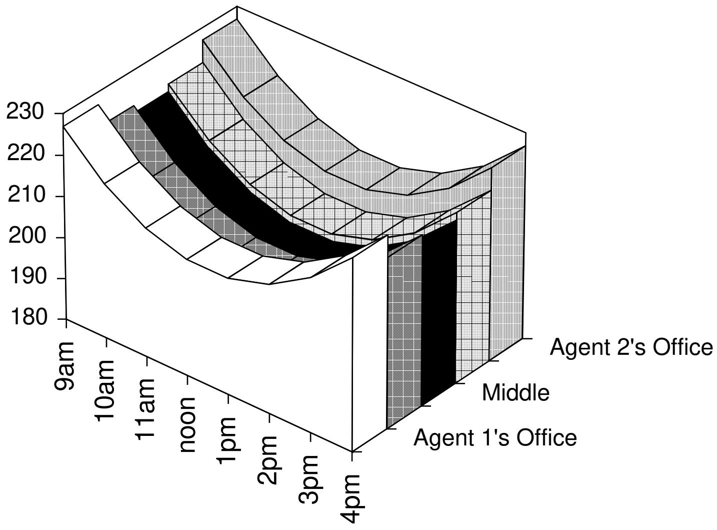

Figure 3: The Sum of the Agents' Utilities

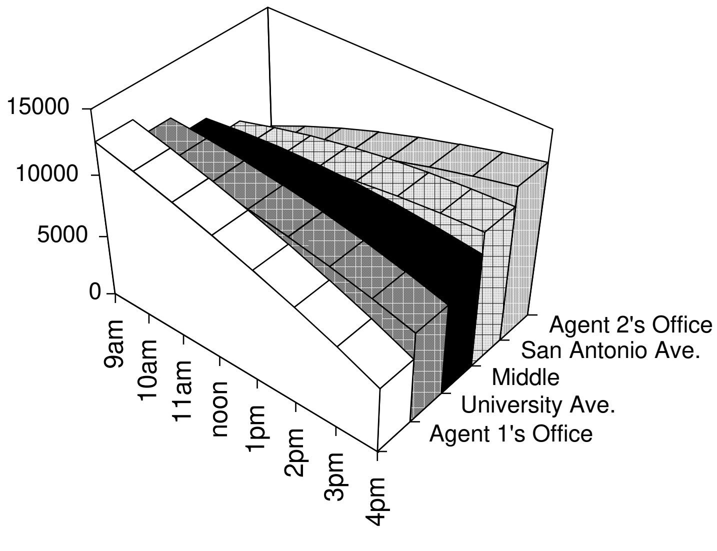

Figure 4: The Product of the Agents' Utilities

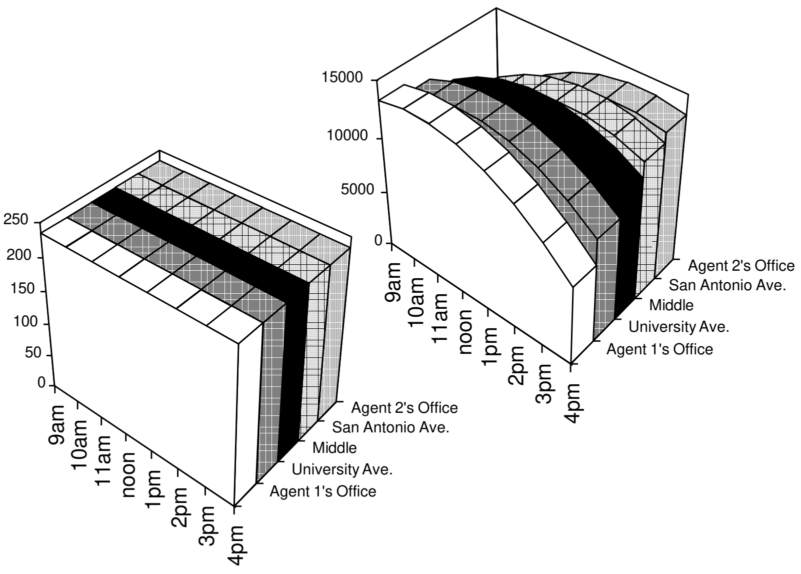

Figure 5: Sum and Product of Utilities with Linear Worth and Cost Functions

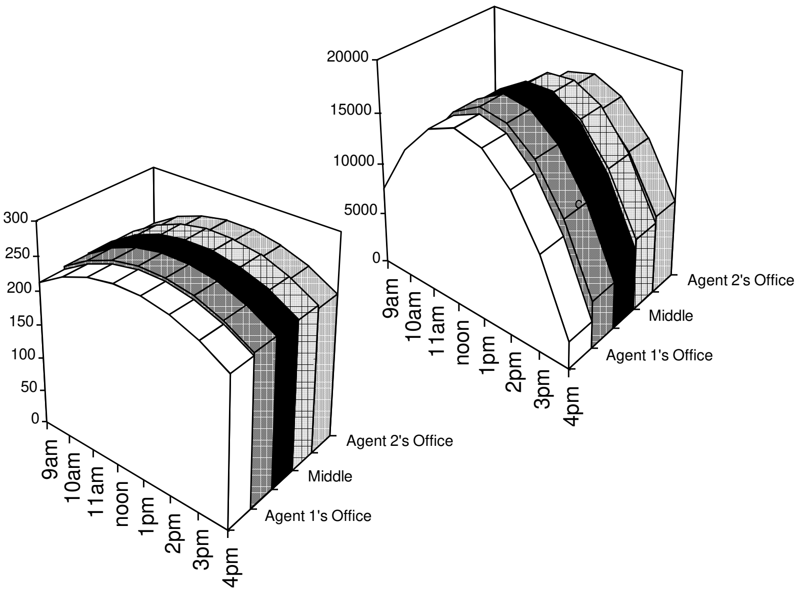

Figure 6: Sum and Product of Utilities with Exponential Worth and Cost Functions

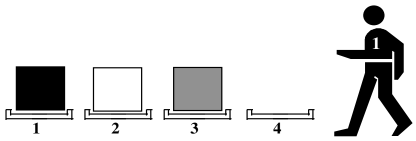

Figure 7: Initial State of the World, One Agent

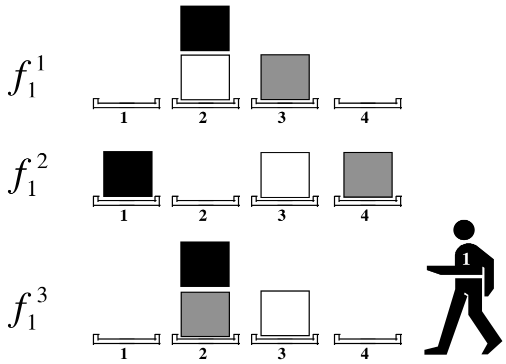

Figure 8: Possible Final States

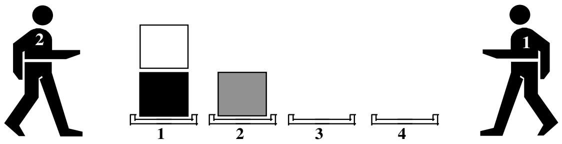

Figure 9: Initial State of the World, Two Agents

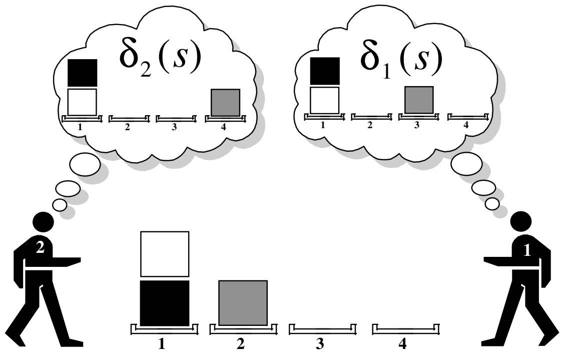

Figure 10: Possible Final States, Two Agents

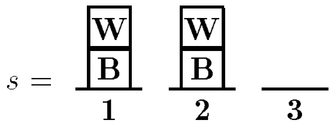

Figure 11: Relationship of the Multi-Plan Deal Type to Mixed Joint Plans

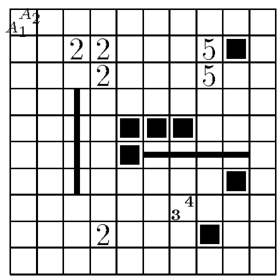

Figure 12: The Multiagent Tileworld

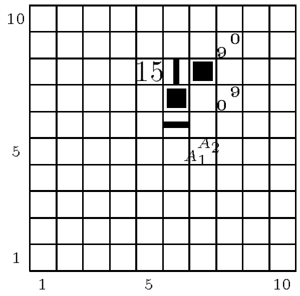

Figure 13: Initial State

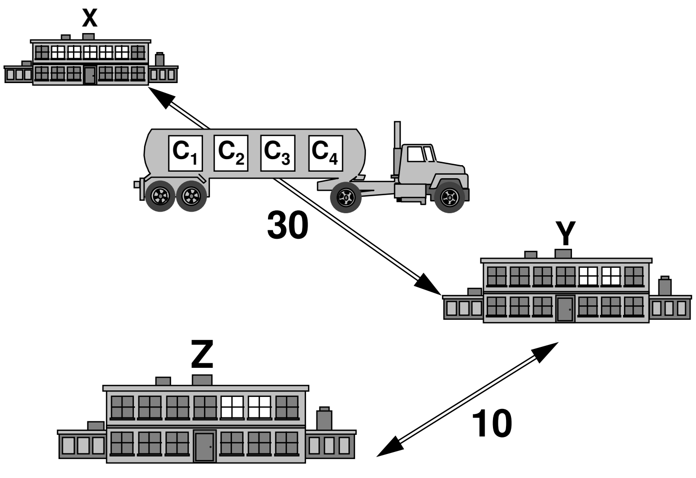

Figure 14: Simple Delivery Problem