Finding Efficient Schedules in Multi-Entity Negotiation*

Mohamed A. Soliman and Ahmed A. Soror

University of Waterloo
ON N2L 3G1
Waterloo
Canada
{m2ali,aakssoro}@cs.uwaterloo.ca

Abstract

Efficient scheduling for potential negotiations that could be initiated among interacting business parties is crucial to satisfy the embedded time or budget constraints that parties might have. An efficient schedule should guarantee the maximum possible utility to all parties taking into account the correlations or inter-dependencies that exist among different negotiable entities. In this project we address the problem of scheduling negotiation over multiple entities in both deterministic and probabilistic settings. We study the problems involved in these settings from both data management and game theory perspectives, and propose a set of heuristic measures and search algorithms to solve the problem in different scenarios. We validate our proposed methods by an extensive experimental study performed on synthesized data sets generated from GAMUT[9] test suite.

1 Introduction

Large-scale B2B transactions usually involve complex and lengthy negotiation among interacting parties to trade a large set of objects or services. Agreements that enable transactions to proceed are usually reached by negotiating over a set of underlying entities. For example implementing a transaction that involves buying a simple product might require reaching an agreement for price, delivery schedule, payment method, additional discounted offers, etc. The negotiable entities might be of different importance with respect to the two parties. For example *Delivery Schedule* might be more beneficial to the seller; while *Payment Method* might be more important to the buyer.

In complex business transactions, it is usually infeasible to go through negotiation over all entities due to different constraints that interacting parties might have. For example it is possible that both parties are required to implement the transaction before some deadline otherwise it would be less beneficial. There might be also a pre-defined amount of money that each party can invest in the transaction. Based on their buget/time constraints, the interacting parties need to restrict negotiation to the most beneficial entities that guarantee the maximal gain according to their constraints.

For example consider Figure 1. It illustrates a possible set of negotiable entities and the perceived utilities of the two interacting parties under different strategies. The indicated payoffs could indicate monetary profit, satisfaction level, convenience, etc. Negotiating over some of the entities is more profitable, to both parties, than others. For example negotiating over *Payments* could yield a payoff pair of (25,25); while negotiating over *Additional Offers* would only result in a maximum payoff of 20 to one of the two parties. If the two parties, for example, are allowed to negotiate over one entity only (because of their possible time limits), they might find negotiating over *Payments* is potentially more beneficial than *Additional Offers*.

Another observation is that the outcome of some negotiations might affect the strategies of both parties for the remaining negotiations. For example adopting the 'Discount' strategy by party 1 while negotiating over *Payments* makes adopting the 'Bonus Offer1' strategy while negotiation over *Additional Offers* relatively unprofitable.

In practical 'online' settings, an overwhelming number of possible interactions (negotiations) could be initiated among business parties. Moreover, the inter-dependencies among negotiable entities could redefine the negotiation space multiple times. The exponential number of possible negotiation orders (all possible permutations), and their inter-dependencies make finding an efficient 'schedule' to conduct negotiation a challenging task.

---

*CS 886 Advanced Topics in AI - MultiAgent System
Winter 2006

Prof. Kate Larson

Party1 \ Party2	One Payment	Installments	Money Order	Bank Transfer
Discount	25,25	2,30	–	–
Coupon	30,2	5,5	–	–
Fast delivery	–	–	20,20	5,30
Slow delivery	–	–	30,5	10,10
Bonus Offer1	–	–	15,15	1,20
Bonus Offer2	–	–	20,1	5,5

Delivery Schedule

Payments

Additional Offers

Figure 1: Possible negotiable entities and their utilities

In this project we address the problem of scheduling negotiations over a huge number of entities in a multi-agent system. We study how to reduce the large negotiation space, by following an efficient negotiation schedule, *with the objective of finding the non-dominated or most beneficial negotiation outcomes with respect to both interacting parties*. The essence of our proposed techniques is to push potentially beneficial negotiations ahead in the schedule such that agents have the chance to get the maximum benefits as early as possible. The intuition behind this objective is that as agents move through the schedule, they might decide to quit negotiation at some point because of their time/budget constraints. An 'efficient' schedule must therefore allow the agents to get the most benefits before they decide to cancel remaining negotiations; or accept their predefined default outcomes.

Unfortunately, discovering the exact outcome of each negotiation is not possible unless that negotiation is actually conducted. Since we aim at providing a schedule for agents to follow, asking agents to conduct negotiations and monitor their outcomes is meaningless. We therefore design search techniques that use heuristics to estimate the goodness of each negotiation without actually conducting that negotiation. The output of these search techniques is a negotiation schedule of maximal 'estimated' profit that acts as a guide for agents to follow. Since there might be multiple possible schedules that all lead to the same set of non-dominated or beneficial outcomes, we emphasize two key properties in the negotiation schedules we are after:

- *Minimality*: The negotiating parties would not be interested in long negotiation schedules because of their time and/or budget constraints. A key requirement for a schedule is to guarantee the maximum possible profit in the minimum possible number of steps.

- *Equilibrium*: A negotiation schedule should lead the interaction space to equilibrium; i.e. adopting the schedule should reduce the interaction space into a smaller space in which going through any of the interactions (negotiations) would not change the set of non-dominated or most beneficial outcomes. In other words, none of the negotiating parties would prefer to *deviate* from the generated schedule.

We model the negotiations that could be initiated among the agents based on the principles of Game Theory. The adopted game theory models capture strategic behavior of agents in different settings during the negotiation process. In other related work [6, 7] negotiations are modeled as 'offer, counter-offer' interactions with possible time and budget constraints. However, we believe that this model does not clearly capture the essence of strategic negotiation since it does not allow encoding different players strategies as in, for example, Nash bargaining model [1].

Formulating negotiations as games depends on the structure of utility matrix; and how players agree about utilities. Different approaches might be adopted when revealing utilities for each negotiable entity. We distinguish two different settings. In a *cooperative* business environment parties might be willing to reveal their true utilities for each action taken while negotiating over each object. On the other hand in a *competitive* environment parties might be more self-interested or strategic such that they reveal fuzzy/probabilistic representations of their utilities for the purpose of confusing the opponent. Selecting appropriate game theory model to adopt in each case affects the way negotiations are formulated, how they are correlated, and how negotiation schedule is generated.

We assume the existence of an *intermediate party*, or a *mediator* where agents submit their strategies and utilities (either explicit or probabilistic) for each negotiable entity. The intermediate party is responsible of coordinating and directing the negotiation with the goal of maximizing players utilities. Under these settings we focus on two-players games with two variants:

- *Cooperative Games*: The intermediate party adopts Nash bargaining model [1]. Explicit utility information is provided by the interacting parties for each negotiable object. This utility information is used to arbitrate the cooperation between agents in order to maximize their corresponding profits.

- *Bayesian Games*: Agents obscure their utilities in the form of probabilistic distributions (that might be already captured by the mediator based on agents previous interactions). Agents submit these fuzzy versions of utility to the intermediate party that has to figure out how to devise an efficient schedule based on such *incomplete* information.

Previous work [5] described a solution for the problem based on the first variant (cooperative games). We present this solution in Section 4.2, and further extend it by our implementation and experimental analysis (described in Sections 6, 7). In this extension we consider the combined complexity of solving games based on Nash bargaining model (the internal arbitration mechanism that solves each cooperative game) and the search techniques; rather than only the search complexity as in [5]. Moreover, we study the second variant (Bayesian games); which applies to a more generic setting. We present a new probabilistic model capturing probabilistic negotiation schedules, and design search techniques based on such model. The main challenges that we tackle in this project are the following:

- Adopting Game Theory principles in negotiation environment. This adoption comes in two different forms: (1) cooperative negotiation where agents reveal their true utilities to a trusted intermediate party, and (2) competitive negotiation where agents are more strategic such that they reveal only a fuzzy version of their utilities.
- Analyzing the complexity of solution procedures of a large number of inter-dependent games in both cooperative and competitive settings.
- Studying the potential effects of negotiation interdependencies in terms of eliminating or redefining players strategies.
- Designing efficient algorithms to generate a schedule for the negotiations to yield the non-dominated or most beneficial outcomes.

2 Background

In this section we present necessary background from theoretical game models that we adopt in our proposed techniques. Section 2.1 presents the principles of Nash bargaining model and cooperative games; where utilities are explicitly available. Section 2.2 discusses the basic definition of Bayesian games; where utilities are given in the form of probabilistic distributions.

2.1 Cooperative Games

The material presented in this section is based on the text [3]. A cooperative game is one in which players are able to make enforceable contracts. Communication of the players is allowed to enable binding agreements to be made. This requires some outside mechanism to enforce the agreements. With the extra freedom to make enforceable binding agreements, the players can generally do much better. For example in the classical prisoners dilemma, the only Nash equilibrium is for both players to defect. In the cooperative theory, they can reach a binding agreement to both use the cooperate strategy, and both players will be better off. Cooperative game theory is divided into two classes of problems depending on whether there is a mechanism to transfer utilities from one player to another, or not. We focus here on cooperative games with nontransferable utility.

When one player with m strategies cooperates with another player with and n strategies in a bimatrix game, they may agree to achieve a payoff vector of any of the $m \times n$ pairs. They might also agree on mixed strategies on these pairs. The set of all such payoff pairs is the convex hull of the $m \times n$ pairs.

Definition 1 ***Feasible Set of Cooperative Game*** *The set of feasible solutions of a cooperative game with an $m \times n$ utility matrix is the convex hull of the $m \times n$ utility pair.*

Normally in a cooperative game, we assume there is a period of *pre-play negotiation* where players agree on the probabilities of choosing each possible joint strategy. If an agreement is reached, no player can be made better off without making the other player worse off. Such an outcome is said to be Pareto optimal. On the other hand, if an agreement is not reached, each player can *threat* with a strategy that is bad to other player. This designates a *threat pair* inside the feasible set representing game solution in case of disagreement. Based on threat point, each player declares that he will accept no less than some minimum profit since this profit can be achieved without agreement. The threat point (u_0, v_0) is usually taken as the maximin utility pair since this is the pessimistic expected payoff pair. Finding the maximin pair for some game requires solving a linear program over the game feasible set.

Nash bargaining model [1, 2] is based on the two elements described above, namely *feasible solution set* and *threat point*. Given a feasible set S and a threat point (u_0, v_0), the space of possible game solutions can be truncated to the region enclosed between (u_0, v_0) and maximum utility point. The problem now is to decide on a feasible outcome vector reflecting the value of the game. Nash axioms define reasonable properties for the outcome vector (u^*, v^*):

- **Feasibility**.$(u^*, v^*) \in S$.
- **Pareto Optimality**. There is no point $(u, v) \in S$ such that $u \geq u^*$ and $v \geq v^*$.
- **Symmetry**. If S is symmetric around the line $u = v$ and if $u_0 = v_0$ then $u^* = v^*$.
- **Independence of irrelevant alternatives**. If X is a convex subset of S such that $(u^*, v^*) \in X$ and $(u_0, v_0) \in X$, then the solution of the game whole feasible set id X is also (u^*, v^*).
- **Invariance under change of location and scale**. Applying linear transformation to S reflects exactly on the solution point.

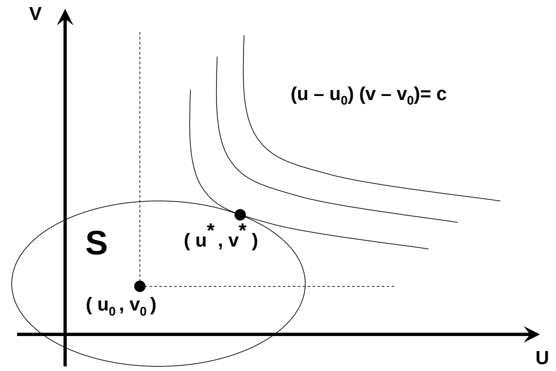

Figure 2: Cooperative Game Solution

Nash bargaining theory proved the existence of a unique solution satisfying the above axioms. Moreover, he proved that the solution is a point $(u, v) \in S$ such that $u > u_0$ and $v > v_0$, and $(u - u_0)(v - v_0)$ is maximized.

To illustrate the above principles, Figure 2 depicts an example cooperative game with an oval feasible set. The set of all rational game outcomes is the oval quarter on the top-right side of the threat point (u_0, v_0). To find a game solution pair satisfying Nash axioms, one can think of moving the curve of the parabola $(u - u_0)(v - v_0) = c$ until it tangents the boundary of the set S at the point (u^*, v^*).

2.2 Bayesian Games

The background in this section is based on [4]. Bayesian games are incomplete information games that allow players to encode their uncertainty about the game being played. The uncertainty is represented as a probability distribution over a set of possible games. However, all these possible games share the same structure in terms of number of players and strategy space of each player.

Agents have posterior beliefs about each other based on a common prior information available to all agents. These beliefs refer to what agents think about their own *types* and the opponents *types*. Each agent can have a set of possible epistemic types; where under each type the agent has a probability distribution over utilities of different strategies. This is formalized by the following definition.

Definition 2 *Bayesian Games: A Bayesian game is a tuple* (N, A, Θ, p, u) *where*

- N *is a set of agents*
- $A = (A_1, A_2 \dots A_n)$*, where* A_i *is the set of actions available to agent* i*.*
- $\Theta = (\Theta_1, \Theta_2 \dots \Theta_n)$*, where* Θ_i *is the type space of agent* i*.*

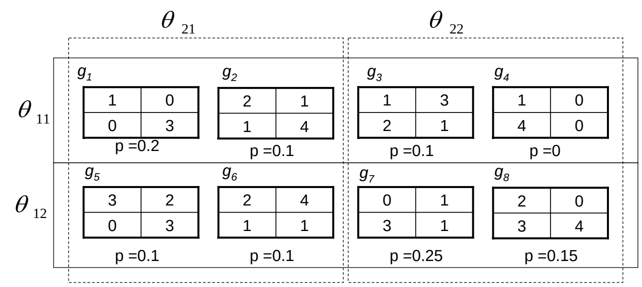

Figure 3: A Bayesian Game

- $p : \Theta \rightarrow [0, 1]$ *is the common prior information about agents types*
- $u = (u_1, u_2, \dots u_n)$*, where* $u_i : A \times \Theta \rightarrow R$ *is the utility function for agent* i*.*

The assumption is that all of the above components are common knowledge for all players and that each player knows his own type. For example consider Figure 3 that depicts an example zero-sum Bayesian game. In this example the first player has two possible types $\{\theta_{11}, \theta_{12}\}$; and the second player has also two different types $\{\theta_{21}, \theta_{22}\}$. For each type each player has a probability distribution over utilities. When the game is played, each one of the two players reveals its own type and selects strategies according to the corresponding probability distribution over utilities. The game can therefore be seen a set of possible games associated with a probabilistic distribution. The selected type pair of both players determine the utilities of different strategies in the played game. For example the utility of the first strategy of player 1, given that players types are θ_{11} and θ_{21}, can be calculated as $\frac{0.2}{0.2+0.1}(1) + \frac{0.1}{0.1+0.2}(2)$.

A Bayesian game can be solved in practice by flattening the possible games structure into an induced single normal form game based on players types. Solution concepts such as Nash equilibrium can be applied to the induced normal form game. For example consider again the game in Figure 3. Assume that the strategies of the row player are $\{U, D\}$; and those of the column player are $\{L, R\}$. Player 1 has four different strategies $\{UU, UD, DU, DD\}$; where UD, for example, means that player 1 plays U if his type is θ_{11} and plays D if his type was θ_{12}. Similarly we can define the strategies of the column player.

We can now define a normal form game based on the extended strategy space. Figure 4 depicts such a game based on the Bayesian game of Figure 3. The indicated utilities are calculated as expectations over players utilities when they adopt each pair of strategies based on their types. For example $u_1(UU, LL) = \sum_{\theta \in \Theta} p(\theta) \times u_1(U, L, \theta) = p(\theta_{11}, \theta_{21}) \times u_1(U, L, \theta_{11}, \theta_{21}) + p(\theta_{11}, \theta_{22}) \times u_1(U, L, \theta_{11}, \theta_{22}) + p(\theta_{12}, \theta_{21}) \times u_1(U, L, \theta_{12}, \theta_{21}) + p(\theta_{12}, \theta_{22}) \times$

$u_1(U, L, \theta_{12}, \theta_{22}) =$
$0.3(\frac{4}{3}) + 0.1(1) + 0.2(\frac{5}{2}) + 0.4(\frac{3}{4}) = 1.3.$

The strategy profile (UD, LR) is the Nash equilibrium for this game.

	LL	LR	RL	RR
UU	1.3	1.45	1.1	1.25
UD	1.8	1.65	1.8	1.65
DU	1.1	0.7	2.0	1.95
DD	1.5	1.15	2.8	2.35

Figure 4: Induced Normal Form for the Bayesian Game in Fig 3

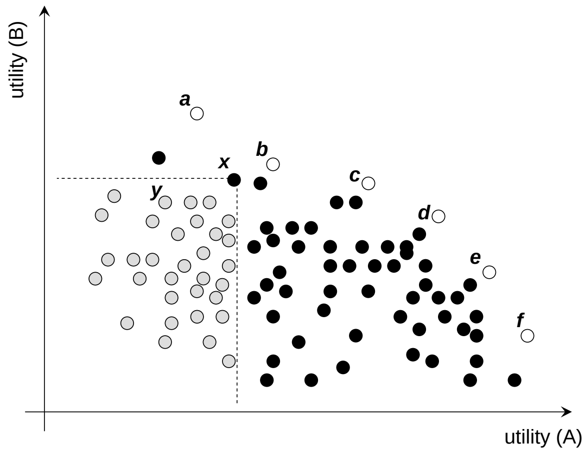

Figure 5: Non-dominated Negotiation Outcomes

3 Problem Definition

In this section we define the problem addressed by this project. In a space of huge number of potential negotiations, many of the negotiations should be avoided if we are looking for only the most profitable ones (because of time/budget constraints). Consider Figure 5 that illustrates a space of negotiation outcomes. An outcome dominates another outcome if it is better in both agents utilities. For example outcome x dominates all gray-colored outcomes such as y. The outcome set $\{a, b, c, d, e, f\}$ is called the 'non-dominated' outcome set, since none of its elements is dominated by any other outcome in the space. Reaching the non-dominated outcome set is not possible directly since we do not know the exact negotiations outcomes unless we actually conduct these negotiations. The main problem we address can be thus stated as:
Find a schedule to order negotiations among agents in order to reach the 'non-dominated' outcome set in the least number of steps avoiding going through weak negotiations.

4 Deterministic Utilities

In this section we discuss the solution of the problem in the deterministic settings where utilities are explicitly exposed. In Section 4.1, we discuss how utilities are expressed, and solutions are found based on Nash bargaining model. In Section 4.2, we describe the previous work of [5] for finding pareto-optimal solutions in electronic bargaining.

4.1 Describing deterministic utilities

The intermediate party generates negotiation schedules for agents based on the information it has regarding agents utilities and preferences. Agents have to submit descriptions for the entities they would like to negotiate over as well as their perceived utility matrices. The intermediate party uses this information to evaluate the goodness of different negotiation schedules and select the best one for the agents to implement.

Agents describe their negotiation preferences by submitting *deterministic interaction descriptors* to the intermediate party as formalized by the following definition.

Definition 3 ***Deterministic Interaction Descriptor:*** *An agent i declares willingness to negotiate with agent j over a set of entities, while exposing his explicit utilities, by providing a descriptor $< E_i, A_i, M_i >$ with the following information:*

- $E_i = \{e_1, \ldots, e_m\}$, *a set of m negotiable entities.*
- $A_i = (A_i^1, \ldots A_i^m)$, *where A_i^l is the set of available actions of agent i while negotiating over entity l.*
- $M_i = (M_i^1, \ldots, M_i^m)$, *where M_i^l is the utility matrix of agent i while negotiating over entity l.*

For example consider Figure 1. Agent 1 could submit a deterministic descriptor of the form:

- $E_1 = \{$'Delivery Schedule'$\}$.
- $A_1 = \{$'Fast Delivery', 'Slow Delivery'$\}$ Vs. $\{$'Money Order', 'Bank Transfer'$\}$.
- $M_1 = \begin{pmatrix} 20 & 5 \\ 30 & 10 \end{pmatrix}$.

This descriptor declares that agent 1 is willing to negotiate over only 'Delivery Schedule' based on two strategies $\{$Fast Delivery', 'Slow Delivery'$\}$ against the two opponent strategies $\{$Money Order', 'Bank Transfer'$\}$. Agent 1 declares explicit utilities for his strategies in the utility matrix M_1.

If agent 2 submits a similar descriptor to declare his willingness to negotiate over 'Delivery Schedule'

Agent2 / Agent1	Money Order	Bank Transfer
Fast delivery	20,20	5,30
Slow delivery	30,5	10,10

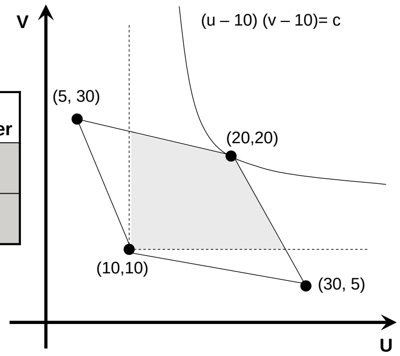

Figure 6: Solving Cooperative Games Based on Nash Bargaining Model

as well, the two descriptors can be joined to create a cooperative game with the payoff matrix depicted in Figure 6. The graph on the side illustrates the solution procedure followed by Nash bargaining model to find game outcome as outlined in Section 2. Notice that the maximin pair for this game is the utility pair (10,10). The convex feasible set can thus be safely truncated to the shaded region. The feasible point that maximize $(u-10)(v-10)$ is the utility pair $(20, 20)$; which would be the outcome of this game.

The formal description of the generated negotiation schedule is given by the following definition:

Definition 4 ***Deterministic Negotiation Schedule:*** *A negotiation schedule $\mathcal{S}$ in deterministic settings is a minimal set of game $\{g_1, \dots g_n\}$ each representing a potential negotiation whose resolving lead the negotiation space to equilibrium, and results in the non-dominated outcome set.*

4.2 Finding Pareto Optimal Solutions in Electronic Bargaining

In this section we present the techniques proposed in [5]. The negotiation space is represented as a set of cooperative games with predefined utilities. The objective is to find the set of non-dominated (aka Skyline), or most beneficial negotiation outcomes. The solutions of cooperative games are known to be located on the Pareto frontier of the feasible solution set based on Nash bargaining model. However, obtaining the exact solution points is an expensive operation since it involves solving a linear program over the set of possible solution as pointed out in Section 4.1. Moreover, solving one of the games might affect the feasible sets of other games, or necessitate resolving other games more than once. The task of an intermediate party in these settings is to generate a game playing schedule that results in the set of Pareto optimal outcomes while doing the least possible number of game evaluations.

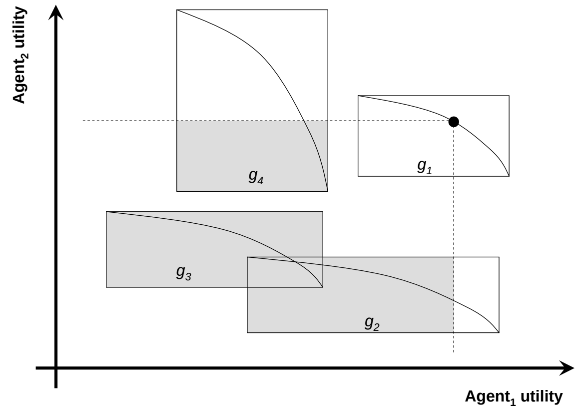

Figure 7: A Space of Cooperative Games Representing Negotiation

Consider for example the game space in Figure 7. The depicted games are represented as rectangular utility areas between the maximin and max-utility pairs of each game; where the max-utility pair is the point with the maximum available players utilities. Notice that if we are looking for the set of non-dominated outcomes, we can ignore solving the game g_3 since its max-utility pair is dominated by the maximin utility pair of g_1. Moreover, the dominance regions of the outcomes of solved games are used to filter the feasible sets of other games. This can be seen as pruning any possible outcomes that are guaranteed to be weaker than other received outcomes. For example the dominance region of g_1 outcome (enclosed within the dotted lines) intersects with the feasible sets of g_2 and g_4, and consequently the dominated portions of g_2 and g_4 have to be clipped. This necessitates redefining g_2 and g_4 maximin pairs since the players strategies have changed by eliminating the weak utilities. Redefining a game leads to having to re-solve that game since the players view of the game feasible set has changed.

The interaction among games in the negotiation space is captured by the following two principles.

Principle 1 ***Game Clipping.*** *The feasible set of a game g is clipped by the outcome of another game g^*, if the dominance region of g^* outcome intersects but not subsumes the feasible set of g.*

Principle 2 ***Game Dominance.*** *A game g is completely eliminated if the dominance region of another game g^* subsumes the max-utility pair of g.*

Applying the above two principles when solving each game causes exponential explosion in the number of games needed to be solved, since we might need to resolve the same game multiple times whenever it is clipped based on the outcome of some other game.

Finding an efficient schedule by trying all possible game orders is clearly infeasible because of the dynamic space configuration described above. We thus need means to select the most important games to solve in order to bring the game space to equilibrium in the least number of steps. Since we have rectangular approximations for the outcome of each game, this information is used to calculate an *importance score* that estimates the pruning power of each game g based on two heuristics:

- *Clipping area:* The area of the rectangle enclosed between the origin point and the solution point of g. The larger the clipping area, the higher the probability that solving g prunes more games.
- *Number of clipped games:* The pruning power of g is represented by the number of games fully/partially dominated by the solution point of g in the game set.

If we know the exact solutions of all games beforehand, we can apply the above heuristics directly until the game set is brought to equilibrium. However we usually do not know the outcome of each game unless we solve that game. Since solving a game is an expensive operation whose invocations need to be optimized, we replace the exact solution of some game g, (u_g^*, v_g^*), by an estimated solution $(\hat{u_g^*}, \hat{v_g^*})$. This estimation is conducted based on two alternatives:

- $(\hat{u_g^*}, \hat{v_g^*})$ is the max-utility pair of g.
- $(\hat{u_g^*}, \hat{v_g^*})$ is the utility pair at the center of g feasible set.

The first alternative is a looser estimate compared to the second one since game solution is believed to be on the Pareto frontier of the feasible set.

The general procedure of finding Pareto optimal negotiation outcomes is summarized as follows:

1. Initialize schedule $\mathcal{S}$ to $\{\}$.
2. For each game, estimate its solution according to either the max-utility or center-utility pairs, and use it to estimate the pruning power of that game based on one of the heuristics.
3. Order games in a descending order according to their pruning power.
4. Solve the top game g^* accurately, prune all dominated games, and clip partially dominated games.
5. Add g^* to $\mathcal{S}$.
6. Re-estimate the pruning power of redefined games and re-insert them in the game list ordered by their new pruning power.
7. If there are still any unsolved or redefined games, go to step 4.

5 Probabilistic Utilities

In this section we describe our formulation of the problem in the more generic setting; where we deal with probabilistic utilities rather than explicit ones. We show how to model the negotiations in this setting as Bayesian games in Section 5.1, and present our proposed scheduling technique in Section 5.2.

5.1 Describing Probabilistic Utilities

In Bayesian games, each one of the interacting agents has a set of possible types. For each type the agent keeps a probability distribution over utilities of different strategies. This defines a set of possible games that could be played by agent under each type. We assume the exclusiveness of these games, i.e, only one of these games is actually played in practice. For example consider Figure 3. If the types of the two agents are θ_{11} and θ_{21}, then the Bayesian game reduces to a probability distribution over games g_1 and g_2. The joint probabilities of these two games would be $\frac{2}{3}$ and $\frac{1}{3}$, respectively. Moreover, an agent cannot mix his utilities based on the two games. In other words, the two agents have to select either g_1 or g_2 to play before they start to interact. Of course this selection is influenced by the joint probabilities (g_1 has as twice chance to be selected as g_2). Notice that the probability distribution $\{\frac{2}{3}, \frac{1}{3}\}$ captures the 'joint' choice of the two players and not their individual choices.

Agents reveal their *true types* to the intermediate party before interaction starts. However, they do not have information about the exact types of other agents. Each agent describes his utilities under different strategies in a probabilistic manner. The intermediate party is required to generate an efficient negotiation schedule for the interacting agents based on this information. This schedule should consider probabilistic agents utilities as opposed to the deterministic utilities discussed in Section 4.

The above described problem is fundamentally different from the cooperative deterministic settings. Specifically, in probabilistic settings the intermediate party does not have an exact picture of the goodness of each game with respect to the two players since the actual playing of this game is by itself nondeterministic. This consideration changes the semantics of our problem from *'Finding an efficient negotiation schedule that results in the non-dominated outcome set'* to *'Finding an efficient negotiation schedule that results in the non-dominated outcome set with a high probability'*.

Taking probability into consideration extends the space of possible games to allow multiple exclusive versions of each game based on utility distributions. In this setting the information provided by each agent to the intermediate party is formulated as a 'probabilistic interaction descriptor' as formalized by the following definition.

Definition 5 ***Probabilistic Interaction Descriptor:*** *An agent i declares willingness to negotiate with agent j over a set of entities without exposing his explicit utilities, by providing a descriptor $< \theta_i, E_i, A_i, \lambda_i >$ with the following information:*

- *θ_i is the chosen type of agent i.*
- *$E_i = \{e_1, \ldots, e_m\}$, a set of m negotiable entities.*
- *$A_i = (A_i^1, \ldots A_i^m)$, where A_i^l is the set of available actions of agent i while negotiating over entity l.*
- *$\lambda_i = (\lambda_i^1, \ldots \lambda_i^m)$, where $\lambda_i^l : \Theta_j \rightarrow \mathcal{P}^l$ maps each possible type of the other agent j to a probability distribution $\mathcal{P}$ over the utility matrices of agent i while negotiating with agent j over entity l.*

For example consider Figure 1. Agent 1 could submit a probabilistic descriptor of the form:

- $\theta_1 = \theta_{11}$.
- $E_1 = \{$'Delivery Schedule'$\}$.
- $A_1 = \{$'Fast Delivery', 'Slow Delivery'$\}$ Vs. $\{$'Money Order', 'Bank Transfer'$\}$.
- $\lambda_1 = (\theta_{21} \rightarrow \{\begin{pmatrix} 30 & 10 \\ 40 & 15 \end{pmatrix} : 0.4, \begin{pmatrix} 20 & 12 \\ 35 & 10 \end{pmatrix} : 0.6\} \; ; \; \theta_{22} \rightarrow \{\begin{pmatrix} 10 & 20 \\ 30 & 22 \end{pmatrix} : 0.8, \begin{pmatrix} 25 & 15 \\ 15 & 12 \end{pmatrix} : 0.2\})$.

This descriptor declares that agent 1 has type θ_{11}, and is willing to negotiate over only 'Delivery Schedule' based on two strategies $\{$Fast Delivery', 'Slow Delivery'$\}$ against the two opponent strategies $\{$Money Order', 'Bank Transfer'$\}$. Agent 1 also declares that he will negotiate with the other agent based on the two given probabilistic distributions over utilities according to the type of the other agent.

Now, assume that agent 2 has also submitted the following descriptor to the intermediate party:

- $\theta_2 = \theta_{21}$.
- $E_2 = \{$'Delivery Schedule'$\}$.
- $A_2 = \{$'Money Order', 'Bank Transfer'$\}$ Vs. $\{$'Fast Delivery', 'Slow Delivery'$\}$.
- $\lambda_2 = (\theta_{11} \rightarrow \{\begin{pmatrix} 20 & 30 \\ 10 & 15 \end{pmatrix} : 1.0\} \; ; \; \theta_{12} \rightarrow \{\begin{pmatrix} 40 & 20 \\ 20 & 22 \end{pmatrix} : 0.3, \begin{pmatrix} 35 & 30 \\ 25 & 12 \end{pmatrix} : 0.7\})$.

Based on this information, the intermediate party considers a potential negotiation between agent 1 and agent 2 over 'Delivery Schedule' based on the Bayesian game of Figure 8. In this Figure negotiation over 'Delivery Schedule' could follow any of the two payoff matrices based on the indicated probabilities. Such probabilities have to be considered while evaluating the potential importance of each game since the importance of some game instance might be conflicting with its probability. For example a highly beneficial game instance with a small probability would not be actually selected for playing in practice. The task of the intermediate party in these settings is to find a negotiation schedule conforming with the probabilities of game instances and guaranteeing the maximal estimated benefits to both players.

Agent2 / Agent1	Money Order	Bank Transfer
Fast delivery	30,20	10,30
Slow delivery	40,10	15,15

p =0.4

Agent2 / Agent1	Money Order	Bank Transfer
Fast delivery	20,20	12,30
Slow delivery	35,10	10,15

p =0.6

Figure 8: Generated Bayesian Game from Probabilistic Agents Descriptors

The negotiation space in probabilistic settings can be seen as a probability distribution over a set of possible space instances. Consider for example the Bayesian game set depicted by Figure 9. In this example the probabilistic negotiations are represented by three games $\{g_1, g_2, g_3\}$; where g_2 represents exact negotiation over some entity while both g_1 and g_3 are probabilistic negotiations of two instances each. The probability tree on the side illustrates the complete space that could be derived from $\{g_1, g_2, g_3\}$. The negotiation space has four possible instances $\{I_1, I_2, I_3, I_4\}$ whose probabilities are calculated assuming the independence of games instances. Notice that any exclusive games instances (e.g. g_{11} and g_{12}) cannot appear in the same space instance together.

A straightforward approach to find an efficient schedule in these settings is to process each space instance separately: Find efficient schedules for each one of $\{I_1, I_2, I_3, I_4\}$ using the techniques discussed in Section 4, weight each generated schedule by the probability of its instance, and finally select the most probable schedule. However, this simple approach is not applicable for even a moderate number of negotiations since the number of space instances is expo-

Game	Game Instances
g_1	$\{g_{11}$:0.3, g_{12}:0.7$\}$
g_2	$\{g_2$:1.0$\}$
g_3	$\{g_{31}$:0.4, g_{32}:0.6$\}$

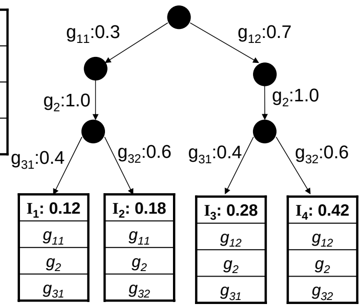

Figure 9: Space of Possible Negotiations Derived from Bayesian Games

nential in the number of negotiations (games) (e.g. a set of 100 Bayesian games of 2 instances each generates 2^{100} space instance). Therefore we need alternate techniques that work on the negotiation space in its compact form rather than having to flatten the space to all its possible instances. Section 5.2 below describes our approach to solve this problem.

5.2 Scheduling Probabilistic Negotiations

A *negotiation schedule*, as defined in Section 4, is a set of sequential game plays that results in the non-dominated outcome set. In deterministic settings the selection of such games was based on game importance scores only. However, in probabilistic settings each game instance has an associated probability that needs to be taken into account as well. The intermediate party selects games to be added to the schedule based on the following two factors:

- *Game Importance*: Each game instance receives an importance score based on the heuristics described in Section 4.
- *Game Probability*: Each game instance has a probability that expresses the likelihood for this game to be selected by the two players.

In the following discussion we use the notion of *'max-probability most beneficial game'*; to refer to the most beneficial game with the maximum probability.

To illustrate the above principles, consider for example the set of games in Table 1; where games are ordered based on their normalized importance scores and only the few top games are shown for brevity. Notice that games g_{11} and g_{12} are exclusive instances since they are both derived from the same game (negotiable object) g_1. This means that g_{11} and g_{12} can never appear together in the same negotiation schedule since they can never be both played. Assume that we are looking for the *'max-probability most beneficial game'* in this set. We show now how to find such game by scanning the game list in the order of importance scores:

Game Instance	Score	Probability
g_{11}	0.9	0.4
g_2	0.7	0.7
g_3	0.6	0.2
g_{12}	0.5	0.6
g_4	0.4	0.3

Table 1: Game Instances and their Probabilities

- In the first step we encounter g_{11} whose probability is 0.4. Although g_{11} has the maximum importance score, it cannot be considered as the *'max-probability most beneficial game'*. This is because there is a probability of $1 - 0.4 = 0.6$ that g_{11} is not played. However we now know that the upper bound probability for any unseen game to be the most beneficial is 0.6 (which corresponds to the case that g_{11} is not played and another 'unseen' game instance with probability 1 is played).
- In the second step we encounter g_2 with probability 0.7. Now the probability that g_2 is the *'max-probability most beneficial game'* is $0.7 \times 0.6 = 0.42$; while the upper bound probability for any unseen game drops to $0.6 \times 0.3 = 0.18$ (which corresponds to the case that both g_{11} and g_2 are not played and another 'unseen' game instance with probability 1 is played). This means that we can report g_2 as the *'max-probability most beneficial game'*, and g_{11} as the second *'max-probability most beneficial game'* with probabilities 0.42 and 0.4, repectively.

The above procedure describes how to find most beneficial games in decreasing probability order. This procedure can be extended to find a schedule for the max-probability beneficial games that should be played sequentially till equilibrium. For example if g_2 is scheduled for playing as the first game, the intermediate party applies the clipping and pruning principles based on g_2 solution point as described in Section 4. The order of the games based on importance score might change as a result of this operation. Therefore, the game set needs to be re-sorted based on the new importance scores (this was done efficiently in our implementation using a priority queue). The game set is scanned again from the top to find the next *'max-probability most beneficial game'* to be played after g_2. The process is repeated until the game set reaches equilibrium.

We next present the formal definition of *negotiation schedule* under the above described probabilistic settings:

Definition 6 ***Probabilistic Negotiation Schedule:*** *A negotiation schedule $\mathcal{S}$ in probabilistic settings is a minimal set of non-exclusive game instances*

$\{g_1, \ldots g_n\}$ *whose playing leads the game set to equilibrium, and is estimated to have the maximum profit to agents at the maximum possible probability.*

The following steps formalize the search technique adopted to generate *negotiation schedule* in probabilistic settings:

1. Initialize schedule $\mathcal{S}$ to $\{\}$.
2. Scan game set based on the importance scores until some game g^* has a higher probability of being the most beneficial game than all other games.
3. Add g^* to $\mathcal{S}$.
4. Solve g^* at $Sol(g^*)$.
5. Remove all exclusive game instances to g^*, and clip/prune other games affected by $Sol(g^*)$.
6. Re-insert redefined games in the game list based on their new importance scores
7. If equilibrium is not reached goto step 2.

Notice that in step 5 we have to remove other exclusive game instances to g^* since a schedule cannot include two or more games whose existence is complementary (e.g. g_{11} and g_{12} in Figure 9); otherwise it would be non-implementable by the agents.

The above scheme addresses the interplay between the importance scores and probabilities of game instances. A schedule that considers only games importance scores could generate a higher outcome than our proposed scheme, however that outcome is only achievable with a relatively smaller probability. On the other hand, a schedule that considers only game instance probabilities could clearly generate less beneficial outcomes to interacting parties.

6 Implementation

During the implantation phase of the proposed solution two main issues aroused; first was the need to efficiently solve games and second was the importance of supporting a general purpose (i.e. generic) game generation process. In Section 6.1 we briefly sketch our implemented Nash bargaining procedure as adopted from [8]. Section 6.2 discusses how we incorporated GAMUT [9] as a test bed generator into our implementation.

6.1 Polynomial Time Nash Bargaining Solver

Although focused on finding Nash equilibrium for repeated games, yet based on the aforementioned Nash's axioms for solving Nash bargaining model in Section 2.1; the authors in [8] construct the polynomial time algorithm sketched below to find such solution:

- The point that maximizes the product of the advantages is found on the outer boundary of the convex hull of the game feasible set. That point can thus be expressed by a weight vector w that has non-zero weight on only one or two points in the payoff set X, since the convex hull is a two dimensional polygonal region bounded by line segments.
- We are trying to identify a payoff pair in the convex hull of X that maximizes the product of its components. Such a point is either in X or on a line segment between two points in X.
- Let $x = (x^1, x^2) \in X$, $y = (y^1, y^2) \in X$ be two points formed by two different action pairs. We want to find a point $z = (z^1, z^2)$ on the edge between x and y, where $z = w_x \times x + (1 - w_x) \times y$ for $0 \leq w_x \leq 1$, such that the product of the payoffs, $z^1 \times z^2$, is maximized. Setting the derivative of this product to zero and solving for w_x, we find that the product is maximized when: $w_x = \frac{-y^2(x^1-y^1)-y^1(x^2-y^2)}{2(x^2-y^2)(x^1-y^1)}$
- Evaluating the above equation results in a weight that is a polynomial-size rational number, since it involves a constant number of basic arithmetic operations.
- If $w_x < 0$ or $w_x > 1$, then the maximum product is achieved at an endpoint.
- By looping through all points and pairs of points in X (a polynomial number of combinations), computing weights, and checking which has the largest product of advantages, we can identify a pair of action pairs (polynomial size) that is a solution to Nash's model.

The above procedure executes in polynomial time $O(n^2)$ and produces a polynomial-size output.

6.2 GAMUT

Well established as a game theory test suite, GAMUT [9] was a very good alternative rather than manually implementing our game generator. GAMUT is a highly modular and extensible software framework. With its 35 implementation of game objects, GAMUT provided us with a very rich test bed for our experiments.

Our implementation tweaked and extended GAMUT. We needed to tweak GAMUT to provide our own normal form presentation that is compatible with our adopted Nash bargaining solution procedure. Furthermore tweaked GAMUT was plugged into our own implementation of Bayesian Games; where there exist several instances of GAMUT games each associated with a certain probability of occurrence. We implemented also the appropriate normalization procedure for Bayesian Games.

Our proposed techniques were implemented on top of GAMUT to generate negotiation schedules for the tested data sets. As a baseline we implemented an exhaustive depth-first search algorithm to try every possible schedule in the negotiation space, and return the optimal schedule in terms of length and profit. Because of space explosion this method was not applicable to large game sets since considering all game permutations is infeasible. However based on relatively small game sets, comparing the generated schedules by proposed methods with the optimal schedules gave important insights as outlined in the next Section.

7 Experiments

We conducted an extensive set of experiments to evaluate the proposed techniques. The focus of the experiments is to evaluate the incurred cost by arbitration mechanisms that were used to resolve games as well as the complexity of the search for the non-dominated outcome set. We primarily evaluated the proposed methods based on the following performance metrics:

- *Schedule Length*: How many games did the agents play in order to reach the non-dominated outcome set.
- *Social Welfare per Negotiation*: The summation of agents payoffs throughout all scheduled games divided by the schedule length.

We extended GAMUT to generate game sets of two different layouts:

- *Dispersed Games* : Games are scattered randomly in 2D space. It is relatively easier in this case to find an efficient schedule since dominated games can be directly identified and eliminated.
- *Correlated Games*: We applied a linear transformation with a random noise to put the games around the -45 line (0,max(Y))-(max(X),0). This layout is more competitive since many games outcomes have a chance to be in the non-dominated outcome set.

We evaluated the effectiveness of different heuristics to order game plays compared to the baseline optimal solution whenever possible. Figure 10 depicts the evaluated heuristics. Max–Area (Avg–Area) refer to the area enclosed between game's max–utility point (center point) and origin. Tightly (loosely) dominated games refer to the number of subsumed games by Max-Area (Avg–Area). Max–X (Max–Y) refer to the maximum utilities of X (Y) players.

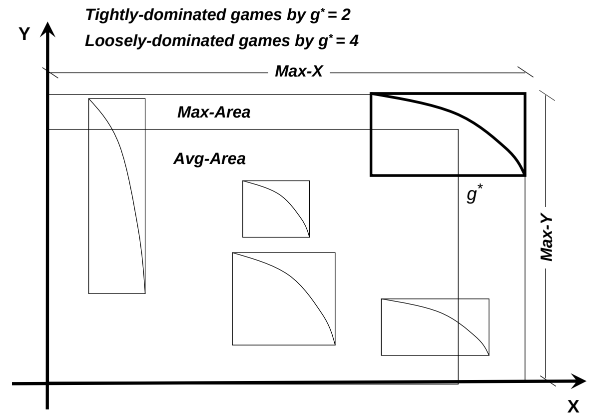

Figure 10: Evaluated Heuristics

7.1 Schedule Length

In this experiment we compare the schedule length generated by different heuristics with the optimal schedule length generated by the exhaustive search algorithm. Figures 11 and 12 show the results obtained for *Dispersed Games* data set. Notice that we compared the length of generated schedules with the optimal schedule length for small number of games only, because of the space explosion of the exhaustive search algorithm as the number of games increases. All of the heuristics showed comparable performance as they generate schedules with lengths near to the optimal. For larger number of games, in Figure 12, the Max-Area heuristic showed the worst performance. This is primarily attributed to the fact that as the number of games increase, the number of games with comparable Max–Area's also increase making deciding the most efficient game inaccurate sometimes. The Avg–Area heuristic shows similar behavior but at a less degree since it is calculated from a point usually nearer to the pareto frontier of game feasible sets.

Figures 13 and 14 show similar results for the *Correlated Games* data set. However, all of the heuristics show similar behavior in this case. This is expected based on the fact that dominating a game by the solution of another game is less likely in this data set (since games are scattered around the –45 line) making solving most of the games necessary in any order.

7.2 Number of Players Strategies

In this experiment we evaluated the effect of increasing the number of players strategies in each game on the length of the generated schedule. Increasing the number of players strategies is equivalent to increasing the area of game feasible set. Games in this setting tend to be more interleaving (or subsumed by each other) making it highly likely to prune multiple games by solving a small number of games yielding schedules with small lengths. Figure 15 illustrates this fact

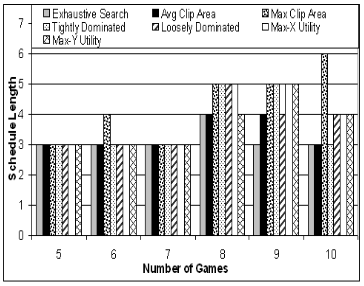

Figure 11: Schedule Length for Dispersed Games (1)

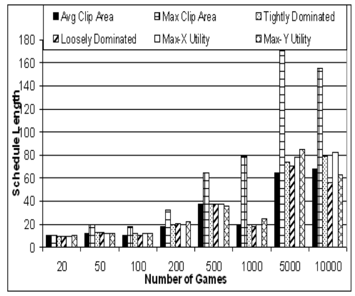

Figure 12: Schedule Length for Dispersed Games (2)

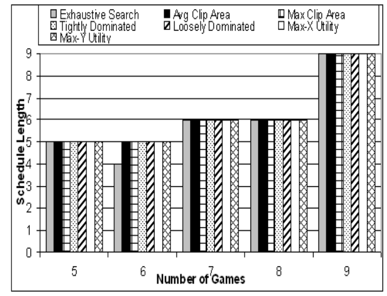

Figure 13: Schedule Length Correlated Games (1)

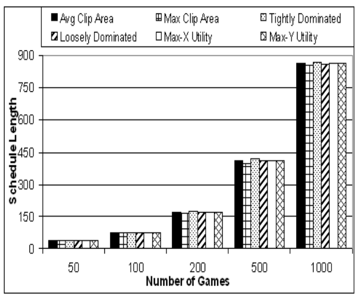

Figure 14: Schedule Length Correlated Games (2)

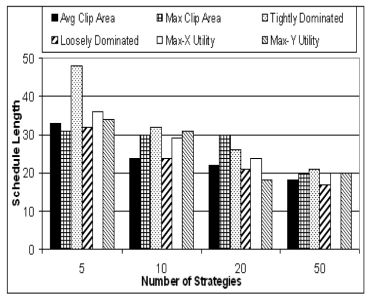

Figure 15: Effect of Number of Strategies on Schedule Length

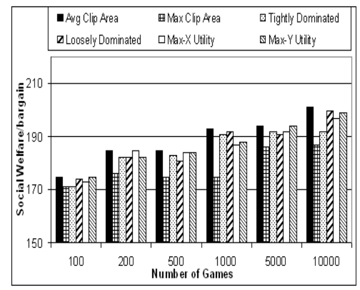

Figure 16: Social Welfare/Bargain of Different Heuristics

for different heuristics for a test made on 5000 games drawn from the *Dispersed Games* data set.

7.3 Social Welfare per Negotiation

In this experiment we compare the average social welfare (defined as the sum of players utilities in all scheduled games divided by the schedule length). Figure 16 shows the results obtained from *Dispersed Games* data set. Generally, the Avg–Area and Tightly Dominated Games heuristics show the best performance confirming the fact that they resemble more accurate estimates for game solutions. However, this behavior does not hold for games with larger areas where a game solution usually deviates from the game center point.

7.4 Social Welfare in Bayesian Games

In this experiment we compare the social welfare per negotiation for Bayesian games with the expected welfare obtained from the normalized version of the Bayesian games. The proposed algorithm for selecting an instance to play from each Bayesian game is based on selecting most beneficial game instances at the maximum possible probability. This can be seen as trying to guarantee a payoff larger than the expected payoff that could be obtained by solving the normalized Bayesian game. Figure 17 confirms this fact by showing the average social welfare obtained by different heuristics compared to that of the normalized Bayesian games. Heuristics generally generate payoffs above the expected payoff of the normalized games since they select the most beneficial games at each step with the highest possible probability.

7.5 Probability of Probabilistic Schedules

Probabilistic schedule captures a possible instantiation of the game space that guarantees the best payoff to the agents at the highest probability. In this experiment we compare the weighted social welfare generated by different heuristics. This is defined as the product of schedule probability and its average social welfare. The schedule probability is the product of the probabilities of schedule game instances (assuming independence among instances of different games). Figure 18 illustrates the results obtained for different number of games drawn from *Dispersed Games* data set. It can be noticed that Max–Area and Tightly Dominated Games heuristics have the maximum weighted welfare among all other heuristics indicating their superiority to other heuristics in probabilistic settings. This can be explained based on the fact that Max–Area allows overestimation of game's solution which leads to safe selection of beneficial games in many cases; while Tightly Dominated Games reflects the accurate game solution in most of the cases.

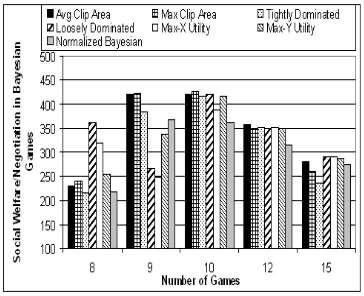

Figure 17: Average Social Welfare of Probabilistic Schedules

Games	Avg-Area	Max-Area	T. Dominated	L. Dominated	Max-X	Max-Y
10	0.1813	3.024	1.3	.008	0.632	0.034
50	0.375	4.34	4.88	3.86	0.379	0.772
100	2.99	3.02	3.19	3.9	2.93	1.7
1000	3.08	4.02	15.6	15.6	1.23	1.7

Figure 18: Average Social Welfare of Probabilistic Schedules Weighted by their Probabilities

8 Related Work

The problems of scheduling negotiations and optimal agenda were addressed in previous work under many semantics. In [10], a model was devised based on the assumptions that players have identical discounting factor to damp their utilities as time passes without incorporating players deadlines. The work of Sandholm and Vulkan [11] considered probabilistic representation of agents deadlines, however agents payoffs do not change based on discounting factors. Other multi-issues negotiation models [7, 6] analyzed the effect of both deadlines and discounting factors. [6] proposed decomposing the negotiable entities into equal stages determined offline while the negotiation order within each stage is determined online.

The problems of discovering the non-dominated outcome set in a stepwise fashion with the least possible number of steps, inter-dependencies among negotiations, modeling agents strategies and preferences based on traditional game theory principles and devising arbitration mechanisms under uncertainty were not clearly addressed/integrated in one solution in the previous approaches.

9 Conclusions and Future Extensions

In this work we addressed the problem of scheduling negotiations over multiple entities based on game theory principles. We identified a number of issues related to the formulation of negotiation descriptors in deterministic and probabilistic settings, converting descriptors into cooperative and Bayesian games, and designing search mechanisms to find efficient negotiation schedules in different settings. Our work raises potential future extensions such as tight integration of time/budget constraints in game solving procedures, studying the applicability of alternative game models, and designing a standardized protocol to allow the communication between negotiating parties and possible intermediate mediators.

References

[1] J. Nash, The bargaining problem, Econometrica, 18(2), 155-162, 1950.

[2] J. Nash, Two-person cooperative games, Econometrica, 21(1), 129-140, 1953.

[3] Thomas S. Ferguson. Game Theory Text. Mathematics Department, UCLA.

[4] Yoav Shoham and Kevin Leyton-Brown, Multi Agent Systems.

[5] Mohamed A. Soliman, Ihab F. Ilyas, and Nick Koudas. Finding Skyline and Top-k Bargaining Solutions. Submitted, 2006.

[6] S. Fatima and M. Wooldridge and N. Jennings. Optimal agendas for multi-issue negotiation. AAMAS, 2003.

[7] S. Fatima and M. Wooldridge and N. Jennings. An agenda based framework for multi-issue negotiation. Artificial Intelligence Journal, 2004.

[8] M. L. Littman and P. Stone. A polynomial-time Nash equilibrium algorithm for repeated games. In EC'03, 2003.

[9] E. Nudelman, J. Wortman, K. Leyton-Brown, Y. Shoham. Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. International joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2004.

[10] C. Freshtman. The importance of agenda in bargaining. In Games and Economic Behavior , 1990.

[11] T. Sandholm and N. Vulkan. Bargaining with deadlines. In AAAI, 1999.