Report Number: WUCS-99-04

1999-01-01

Algorithms for Optimizing Leveled Commitment Contracts

Thomas Sandholm, Sandeep Sikka, and Samphel Norden

In automated negotiation systems consisting of self-interested agents, contracts have traditionally been binding. Leveled commitment contracts - i.e. contracts where each party can decommit by paying a predetermined penalty - were recently shown to improve Pareto efficiency even if agents rationally decommit in Nash equilibrium using inflated thresholds on how good their outside offers must be before they decommit. This paper operationalizes the four leveled commitment contracting protocols by presenting algorithms for using them. Algorithms are presented for computing the Nash equilibrium decomitting thresholds and decommitting probabilities given the contract price and the penalties. Existence and uniqueness of the... Read complete abstract on page 2.

Algorithms for Optimizing Leveled Commitment Contracts

Thomas Sandholm, Sandeep Sikka, and Samphel Norden

Complete Abstract:

In automated negotiation systems consisting of self-interested agents, contracts have traditionally been binding. Leveled commitment contracts - i.e. contracts where each party can decommit by paying a predetermined penalty - were recently shown to improve Pareto efficiency even if agents rationally decommit in Nash equilibrium using inflated thresholds on how good their outside offers must be before they decommit. This paper operationalizes the four leveled commitment contracting protocols by presenting algorithms for using them. Algorithms are presented for computing the Nash equilibrium decomitting thresholds and decommitting probabilities given the contract price and the penalties. Existence and uniqueness of the equilibrium are analyzed. Algorithms are also presented for optimizing the contract itself (price and penalties). Existence and uniqueness of the optimum are analyzed. Using the algorithms we offer a contract optimization service on the web as part of eMediator, our next generation electronic commerce server. Finally, the algorithms are generalized to contracts involving more than two agents.

Algorithms for Optimizing Leveled Commitment Contracts

Tuomas Sandholm, Sandeep Sikka, Samphel Norden

WUCS-99-04
February 1, 1999

Algorithms for Optimizing Leveled Commitment Contracts

Tuomas Sandholm,* Sandeep Sikka, and Samphel Norden
Department of Computer Science
Washington University
St.Louis, MO 63130
{sandholm,sikka,samphel}@cs.wustl.edu

February 1, 1999

Abstract

In automated negotiation systems consisting of self-interested agents, contracts have traditionally been binding. Leveled commitment contracts — i.e. contracts where each party can decommit by paying a predetermined penalty — were recently shown to improve Pareto efficiency even if agents rationally decommit in Nash equilibrium using inflated thresholds on how good their outside offers must be before they decommit. This paper operationalizes the four leveled commitment contracting protocols by presenting algorithms for using them. Algorithms are presented for computing the Nash equilibrium decommitting thresholds and decommitting probabilities given the contract price and the penalties. Existence and uniqueness of the equilibrium are analyzed. Algorithms are also presented for optimizing the contract itself (price and penalties). Existence and uniqueness of the optimum are analyzed. Using the algorithms we offer a contract optimization service on the web as part of *eMediator*, our next generation electronic commerce server. Finally, the algorithms are generalized to contracts involving more than two agents.

*This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IRI-9610122, and Grant IIS-9800994.

1 Introduction

In multiagent systems consisting of self-interested agents, contracts have traditionally been binding [Rosenschein and Zlotkin, 1994, Sandholm, 1993, Kraus, 1993]. Once an agent agrees to a contract, she has to follow through no matter how future events unravel. Although a contract may be profitable to an agent when viewed *ex ante*, it need not be profitable when viewed after some future events have occurred, i.e. *ex post*. Similarly, a contract may have too low expected payoff *ex ante*, but in some realizations of the future events, it may be desirable when viewed *ex post*. Normal full commitment contracts are unable to take advantage of the possibilities that such future events provide.

On the other hand, many multiagent systems consisting of cooperative agents incorporate some form of decommitment in order to allow agents to accommodate new events. For example, in the original Contract Net Protocol [Smith, 1980], the agent that contracts out a task could send a termination message to cancel the contract even when the contractee had partially fulfilled it. This was possible because the agents were not self-interested: the contractee did not mind losing part of its effort without a monetary compensation. Similarly, the role of decommitment among cooperative agents has been studied in meeting scheduling [Sen, 1993].

Contingency contracts have been suggested for utilizing the potential provided by future events among self-interested agents [Raiffa, 1982]. The contract obligations are made contingent on future events. In some games this increases the expected payoff to both parties compared to any full commitment contract. However, contingency contracts are often impractical because the space of combinations of future events may be large and unknown. Also, when events are not mutually observable, the observing agent can lie about what transpired.

Leveled commitment contracts are another method for capitalizing on future events [Sandholm and Lesser, 1996]. Instead of conditioning the contract on future events, a mechanism is built into the contract that allows unilateral decommitting. This is achieved by specifying in the contract the level of commitment by decommitment penalties, one for each agent. If an agent wants to decommit—i.e. to be freed from the obligations of the contract—it can do so simply by paying the decommitment penalty to the other party. The method requires no explicit conditioning on future events: each agent

can do her own conditioning dynamically. No event verification mechanism against lying is required either.

Principles for assessing decommitment penalties have been studied in law [Calamari and Perillo, 1977, Posner, 1977], but the purpose has been to assess a penalty on the agent that has breached the contract *after the breach has occurred.* Similarly, penalty clauses for partial failure—such as not meeting a deadline—are commonly used in contracts, but the purpose is usually to motivate the agents to follow the contract. Instead, in leveled commitment contracts, explicitly allowing decommitment from the contract for a predetermined price is used as an active method for utilizing the potential provided by an uncertain future.[1] The decommitment possibility increases each agent's expected payoff under very general assumptions [Sandholm and Lesser, 1996].

We analyze contracting situations from the perspective of two risk neutral agents each of which attempts to maximize his own expected payoff: the *contractor* who pays to get a task done, and the *contractee* who gets paid for handling the task. Handling a task can mean taking on any types of constraints. The method is not specific to classical task allocation. The contractor tries to minimize the contract price ρ that he has to pay. The contractee tries to maximize the payoff ρ that she receives.

We study a setting where the future of the agents involves uncertainty. Specifically, the agents might receive outside offers.[2] The contractor's best outside offer $\breve{a}$ is only probabilistically known *ex ante* by both agents, and is characterized by a probability density function $f(\breve{a})$. If the contractor does not receive an outside offer, $\breve{a}$ corresponds to its best outstanding outside offer or its fall-back payoff, i.e. payoff that it receives if no contract is made. The contractee's best outside offer $\breve{b}$ is also only probabilistically known *ex ante*, and is characterized by a probability density function $g(\breve{b})$. If the contractee does not receive an outside offer, $\breve{b}$ corresponds to its best outstanding outside offer or its fall-back payoff.[3] The variables $\breve{a}$ and $\breve{b}$ are assumed statistically

[1] Decommitting has been studied in other settings, e.g. where there is a constant inflow of agents, and they have a time cost for searching partners of two types: good or bad [Diamond and Maskin, 1979].

[2] The framework can also be interpreted to model situations where the agents' cost structures for handling tasks and for getting tasks handled change e.g. due to resources going off-line or becoming back on-line.

[3] Games where at least one agent's future is certain, are a subset of these games. In

independent.

The contractor's options are either to make a contract with the contractee or to wait for $\breve{a}$. Similarly, the contractee's options are either to make a contract with the contractor or to wait for $\breve{b}$. The two agents could make a full commitment contract at some price. Alternatively, they can make a leveled commitment contract which is specified by the contract price, ρ, the contractor's decommitment penalty, a, and the contractee's decommitment penalty, b. We restrict our attention to contracts where $a \geq 0$ and $b \geq 0$, i.e. agents do not get paid for decommitting. The contractor has to decide on decommitting when he knows his outside offer $\breve{a}$ but does not know the contractee's outside offer $\breve{b}$. Similarly, the contractee has to decide on decommitting when she knows her outside offer $\breve{b}$ but does not know the contractor's. This seems realistic from a practical automated contracting perspective.

The theory of these leveled commitment protocols was presented by [Sandholm and Lesser, 1996], but to date no algorithms have been presented for agents to compute when they should decommit given a contract, or for agents to choose beneficial contracts. This paper operationalizes leveled commitment contracts by presenting an algorithm for computing how the agents should decommit (Section 2), and an algorithm for constructing the optimal leveled commitment contract for any given setting defined by $f(\breve{a})$ and $g(\breve{b})$ (Section 3).

2 Nash equilibria for a given contract

One concern is that a rational agent is reluctant in decommitting because there is a chance that the other party will decommit, in which case the former agent gets freed from the contract, does not have to pay a penalty, and collects a penalty from the breacher. [Sandholm and Lesser, 1996] showed that despite such insincere decommitting the leveled commitment feature increases each contract party's expected payoff, and enables contracts in settings where no full commitment contract is beneficial to all parties. To set the context, we first review their analysis of how rational agents would decommit, i.e. we derive the Nash equilibrium (NE) [Nash, 1950] of the decommitting game

such games all of the probability mass of $f(\breve{a})$ and/or $g(\breve{b})$ is on one point.

where each agent's decommitting strategy is a best response to the other agent's decommitting strategy. The new contributions begin in Section 2.3.

The contractor decommits if he gets a low enough outside offer, e.g., he can get his task handled at a low cost. We denote his decommitting threshold by $\breve{a}^*$, so his decommitting probability is

$$p_a = \int_{-\infty}^{\breve{a}^*} f(\breve{a})da \tag{1}$$

The contractee decommits if she gets a high enough outside offer, e.g., gets paid for handling a task. We denote her decommitting threshold by $\breve{b}^*$, so her decommitting probability is

$$p_b = \int_{\breve{b}^*}^{\infty} g(\breve{b})db \tag{2}$$

2.1 Sequential decommitting (SEQD) game

In our sequential decommitting (SEQD) game, one agent has to reveal her decommitting decision before knowing whether the other party decommits. While our implementation analyzes both orders of decommitting, due to space limitations we only discuss the setting where the contractee has to decide first. The case where the contractor decides first is analogous. There are two alternative leveled commitment contracts that differ on whether or not the agents have to pay the penalties if both decommit.

If the contractee has decommitted, the contractor's best move is not to decommit because $-\breve{a} - a + b \leq -\breve{a} + b$ (because $a \geq 0$). This also holds for a contract where neither agent has to pay a decommitment penalty if both decommit since $-\breve{a} \leq -\breve{a} + b$. In the subgame where the contractee has not decommitted, the contractor's best move is to decommit if $-\breve{a} - a > -\rho$, i.e.

$$\breve{a}^* = \rho - a \tag{3a}$$

The contractee gets $\breve{b} - b$ if she decommits, $\breve{b} + a$ if she does not but the contractor does, and ρ if neither decommits. Thus the contractee decommits if $\breve{b} - b > p_a(\breve{b} + a) + (1 - p_a)\rho$. If $p_a = 1$, this is equivalent to $-b > a$ which is false because $a \geq 0$ and $b \geq 0$. In other words, if the contractee surely decommits, the contractor does not. On the other hand, the above is equivalent to

$$\breve{b} > \rho + \tfrac{b + a p_a}{1 - p_a} \stackrel{\text{def}}{=} \breve{b}^* \text{ when } p_a < 1 \tag{4a}$$

2.2 Simultaneous decommitting games

In our simultaneous decommitting games, agents have to reveal their decommitment decisions simultaneously. We first discuss the variant (SIMUDBP) where both have to pay the penalties if both decommit. The contractor decommits if $p_b \cdot (-\breve{a} + b - a) + (1 - p_b)(-\breve{a} - a) > p_b \cdot (-\breve{a} + b) + (1 - p_b)(-\rho)$. If $p_b = 1$, this equates to $a < 0$, but we already ruled out contracts where an agent gets paid for decommitting. On the other hand, this equates to

$$\breve{a} < \rho - \tfrac{a}{1-p_b} \stackrel{\text{def}}{=} \breve{a}^* \text{ when } p_b < 1 \tag{3b}$$

The contractee decommits if $(1 - p_a)(\breve{b} - b) + p_a(\breve{b} - b + a) > (1 - p_a)\rho + p_a(\breve{b} + a)$. If $p_a = 1$, this equates to $b < 0$, but we ruled out contracts where an agent gets paid for decommitting. However, this equates to

$$\breve{b} > \rho + \tfrac{b}{1-p_a} \stackrel{\text{def}}{=} \breve{b}^* \text{ when } p_a < 1 \tag{4b}$$

In a SIMUDNP game, neither agent has to pay if both decommit. The contractor decommits if $p_b \cdot (-\breve{a}) + (1 - p_b)(-\breve{a} - a) > p_b \cdot (-\breve{a} + b) + (1 - p_b)(-\rho)$. If $p_b = 1$, this equates to $b < 0$, but we already ruled out contracts where an agent gets paid for decommitting. On the other hand, this equates to

$$\breve{a} < \rho - a - \tfrac{bp_b}{1-p_b} \stackrel{\text{def}}{=} \breve{a}^* \text{ when } p_b < 1 \tag{3c}$$

The contractee decommits if $(1 - p_a)(\breve{b} - b) + p_a\breve{b} > (1 - p_a)\rho + p_a(\breve{b} + a)$. If $p_a = 1$, this equates to $a < 0$, but we ruled out contracts where an agent gets paid for decommitting. However, this equates to

$$\breve{b} > \rho + b - \tfrac{ap_a}{1-p_a} \stackrel{\text{def}}{=} \breve{b}^* \text{ when } p_a < 1 \tag{4c}$$

For each game, calculating the Nash equilibria amounts to solving the simultaneous equations (3) and (4) which use (1) and (2).

2.3 Existence of a Nash equilibrium

We now discuss existence of the NE. Denote the support of $f(\breve{a})$ by $[\breve{a}_1, \breve{a}_n]$, and the support of $g(\breve{b})$ by $[\breve{b}_1, \breve{b}_m]$.

Theorem 2.1
SEQD: *A NE exists if* $\breve{a}_1 \leq \rho - a \leq \breve{a}_n$. *No NE exists if* $\rho - a > \breve{a}_n$. *Only a trivial NE with* $p_a = 0$ *exists if* $\rho - a < \breve{a}_1$.
SIMUDBP: *A NE exists if* $\rho - a \geq \breve{a}_n$ *and* $\rho + b \leq \breve{b}_1$. *Only a trivial NE with* $p_a = 0$ *exists if* $\rho - a < \breve{a}_1$ *and* $\rho + b > \breve{b}_1$. *Only a trivial NE with* $p_b = 0$ *exists if* $\rho + b > \breve{b}_m$ *and* $\rho - a < \breve{a}_n$. *Otherwise a NE may or may not exist.*

SIMUDNP: *A NE exists if* $\rho - a \geq \breve{a}_n$ *and* $\rho + b \geq \breve{b}_m$. *Only a trivial NE with* $p_a = 0$ *exists if* $\rho - a < \breve{a}_1$ *and* $\rho + b > \breve{b}_1$. *Only a trivial NE with* $p_a = 0$ *and* $p_b = 0$ *exists if* $\rho - a < \breve{a}_1$ *and* $\rho + b > \breve{b}_m$. *No NE exists if* $\rho + b < \breve{b}_1$. *Otherwise a NE may or may not exist.*

Proof. For each of the three games, the curves defined by (3) and (4) are continuous.
SEQD: From (3a): $\breve{a}^* = \rho - a$. For (4a), when $\breve{a}^* = \breve{a}_1, \breve{b}^* = \rho + b$ and in the limit $\breve{a}^* \to \breve{a}_n$, $\breve{b}^* \to \infty$. Thus the curves intersect if $\breve{a}_1 \leq \rho - a \leq \breve{a}_n$. For (4a) if $\rho - a < \breve{a}_1$ then $\breve{b}^* = \rho + b$, and so (3a) intersects with (4a) such that $p_a = 0$. For $\rho - a > \breve{a}_n$ (3a) and (4a) do not intersect.
SIMUDBP: For (3b), in the limit $\breve{b}^* \to \breve{b}_1$, $\breve{a}^* \to -\infty$ and when $\breve{b}^* = \breve{b}_m, \breve{a}^* = \rho - a$. For (4b), when $\breve{a}^* = \breve{a}_1, \breve{b}^* = \rho + b$ and in the limit $\breve{a}^* \to \breve{a}_n, \breve{b}^* \to \infty$. Thus the curves intersect if $\rho - a \geq \breve{a}_n$ and $\rho + b \leq \breve{b}_1$ For (4b) if $\breve{a}^* < \breve{a}_1$ then $\breve{b}^* = \rho + b$, and so the two curves will intersect with $p_a = 0$ if $\rho + b > \breve{b}_1$. Similarly for (3b) if $\breve{b}^* > \breve{b}_m$ then $\breve{a}^* = \rho - a$, and so the two curves will intersect with $p_b = 0$ if $\rho - a < \breve{a}_n$. Otherwise they may or may not intersect.
SIMUDNP: For (3c), in the limit $\breve{b}^* \to \breve{b}_1$, $\breve{a}^* \to -\infty$ and when $\breve{b}^* = \breve{b}_m, \breve{a}^* = \rho - a$. For (4c), when $\breve{a}^* = \breve{a}_1, \breve{b}^* = \rho + b$ and in the limit $\breve{a}^* \to \breve{a}_n$, $\breve{b}^* \to -\infty$. Thus the curves intersect if $\rho - a \geq \breve{a}_n$ and $\rho + b \geq \breve{b}_m$. For (4c) if $\breve{a}^* < \breve{a}_1$ then $\breve{b}^* = \rho + b$, and so the two curves intersect with $p_a = 0$ if $\rho + b > \breve{b}_1$. Further if $\rho + b > \breve{b}_m$ then p_b also equals 0. The curves do not intersect if $\rho + b \leq \breve{b}_1$. Otherwise they may or may not intersect. □

2.4 Uniqueness of the Nash equilibrium

We now consider uniqueness of the Nash equilibrium.

Theorem 2.2 *For the sequential games, the NE is unique. For the simultaneous games the NE need not be unique.*

Proof. For the sequential games, uniqueness follows from the forms of (3) and (4): One threshold is expressed as a function of ρ and a penalty, and the other is defined as a function of this threshold. Fig. 1 shows an example where multiple equilibria exist for a SIMUDBP game. We constructed a similar example for SIMUDNP. □

$$f(\breve{a})=\begin{cases}0.01 & \text{if } 0.1 \leq \breve{a} < 0.5\\ 1.67\breve{a}-0.82 & \text{if } 0.5 \leq \breve{a} < 0.8\\ 0.42\breve{a}+0.17 & \text{if } 0.8 \leq \breve{a} \leq 2.0\\ 0 & \text{otherwise}\end{cases}$$

$$g(\breve{b})=\begin{cases}-1195.20\breve{b}+2034.87 & \text{if } 1.69 \leq \breve{b} < 1.7\\ 3.03 & \text{if } 1.7 \leq \breve{b} \leq 2.0\\ 0 & \text{otherwise}\end{cases}$$

$\rho = 1.5$
$a = 0.1$
$b = 0.2$

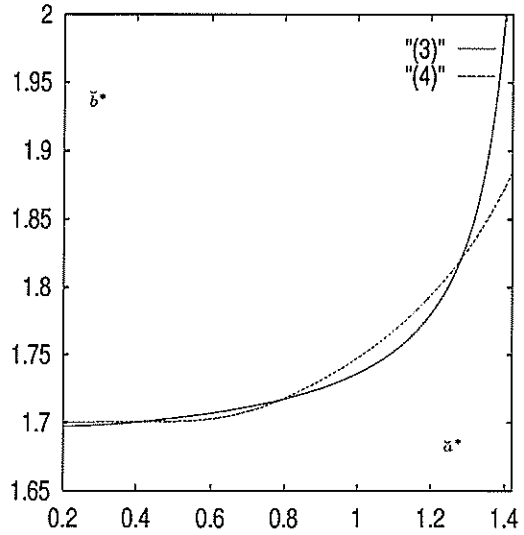

Figure 1: Example with 3 Nash equilibria for SIMUDBP.

2.5 Algorithm for finding the Nash equilibria

We now discuss the algorithm for computing the Nash equilibria. We assume that the probability density functions $f(\breve{a})$ and $g(\breve{b})$ are piecewise linear with n and m pieces respectively. This is justified since any continuous function can be approximated arbitrarily closely by a piecewise linear curve. Also, in our implementation we use piecewise linear curves because they are easy for the user to input.

The algorithm for computing the equilibrium of a sequential game is straightforward. It simply substitutes the contract parameters (ρ, a, b) into (3) and (4).

In the simultaneous games this substitution gives a system of two non-linear simultaneous equations. These equations may have multiple solutions and all of them need to be found. This rules out iterative procedures that only find one solution. That motivated us to design a fast analytic algorithm that finds all solutions. We decompose the search space into rectangles. Since $f(\breve{a})$ and $g(\breve{b})$ are piecewise linear, the simultaneous equations can be solved analytically within each rectangle. The rectangles are $[\breve{a}_i, \breve{a}_{i+1}] \times [\breve{b}_j, \breve{b}_{j+1}]$ where $1 \leq i < n$ and $1 \leq j < m$. Within each rectangle, the algorithm

1. Solves (1) and (2) to get $p_a(\breve{a}*)$ and $p_b(\breve{b}*)$. These functions are quadratic

because $f(\breve{a})$ and $g(\breve{b})$ are linear within the rectangle.

2. Substitutes $p_a(\breve{a}*)$ into (4), and $p_b(\breve{b}*)$ into (3).

3. Substitutes $\breve{a}^*$ from (3) into (4).

4. Reduces the resulting equation to a cubic polynomial in $\breve{b}^*$.

5. Solves for the roots of this polynomial.

6. Calculates the corresponding values for $\breve{a}^*$ using (3).

7. Accepts the solutions that lie within the rectangle.

The algorithm runs in $O(nm)$ time because it is $O(1)$ within each rectangle, and there are $O(nm)$ rectangles.

Of all the equilibria found, it presents social welfare maximizing one(s) to the user, i.e. the one(s) that maximize the sum of the agents' expected payoffs. It can also present all equilibria.

3 Optimizing the contract

So far we discussed how rational agents would decommit under a given contract. Now we take this further by optimizing the contract itself—taking into account that agents will decommit insincerely in Nash equilibrium. Specifically, we present an algorithm and analysis for finding the optimal contract price and decommitment penalties in any given setting for all three protocols.

3.1 Analysis

We first describe the optimization problem. Let π_a be the contractor's expected utility, and π_b the contractee's. For shorthand, we define T_1, T_2, T_3, and T_4:
$T_1 = \int_{\breve{b}^*}^{\infty} g(\breve{b})db$, $T_2 = \int_{\breve{b}^*}^{\infty} bg(\breve{b})db$
$T_3 = \int_{\breve{a}^*}^{\infty} f(\breve{a})da$, $T_4 = \int_{\breve{a}^*}^{\infty} af(\breve{a})da$
We now derive π_a and π_b for each of the games.
SEQD:

$$
\begin{aligned}
\pi_a &= \int_{\check{b}^*(\rho,a,b)}^{\infty} g(\check{b}) \int_{-\infty}^{\infty} f(\check{a})[-\check{a}+b]d\check{a}d\check{b} \\
&\quad + \int_{-\infty}^{\check{b}^*(\rho,a,b)} g(\check{b})[\int_{-\infty}^{\rho-a} f(\check{a})[-\check{a}-a]d\check{a} + \int_{\rho-a}^{\infty} f(\check{a})[-\rho]d\check{a}]d\check{b} \\
&= \check{b}^* T_3 T_1 - E[\check{a}] + T_4 - T_1 T_4 - \check{a}^* T_3 + \check{a}^* - \rho \\
\pi_b &= \int_{\check{b}^*(\rho,a,b)}^{\infty} g(\check{b})[\check{b}-b]d\check{b} + \int_{-\infty}^{\check{b}^*(\rho,a,b)} g(\check{b})[\int_{-\infty}^{\rho-a} f(\check{a})[\check{b}+a]d\check{a} + \int_{\rho-a}^{\infty} f(\check{a})\rho d\check{a}]d\check{b} \\
&= -\check{b}^* T_3 T_1 - E[\check{b}](1-T_3) + T_2 T_3 + \check{a}^* T_3 - \check{a}^* + \rho
\end{aligned}
$$

SIMUDBP:

$$
\begin{aligned}
\pi_a &= \int_{\check{b}^*(\rho,a,b,\check{a}^*)}^{\infty} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[-\check{a}+b-a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[-\check{a}+b]d\check{a}]d\check{b} \\
&\quad + \int_{-\infty}^{\check{b}^*(\rho,a,b,\check{a}^*)} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[-\check{a}-a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[-\rho]d\check{a}]d\check{b} \\
&= -E[\check{a}] + \check{b}^* T_1 T_3 + \check{a}^*(1-T_1)(1-T_3) + T_4(1-T_1) \\
&\quad -\rho(1-T_3+T_1T_3) \\
\pi_b &= \int_{\check{b}^*(\rho,a,b,\check{a}^*)}^{\infty} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[\check{b}-b+a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[\check{b}-b]d\check{a}]d\check{b} \\
&\quad + \int_{-\infty}^{\check{b}^*(\rho,a,b,\check{a}^*)} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[\check{b}+a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[\rho]d\check{a}]d\check{b} \\
&= E[\check{b}](1-T_3) - \check{b}^* T_1 T_3 - \check{a}^*(1-T_1)(1-T_3) + T_3 T_2 \\
&\quad +\rho(1-T_3+T_1T_3)
\end{aligned}
$$

SIMUDNP:

$$
\begin{aligned}
\pi_a &= \int_{\check{b}^*(\rho,a,b,\check{a}^*)}^{\infty} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[-\check{a}]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[-\check{a}+b]d\check{a}]d\check{b} \\
&\quad + \int_{-\infty}^{\check{b}^*(\rho,a,b,\check{a}^*)} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[-\check{a}-a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[-\rho]d\check{a}]d\check{b} \\
&= -E[\check{a}] + T_4(1-T_1) + a(-1+T_1+T_3-T_1T_3) + bT_1T_3 \\
&\quad -\rho(1-T_1)T_3 \\
\pi_b &= \int_{\check{b}^*(\rho,a,b,\check{a}^*)}^{\infty} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[\check{b}]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[\check{b}-b]d\check{a}]d\check{b} \\
&\quad + \int_{-\infty}^{\check{b}^*(\rho,a,b,\check{a}^*)} g(\check{b})[\int_{-\infty}^{\check{a}^*(\rho,a,b,\check{b}^*)} f(\check{a})[\check{b}+a]d\check{a} + \int_{\check{a}^*(\rho,a,b,\check{b}^*)}^{\infty} f(\check{a})[\rho]d\check{a}]d\check{b}
\end{aligned}
$$

$$= E[\breve{b}](1-T_3)+T_2T_3-a(-1+T_1+T_3-T_1T_3)-bT_1T_3 \\ +\rho(1-T_1)T_3$$

For each of the three protocols the social welfare, π, is

$$\pi = \pi_a + \pi_b = -E[\breve{a}] + E[\breve{b}](1-T_3) + T_2T_3 + T_4 - T_1T_4$$

We define the optimal contract to be the contract that maximizes social welfare, i.e. $max_{(\rho,a,b)}\pi$. The expression above shows that π is the same for all three games. Also, π is a function of $(\breve{a}^*, \breve{b}^*)$. So, the original problem of optimizing over a 3-dimensional space (ρ, a, b) reduces to optimizing over a 2-dimensional space $(\breve{a}^*, \breve{b}^*)$.

A rational agent only accepts a contract if it does not decrease his expected payoff. These individual rationality (IR) constraints, $\pi_a \geq E[-\breve{a}]$ and $\pi_b \geq E[\breve{b}]$, define a feasible set in the 3-dimensional space $(\breve{a}^*, \breve{b}^*, \rho)$.

We first derive the unconstrained optima, and then verify that they belong to the feasible set. The necessary conditions for unconstrained optima are: [4]
$\frac{\partial \pi}{\partial \breve{a}^*} = 0 \Leftrightarrow f(\breve{a}^*)(E[\breve{b}] - T_4 - \breve{a}^* + \breve{a}^*T_3) = 0$
$\frac{\partial \pi}{\partial \breve{b}^*} = 0 \Leftrightarrow g(\breve{b}^*)(\breve{b}^*T_1 - T_2) = 0$
Four cases satisfy these equalities. 1-3 are special cases for dealing with *gaps* in $f(\breve{a})$ or $g(\breve{b})$, i.e. regions where the probability density is 0 inside $[\breve{a}_1, \breve{a}_n]$ or $[\breve{b}_1, \breve{b}_m]$:

Case 1: $f(\breve{a}^*) = 0$, $g(\breve{b}^*) = 0$: Each such pair of gaps constitutes a continuous set of locally optimal solutions. The value of π is constant in that region because the values for the definite integrals T_1, T_2, T_3, and T_4 are constant there. Thus, to find that locally optimal value, it suffices to evaluate π at any point in that region. This is done for each pair of gaps. Satisfaction of the IR constraints is verified for each local optimum found as follows. By substituting the values of T_1, T_2, T_3, and T_4 into the equations for π_a and π_b, the IR constraints reduce to a system of two linear inequalities in the two unknowns, $\breve{a}^*$ and $\breve{b}^*$. It is straightforward to solve this system and verify whether any part of the resulting solution lies in the region defined by the pair of gaps. If so, the local optimum is feasible, and is considered as a candidate for being globally optimal.

Case 2: $f(\breve{a}^*) = 0$, $\breve{b}^*T_1 - T_2 = 0$: The algorithm does the following for each gap in $f(\breve{a})$. T_1 and T_2 are constant within the gap. These give a value for

[4] In deriving the formulas on the right, the derivative of an integral with respect to its limits is calculated using the Leibnitz formula [Arfken and Weber, 1995].

$\breve{b}^*$ via the second condition. These are used to calculate the locally optimal π value. Then the values for T_1, T_2, T_3, T_4, and $\breve{b}^*$ are substituted in the equations for π_a and π_b. As a result, the IR constraints give two inequalities for $\breve{a}^*$. If any part of the gap is in that feasible region, the local optimum is feasible, and is considered as a candidate for being globally optimal.

Case 3: $g(\breve{b}^*) = 0, E[\breve{b}] - T_4 - \breve{a}^* + \breve{a}^* T_3 = 0$: Analogous to case 2.

Case 4: $E[\breve{b}] - T_4 - \breve{a}^* + \breve{a}^* T_3 = 0, \breve{b}^* T_1 - T_2 = 0$: These two conditions give a pair of simultaneous equations:

$$\breve{a}^* = \frac{E[\breve{b}] - T_2}{1 - T_1} \tag{5}$$

$$\breve{b}^* = \frac{T_4}{T_3} \tag{6}$$

Equation (5) describes $\breve{a}^*$ as a function of $\breve{b}^*$ and Equation (6) gives $\breve{b}^*$ as a function of $\breve{a}^*$. A local optimum of π exists where the curves intersect. They may or may not intersect, and they can intersect at multiple points:

Theorem 3.1 *Curves (5) and (6) intersect if* $E[\breve{a}] \geq \breve{b}_1$ *and* $E[\breve{b}] \leq \breve{a}_n$. *They do not intersect if* $\breve{b}_1 > \breve{a}_n$. *Otherwise they may or may not intersect. The intersection point is unique if* $\breve{b}_m < E[\breve{a}]$ *or* $\breve{a}_1 > E[\breve{b}]$.

Proof. From (5), in the limit $\breve{b}^* \to \breve{b}_1$, $\breve{a}^* \to \breve{b}_1$ and when $\breve{b}^* \geq \breve{b}_m$, $\breve{a}^* = E[\breve{b}]$. From (6), when $\breve{a}^* \leq \breve{a}_1$, $\breve{b}^* = E[\breve{a}]$ and in the limit $\breve{a}^* \to \breve{a}_n$, $\breve{b}^* \to \breve{a}_n$. Both curves represent strictly increasing functions (the first derivatives are always positive). Note that $T_1 \neq 1 \Rightarrow \breve{b}^* > \breve{b}_1$, and $T_3 \neq 0 \Rightarrow \breve{a}^* < \breve{a}_n$. Thus they intersect if $E[\breve{a}] \geq \breve{b}_1$ and $E[\breve{b}] \leq \breve{a}_n$. This intersection point is unique if $\breve{b}_m < E[\breve{a}]$ or $\breve{a}_1 > E[\breve{b}]$. They do not intersect if $\breve{b}_1 > \breve{a}_n$. Otherwise they may or may not intersect. □

Figure 2 shows that (5) and (6) can intersect at multiple points, i.e. that π can have multiple (local) optima.

We now show that if these locally optimal solutions exist, they lie within the feasible set, i.e. there exists a nonempty range $[LB, UB]$ for the contract price ρ which satisfies the IR constraints. We derive LB and UB using the expressions for π_a, π_b and the IR constraints, $\pi_a \geq -E[\breve{a}]$, $\pi_b \geq E[\breve{b}]$, to get:

SEQD:

$$LB = \breve{b}^* T_3 T_1 + E[\breve{b}] T_3 - T_2 T_3 - \breve{a}^* T_3 + \breve{a}^*$$

$$UB = \breve{b}^* T_3 T_1 + T_4 - T_1 T_4 - \breve{a}^* T_3 + \breve{a}^*$$

SIMUDBP:

$$LB = \frac{\breve{b}^* T_3 T_1 + \breve{a}^* (1 - T_1)(1 - T_3) + E[\breve{b}] T_3 - T_3 T_2}{1 - T_1 + T_3 T_1}$$

$$f(\breve{a})=\begin{cases}0.95\breve{a}-0.52 & \text{if } 0.7 \leq \breve{a} < 0.8\\ 0.24 & \text{if } 0.8 \leq \breve{a} < 0.9\\ 1.43\breve{a}-1.05 & \text{if } 0.9 \leq \breve{a} < 1.0\\ 2.47\breve{a}-2.09 & \text{if } 1.0 \leq \breve{a} < 1.5\\ 5.23\breve{a}-6.22 & \text{if } 1.5 \leq \breve{a} \leq 1.7\\ 0 & \text{otherwise}\end{cases}$$

$$g(\breve{b})=\begin{cases}0.02 & \text{if } 0.1 \leq \breve{b} < 1.41\\ 24.27\breve{b}-34.19 & \text{if } 1.41 \leq \breve{b} < 1.44\\ 0.75 & \text{if } 1.44 \leq \breve{b} \leq 2.72\\ 0 & \text{otherwise}\end{cases}$$

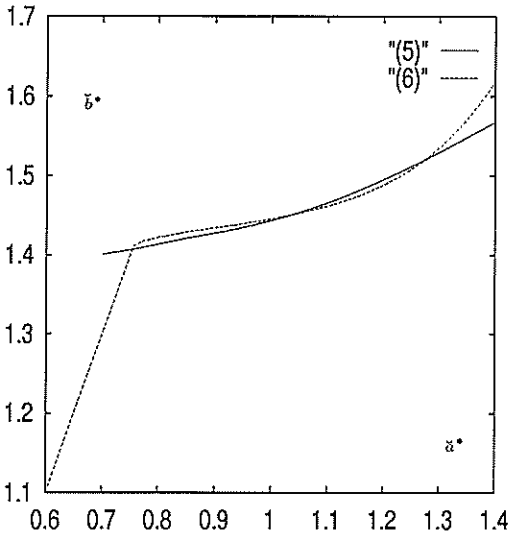

Figure 2: Example with 3 local optima.

$UB = \frac{\breve{b}^* T_3 T_1 + \breve{a}^*(1-T_1)(1-T_3)+T_4(1-T_1)}{1-T_1+T_3T_1}$

SIMUDNP:

$LB = \frac{bT_3T_1 - a(1-T_1)(1-T_3)+(E[\breve{b}]-T_2)T_3}{T_3(1-T_1)}$

$UB = \frac{bT_3T_1 - a(1-T_1)(1-T_3)+T_4-T_1T_4}{T_3(1-T_1)}$

For SIMUDBP games, $1 - T_1 + T_3T_1 \neq 0$. For SIMUDNP, $(1 - T_1)T_3 \neq 0$. This occurs from the form of the integrals, and the fact that they are always between 0 and 1 because they are defined over probability density functions. For each game above, $LB \leq UB$ reduces to $T_3(E[\breve{b}] - T_2) \leq T_4(1 - T_1) \Leftrightarrow \frac{E[\breve{b}]-T_2}{1-T_1} \leq \frac{T_4}{T_3} \Leftrightarrow \breve{a}^* \leq \breve{b}^*$ which can be shown true by the form of (5) and (6).

3.2 Algorithm for finding optimal contracts

Our algorithm for finding optimal contracts first checks the special cases 1-3 as described above. This takes $O(nm)$ time because there are at most nm pairs of gaps. The optima of case 4 are determined by computing the intersection points of curves (5) and (6). Multiple solutions might exist, which rules out the use of iterative methods that only find one solution. Our algorithm uses the same rectangular decomposition as our equilibrium finding algorithm presented above. In each rectangle it

1. Solves T_1, T_2, T_3, and T_4 to get quadratic equations for them (because $f(\breve{a})$ and $g(\breve{b})$ are linear within the rectangle).
2. Substitutes these into (5) and (6).
3. Substitutes $\breve{a}^*$ from (5) into (6).
4. Reduces the resulting equation to a 9th degree polynomial in $\breve{b}^*$.
5. Solves for the roots of this polynomial.
6. Calculates the corresponding values of $\breve{a}^*$ from (5).
7. Accepts solutions $(\breve{a}^*, \breve{b}^*)$ that lie in the rectangle.

The algorithm runs in $O(nm)$ time because it is $O(1)$ within each rectangle, and there are $O(nm)$ rectangles. Finally, the candidate set of local optima resulting from cases 1-4 is scanned for the global optima.

3.3 The fair optimal contract

An optimal pair $(\breve{a}^*, \breve{b}^*)$ defines a set of welfare maximizing contracts (ρ, a, b) which differ based on how the *excess*—i.e. expected gain over full commitment contracts—is divided among the contract parties. Our implementation shows all the optimal feasible contracts to the user and visualizes how the different choices lead to different division of expected payoff (π_a and π_b as a function of $\rho \in [LB, UB]$). It also suggests a *fair* contract which divides the excess equally. It turns out that the fair contract price is $\rho = \frac{LB+UB}{2}$. The fair values for the penalties are then calculated using (3) and (4).

4 Implementation

Using the algorithms of this paper, we provide a client-server based contract optimizing service on the web (`http://ecommerce.cs.wustl.edu/contracts.html`) as part of *eMediator*, our next generation electronic commerce server. The client, a Java applet, implements the GUI and communicates to the server over the web using CGI/Perl. The server, written in C, executes the computations. The client accepts piecewise linear functions

$f(\breve{a})$ and $g(\breve{b})$. The user can input these graphically or as text. Each probability density functions is automatically scaled so that its integral is 1. The output interface displays either the decommitting thresholds and probabilities for a given contract, or the optimal feasible contracts together with the fair optimal feasible contract.

The server computes the roots of polynomials in our rectangular decomposition algorithms. We use Laguer's root finding method to compute the roots [Press *et al.*, 1993]. It finds all roots simultaneously and polishes roots to reduce numerical imprecision. If the roots are close to the boundaries of the rectangle, imprecision may result in these roots being calculated just outside the rectangle. To accommodate for this, we extend the rectangle in all directions by ϵ. That allows us to capture all those roots that would otherwise be overlooked.

5 Generalization to 3-agent contracts

The method can be generalized to contracts involving more than two agents. We demonstrate this via 3-agent games. Let there be one contractor, a, and two contractees, b and c. If even one party decommits the contract breaks down between all 3 parties. Let the density functions of outside offers be $f(\breve{a})$, $g(\breve{b})$, and $h(\breve{c})$. The IR constraints are $\pi_a \geq E[-\breve{a}]$, $\pi_b \geq E[\breve{b}]$, and $\pi_c \geq E[\breve{c}]$. The contract price, ρ, paid by the contractor, is paid in two parts, ρ_1 to b, and $\rho - \rho_1$ to c. The decommitment penalties are a_b, a_c, b_a, b_c, c_a, and c_b, see Figure 3.

The sequential game will have 3! variants depending on the decommitting order. The simultaneous games differ based on whether two parties on simultaneous decommitment pay each other or not. Due to limited space we only present the formulae for SIMUDBP:
$p_a = -\int_{-\infty}^{\breve{a}^*} f(\breve{a})da, p_b = \int_{\breve{b}^*}^{\infty} g(\breve{b})db, p_c = \int_{c*}^{\infty} h(c)dc$
$\breve{a}^* = \rho - \frac{a_b+a_c}{(1-p_b)(1-p_c)}$ when $p_b < 1$ and $p_c < 1$
$\breve{b}^* = \rho_1 + \frac{b_a+b_c}{(1-p_a)(1-p_c)}$ when $p_a < 1$ and $p_c < 1$
$\breve{c}^* = \rho - \rho_1 + \frac{c_a+c_b}{(1-p_a)(1-p_b)}$ when $p_a < 1$ and $p_b < 1$
These expressions are analogous to 2-agent SIMUDBP. We get similar expressions for π_a, π_b, π_c,and π. Our rectangular decomposition algorithms for computing the Nash equilibria extend by using cuboids instead of rectangles. Nonuniqueness extends to this case. The optimization problem also has sim-

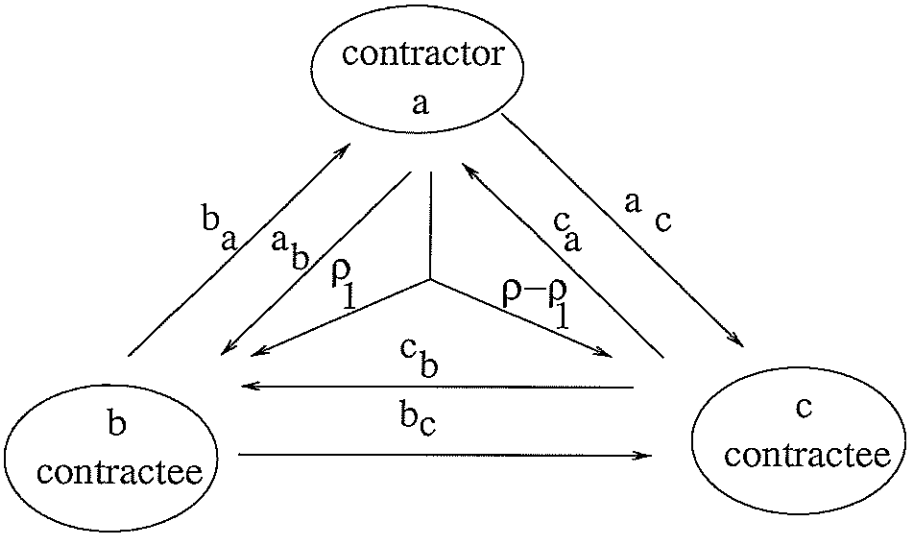

Figure 3: 3-agent contract

ilar characteristics. π is a function of $\breve{a}^*, \breve{b}^*, \breve{c}^*$. Solving for the roots of the first derivatives yields three equations in three variables. Our rectangular decomposition algorithms for calculating the optima extend to this case by using cuboids instead of rectangles. Similar expressions/results occur for the 3-agent generalization of SIMUDNP.

6 Conclusions

Leveled commitment contract are a practical way of capitalizing on future uncertainties in negotiation among self-interested agents. This paper operationalizes them by presenting two algorithms. The first one takes a given contract and computes the Nash equilibrium decommitting thresholds and decommitting probabilities for rational agents. The second algorithm optimizes the contract price and penalties so as to maximize the sum of the agents' expected payoffs —taking into account that agents decommit insincerely. Using these algorithms, a contract optimizing service is provided on the web.

References

[Arfken and Weber, 1995] George B. Arfken and Hans-Jurgen Weber. *Mathematical Methods for Physicists.* Academic Press, 1995.

[Calamari and Perillo, 1977] John D Calamari and Joseph M Perillo. *The Law of Contracts.* West Publishing Co., 2nd edition, 1977.

[Diamond and Maskin, 1979] Peter A Diamond and Eric Maskin. An equilibrium analysis of search and breach of contract, I: Steady states. *Bell J. of Economics*, 10:282–316, 1979.

[Kraus, 1993] Sarit Kraus. Agents contracting tasks in non-collaborative environments. In *Proc. Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 243–248, July 1993.

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

[Posner, 1977] Richard A Posner. *Economic Analysis of Law.* Little, Brown and Company, 2nd edition, 1977.

[Press *et al.*, 1993] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge University Press, 1993.

[Raiffa, 1982] H. Raiffa. *The Art and Science of Negotiation.* Harvard Univ. Press, Cambridge, Mass., 1982.

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

[Sandholm and Lesser, 1996] Tuomas W Sandholm and Victor R Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 126–133, Portland, OR, August 1996.

[Sandholm, 1993] Tuomas W Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.

[Sen, 1993] Sandip Sen. *Tradeoffs in Contract-Based Distributed Scheduling.* PhD thesis, Univ. of Michigan, 1993.

[Smith, 1980] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.