Equilibrium Analysis of the Possibilities of Unenforced Exchange in Multiagent Systems

Tuomas W. Sandholm and Victor R. Lesser
{sandholm, lesser}@cs.umass.edu
University of Massachusetts at Amherst
Computer Science Department
Amherst, MA 01003

Abstract

In multiagent systems, interaction protocols are usually enforced by law. Enforcement is problematic among computational agents, because they may operate under incomplete or different laws, the laws may not be uniformly enforced, and the agents can vanish easily. This paper presents an enforcement free method for carrying out exchanges so that both agents are motivated to abide to their contract. This is achieved by splitting the exchanged goods into partial exchanges so that at each step, both agents benefit more from the future of the exchange than from vanishing with the goods or payment. The conditions for such exchange are presented in general, and the maximum deliveries and payments—for any point in the exchange—are solved for. Similar analysis is carried out for the case, where the agents' current actions affect their future contracts. Strategic delaying is also discussed. The paper presents a fast algorithm that will find a sequence of independent partial deliveries in a way that enables unenforced exchange if such a sequence exists. This problem cannot be solved in polynomial time if the partial deliveries are interdependent. Finally, the paper shows that the unenforced exchange scheme hinders unfair renegotiation.[1]

1 Introduction

In cooperative distributed problem solving [Durfee *et al.*, 1989], the system designer imposes an interaction *protocol* and a *strategy* (a mapping from state history to actions; a way to use the protocol) for each agent. In multiagent systems [Sandholm and Lesser, 1995c; 1995a; 1995b; Rosenschein and Zlotkin, 1994; Durfee *et al.*, 1993; Kraus *et al.*, 1992; Wellman, 1992], the agents are provided with an interaction protocol, but each agent may choose its own strategy. This allows the agents to be constructed by separate designers and/or represent different real world parties. Agents in such systems often act based on self-interest, and the protocols have to be constructed accordingly. An example interaction protocol is the *auction*, where some agents bid to take responsibility for a task, which is awarded to the lowest price bidder. The bids are binding: if an agent makes a bid and the task is awarded to it, it must take responsibility for the task at that price. Among real world agents, this protocol is enforced by law.

Such enforced protocols are problematic when used among computational agents. First, there may be a lack of laws for interactions of computational agents, or the agents may be governed by different laws—e.g. sited in different countries. It may also be the case that the laws are not strictly enforced or that enforcing them (e.g. by litigation) is impractically expensive. We would like the agents' interactions to work properly independent of fluctuations in enforcement. Secondly, a computational agent may vanish at any point in time, e.g. by killing its own process. Thus, the laws cannot be enforced unless the terminated agent represented some real world party and the connection between the agent and the real world party can be traced. For example, the Telescript technology [General Magic, Inc., 1994] follows the approach of strictly trying to tie each agent to its real world party. On the contrary, we analyze exchanges among more autonomous agents and study possibilities of exchange without enforcement (e.g. with unknown real world parties or no litigation possibility). In cases where this type of exchange is possible, it is clearly preferable to the strictly enforced mode of exchange due to savings in enforcement costs and lack of enforcement uncertainty.

The fulfillment of a mutual contract can be viewed as one agent delivering and the other agent paying. We propose a method for carrying out such an exchange without enforcement. The exchange is managed so that for both agents—supplier and demander—at any point in the exchange, the future gains from carrying out the rest of the exchange (*cooperating* according to the contract) are larger than the gains from *defecting*. Defection is equivalent to prematurely terminating the exchange by vanishing. For example, defection may be beneficial to a demander agent if the supplier agent has delivered much more than what the demander has yet paid for. By intelligently splitting the exchange into smaller portions, the agents can avoid situations where at least one of them is motivated to defect. We will call a sequence of deliveries and payments *safe* if neither agent is motivated to defect at any point in the exchange. The basic idea of enhancing cooperation by making the present less im-

[1]Supported by ARPA contract N00014-92-J-1698. T. Sandholm also supported by the Finnish Culture Foundation, Honkanen Foundation, and Ella and George Ehrnrooth Foundation. The content does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. An early version of this paper appeared in a workshop [Sandholm and Lesser, 1994].

portant compared to the future has been suggested for example in [Axelrod, 1984].

We propose an *exchange strategy manager* module to be added to each agent's architecture. This module is potentially different for each agent. Its role is to schedule the agent's deliveries (or payments) in such a way that the opponent is not motivated to defect at any point in the exchange. This is in the agent's self-interest. The exchange strategy manager also provides the agent's negotiator module with information on whether a certain proposed contract can be carried out safely. Unless protocol enforcement is guaranteed, the negotiator should only agree to contracts that can be executed so that the opponent is not motivated to defect at any point of the exchange. Automated negotiation has been mostly studied with respect to *ex ante* rationality: what contracts seem desirable to the agents before they are carried out [Sandholm and Lesser, 1995b; 1995c; 1995a; Sandholm, 1993; Rosenschein and Zlotkin, 1994; Kraus *et al.*, 1992; Wellman, 1992; Durfee *et al.*, 1993]. We suggest that contracts should also fulfill the condition of *ex post* rationality: abiding to the contract should be desirable to the agents at each step of the carrying out of the contract. *Ex post* conditions were studied in multiagent planning without payments in [Brainov, 1994].

This paper is organized as follows. Section 2 handles one exchange in isolation. Conditions for safe exchange are derived and an inherent restriction concerning the completion of the exchange is identified. Section 3 takes the agents' future transactions into account in describing safe exchange in order to solve the completion problem. The role of time in an exchange is discussed in Section 4. Section 5 analyses the case, where the delivering order of independent goods can be varied. A quadratic sequencing algorithm is presented that finds a safe sequence if one exists. Section 5.1 studies sequencing of interdependent goods. Section 6 describes the advantages of safe exchange with respect to unfair renegotiation, and Section 7 concludes.

2 Exchanging goods and payments

Our model analyzes exchanging goods—information, computation services, or other types—for payments. The exchange proceeds on two axis: the portion of goods of the contract delivered by exchange step n is called $x_n \in [0,1]$, and the cumulative payment so far is $p_n \in [0, p^{contr}]$. p^{contr} is the total payment specified in the contract. The agents can make simultaneous moves and they observe the other agent's moves so far. They value the goods x according to nondecreasing functions that are in equivalent units of payment p. The supplier's value function $v_s(x)$ describes how much cost the supplier incurs by generating and delivering x. The demander's value function $v_d(x)$ describes what the goods x are worth to the demander. Trivially, $v_s(0) = v_d(0) = 0$.

At any point in an exchange, an agent has the options of defecting or cooperating. Defecting gives no added gains that have not already been received (when already accounting for the opponent's move on the current step of the exchange) and no added costs, so its net benefit is 0. Therefore a net benefit maximizing supplier agent will cooperate throughout the rest of the exchange from an arbitrary point (x,p) of the exchange if its future compensation is at least as great as its future cost[2], i.e. $p^{contr} - p \geq v_s(1) - v_s(x)$. This assumes that the demander will actually finally increase cumulative payment to p^{contr}. An equilibrium analysis with respect to this issue will be presented shortly. To facilitate that analysis, $p^{max}(x)$ is defined based on the above intuition:

$$p^{max}(x) \stackrel{def}{=} p^{contr} - v_s(1) + v_s(x). \qquad (1)$$

A rational demander agent will cooperate throughout the rest of the exchange from an arbitrary point (x,p) if the future compensation it has to pay is smaller than or equal to its added value, i.e. $p^{contr} - p \leq v_d(1) - v_d(x)$. This assumes that the supplier will finally increase total delivery to 1, which will be shown to be an equilibrium shortly. Now, $p^{min}(x)$ is defined, Fig. 1 left:

$$p^{min}(x) \stackrel{def}{=} p^{contr} - v_d(1) + v_d(x). \qquad (2)$$

Clearly, $p^{max}(x)$ and $p^{min}(x)$ are nondecreasing in x. For the supplier to have agreed to the contract, $p^{max}(0) \geq 0$, and for the demander, $p^{min}(0) \leq 0$.

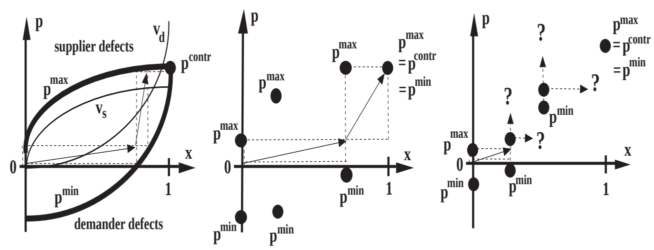

Figure 1: *Left: example of safe exchange with continuous goods. Middle: safe exchange of discrete goods possible. Right: safe exchange of discrete goods not possible.*

If the agents do not know each other's value functions, they can use bounds they know. The supplier is safe using an upper bound for $p^{min}(x)$, i.e. a lower bound for $v_d(1)$ and an upper bound for $v_d(x)$. The demander is safe using a lower bound for $p^{max}(x)$. Although the agents are safe using these bounds, even possible exchanges are disabled if the bounds are too far off.

The next sections present an equilibrium study of when safe exchange can actually occur. The analysis is slightly different for discrete and continuous goods.

2.1 Discrete goods

Discrete goods are goods that are inherently split into atomic chunks. Such chunks cannot be further split, and we assume in this section that the delivery order of the chunks is fixed. For example in the TRACONET (TRAnsportation COoperation NET) multiagent system [Sandholm, 1993], agents representing dispatch centers negotiated over who's vehicles should transport

[2] If equality holds, the agent is indifferent between cooperating and defecting. Throughout this paper we assume that indifferent agents will cooperate. Note also that throughout the paper we analyze only remaining payoffs, not total payoffs, because the already incurred payoffs are constant with respect to the remaining game.

which parcels. Taking care of one parcel is an atomic chunk because the task cannot be split. Sometimes a contract between two agents involved multiple tasks (in order to avoid local optima in distributed task allocation [Sandholm, 1993]) so the total exchange could have been split into smaller parts. The following theorems describe the conditions for safe exchange of discrete goods.

Definition 2.1 *Supplier's strategy S_s: At any point of the exchange, if $p_n \le p^{max}(x_n)$ deliver an amount such that cumulative delivery $x_{n+1} = max\{x \in X | p^{min}(x) \le p_n\}$. If $p_n > p^{max}(x_n)$, exit.*

Definition 2.2 *Demander's strategy S_d: At any point of the exchange, if $p_n \ge p^{min}(x_n)$, pay an amount such that cumulative payment $p_{n+1} = p^{max}(x_n)$. If $p_n < p^{min}(x_n)$, exit.*

Theorem 2.1 [3] *With a finite number of discrete goods (discrete $X \subset [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies S_s and S_d form a subgame perfect Nash equilibrium if for every two consecutive amounts of cumulative delivery $x, x' \in X$, $p^{max}(x) \ge p^{min}(x')$.*[4] *Furthermore, the exchange is completed. Fig. 1 middle.*

Theorem 2.2 *With a finite number of discrete goods (discrete $X \subset [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, there is no subgame perfect Nash equilibrium leading to completion of the exchange if for some two consecutive amounts of cumulative delivery $x, x' \in X$, $p^{max}(x) < p^{min}(x')$. See Fig. 1 right.*

Nash equilibrium [Nash, 1950; Kreps, 1990] means that each agent is motivated to abide to its specified strategy given that the other agent abides to its specified strategy. Subgame perfection [Selten, 1965; Kreps, 1990] means that the equilibrium is a Nash equilibrium at any point (x_n, p_n) of the exchange, not only the beginning of it. This means that the equilibrium remains an equilibrium after the agents have partially carried out the exchange. Furthermore, it is an equilibrium at points (x, p) of the exchange that will actually not be reached by the agents in the exchange process. For these reasons, subgame perfection precludes incredible threats/promises and provides some robustness against external perturbances.

From the condition $p^{max}(x) \ge p^{min}(x')$ and the fact that $p^{min}(x)$ is nondecreasing we see that the following has to hold for safe exchange: $p^{max}(x) \ge p^{min}(x)$. In terms of the agents' value functions this can be written as $v_d(x) - v_s(x) \le v_d(1) - v_s(1)$. This means that the agents' combined profit must be higher (or equal) at $x = 1$ than at any other $x \in [0, 1[$. If the agents would have been better off by making the contract for a smaller amount of goods, an isolated safe exchange is impossible. Furthermore, at $x = 0$ this gives $v_s(1) \le v_d(1)$, which is an intuitive condition for the contract to have been made in the first place. Specifically, $v_s(1) \le p^{contr} \le v_d(1)$, Fig. 1 left.

Theorems 2.1 and 2.2 state that rather stringent conditions have to be met to enable unenforced isolated exchange of discrete goods. Substituting $x = 1$ in the definitions of $p^{max}(x)$ and $p^{min}(x)$ gives $p^{max}(1) = p^{min}(1) = p^{contr}$. According to the theorems, safe exchange is possible only if $p^{max}(x) \ge p^{min}(x')$ for any two consecutive x and x'. Let us call the size of the last delivery Δx. So for safe exchange the following has to hold: $p^{max}(1 - \Delta x) \ge p^{min}(1) = p^{max}(1)$. Thus the increasing function $p^{max}(x)$ has to be constant during the last step (Fig. 1 middle). This means that the supplier's value function $v_s(x)$ is constant. So an isolated safe exchange is possible only if the supplier does not incur any cost from generating and delivering the last chunk. This occurs for example when the supplier has had to acquire a number of the last deliverables atomically. Its cost of acquiring the deliverables can be entirely attributed to the first one, while it can deliver these items separately with only the first one increasing $v_s(x)$ (assuming negligible costs of physically delivering). This may not occur very often in practise (Fig. 1 right). Intuitively, when there is no future benefit to be gained from exchanging, agents are better off defecting on the current move.

If this problem occurs in an isolated exchange of a finite number of discrete goods, it spoils the entire exchange. On the last move the supplier does not want to increase delivery to $x = 1$, because the demander would defect. Similarly, the demander does not want to increase cumulative payment above $p^{max}(1 - \Delta x)$, because the supplier would defect. Both agents know that the last part of the exchange will not take place due to this. So they can analyze the exchange as if it did not have the last part. Now the second to last part has the same problem (unless the supplier can deliver that part without cost): neither agent wants to initiate that part. Again, both agents know this and so on. This backward induction can be carried out up to the first exchange. So, neither agent will make any move, and the exchange will not take place. Theoretically, there can be an infinite number of discrete goods. In such cases this exchange spoiling backward induction argument does not apply because at no point can an agent say that the next move is the last. Backward induction is inapplicable also with continuous goods. This facilitates safe unenforced exchange of continuous goods, as discussed in the next section. In both the discrete and the continuous case, the problem of requiring that the supplier can deliver the last part without costs can be overcome by considering related future interactions of the agent, Sec. 3.

2.2 Continuous goods

This section analyzes the exchange of continuous goods, i.e. goods that can be split arbitrarily. First, two conditions for safe exchange are presented. Intuitively, the first one states the conditions under which a safe exchange can proceed to some amount of cumulative delivery. The second one states the conditions under which safe exchange can proceed from some amount of cumulative delivery.

[3]Proofs of the theorems are omitted due to limited space. They will be published in an extended version of this paper.

[4]The equilibria in Theorems 2.1, 2.3, 4.1 and 4.2 are not unique. For example, the strategies that specify that the demander never pays anything and the supplier never delivers anything at any point of the exchange also form a subgame perfect Nash equilibrium.

Condition 2.1 ***Reachability.*** *(Fig. 2) For every point $x^* \in \,]0,1]$,*

1. *$p^{max}(x^*) = p^{min}(x^*)$, and $p^{max}(x)$ is constant in some left neighborhood of x^*, or*
2. *$p^{max}(x^*) > p^{min}(x^*)$, and $\lim_{x \to x^*-} p^{max}(x) > p^{min}(x^*)$, or*
3. *$p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*-} p^{max}(x) = p^{min}(x^*)$, and $p^{max}(x)$ is constant in some left neighborhood of x^*.*

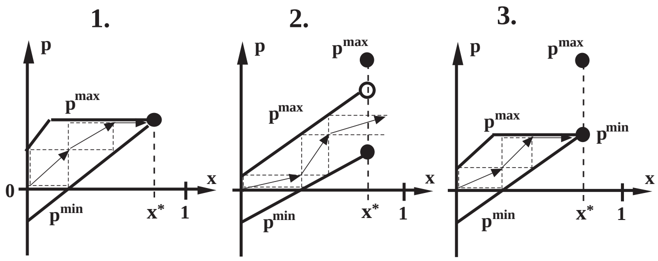

Figure 2: *Exchanging continuous goods: reaching a point.*

Condition 2.2 ***Departability.*** *(Fig. 3) For every point $x^* \in [0,1[$,*

1. *$p^{max}(x^*) = p^{min}(x^*)$, and $p^{min}(x)$ is constant in some right neighborhood of x^*, or*
2. *$p^{max}(x^*) > p^{min}(x^*)$, and $\lim_{x \to x^*+} p^{min}(x) < p^{max}(x^*)$, or*
3. *$p^{max}(x^*) > p^{min}(x^*)$, $\lim_{x \to x^*+} p^{min}(x) = p^{max}(x^*)$, and $p^{min}(x)$ is constant in some right neighborhood of x^*.*

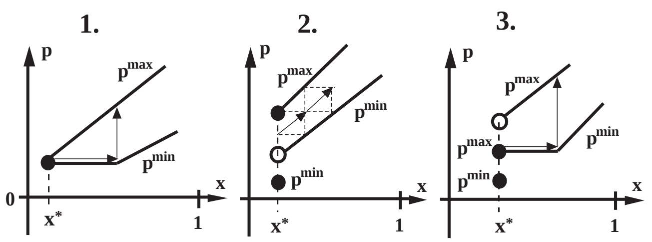

Figure 3: *Exchanging continuous goods: departing from a point.*

Theorem 2.3 *With continuous goods ($X = [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies S_s and S_d form a subgame perfect Nash equilibrium if conditions 2.1 and 2.2 hold. Furthermore, the exchange is completed.*

Theorem 2.4 *With continuous goods ($X = [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, there is no subgame perfect Nash equilibrium leading to the completion of the exchange if conditions 2.1 and 2.2 do not hold.*

Theorems 2.3 and 2.4 state that isolated unenforced exchange of continuous goods is safe if some initial delivery can be made, every intermediate amount of delivery can be reached and departed, and the final amount of delivery can be reached without exceeding $p^{max}(x)$ or moving below $p^{min}(x)$. These theorems do not assume continuity of $p^{max}(x)$ (equivalently $v_s(x)$) or $p^{min}(x)$ (equivalently $v_d(x)$). Neither do they assume that $p^{max}(x)$ or $p^{min}(x)$ is strictly increasing. If $p^{max}(x)$ and $p^{min}(x)$ are continuous, the exchange can be carried out safely if and only if $\forall x \in [0,1]$, either $p^{max}(x) > p^{min}(x)$ or $p^{max}(x) = p^{min}(x)$ and $p^{max}(x)$ is constant in a left neighborhood of x and $p^{min}(x)$ is constant in a right neighborhood of x. If in addition to continuity, $p^{max}(x)$ and $p^{min}(x)$ are strictly increasing, the exchange can be made safely if and only if $\forall x \in [0,1], p^{max}(x) > p^{min}(x)$.

Isolated safe exchange can be problematic also in the case of continuous goods. Substituting $x = 1$ into the definitions of $p^{max}(x)$ and $p^{min}(x)$, we see that $p^{max}(1) = p^{contr} = p^{min}(1)$. From case 1 of condition 2.1 we see that full delivery ($x = 1$) can be reached only if $p^{max}(x)$ is constant in some left neighborhood of $x = 1$. If the value function of the supplier $v_s(x)$ is not constant in the end of the exchange, the exchange cannot be completed. So an isolated safe exchange is possible only if the supplier does not incur any cost from generating and delivering the last portion of the goods, which was discussed in Section 2.1. This problem is less severe than in the case of finitely many discrete goods because the size of the deliveries can be made decreasing and arbitrarily small—thus making the number of deliveries infinite (Fig. 1 left). This allows the agents to reach a cumulative delivery that is arbitrarily close to 1 because the backward induction argument that disabled the entire exchange in the case of discrete goods does not hold. There is no particular exchange step at which the agents could reason that neither will make a move.

3 Extension 1: Non-isolated exchange

Often an agent interacts with other agents more than once. One interaction may affect the agent's future interactions. For example, if an agent defects in the current exchange, its counterpart may not want to take on future contracts with that agent. Moreover, the counterpart can notify other agents that the agent defected. Thus, the agent's interactions with third parties may also be hindered by defecting in the current exchange. The hindering future impact of a defection can be thought of as an extra cost. This future cost may motivate agents to cooperate in the current exchange even if it would be rational to defect in it when considered in isolation. The methods for calculating defection impacts on the future are beyond the scope of this paper. We assume that both agents know their own and their opponent's defection costs. We denote the supplier's defection cost by c_s^{def} and the demander's by c_d^{def}. The defection costs can be incorporated into the model by redefining $p^{max}(x)$ and $p^{min}(x)$, Fig. 4:

$$p^{max'}(x) \stackrel{def}{=} p^{contr} - v_s(1) + v_s(x) + c_s^{def}. \qquad (3)$$

$$p^{min'}(x) \stackrel{def}{=} p^{contr} - v_d(1) + v_d(x) - c_d^{def}. \qquad (4)$$

In isolated exchange, substituting $x = 1$ in the definitions of $p^{max}(x)$ and $p^{min}(x)$ gave $p^{min}(1) = p^{contr} = p^{max}(1)$. This led to the problem that the exchange could not be carried out to completion—unless $p^{max}(x)$ was constant in the end of the exchange. In non-isolated exchange, substituting $x = 1$ gives $p^{min'}(1) = p^{contr} - c_d^{def}$ and $p^{max'}(x) = p^{contr} + c_s^{def}$. The defection penalties give leeway to the exchange, thus possibly enabling safe exchange to be completed even if $p^{max}(x)$ is not constant in the end. The contract price p^{contr} could be exceeded due to this leeway. To avoid this, the demander's strategy can be modified so that at any point in the exchange, the demander increases cumulative payment to $min(p^{contr}, p^{max'}(x))$ instead of $p^{max'}(x)$. This will not hinder exchange, because the condition takes effect after the full contract price has been paid. Non-isolated exchange is more fruitful than isolated exchange, because it facilitates safe completion. The theorems on the possibilities of subgame perfect Nash equilibrium exchange (2.1, 2.2, 2.3, and 2.4) apply directly to the case of non-isolated exchange with the new definitions $p^{max'}(x)$ and $p^{min'}(x)$ substituted in place of $p^{max}(x)$ and $p^{min}(x)$, and the minor modification in the demander's strategy.

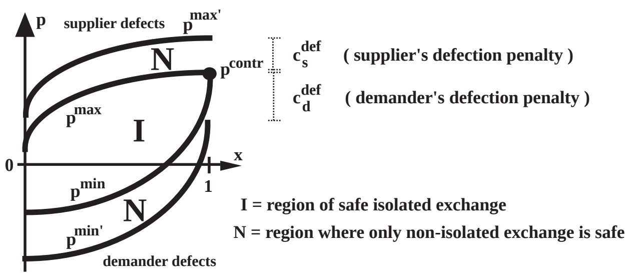

Figure 4: *Defection penalties of non-isolated exchange give leeway to safe moves.*

If the agent does not know the defection cost of the opponent, it can use a lower bound for that cost. This way the agent is safe, but if the bound is too far off, even possible exchanges are disabled.

4 Extension 2: The role of time

This section addresses real time in the exchange: will an exchange take place immediately, or will it be infinitely postponed, or something in between? Nonincreasing discount functions $f(t_n)$, $(0 \leq f(t_n) \leq 1, f(0) = 1)$ are assumed. Subscripts s and d distinguish between the supplier and the demander, and superscripts p and v characterize whether the discount applies to payment or the value of goods. For example, using constant interest rate (r) compounded interest, the discount function is $f(t_n) = e^{-rt_n}$. First, the role of time in isolated exchange is analyzed. Real time is incorporated into the model by allowing the agents to postpone their moves. During the time that one agent is postponing, the other agent can make a delivery or a payment, at which point the first agent can redecide its postponing decision. The players' strategies are redefined to handle time:

Definition 4.1 *Supplier's strategy S_s^{timed}: At any point of the exchange, immediately deliver an amount such that cumulative delivery $x_{n+1} = max\{x \in X | p^{min}(x) \leq p_n\}$ if $p_n \leq p^{max}(x_n)$. Exit if $p_n > p^{max}(x_n)$.*

Definition 4.2 *Demander's strategy S_d^{timed}: At any point of the exchange, immediately pay an amount such that cumulative payment $p_{n+1} = p^{max}(x_n)$. Exit if $p^{min}(x_n) > p_n$.*

The following theorem states that neither agent is motivated to unilaterally deviate from immediate exchange if certain conditions hold on the discount factors.

Theorem 4.1 *With a finite number of discrete goods (discrete $X \subset [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies S_s^{timed} and S_d^{timed} form a Nash equilibrium if for every two consecutive amounts of cumulative delivery x and x', $p^{max}(x) \geq p^{min}(x')$, and $\forall t_n \geq 0, f_s^p(t_n) \leq f_s^v(t_n), f_d^p(t_n) \geq f_d^v(t_n)$. The equilibrium is a Nash equilibrium in every subgame that is reached and the exchange is completed immediately.*

So, isolated unenforced exchange is feasible if the supplier discounts payments more sharply (or equally) than production costs and the demander discounts the value of goods more sharply (or equally) than payment. The condition on the supplier's discount functions is rather natural. For example in a stable environment, the supplier's current value of producing an item should remain constant, but obviously payment is discounted. The condition on the demander's discount functions is more stringent. It is realistic in the case where the demander needs the goods urgently. An agent need not know the opponent's exact discount functions. It is sufficient to know whether they fulfill the conditions. The equilibrium concept of the theorem is slightly weaker than subgame perfection because it only guarantees that the equilibrium is a Nash equilibrium in subgames that are reached—not all subgames. In practise this means that if, for some unknown reason, the exchange has been delayed, it is not guaranteed that the agents are motivated to proceed immediately or at all. For example, in a subgame where $f_s^p(t_n) = 0, f_s^v(t_n) > 0$ for the current t_n, the supplier is not motivated to proceed immediately because no payment by the demander can compensate for any cost incurred by the supplier's delivering.

Clearly, by Theorem 2.2, immediate exchange is not possible if the condition on the consecutive x's does not hold. Similarly, by Theorem 2.4, immediate exchange is not possible with continuous goods if conditions 2.1 (reachability) and 2.2 (departability) do not hold. If they do hold, immediate exchange is possible also in the continuous case:

Theorem 4.2 *With continuous goods ($X = [0,1]$), for nondecreasing $p^{min}(x)$ and $p^{max}(x)$, the strategies S_s^{timed} and S_d^{timed} form a Nash equilibrium if conditions 2.1 and 2.2 hold, and $\forall t_n \geq 0, f_s^p(t_n) \leq f_s^v(t_n), f_d^p(t_n) \geq f_d^v(t_n)$. Furthermore, the equilibrium is a Nash equilibrium in every subgame that is reached and the exchange is completed immediately.*

If the conditions on the discount functions do not hold, the outcomes vary. For example, a supplier wants to carry out the exchange at a time t_n when its $f_s^p(t_n)$ is high and $f_s^v(t_n)$ is low. This may or may not coincide with the time when the demander wants to move. The exact forms of the discount functions define whether the exchange can be carried out in equilibrium immediately, by slightly postponing (different moves in the exchange may be postponed different amounts), or only by postponing indefinitely.

Next, it is shown that time discounts reduce the advantages of non-isolated exchange. We assume that the *current value* of each agent's defection cost does not change—which seems realistic. If an agent discounts payments, this means that its *absolute value* of the defection cost increases with time. The following theorem states that with certain types of discount functions, the exchange cannot proceed outside of the region of isolated exchange (I in Fig. 4) without being delayed. The result that the discount factors on payments need to reach 0 usually means that the delay is indefinitely long. Thus, in such settings, taking advantage of the defection penalties of non-isolated exchange (by moving into region N in Fig. 4) to facilitate safe exchange is usually infeasible. Intuitively, an agent wants to postpone a negative net benefit into the future where it is heavily discounted.

Theorem 4.3 *If $\lim_{t\to\infty} f_s^v(t) = 0$ in any subgame where $p^{max}(x_n) < p_n \leq p^{max'}(x_n)$, and $\forall t \geq t_n, f_s^p(t) \leq f_s^v(t)$, there is no subgame perfect Nash equilibrium that results in reaching $(1, p^{contr})$ before (supplier's) delays have caused $f_s^v(t) = f_s^p(t) = 0$. Similarly, if $\lim_{t\to\infty} f_d^p(t) = 0$ in any subgame where $p^{min'}(x_n) \leq p_n < p^{min}(x_n)$, and $\forall t \geq t_n, f_d^p(t) \geq f_d^v(t)$, there is no subgame perfect Nash equilibrium that results in reaching $(1, p^{contr})$ before (demander's) delays have caused $f_d^p(t) = f_d^v(t) = 0$.*

The conditions $\lim_{t\to\infty} f_d^p(t) = 0$ and $f_s^p(t) \leq f_s^v(t)$ are almost always true. The condition $f_d^p(t) \geq f_d^v(t)$ is true if the demander needs the goods urgently. The supplier's discount function for its goods need not approach 0 however. Its cost of producing goods (discounted to present) may not even decrease with the production date. This may sometimes allow the demander to facilitate exchange by safely over-paying and moving into the upper region N in Figure 4.

The negative result (Theorem 4.3) stems from not considering indefinite postponing a violation of the contract. This can be changed by specifying *deadlines* or *lateness penalty schedules* for the agents in the contract. If the contract is not abided to (e.g. deadlines not honored or lateness penalties not paid), the defecting agent will suffer the defection penalty (c_s^{def} or c_d^{def}) due to how its defection will affect its future contracts. So, strictly speaking a contract matters only in non-isolated exchange, and therefore forcing timely exchange by deadlines or lateness penalties is possible only in such cases. This highlights the value of Theorems 4.1 and 4.2 for isolated exchange that guarantee that immediate exchange is an equilibrium and does not need to be forced. Even in non-isolated exchange, deadlines and lateness penalties are meaningful only as long as abiding to the deadline or paying the lateness penalty is less expensive than suffering the defection penalty. Lateness penalty schedules are preferable to strict deadlines because they are less risky for the agent who is potentially subject to them, but the other agent can still tailor the lateness penalty schedule to motivate the former to move immediately.

5 Extension 3: Delivery sequencing

So far we have discussed exchanges in which the delivering order of the goods is fixed beforehand. In this section we analyze an exchange where discrete partial deliveries (individual goods or atomic chunks) can be delivered in any order, as long as all of them get delivered. It is assumed (this is relaxed in Section 5.1) that the demander's added value from one chunk does not depend on the other chunks delivered so far, and that the supplier's cost for delivering a chunk does not depend on other chunks delivered earlier. This enables us to associate each chunk c with two values, Δp_c^{max} and Δp_c^{min}, that fully characterize how much the maximum and the minimum cumulative payments change as c is delivered.

For example, an agent could make a contract to carry out a number of matrix multiplications. Multiplying two matrices neither facilitates nor hinders multiplying some other two, so the chunks are independent with respect to the supplier. The chunks are truly independent if they are independent with respect to the demander also—based on the uses of the multiplication results.

Call a delivery sequence safe if $min(p^{max}(x), p^{contr}) \geq p^{min}(x')$ for all consequtive x and x'. We provide a fast greedy algorithm that finds a safe ordering if one exists. The algorithm takes six inputs: a set of chunks C, a vector of Δp_c^{max} values, a vector of Δp_c^{min} values, the contract price p^{contr}, and the defection penalties ($c_s^{def} = c_d^{def} = 0$ in the case of isolated exchange).

Algorithm 5.1 ***SEQUENCE-CHUNKS***(*C, $\boldsymbol{\Delta p^{max}}$, $\boldsymbol{\Delta p^{min}}$, p^{contr}, c_s^{def}, c_d^{def}*)

1. $p_{init}^{max} = p^{contr} + c_s^{def}, p_{init}^{min} = p^{contr} - c_d^{def}$.
2. *For every c in C do /* Sets bounds for p at $x = 0$ */*
   $p_{init}^{max} = p_{init}^{max} - \Delta p_c^{max}, p_{init}^{min} = p_{init}^{min} - \Delta p_c^{min}$.
3. *If $p_{init}^{max} < 0$ or $p_{init}^{min} > 0$ return "NO SOLUTION".*
4. *Divide C into two sets POS and NEG s.t.*
   $POS = \{c \in C | \Delta p_c^{max} - \Delta p_c^{min} \geq 0\}$ *and*
   $NEG = \{c \in C | \Delta p_c^{max} - \Delta p_c^{min} < 0\}$.
5. $p^{max} = p_{init}^{max}, p^{min} = p_{init}^{min}, n_p = |POS|, n_n = |NEG|$
6. *For $i = 1$ to n_p*
   $FEASIBLES = \{c \in POS | p^{min} + \Delta p_c^{min} \leq p^{max}\}$
   If $FEASIBLES = \emptyset$ return "NO SOLUTION".
   $c^* = \underset{c \in FEASIBLES}{\operatorname{argmax}} \Delta p_c^{max} - \Delta p_c^{min}$.
   $chunk[i] = c^*$.
   $p^{max} = p^{max} + \Delta p_{c^*}^{max}$, $p^{min} = p^{min} + \Delta p_{c^*}^{min}$.
   $POS = POS - \{c^*\}$.
7. $p^{max} = p^{contr} + c_s^{def}, p^{min} = p^{contr} - c_d^{def}$.
8. *For $i = n_n + n_p$ down to $n_p + 1$*
   $FEASIBLES = \{c \in NEG | p^{min} \leq p^{max} - \Delta p_c^{max}\}$
   If $FEASIBLES = \emptyset$ return "NO SOLUTION".
   $c^* = \underset{c \in FEASIBLES}{\operatorname{argmax}} \Delta p_c^{min} - \Delta p_c^{max}$.
   $chunk[i] = c^*$.
   $p^{max} = p^{max} - \Delta p_{c^*}^{max}$, $p^{min} = p^{min} - \Delta p_{c^*}^{min}$.
   $NEG = NEG - \{c^*\}$.
9. *Return the ordered vector "chunk". First chunk to be delivered is in "chunk$[1]$".*

Step 6 of the algorithm sequences the chunks with positive $\Delta p_c^{max} - \Delta p_c^{min}$ in a forward passing greedy manner to try to increase p^{max} as much as possible while increasing p^{min} as little as possible. Intuitively, the algorithm tries to maximize the range of possible safe prices at each x. Step 7 just computes p^{max} and p^{min} at the end of the whole sequence of chunks. Step 8 makes a greedy backward pass. It tries to allocate the chunks with negative $\Delta p_c^{max} - \Delta p_c^{min}$ so as to use as little as possible of the beneficial difference $\Delta p_c^{max} - \Delta p_c^{min}$ in the end of the sequence. Intuitively, this difference is saved for the middle of the sequence, from where it has time to affect more chunks (lying later in the sequence).

To solve our sequencing problem, we tried several greedy algorithms starting with the intuitive ones. Most of them do not guarantee that a safe sequence is found even if one exists. For example, the algorithms that greedily pass only forward and maximize $\Delta p_c^{max} - \Delta p_c^{min}$ or minimize Δp_c^{min} can be defeated by counterexamples with just two chunks. Our algorithm cannot be defeated:

Theorem 5.1 *Algorithm 5.1 finds a safe ordering if one exists and always terminates in $O(|C|^2)$ time.*

Sometimes the division of the exchange into chunks is not externally fixed but can be decided by the agents, e.g. at contract time. This can be done *top down* by generating a chunking and then testing its safety by running algorithm 5.1. If it is not safe, the chunking can be refined by splitting chunks further. Splitting is monotonic in the sense that no split can make a safe exchange unsafe. Therefore this splitting algorithm does not need to backtrack. The top down method can be used for continuous goods also. The minus side of the approach is the need to guess the splits. If they are guessed badly, possibly many more chunks are generated than are necessary to enable safe exchange. A *bottom up* approach for chunking is to sequence the smallest possible atomic chunks using algorithm 5.1. Next, the agents can see how many atomic chunks they can deliver at once at each step without changing the order and while still keeping the exchange safe. Bottom up chunking requires no guessing of splits but it can be computationally complex if the number of smallest possible chunks is large. It cannot be applied to continuous goods because the number of smallest possible chunks is infinite.

5.1 Sequencing interdependent deliveries

Sometimes partial deliveries are interdependent. The value of a chunk may depend on which chunks have been delivered before it. For example in manufacturing, the first products can be thought of as more costly than subsequent ones because the fixed costs (e.g. rent, acquired equipment) can be associated with the earlier products. Similarly, a data retrieval agent may incur large costs in searching for certain information. Once the information is found, subsequent searches of related information are less expensive. The demander may also value a chunk differently depending on the other chunks delivered so far. In TRACONET (see Section 2.1), the chunks (transport tasks) were interdependent for both the supplier and the demander. Transporting a parcel often affects the marginal cost of transporting others. For example, a vehicle may be able to carry two parcels to adjacent locations, thus reducing the marginal cost of both tasks. Conversely, one parcel may fill up the vehicle so that another task must be handled by a more costly vehicle. Some contracts involved multiple tasks. So if the safe exchange mechanisms of this paper had been used, sequencing of interdependent chunks would have been required. This was not crucial because the agents represented real world dispatch centers whose contracts were enforced by law.

In general, interdependent goods cannot be sequenced in polynomial time in the number of chunks if it is required that a safe solution is found if one exists. Just representing the problem requires $\Theta(2^{|C|})$ space because for each set of chunks in the power set of all chunks, p^{max} and p^{min} have to be represented—and in the worst case this information cannot be compressed. Nevertheless, if the number of chunks per contract is small—as in TRACONET—exponential search among sequences of chunks is viable. In such cases the advantages of safe unenforced exchange outweigh the extra computational load. Furthermore, special cases of the problem may be solvable in polynomial time, e.g. the case of independent chunks discussed earlier.

6 Renegotiation risk

After an irrevocable delivery or payment, the agent that gained from it may want to renegotiate the contract. For example after the first partial delivery, the demander may want to renegotiate the contract for a lower price. The demander knows that the original contract price was safe for the supplier, so now that the supplier has already "lost" the first delivery, the supplier should be willing to carry out the rest of the exchange at a lower price. On the other hand, the supplier knows that any point in the exchange is safe for the demander. Therefore, if the supplier can commit to not renegotiating, the demander is motivated to follow the original contract and to not vanish.

Renegotiation is more likely in unsafe exchange [Lax and Sebenius, 1981; Raiffa, 1982]. For example, when an international company initiates a mining venture in a developing country, it has to invest most of the capital up-front. This unsafe move motivates the developing country to renegotiate the conditions of the mining venture (profit division etc.). Due to expropriation risk the company cannot avoid renegotiation.

7 Conclusions and future research

This paper presented a method for carrying out mutual exchanges among self-motivated agents without third party enforcement. Larger exchanges were split into smaller parts so that at no point was either agent motivated to defect (in equilibrium). The maximum size delivery that the supplier can safely make at any point in the exchange was shown as well as the maximum amount that the demander can safely pay. The possibility of safe exchange depends on the demander agent's and the supplier agent's value functions for the goods of the contract. Safe exchange is enhanced if the supplier incurs most of its cost from the early portion of the exchange, while the possibility that the demander acquires most of its value already from the early parts hinders safe exchange.

Isolated safe exchange can be carried out entirely only if the supplier can deliver the last part without cost.

With continuous goods it can be carried out arbitrarily close to completion even if this is not the case. Considering defection's adverse effect on future negotiations often enables completing the exchange.

Under the presented conditions on their discount functions, agents are motivated to carry out isolated exchanges immediately. Time discounts reduce the viability of taking advantage of non-isolated exchange. In such cases, immediate moves can be forced by deadlines or lateness penalties.

Some domains allow goods to be delivered in different orders. The presented quadratic algorithm finds a safe ordering for independent goods if one exists. The problem cannot be solved in polynomial time for interdependent goods. Finally, we showed that safe exchange helps prevent unfair renegotiation.

In this paper we looked at totally safe exchanges, where each agent knew its opponent's value function, discount functions, and defection penalty (i.e. cost of making reputation worse). We explained how agents could use bounds for these if they are not exactly known. If the bounds were too far off, even possible exchanges were disabled. Often it is the case that agents can estimate a distribution for each of these, although strict bounds are not available or they are too far off. Using these distributions the agents can take a calculated risk of making moves that are unsafe with a certain probability. This approach of using distributions is also useful to the agent in trying to model the possibility of changes in the opponent's value function, discount functions or defection penalty that happen during the exchange due to the opponent interacting in its environment (getting other offers, contracts etc.).

Another approach is to try to bound ones losses by making the partial exchanges small enough so that even if the opponent defects, the loss will be within a bound. In both the probabilistic risk method and the loss bounding method there is a tradeoff between making the exchange safer by using small partial exchanges and minimizing partitioning costs (e.g. physical per part delivery costs) by using large ones. Finally, either a probabilistic approach or a loss bounding approach can be used to address the risk of the opponent accidentally defecting—e.g. loosing contact due to a technical fault.

Acknowledgments

We thank Prof. Neil Immerman (Umass Comp Sci) for the backward pass idea for algorithm 5.1, and Prof. Herb Gintis (Umass Economics) for game theoretic comments.

References

[Axelrod, 1984] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 1984.

[Brainov, 1994] Sviatoslav Brainov. Deviation-proof plans in open multiagent environments. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 274–278, August 1994.

[Durfee *et al.*, 1989] E Durfee, V Lesser, and D Corkill. Cooperative distributed problem solving. In A Barr, P Cohen, and E Feigenbaum, editors, *The Handbook of Artificial Intelligence*, vol. IV. Addison Wesley, 1989.

[Durfee *et al.*, 1993] Edmund H Durfee, J Lee, and Piotr J Gmytrasiewicz. Overeager reciprocal rationality and mixed strategy equilibria. In *AAAI-93*, pages 225–230, Washington DC, July 1993.

[General Magic, Inc., 1994] General Magic, Inc. Telescript technology: The foundation for the electronic marketplace, 1994. White paper.

[Kraus *et al.*, 1992] Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. Technical Report 2975, Univ. of Maryland, College Park, Computer Science Dept., 1992.

[Kreps, 1990] David M Kreps. *A course in microeconomic theory.* Princeton University Press, 1990.

[Lax and Sebenius, 1981] D A Lax and J K Sebenius. Insecure contracts and resource development. *Public Policy*, 29(4):417–436, 1981.

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

[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.* MIT Press, 1994.

[Sandholm and Lesser, 1994] Tuomas W Sandholm and Victor R Lesser. An exchange protocol without enforcement. In *Proc. 13th International Workshop on Distributed Artificial Intelligence (DAI-94)*, pages 305–319, Seattle, WA, July 1994.

[Sandholm and Lesser, 1995a] Tuomas W Sandholm and Victor R Lesser. Advantages of a leveled commitment contracting protocol. Technical report, Univ. of Massachusetts at Amherst, Comp. Sci. Dept., 1995. In preparation.

[Sandholm and Lesser, 1995b] Tuomas W Sandholm and Victor R Lesser. Coalition formation among bounded rational agents. In *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 1995.

[Sandholm and Lesser, 1995c] Tuomas W Sandholm and Victor R Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *Proc. First International Conference on Multiagent Systems (ICMAS-95)*, San Fransisco, June 1995.

[Sandholm, 1993] Tuomas W Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, July 1993.

[Selten, 1965] R Selten. Spieltheoretische behandlung eines oligopolmodells mit nachfragetragheit. *Zeitschrift fur die gesamte Staatswissenschaft*, 12:301–324, 1965.

[Wellman, 1992] Michael Wellman. A general- equilibrium approach to distributed transportation planning. In *Proc. 10th National Conference on Artificial Intelligence (AAAI-92)*, 1992.