

Trust-Aware Delivery of Composite Goods

Zoran Despotovic, Karl Aberer

Department of Communication Systems
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland
{zoran.despotovic, karl.aberer}@epfl.ch

Abstract

The vast majority of the interactions in typical online communities nowadays is between complete strangers. In such settings reputation reporting and trust management models play a crucial role for proper functioning of those communities. A lot of work has been done on the issues of collecting and spreading reputations and subsequent computation of trust. The application of such data for decision making, however, is far less explored. In this paper we present a solution for scheduling exchanges among participants of an online community which takes into account their trustworthiness. In this way we can enable exchanges that would otherwise not be taking place. Thus this work also demonstrates that trust can in fact increase economic activity.

Keywords. *Reputation Reporting, Trust Management, Safe Exchange*

1 Introduction - Reputation and Trust

The rapid growth of the Internet and accompanying computing and networking technologies have made it possible for any computer around the globe to participate in a collaborative scenario such as e-commerce, teamwork, Peer-To-Peer file sharing, etc. As a result, online communities with highly autonomous members are emerging. This autonomy does not stem only from recently developed technological innovations in the area of distributed computing (P2P in particular) but also from the fact that today's typical online environments span the globe making the enforcement of the common law among their members impossible or intolerably expensive. In such a situation reputation reporting and trust assessment mechanisms play a crucial role for deterring opportunistic behaviour and encouraging trusting interactions in those environments enabling thus their proper functioning (e.g., in the area of e-commerce we refer to [6] for a detailed analysis of the importance of eBay's reputation management scheme, Feedback Forum, for the success of eBay).

Though they represent the most typical examples, auction and, more generally, e-commerce marketplaces are not the only areas in need of reputation and trust management mechanisms. Essentially, trust and reputation come into play in any interaction involving at least one vulnerable participant. Opportunistic behavior of the other side is just a way to realize that vulnerability as it is the case in exchanges of goods for money. The reasoning behind this statement is rather simple. If you *depend* on someone's actions, good will etc., then you need to trust. On the other hand, if you are indifferent to what someone is doing or talking to you then your trust is not necessary. Hence the need for these mechanisms is not only in e-commerce settings but as well in teamwork environments, expert sites etc.

Numerous works on reputation and trust management in online communities have appeared recently [1, 4, 6]. In our opinion, an inter-disciplinary approach taking into account works on the trust issues from

the areas of sociology, psychology, and economics is needed here. Figure 1 shows the common reference model that most approaches apply. The reputation management module is responsible for collecting information about the past behavior of the members of the community under consideration as well as making this information available for others to use. Once the data about past interactions of potential partners has been collected, the trust management module comes into play by calculating predictions of their future behavior. Finally, the decision making module, analysing the performance of those predictions in the interaction that is about to happen and having the risk averseness related inputs from the user, makes decision whether to interact or not. The results of the interactions are then fed back into the reputation management module for the future use.

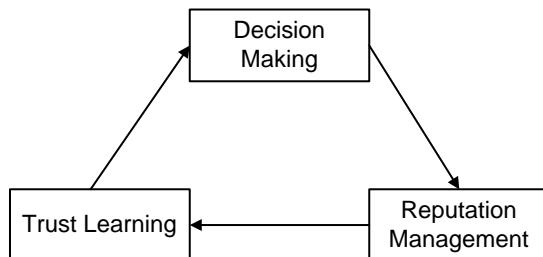


Figure 1. Reputation and trust management reference model

At the moment research focuses on how to organize reputation reporting models and, subsequently, compute the trustworthiness of potential exchange partners. What is missing in the literature is the analysis of whether and under what conditions to interact after the trust assessments have been made. In other words, the question how to use trust data to select optimal strategies in interactions is not addressed. In a typical example of exchanges of goods for money, a seller may require an insufficiently trustworthy buyer to deliver the payment before the delivery of the goods. On the other hand, this may be unacceptable to the buyer because of the seller’s bad reputation and the buyer may as well require the seller to deliver the goods before the money is delivered. In most cases the reputation records of the two partners will suggest, however, that it is not necessary to go into these two extremes and that the deliveries of the goods and the payments can be arranged in such a way that the partners’ expectations are met with high probability. For this purpose a closer analysis of the quality of the exchange strategies with respect to the trustworthiness of the partners is needed. This is where our approach comes into play. It represents a trust-related decision making scheme that can be used in many online exchange scenarios.

2 The Setting - Safe Exchange

[7] presents the basis for our work. We consider exchanges between two participants in an e-commerce scenario. The starting assumption is that one of the two participants (the “supplier”) is supplying and selling a set of “goods” $C = \{c_1, \dots, c_n\}$ to the other partner (the “consumer”). We further assume that they agreed about the overall price the consumer will have to pay for the goods (P^{contr}) as well as that the consumer may deliver the payments in arbitrarily sized chunks. The set of goods C consists of a number of items and it is assumed that the supplier’s value function $V_s(x)$, describing the supplier’s cost for generating and delivering any item x of the goods (expressed in monetary units or on a different scale), as well as the consumer’s value function $V_c(x)$, describing what the good x is worth to the consumer are both known to the both partners. As well, the order in which the items are delivered does not matter, so that the above functions can be naturally extended to all subsets of the original set of items - by summing up the values of the goods contained in the subsets. (Or precisely, for any $A = \{c_1, \dots, c_k\} \subset C$, we have $V_s(A) = V_s(c_1) + \dots + V_s(c_k)$ and $V_c(A) = V_c(c_1) + \dots + V_c(c_k)$).

In this setting, the main problem can be now stated as follows: how to combine the deliveries of the goods and the payments in such a way that, at any point during the exchange, future gains of both partners are greater than their gains from the instant defection. With such an exchange strategy the partners have no rational incentive to break the exchange at any point before its end. Therefore, an exchange sequence that meets these conditions is called *safe* (exchange) sequence. Note that the term rational exchange is also used in the literature [3].

As far as the existence of a safe exchange sequence is concerned, the main result of [7] can be described in the following way. Assuming that the supplier delivers the goods in the order c_{i_1}, \dots, c_{i_n} ($n = |C|$) then this order is safe if and only if the payments and the goods are delivered in such a way that the gains of the both partners at any point k ($1 \leq k \leq n$) in the exchange lie between these two bounds:

$$P^{min}(k) = V_c(\{c_{i_1}, \dots, c_{i_k}\}) + P^{contr} - V_c(C)$$

and

$$P^{max}(k) = V_s(\{c_{i_1}, \dots, c_{i_k}\}) + P^{contr} - V_s(C)$$

where P^{contr} is the overall payment to be delivered.¹ The line $P^{min}(k)$ defines the lower bound and the seller must deliver the goods in such a way that this bound is not crossed. Similarly, the consumer has to deliver the payments without crossing the upper bound $P^{max}(k)$.

With these two functions defined the main theorem states that a safe (exchange) sequence exists if and only if there exists an order c_{i_1}, \dots, c_{i_n} such that, for any $1 \leq k < n$:

$$P^{min}(k+1) \leq P^{max}(k).$$

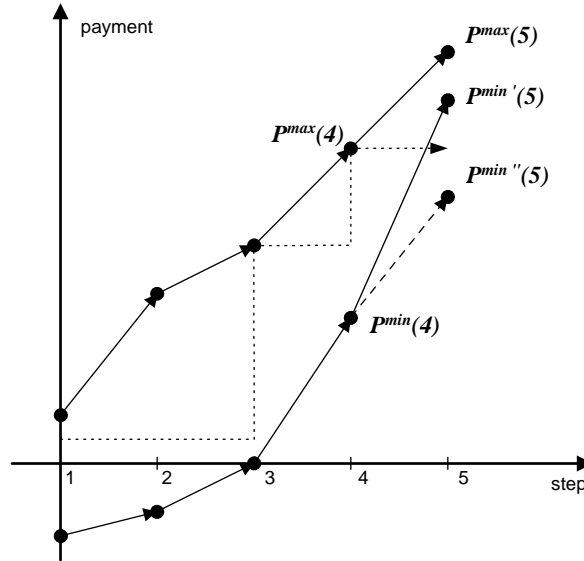


Figure 2. Safe Exchange Example

But, this result is rather negative for isolated exchanges. Because $P^{min}(n) = P^{max}(n) = P^{contr}$ whichever of the two partners is supposed to make the last delivery he is better off by leaving the exchange at the

¹For a given order, such as c_{i_1}, \dots, c_{i_n} , to make the notations more compact we write $P^{min}(k)$ instead of $P^{min}(\{c_{i_1}, \dots, c_{i_k}\})$. Similar notations will be used for $P^{max}(k)$, $V_s(k)$ and $V_c(k)$. $V_s(n)$ and $V_c(n)$ will be denoted by V_s and V_c .

penultimate step. Therefore, a mechanism for moving apart the $P^{min}(n)$ and $P^{max}(n)$ is needed and, assuming that the participants will be engaged in repeated exchanges (possibly with different partners), reputation effects (modeled through “defection costs”) are suggested as one such mechanism. A more detailed discussion on this view on the notion of reputation is given in the next section.

Figure 2 presents an example. Here, the solid line (from $P^{min}(4)$ to $P^{min'}(5)$) shows a situation in which a safe exchange does not exist. In this case we have that $P^{max}(4) < P^{min'}(5)$, which is not acceptable. At the point $k = 4$, if the supplier delivers the next item, then the consumer should immediately defect because his utility at the next point would be higher than his utility at the end of the exchange. On the other hand, the dashed line shows an example in which a safe sequence exists ($P^{max}(4) > P^{min''}(5)$).

3 Trust Aware Safe Exchange

The results described in the previous section can be applied in many different settings. The necessary condition for the applicability of the approach is that the good being sold is divisible into a set of chunks whereby the valuations of these chunks are known to the both partners (or, a number of independent items must be collected and sold together). The condition can be met in many practical situations such as trades in eBay’s auctions, exchanges of MP3 files for money in a P2P system or trades of services in a teamwork environment. But, a serious practical problem associated with the original approach is that a fully safe exchange sequence for the deliveries of the chunks of goods and the payments may not exist in many cases. Assuming that interactions in the mentioned systems are backed by underlying reputation and trust management models, a trust aware extension of the above results is required as it may help schedule exchanges between (sufficiently) honest partners in these cases.

We mentioned in the previous section that [7] suggests reputation effects modeled through “defection costs” as a mechanism that may enable the existence of a safe exchange sequence of the deliveries of goods and payments. The question how to compute these costs was not answered. In the following we sketch the reasoning behind this statement and introduce our view on reputations and trust by differentiating them from the way they are used in [7].

Namely, the mentioned suggestion is based on the idea that it may not be beneficial to defect in the present if the likelihood of having interactions in the future is sufficiently large (e.g., [2] presents a detailed discussion on the importance of this parameter in the repeated Prisoner’s Dilemma game). In principle, it should be possible, though not easy, to assess one’s view on this likelihood based on their past behavior and compute the threshold whether defecting is beneficial (defection costs). But, even by including this “shadow of future” in the model we still remain in the domain of *safe* exchanges, i.e, as long as the partners are acting rationally, risks do not exist and that trust per-se is not necessary. On the other hand, as many analyses as well as common sense show, people *do* take risks and *do* use trust while interacting with strangers. In this work, we provide a mechanism that, having the levels of risk averseness the two participants of the previously described scenario are willing to accept, finds an exchange strategy satisfying these levels.

Every set of goods that the supplier wants to sell to the consumer and their valuations, $V_s(x)$ and $V_c(x)$, have associated gains of the two partners. The seller’s gain is the difference between the overall price of the goods and the seller’s costs caused by generating and delivering the goods ($P^{contr} - V_s(C)$). Similarly, the consumer’s gain is the difference between his valuations of the goods and the price he has to pay ($V_c(C) - P^{contr}$). If the exchange can be carried out in the safe manner these gains are guaranteed, but, in situations when a fully safe exchange sequence does not exist (due to the mentioned valuations) these gains cannot be achieved. Yet, if their opponents are sufficiently trustworthy, the partners can accept even unsafe exchanges. But in this case an adaptation of the original approach that takes into account the expectations of the two partners is needed. Therefore, we proceed in the following way: We assume that the two parties can compute probabilistic estimates of a specific (in the simplest case honest and dishonest) behavior of the other side. This is the task of the underlying trust computation module. [4] presents a theoretically well-founded solution for this problem. [1] is a practical approach that can be used in P2P environments. Now, we expect the partners to refine their expectations from the exchange by decreasing the expected gains. The question of how much to decrease the expected gains is left to the partners themselves. Obviously, these decreases

will be based on the risk averseness of the two parties and the trustworthiness of the other side. Without elaborating further on that, note that these expectations easily translate into two bounds representing the values that the partners accept to be indebted. Having these new, decreased, expected gains we provide a provably correct quadratic-time algorithm that finds an exchange sequence, if one exists, that satisfies these expectations.

4 Algorithms

In this section we give a detailed description of the algorithm for scheduling exchanges between two partners taking into account their trustworthiness.

Let us assume that the seller delivers the goods $C = \{c_1, \dots, c_n\}$ in the order c_{j_1}, \dots, c_{j_n} ($\{j_1, \dots, j_n\}$ is a permutation on the set $\{1, \dots, n\}$). Assume as well that the seller has specified that he wants his expected utility (with respect to the trustworthiness of the consumer) to be higher than U_s^{given} :

$$U_s^{exp} \geq U_s^{given}$$

On the other hand, whenever the seller delivers an item causing the lower bound P^{min} to be crossed he exposes himself to the risk that the consumer will leave the exchange. But the consumer's rationality implies that this will happen only at the step i for which the difference $P^{min}(i) - P^{max}(i-1)$ is maximal. Therefore, the seller's expected utility becomes:

$$U_s^{exp} = p_c^H [P^{contr} - V_s] + (1 - p_c^H) [P^{max}(i-1) - V_s(i)],$$

where i is the step during the exchange that achieves the maximum of $P^{min}(i) - P^{max}(i-1)$ ($P^{min}(i) > P^{max}(i-1)$) and p_c^H is the probability of the consumer's honest behavior and we are assuming that it can be obtained from the underlying trust model.

Further, because

$$P^{max}(i-1) - V_s(i) = P^{max}(i-1) - V_s(i-1) - V_s(c_{j_i})$$

and, for any i ,

$$P^{max}(i-1) - V_s(i-1) = P^{contr} - V_s = G_s$$

is a constant independent of i (the seller's gain from the exchange) we have:

$$p_c^H G_s - (1 - p_c^H) [V_s(c_{j_i}) - G_s] \geq U_s^{given},$$

which leads to the following inequality:

$$V_s(c_{j_i}) \leq C_s,$$

where

$$C_s = \frac{G_s - U_s^{given}}{1 - p_c^H}.$$

Thus, whenever the seller has to cross the boundary P^{min} and deliver an item c then the value of the goods c must satisfy:

$$V_s(c) \leq C_s. \tag{1}$$

The following algorithm finds an ordering of the items (if one exists) with the property that whenever the seller goes below the lower bound P^{min} the item being delivered at that step satisfies: $V_s(c) \leq C_s$.

Algorithm 1.

Input: the set of goods C , for each $c \in C$ the consumer's and supplier's values of the good c : $V_c(c)$ and $V_s(c)$, the seller's parameter C_s , $P_{init}^{max} = P - V_s(C) = G_s$ and $P_{init}^{min} = P - V_c(C) = -G_c$

1. Divide C into two sets POS and NEG s.t.

- $POS = \{c \in C \mid V_s(c) - V_c(c) \geq 0\}$ and
 $NEG = \{c \in C \mid V_s(c) - V_c(c) < 0\}$
2. $P^{max} = P^{max}_{init}$, $P^{min} = P^{min}_{init}$, $p = |POS|$, $n = |NEG|$
 3. For $i = 1$ to p {
 - $F = \{c \in POS \mid P^{min} + V_c(c) \leq P^{max}\}$
 - If $F = \emptyset$ then
 - $F = \{c \in POS \mid V_s(c) \leq C_s\}$
 - If $F = \emptyset$ then return *NO SOLUTION*
 - $c^* = \operatorname{argmax}_{c \in F} (V_s(c) - V_c(c))$
 - $chunk[i] = c^*$
 - $P^{max} = P^{max} + V_s(c^*)$, $P^{min} = P^{min} + V_c(c^*)$
 - $POS = POS - \{c^*\}$
 4. For every c in NEG : $P^{max} = P^{max} + V_s(c)$, $P^{min} = P^{min} + V_c(c)$
 5. For $i = n + p$ down to $p + 1$ {
 - $F = \{c \in NEG \mid P^{min} \leq P^{max} - V_s(c)\}$
 - If $F = \emptyset$ then
 - $F = \{c \in NEG \mid V_s(c) \leq C_s\}$
 - If $F = \emptyset$ then return *NO SOLUTION*
 - $c^* = \operatorname{argmax}_{c \in F} (V_c(c) - V_s(c))$
 - $chunk[i] = c^*$
 - $P^{max} = P^{max} - V_s(c^*)$, $P^{min} = P^{min} - V_c(c^*)$
 - $NEG = NEG - \{c^*\}$
 6. Return the vector *chunk*.

The algorithm first divides the set of goods into two groups, one containing the goods with the values that move apart P^{min} and P^{max} ($V_s(c) - V_c(c) \geq 0$) and the other one consisting of the goods that make P^{min} and P^{max} closer ($V_s(c) - V_c(c) < 0$). Then, it makes a forward pass (step 3) selecting an item c satisfying the condition $V_s(c) \leq C_s$ and, at the same time, maximizing the difference $V_s(c) - V_c(c) \geq 0$. Obviously, this makes more room for items with large $V_c(c)$ that are pushed towards the end of this part of the sequence. Step 5 presents a backward pass adopting the same idea as step 3. The algorithm runs in $O(|C|^2)$ time.

The correctness of the above algorithm can be formally proven as follows.

First, if a safe sequence exists then the algorithm will find it (Theorem 5.10 in [7]). Assume now that a safe sequence does not exist and the algorithm terminates with *NO SOLUTION* (in which case we have only a partial sequence s) but there exists a sequence s' that is acceptable.

The sequence s' can be reordered in such a way that all chunks with $V_s(c) - V_d(c) \geq 0$ come before all chunks with $V_s(c) - V_d(c) < 0$. To see why, take any pair of chunks c_i and c_j such that $V_s(c_i) - V_d(c_i) < 0$, $V_s(c_j) - V_d(c_j) \geq 0$ and c_i is immediately before c_j in s' . We can swap these two chunks. Before and after the pair, P^{max} and P^{min} will remain unchanged. If we denote with l , $l + 1$ and $l + 2$ the (consecutive) points where, respectively, c_i starts, and c_i and c_j end (Figure 3), then $P^{min}(l + 1) \geq P^{max}(l)$ implies that $V_s(c_i) \leq C_s$ (the sequence s' is acceptable) and thus the swap will not cause any problem at $l + 2$ because the swap can only make the difference $P^{max}(l + 1) - P^{min}(l + 1)$ bigger. For the same reason, if $P^{min}(l + 1) < P^{max}(l)$ before the swap then it must be $P^{min}(l + 2) < P^{max}(l + 1)$ after. Therefore, we see that the swap cannot make trouble at $l + 2$. Reasoning similarly, it can be seen that the swap is acceptable at $l + 1$ as well.

Now, applying these swaps repeatedly, we get a sequence (denote it by s'') in which all chunks with $V_s(c) - V_d(c) \geq 0$ lie before all chunks with $V_s(c) - V_d(c) < 0$. But, both parts of this sequence can be reordered according to the ordering of our algorithm. Take for example, the first part, s''_{pos} consisting of the chunks with $V_s(c) - V_d(c) \geq 0$. Assume that the corresponding part of our (partial) sequence s , s_{pos} , and s''_{pos} differ first at the position i and that the chunk on that position in our sequence is located at the position $k > i$ in s''_{pos} . This chunk can be moved from k -th position to the position i because our algorithm

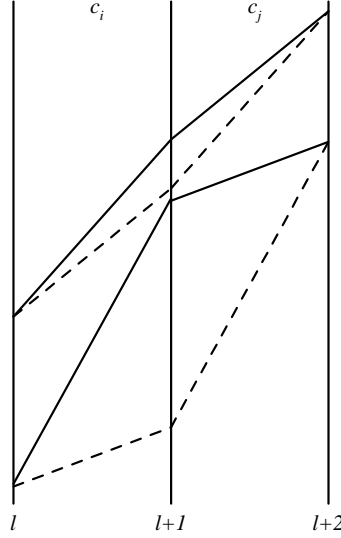


Figure 3. Swapping Two Consecutive Steps in an Exchange

selected it as acceptable. At the same time, as the chunk from the position i in s'' has smaller $V_s(c)$ than that of the position k , the swap will increase the differences $P^{max}(l) - P^{min}(l)$, for $i < l < k$, and the new sequence cannot become unacceptable due to a problem at any step l ($i < l < k$). Using the same line of reasoning as in the previous paragraph, we can see that the swap is acceptable at the position k as well. As for negative parts of s and s'' , s_{neg} and s''_{neg} , we see that the algorithm proceeds in the reverse order and therefore, using the same idea as in the previous paragraph, we conclude that s''_{neg} can be reordered according to the ordering of our algorithm.

Thus, by contradiction, if an acceptable sequence exists the algorithm will find it.

The problem of the non-existence of a safe sequence of the deliveries can be overcome by the consumer's reaction as well. At any point k such that $P^{max}(k) < P^{min}(k + 1)$ the consumer may deliver $P^{min}(k + 1) - P^{max}(k)$ more money and thus enable the exchange to reach its end. But any such action from the consumer's side is unavoidably related to a certain level of risk due to the supplier's eventual dishonesty. Let us take a closer look at what the consumer may expect from the exchange in this case.

For a given order of deliveries c_{i_1}, \dots, c_{i_n} such that $P^{max}(k) < P^{min}(k + 1)$ the consumer's expected utility can be expressed as:

$$U_c^{exp} = p_s^H [V_c - P^{contr}] + (1 - p_s^H) [V_c(k) - P^{min}(k) - V_c(c_{i_{k+1}})]$$

As $V_c - P^{contr} = V_c(k) - P^{min}(k) = G_c$ (the consumer's gain from the exchange) we have:

$$U_c^{exp} = G_c - (1 - p_s^H) V_c(c_{i_{k+1}})$$

Then the requirement that this expected value satisfies $U_c^{exp} \geq U_c^{given}$ is equivalent to:

$$V_c(c_{i_{k+1}}) \leq C_c$$

where $C_c = \frac{G_c - U_c^{given}}{1 - p_s^H}$.

Given C_c an algorithm that finds an ordering of the deliveries such that $V_c(c_{i_{k+1}}) \leq C_c$ whenever $P^{max}(k) < P^{min}(k + 1)$ can be constructed by following the same line of reasoning as in the Algorithm

1. The only change is that when no safe step can be found we select the items c satisfying $V_c(c) \leq C_c$.

Assume now that, for a given set of goods C and the consumer's and seller's valuations of these goods, $V_c(c)$ and $V_s(c)$ ($c \in C$), not only that there exists no fully safe sequence, but moreover, let us assume that due to the 'peculiar' values $V_c(c)$ and $V_s(c)$ ($c \in C$) and/or too high expected thresholds U_s^{given} and U_c^{given} , it is impossible to get over all 'unsafe' cases ($P^{max}(i) < P^{min}(i+1)$) by the exclusive reaction of either the seller or the consumer. What is left to check in this case is whether these problems can be overcome by a cooperative action of the both partners.

The situation we are facing now is shown in Figure 4. Splitting the interval $x = P^{min}(i+1) - P^{max}(i)$ as shown in this figure the consumer would reach a point higher than $P^{max}(i)$ (x_c higher) and the seller would reach a point x_s units lower than $P^{min}(i+1)$.

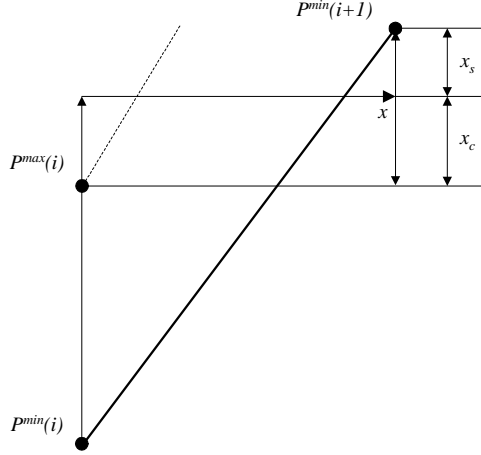


Figure 4. Joint Overcoming of an Unsafe Step

Now, applying the same reasoning on the partners' expectations as before, we can see the requirements that $U_s^{exp} \geq U_s^{given}$ and $U_c^{exp} \geq U_c^{given}$ are equivalent to:

$$V_s(c_{j_{i+1}}) \leq x_c + C_s \quad \text{and} \quad V_c(c_{j_{i+1}}) \leq x_s + C_c \quad (2)$$

where C_s and C_c have the same meaning as before. It is important to note here that if $V_s(c_{j_{i+1}})$ and C_s have such values that the condition $V_s(c_{j_{i+1}}) \leq x_c + C_s$ is met only with $x_c > x$ then the seller can relax this requirement by letting $x_c = x$ because he does not take any risk for $x_c \geq x$.

The following algorithm finds a sequence that satisfies these two conditions.

Algorithm 2.

Input: the set of goods C , for each $c \in C$ the consumer's and supplier's values of the good c : $V_c(c)$ and $V_s(c)$, the seller's parameter C_s , $P_{init}^{max} = P - V_s(C) = G_s$ and $P_{init}^{min} = P - V_c(C) = -G_c$

1. Divide C into two sets POS and NEG s.t.
 $POS = \{c \in C \mid V_s(c) - V_c(c) \geq 0\}$ and
 $NEG = \{c \in C \mid V_s(c) - V_c(c) < 0\}$
2. $P^{max} = P_{init}^{max}$, $P^{min} = P_{init}^{min}$, $p = |POS|$, $n = |NEG|$
3. For $i = 1$ to p {
 $F = \{c \in POS \mid P^{min} + V_c(c) \leq P^{max}\}$
If $F = \emptyset$ then

- For every c in POS
- $$x = P^{min} + V_c(c) - P^{max}$$
- $$x_c = V_s(c) - C_s$$
- If $x_c > x$ then $x_c = x$
- $$x_s = x - x_c$$
- If $V_c(c) \leq x_s + C_c$ then
- $$F = F \cup \{c\}$$
- If $F = \emptyset$ then return *NO SOLUTION*
- $$c^* = \operatorname{argmax}_{c \in F} (V_s(c) - V_c(c))$$
- $$\operatorname{chunk}[i] = c^*$$
- $$P^{max} = P^{max} + V_s(c^*), \quad P^{min} = P^{min} + V_c(c^*)$$
- $$POS = POS - \{c^*\}$$
4. For every c in NEG : $P^{max} = P^{max} + V_s(c), \quad P^{min} = P^{min} + V_c(c)$
5. For $i = n + p$ down to $p + 1$ {
- $$F = \{c \in NEG \mid P^{min} \leq P^{max} - V_s(c)\}$$
- If $F = \emptyset$ then
- For every c in NEG
- $$x = P^{min} - P^{max} + V_s(c)$$
- $$x_c = V_s(c) - C_s$$
- If $x_c > x$ then $x_c = x$
- $$x_s = x - x_c$$
- If $V_c(c) \leq x_s + C_c$ then
- $$F = F \cup \{c\}$$
- If $F = \emptyset$ then return *NO SOLUTION*
- $$c^* = \operatorname{argmax}_{c \in F} (V_c(c) - V_s(c))$$
- $$\operatorname{chunk}[i] = c^*$$
- $$P^{max} = P^{max} - V_s(c^*), \quad P^{min} = P^{min} - V_c(c^*)$$
- $$NEG = NEG - \{c^*\}$$
6. Return the vector *chunk*.

In the above algorithm, it cannot be determined how the “unsafe interval” has been split (or equivalently, what x_c and x_s satisfying the conditions (2) have been actually chosen) because of the two greedy steps that choose chunks maximizing the differences $V_s(c) - V_c(c)$ and $V_c(c) - V_s(c)$ respectively. But this potential drawback can be easily overcome by constraining the content of the feasible set F . Essentially, what we are facing here is a bargaining problem and depending on the sort of the bargaining solution we want to have (maximal overall net utility is a good example) we can derive new conditions that further constrain the set of acceptable chunks. These constraints can be easily integrated into the algorithm without affecting its main properties, its greedy nature and the complexity in particular.

But, this reasoning may prompt the reader to ask the following question: how will the shown interval be split in an actual exchange when there is no third party to impose a particular splitting strategy? To answer it we recall our starting assumption that does not require the expected utilities of the two partners to be maximal but just above given thresholds. It is this assumption that makes room for the acceptability of more than one strategy, which can be clearly seen on the mentioned problem. Therefore an agreement between the partners is necessary here. A game theoretic setting in which maximal expected utilities are required by the partners will be the subject of future work.

5. Discussion on Potential Limitations

In this section we elaborate on the potentially limiting factors of the setting introduced in Section 2 and our trust aware extension presented in the sections 3 and 4. We first mention the most important factors that may limit the usability of the approach in practice and then comment on each of them:

- The goods must be divisible;
- The ordering of the items does not matter;
- Valuations of the items as seen by the two partners must be known to both of them;
- The probabilities obtained from the trust model are constant in the past defection levels;

Divisibility of the goods. It should be clear from the previous discussion that the practical usability of the approach is highly dependent on the possibility to divide the goods into a number of chunks. While this is certainly not possible in most of exchanges of material (tangible) goods (we cannot deliver the sleeves, buttons and the rest of a shirt separately), it is feasible with electronic goods, which can be split arbitrarily and assembled upon reception without any effort. As far as this kind of exchanges is concerned, a typical scenario for application of the approach is exchanges of movie or music files for money. A movie file can be divided into a number of segments each of them having the value of its own. But the fact that the whole movie has been delivered has an additional value that is not present among the values of the isolated segments. This situation can be easily overcome by introducing a fictitious chunk that must be left for the end of the exchange. Note that a modification of the above algorithms that checks the satisfiability of this last chunk is necessary in this case.

We also remark that there is one more interesting possibility to model this sort of exchanges in which one fictitious chunk would come after each delivery of a really existing chunk. But the above algorithms cannot be normally applied in this case, unless the values associated with fictive chunks are all equal independently of their positions. What is necessary in this case is an order constraint based extension of the algorithms.

Ordering. The fact that the ordering of the items does not matter practically means that the supplier's and the consumer's current utilities are additive and change with a delivery of an item only for the partners' corresponding valuations of the item being delivered. This assumption leads to the low complexity of our algorithms. If it was removed a backtracking step to check the previously delivered items would be necessary at each delivery. This would lead to an increased (if not exponential) complexity of the algorithms. Fortunately, with the extension described above (a fictitious chunk at the end of the exchange) this does not present a major problem in the most typical scenario of electronic exchanges.

Known valuations of the goods. The partners must know the two bounds (P^{max} and P^{min}) in order to be able to perform only acceptably safe steps. As the value functions (V_s and V_c) are explicitly present in these two bounds it is clear that the partners must know each other's valuations of the goods. In practice, they will hardly ever be able to know these values exactly. On the other hand, we believe that close assessments of the valuations can be made. On the sellers' side these utilities are related to the price a seller can obtain on a market, considering additional factors such as the expected time needed to sell the good. On the consumers' side these utilities could for example be related to the popularity of the goods, taking into account specific characteristics of the consumer, e.g. being part of a specific consumer group.

Constant probabilities. Presently, defection probabilities obtained from the underlying trust model do not depend on the possible gains of the defecting parties. Put differently, we do not take into account the fact that people are more inclined to defect if their gains from defections are greater. This has one important implication on the two presented algorithms. Namely, the constant probabilities of defections practically mean that the punishment for the past defections is independent of the defection levels. Therefore, it is irrational to defect at lower, non-maximal utilities. If these probabilities were not constant then the expected utilities become functions of the whole sequences and the algorithms cannot be applied.

6. Conclusions and Future Work

In this paper we propose a mechanism for scheduling exchanges of goods for money that is aware of the trustworthiness of the exchange participants. It is based on the risk averseness related inputs from the two exchange partners. The main contribution of the work is that it provides the theoretical foundations

that enable sufficiently trustworthy partners carry out exchanges in cases when they cannot proceed in a completely safe manner.

Further, we elaborated on the practical usability of the approach giving some clues for how to overcome potentially limiting factors for its applicability in practice. These problems will be explored in more detail in our future work. As well, what is left to be checked is how the model behaves on real data from typical online settings. A particularly important question we must answer here is that of the relationship between the granularity levels of the partitioned goods, on the one hand, and the fraction of the cases in which our algorithms will find a sufficiently safe sequence, on the other hand, given different values of the trustworthiness of the participants as a parameter. Having this question answered we will be able to present closer assessments of the practical usability of the approach.

Finally, a game-theoretic extension of this work arising when the partners are interested in maximizing their expected gains from the exchanges will be considered in the future work. In particular, we will address the problem of the uniqueness of the optimal strategies available to the exchange partners under this assumption.

References

- [1] K. Aberer, Z. Despotovic: *Managing Trust in a Peer-2-Peer Information System*, Ninth International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, USA, 2001.
- [2] R. Axelrod: *The Evolution of Cooperation*, Basic Books, New York, 1984.
- [3] L. Buttyan: *Building Blocks for Secure Services: Authenticated Key Transport and Rational Exchange Protocols*, Ph.D. Thesis No. 2511 (2001), Swiss Federal Institute of Technology, Lausanne (EPFL), December 2001.
- [4] L. Mui, M. Mohtashemi, A. Halberstadt: *A Computational Model of Trust and Reputation*, 35th Hawaii International Conference on System Science (HICSS), 2002.
- [5] R. B. Myerson: *Game Theory: Analysis of Conflict*, Harvard University Press, Cambridge, MA, 1991.
- [6] P. Resnick, R. Zeckhauser: *Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System*, Working Paper for the NBER workshop on empirical studies of electronic commerce, Jan 2001.
- [7] T. W. Sandholm: *Negotiation Among Self-Interested Computationally Limited Agents*, PhD Thesis, University of Massachusetts, Amherst, 1996.
- [8] T. Sandholm, V. Ferrandon: *Safe Exchange Planner*, International Conference on Multi-Agent Systems (ICMAS), Boston, MA, 2000.
- [9] T. Tesch, K. Aberer: *Scheduling Non-Enforceable Contracts among Autonomous Agents*, Third IFCIS Int. Conference on Cooperative Information Systems, New York, USA, 1998.