Achieving Cooperation among Selfish Agents in the Air Traffic Management Domain using Signed Money*

Paper ID: #336†

ABSTRACT

We present a monetary system by which selfish agents can cooperate reciprocally. We show that a straight-forward market mechanism can lead to unfair situations when agents misuse key positions. We show that it is not easy to retaliate wrongdoers, as there is a dominant strategy that deviates from the retaliating strategy. We present a monetary system in which every user can issue money and every user is required to sign each credit it issues or circulates. By using a trust-based credit-valuation function, wrongdoers are retaliated and it is no longer dominant to deviate from the retaliating strategy.

1. INTRODUCTION

Our aim in this paper is to construct a mechanism by which airline-representing agents can jointly decide on changes in the flight plan. Our domain is the phase of planning known as *tactical airport planning*. This phase of planning is concerned with the sequencing of arriving and departing aircraft and their scheduling on the gates. A predetermined plan exists, but deviations that occur at the last moment may make it infeasible. In that case the plan needs to be repaired. There are usually several ways in which a plan can be repaired. We will call these *repair candidates*. A candidate consists of actions for one or more agents, such as delays, runways changes and gate changes.

We view the problem as a *repeated resource allocation* problem. In a single round, there are a number of repair candidates, from which one has to be elected and enforced. The resources that are allocated are the events that make up the elected candidate. The fact that usually many planning conflicts have to be repaired on a single day makes it a repeated allocation problem.

Two criteria are most important. First, the allocations needs to be efficient in the sense that the elected candidates should require as little effort from the agents as possible. Secondly, the allocations should be fair in the sense that the total effort is distributed fairly over the agents. Fairness is very important in ATM planning [12] and, more generally, in joint decision making [31, 4]. Note that these criteria need to be satisfied over a large number of rounds.

A key assumption is that agents are selfish, and will try to maximize their own utility at the cost of others if possible. Just as real airlines, agents are often dependent on each other. They may choose to help another agent if requested but may also refuse. We assume that in most cases, the cost of helping is much lower than the cost that is saved as a result.

Tactical airport planning is currently done by human air traffic controllers. However it can be expected that the process will be automized more and more in the future. Currently there is a trend in Air Traffic Management (ATM) research towards on distributed planning with more autonomy for individual participants. This has been coined the 'paradigm shift' [16]. A good example is the *Free Flight* project [21, 11], in which aircraft choose their own flight paths and coordinate with their neighbours. This paper follows the trend by proposing an airline-centered decision mechanism for tactical planning.

In the following section, we will define the criteria of the resource allocation problem. We will then describe some approaches that can be taken to the problem, like voting-based mechanisms, reciprocity-based mechanisms, monetary systems and market systems. This will lead to an initial, straight-forward market mechanism for plan repair. We will show that it is vulnerable to exploitation, and show a natural solution, that of retaliation. The problem with this strategy however is that it is dominant to deviate from it. We will solve this problem by changing the the monetary system that lies at the basis of the market mechanism.

2. REPEATED RESOURCE ALLOCATION

2.1 Efficiency and fairness

We compare repair candidates in terms of agents' utilities. We use $u_a(r)$ to denote agent a's utility for repair candidate r. We use the term *effort* to denote the negation of utility, i.e., if for example $u_a(r) = -10$ we say that repair candidate r requires an effort from agent a of 10. For a sequence of elected candidates $R = \langle r_1, r_2, \dots, r_n \rangle$ we use $u_a(R)$ to denote $\sum_{i=1}^{n} u_a(r_i)$.

As we mentioned previously, ATM plan repair needs to require as little effort of the agents as possible. Thus, we aim to maximize *efficiency*: $\text{eff}(R) = \sum_{i=1}^{k} u_i(R)$ where R is the sequence of elected repair candidates and k the number of agents.

When a group of agents realises a utility gain as a result of cooperation, the rational thing to do is to divide the gain among them. This can be done in several ways and is far from trivial [31]. However, for the ATM case this is not natural. This is because utility gains are hard to measure and hard to distribute. During the

*This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs. Project DIT5780: Distributed Model Based Diagnosis and Repair

†The first author is a student

tactical planning phase many conflicts potentially lead to crashes or near-crashes. A simple rescheduling can often resolve them. Such a rescheduling has a huge utility gain, e.g. from crash to non-crash. In order to divide these kind of gains over many agents, one would have to accurately assess utilities of potential crashes and near-crashes. Also, one would have to divide these utility gains among the agents somehow. It is hard to see how this could be done, as there is nothing of utilities of that magnitude that can be given to agents, e.g., the utility of a preferential treatment in the departure scheduling doesn't compare to the utility of an aircraft not crashing. It is certainly unthinkable to divide the non-crashes over the agents; crashes and near-crashes are simply out of the question at any time.

From our discussion with experts we know that fairness in ATM is measured with respect to effort. If, as a result of bad weather, the plan becomes infeasible and needs to be repaired, the effort should be divided equally over the airlines. However, if one airline is responsible for a conflict, this airline should preferably be the only one involved in the repair.

In our scenario, agents are allowed to spend effort to solve each others problems. We propose the following equation as defining a fair distribution of effort:

$$\text{for each agent } a\text{:} \quad \sum_{i=1}^{k} E_{a,i} = \sum_{i=1}^{k} E_{i,a} \tag{1}$$

where $E_{a,b}$ represents the total effort that agent a has spend to help agent b after some large number of rounds. $E_{a,a}$ represents the effort that agent a has spent to solve its own problems and $\sum_{i=1}^{k} E_{i,a}$ represents all the effort that was needed to solve problems that agent a was responsible for. We assume that responsibility can be shared between agents for a single conflict and if no one can be held responsible, as in the bad weather example, everyone is considered responsible and should bear an equal share of the repair burden.

Note that it is not appropriate to simply divide all the spent effort equally over all agents. If that were done, an airline would hardly be motivated to prevent conflicts from occurring; the effort needed to solve it would be divided over all agents, so it would experience only a fraction of the consequences.

To measure the fairness of allocations, we will for each agent calculate the balance of given and received effort, i.e., $\sum_{i=1}^{k} E_{a,i} - \sum_{i=1}^{k} E_{i,a}$. If the variance in these balances is small, the allocation is fair; if the variance is high, the allocation is unfair.

In many scenario's, including the ATM scenario, it is not possible to find an allocation that is both optimally fair and efficient [31, 1]. In that case a trade-off between the two has to be found which can be done in several ways [15, 13]. In this paper we will aim for a sensible trade-off, leaving some room as where it should lie exactly.

2.2 Efficient Allocation

When a number of agents needs to choose the most efficient from a set of repair candidates, the simplest mechanism to use is that of *weighted voting*. In every round, every agent declares its effort for each of the candidates. The efforts are added up for each candidate and the one scoring best is elected as the winner. If all agents declare their efforts honestly, this will lead to the efficient candidate being chosen.

However, this mechanism is very sensitive to manipulation. An agent will be tempted to exaggerate its preferences trying to get a candidate selected in which its contribution is little or none. If this happens, the mechanism is no longer efficient.

The area of *mechanism design* is concerned with the design of mechanism in which agents are best off when they are truthful. The well known Vickrey-Clarke-Groves (VCG) mechanisms [19] introduce a taxing rule in the weighted voting system that nullifies the advantage of manipulation. The VCG mechanisms have been applied to repeated auctions as well [6, 9]. Although these mechanisms are elegant, there are a number of practical difficulties. For instance, the VCG mechanisms require preference functions to be *quasi-linear* (see [25]), i.e., agents are willing to exchange utility and money in a fixed rate. This assumption is invalidated when an agents wealth is limited, as in [6], and it is poor, i.e., it has only few credits. An agent will then spend its credits strategically, i.e., will ascribe more utility to a single credit. Another difficulty is the fact that VCG mechanisms maximize efficiency and ignore fairness [20, 9].

As the VCG mechanisms don't apply in our case, we have to look for other ways to solve the manipulation problem.

2.3 Incentive to cooperate

The problem we face is caused by the fact that selfish agents need to cooperate and that cooperation requires a non-selfish act from one or more agents. In the weighted voting case, an agent that states its utilities truthfully acts cooperatively by not manipulating the mechanism and willingly running the risk of being chosen to perform some action. An agent that exaggerates its preferences acts selfishly, trying to prevent the candidates in which it participates from being elected.

An agent might want to avoid helping another agent in a single instance. In a repeated situation however, an agent might hurt himself by acting selfishly if other agents will, as a result, refuse to help him when he is in need. Similarly, an agent will be willing to help if it expects to receive help in return another time. Thus, *reciprocity* is an incentive for selfish agents to cooperate.

It is interesting to see that reciprocity is closely related to our definition of fairness. If after some number of rounds all agents would have reciprocated each other perfectly in terms of effort, equation 1 would be satisfied.

Reciprocal behaviour is a phenomenon much researched in social, economical and biological sciences. Experiments have shown that humans have strong reciprocal tendencies [3, 14]. The same is true for companies [8, 10].

Reciprocity has also been tested in artificial environments. In 1980, Axelrod opened a competition to which participants could send in agents that would play a game known as the *iterated Prisoners Dilemma* against each other [1]. The iterated Prisoners Dilemma shares with the ATM scenario the property that agents have a choice between defecting, which is appealing in the short-term, and cooperating, which is better in the long-term but only if done mutually. In the first two competitions a very simple, purely reciprocal strategy called *tit-for-tat* emerged as the winner, outperforming all other strategies. The experiment showed the strength of reciprocity as a strategy when faced with selfish opponents.

More recently, Sen examined the role of reciprocity in a packet delivery scenario [26]. Just as in the iterated Prisoners Dilemma, agents repeatedly find themselves in two-player games where they have the option to either cooperate or defect. This scenario is a generalization of Axelrod's scenario in two respects. First, cooperation need not be symmetric. In Sen's scenario, in each game one agent, the beneficiary, requests another agent, the benefactor, to carry a packet for him. If the benefactor cooperates he incurs costs while the beneficiary saves costs. Distribution of these roles is not necessarily uniform - it might be the case that after many rounds, one agent has been in the role of benefactor more often than another. Secondly, the game that agents play doesn't need to be the same every time, there may be different circumstances or pay-offs that may lead an agent to different decisions. Therefore, agents need some measure by which to value and compare services.

As these assumptions are relaxed, Sen argues that strict tit-for-tat is no longer the optimal strategy. Instead he proposes that reciprocal agents use a stochastic function in which an agent bases its decision to help another on the balance of exchanges the other has with the collective of agents. If an agent has given an equal or higher amount of help to others than it has received, the chance of honoring its request is high. If an agent has given less than it has received, the chance that its request will be honored is low.

The ATM scenario we use in this paper is, in its turn, a generalization of Sen's scenario. We assume that agents might disagree on the measure of help that is provided. This is something that is likely to occur. In a reciprocity mechanism, an agent would like its cooperative act to be assessed as high as possible by others, so that it will have a good balance. The receiving agent would like to assess it realistically or even lower, so that his balance will be better. It is thinkable that negotiation over this measure precedes cooperation. In this paper we will let the benefactor set the measure of help provided. The beneficiary has the option to then accept the offer or not. We will examine different strategies of pricing by the benefactor.

2.4 Monetary systems

Money can be seen as an administrative system for reciprocity. Someone who has provided goods or services to others earns money, which can then be spent to obtain goods or services. Money eliminates the need for the *double coincidence of wants* that occurs in barter; for an exchange to occur in a barter system, both sides must have an object that the other wants, and must want the other's object more than its own. This is often too strong a requirement in reality. Money overcomes this requirement.

When a monetary system utilizes a single currency, we call it a *single issuer* monetary system, as currency is issued by a single authority, such as a bank. When each user may issue currency, we call the system a *multi issuer* monetary system. There are a few examples of multi issuer systems. One example is the situation that occurred in the United States during the westward expansion in the nineteenth century. The government did not have its own money yet and allowed banks, companies, stores, churches and even individuals to issue their own paper money. This system led to great insecurity as some banks would issue more notes than they could back up with gold or silver, thereby earning the nickname 'wildcat banks'.

Several multi issuer monetary systems have been proposed for and used in peer-to-peer systems [17, 27]. By coupling the rate of a users currency to its reputation, malicious users can be put out of the action. We make some comments on this in section 6. Another advantage is that a secure multi issuer system doesn't need a central administrative authority.

The current situation of currencies in the world is somewhere in between a single and a multi issuer system; there are more than one issuers of money, but not everyone is allowed to issue.

Multi-issuer monetary systems potentially require a lot of administration from its users. Users must keep track of the reliability of many issuers to be able to assess the reliability of credits. If different credits may have different values, determining which credits to use for a payment becomes non-trivial (see section 5.2). Clearly it would be highly impractical for human beings to use such a system. However, for computational agents it might be feasible, as they do have the computational power necessary.

2.5 Market mechanisms

The term *market mechanism* refers to a mechanism in which users buy and sell goods or services, using money for payment, leading to an efficient distribution of these goods or services. Prices are a result from the supply of and demand for a particular good or service, assuming perfect competition between suppliers and between consumers. A market mechanism essentially combines two principles we discussed before: efficiency and reciprocity. Efficiency is achieved by the principle that the person who values an item the most will pay the most and therefore obtain it. Reciprocity is a direct result of the usage of money, as we discussed before.

Although a pure market mechanism is usually said to achieve an efficient distribution of goods, viewing money as a reciprocity mechanism in fact cancels this property. For money to enable reciprocal exchange, it needs to be scarce - if it wasn't, those with little wealth would never be compelled to reciprocate. If it is scarce, a buyer could have too little to buy a good he desires. Or he could have enough but feel the urge to be thrifty. So, a good doesn't necessarily go to the person desiring it the most.

A market mechanism in fact achieves a trade-off between efficiency and fairness. The more agents' wealths are allowed to vary, the more efficient the resulting distribution will be. The less agents wealths' are allowed to vary, the more fair the resulting distribution will be.

Market mechanisms have been applied to computer science problems in areas such as e-commerce, resource allocation [28] and distributed planning [30]. Usually some type of *auction* is used as pricing mechanism. It has been shown that the distributed auction model has significant scaling advantages over the centralized auction model [18].

We will use the auction-based market mechanism as a starting point for our ATM plan repair mechanism. The weighted voting procedure we described before is in fact a *reverse auction* with *externalities*. A reverse auction is an auction with one buyer and many sellers, also called *procurement auction*. Externalities are said to exist when a certain combination of items has more or less utility to the owner(s) than the sum of the utilities of the individual items.

An important choice to be made is whether we want to use real world money as the medium of exchange or some other currency. Using real currency has one disadvantage: it favours rich participants. When participants may bring money into the mechanism or take it out of it, the reciprocity process is disturbed. In the ATM scenario this would mean that rich airlines can get better slots, gates and runways than others. Some might argue that that is the way of the world, but we stress that ATM plan repair is supposed to be fair to all participants. Therefore, a closed system with a private currency is preferable.

3. ISSUES WITH MARKET MECHANISMS

3.1 Exploitation

Although a monetary system facilitates reciprocal exchange, it does not guarantee a fair distribution of utility. Even if agents would start with the same amount of money and, after many exchanges, would all end up with exactly the same amount again, the distribution could be unfair. This happens when one or more prices that have been paid did not correspond to the utility that was transfered. This is particularly likely to occur in the following situation. A consumer needs a certain good or product, and there is only one supplier (or a few suppliers forming a cartel) that offers this good or service. The supplier can then raise its price up until just below the point where the consumer is not willing to buy any more. For instance, if an arriving aircraft is out of fuel, needs to land immediately and asks another aircraft to make room, the other aircraft could ask a ridiculously high price, knowing that the landing aircraft will pay anything in order to prevent a crash. We will call this phenomenon *exploitation*. Exploitation in a market mechanism leads to unfairness. An exploiting agent gains credits with more value than it loses with supplying the good or service, which it can spend again to gain other goods or services. In the end he will have gained goods or services with more value than it supplied, hence the unfairness.

To measure the effect of exploitation, we have set up a benchmark experiment in which aircraft repeatedly find themselves in planning conflicts, and need help from others to solve them. One experiment consists of 1500 rounds, with 15 airline-representing agents. To keep things simple, we assume that for each conflict there is one agent responsible, the *problem owner*. Each agent is problem owner 100 times. For each problem there are 15 repair candidates generated, such that there is a default candidate of doing nothing, with a very low utility for the problem owner, and a number of candidates which involve one, two or three other agents with randomized utilities. The candidates that involve other agents have, by average, a lower total cost than the default candidate. In every round, the problem owner opens an auction for the candidates, receives the other agents' price submissions and buys the cheapest one. After all the rounds the scores are calculated for the agents. The score of an agent is the average balance of effort per round, i.e. received effort minus given effort, plus its monetary balance. The monetary balance is added to the score because we assume that earned money will still be spend in the future (cf. [7]). If an agent's balance is in the red, he will still have to help others to pay off its debt.

Although we do assume that monetary surpluses and debt are evened out in the future, we do not specify how. It could be achieved by setting limits on debts or raising taxes, but in this paper we abstract from this and assume that it will just happen. This doesn't mean that it is not important - as soon as such a rule is enforced, money becomes scarce and agents will not be able to buy anything they want any more; the trade-off between efficiency and fairness comes into play. We will not do that now however, but only focus on exploitation.

There are two types of agents: coalition agents and exploiters. Coalition agents are agents that are truthful when submitting prices. We named them 'coalition agents' because throughout this paper they will try to defeat the 'wrongdoers', of which there are two types: exploiters and, as we will later see, forsakers. Exploiters exhibit the described exploitation behaviour. If there are more than one exploiters involved in the cheapest candidate, they will work in perfect unison; they will each raise their price, but still make sure that the total price doesn't become too high. Their 'profit' is split randomly over the exploiters.

We conducted the experiment six times, with an increasing proportion of the agent population being exploiters. The results can be seen in the left chart in figure 1. In the first column, with only coalition agents, the variance is close to zero. When exploiters start to join in, they score much better than coalition agents, which is to be expected. If one assumes that agents will change their strategy to a dominant strategy, the ratio of exploiters can be expected to increase and we move up to the right in the graph, ending up in the column on the very right. In this scenario the variance is high, indicating an unfair, undesirable situation.

The reason that the variance is high in the last scenario is that we have explicitly modelled the fact that some agents have more exploitation opportunities than others, which we believe is plausible for real life situations. For instance, an airline that has many flights on gates next to each other can easily make changes and thereby

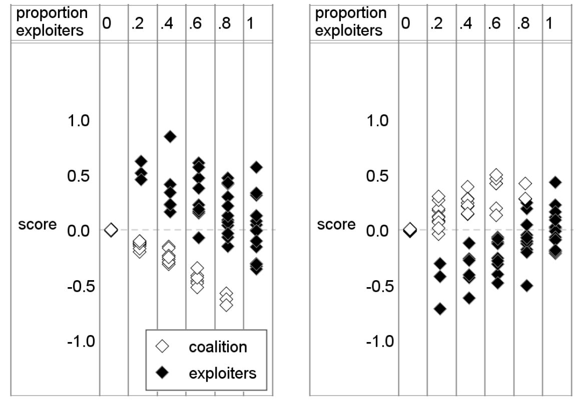

Figure 1: Results of the experiments with coalition agents and exploiters. On the left the experiments of the benchmark scenario. On the right the experiments with collective relatiation. Every column denotes one experiment.

accommodate for someone else. It needs to spend less effort to help than others, so it should be a lot cheaper, but it can easily exploit this situation. Because some agents exploit better than others, if all agents would exploit to their fullest, some agents will gain much more from this than others. The result is the unfair situation we saw in the last experiment.

3.2 Collective retaliation

A straightforward remedy against exploitation, is what we will call *collective retaliation*: agents ask higher prices to the exploiters to nullify the profit they made from exploitation. Agents that are being exploited by an agent should estimate the measure of exploitation and pass this information on to all other agents. Every agent then calculates for every other agent a trust rate, where a low trust rate means that an agent asks too high prices. The trust $t_{a,b}$ that agent a has in agent b is

$$t_{a,b} = \frac{E_b}{P_b}$$

where E_b is the sum of the estimations of the realistic prices agent b should have asked and P_b is the sum of the prices agent b did ask. When problem owner w now asks for price submissions for candidate r, an agent a calculates the price $p_{r,a}$ for its part in r by

$$p_{r,a} = \frac{-u_{r,a}}{t_{a,w}} - S$$

where $u_{r,a}$ is the utility of a for its part in r, $t_{a,w}$ is the trust agent a has in the problem owner w, and S is a punishment factor to make sure that exploiters are not only compensated but also punished a bit, to discourage them from exploiting. S should be greater than zero for exploiters (i.e. when $t_{a,w} < 1$) and zero for non-exploiters.

We have implemented this strategy in a second experiment, of which the results are shown in the right chart in figure 1. It can be seen that the strategy is successful; exploiting is dominated by the coalition strategy.

The success of the collective retaliation relies on two factors. First, the exploited agents should correctly estimate and truthfully communicate the amount of exploitation. We will make this assumption in this paper - an airline should be able to make a reasonable estimation of someone else's costs, especially when agents are allowed to give supporting arguments for their prices and discussion is possible. Also, there is no strong direct advantage for an agent to exaggerate another agent's exploitation. Only if an agent consequently lies about all other agents could it cause a relative advantage for itself, but this can hardly go unnoticed and will harm this agent's reputation. The second requirement for collective retaliation to be successful is that all agents correctly enforce the pricing rule. This is something we cannot assume, since agents have reasons not to, as we will see in the next section.

3.3 Forsaking

The method of collective retaliation works well if all agents enforce the pricing rule. But, a single agent who should at a certain moment raise its price as part of collective retaliation, can have an incentive to raise it less than it should. If there are more suppliers of a desired good or service, an agent might want to lower its price to win the deal. This is especially attractive if others raise their price as a result of collective retaliation; he can raise its price slightly less, win the deal, and make an attractive profit. We will call this kind of behaviour *forsaking*: deliberately asking a price between the realistic price and the one that should be asked as a result of collective retaliation, with the aim of winning the auction and making a profit.

If there are more than one forsakers, they will compete against each other. They will each try to win the deal by setting their price lower than that of the other. This will drive the price down, until all but one forsaker are at their realistic price. If there are enough forsakers, the effects of the collective retaliation rule can in this way be fully nullified.

In order to test the effects of forsaking, we introduce a new type of agent in our experiment, the forsaker, and let it compete against coalition and exploiting agents. The coalition agents and exploiters use the collective retaliation pricing rule, the forsakers sell just below that price if the chance occurs. We've explicitly generated conflicts that are prey to exploiters, as well as conflicts that are prey to forsakers. In the first type of conflict there is one candidate with at least one exploiter involved that is cheaper than all other candidates. In the second type there are several cheap candidates, but in at least one of those there is a forsaker involved. We tested the strategy in 6×6 different distributions of exploiters, forsakers and coalition agents. The results can be seen in figure 2. The chart shows that forsaking is a dominant strategy in every situation. Thus, the ratio of forsakers can be expected to increase. The last six experiments show that, when everyone forsakes, it is also often dominant to exploit. Thus, the ratio of exploiters can be expected to go up, in which case we will end up in the very last column, which shows an unfair, hence undesired situation.

A possible remedy against forsaking could be that the coalition retaliates them too. But this is far-fetched. Forsaking is harder to detect than exploitation, since the drop in price can be quite small. A forsaker could claim that it is not forsaking, but that it is asking a realistic price, which happens to be lower than expected. If the coalition fails to recognize a forsaker, the consequences can be much greater than if it fails to recognize an exploiter. Suppose that the coalition doesn't recognize exploiters and forsaker that stay within 5% of their realistic price. Then, an exploiter that alters its price by 5% will only gain 5%, while the forsaker might gain a lot more, namely the difference between the retaliation price and its realistic price minus 5%. Also, neither the forsaker nor the exploiter who is being retaliated has any incentive to complain about the forsaking - the exploiter likes forsakers. So, coalition agents should then check each bid that is submitted in auctions even in which they have no part. This is a lot of work and it requires that agents reveal pricing information to more agents than only the problem owner, which is less desirable since agents like their information to remain as private as possible. We don't find this remedy feasible.

The reason that forsaking occurs is the fact that the credits that are unfairly earned can be spent again without any problems. We will solve this problem by introducing a monetary system in which this 'dirty money' cannot be spent that easily any more. The system is inspired by an existing monetary system called WAT, which we will describe in the following section.

4. LOCAL EXCHANGE TRADING SYSTEMS

Local Exchange Trading Systems (LETS) are local, non-profit exchange systems in which goods or services are being traded without the use of real money. Instead, a *complementary currency* is used. In most LETS users start with a balance of zero credits and can earn and spend these immediately. Being in the red is allowed and interest-free. In fact, about half of the users of the system will have debts, while the other half has a positive balance. Usually there is a certain limit on debt, forcing the debtors to give goods or services back to the community.

4.1 The WAT-system

The WAT-system [29] is a LETS designed by Eiichi Morino in 2000, which has been actively used in Japan since. Its most distinctive feature is the fact that it doesn't need a central bank or administration to keep the books for its users. Instead, users *issue* their own credits by ordering or printing a WAT-ticket and putting their name and signature on it. By signing a ticket, a user vouches for its value, i.e., he promises to exchange some goods or services in return when asked. When a user accepts a ticket and wants to spend it in another deal, he puts his name and signature on the ticket as well. In this way the list of users on a ticket grows as it *circulates*. All these users vouch for the value of the ticket. If the last one should fail to keep its promise, the second to last is liable, and so forth. The longer the list of names on a ticket, the more trust a user will have in it. Thus, the WAT-system is essentially a trust-based multi issuer monetary system. Finally, when a ticket travels back to its issuer, it is invalidated, which is called *redemption*. Every WAT-ticket goes through the same three stages: issuing, circulation and redemption.

An electronic version of the WAT-system called *i*-WAT was developed by Saito in 2003 [24]. The system lets users issue, circulate and redeem tickets electronically. To prevent against fraud, OpenPGP is used and it is required that spending of a credit is approved by the issuer of that credit each time.

In the WAT-system, as in most monetary systems, each credit is assumed to have the same value. Although this is highly practical, it is not theoretically sound. A user could very well trust the issuer of one credit more than that of another, and therefore valuate the ticket higher. It would be infeasible for human users to have to assess the value of every credit individually. For computational agents however, this is possible. In the following section we will introduce a WAT-like monetary system with individual credit-valuation, and show how it can be applied to the ATM planning problem.

5. AN EXPLOITER- AND FORSAKER-FREE MECHANISM FOR ATM PLAN REPAIR

5.1 Fine-tuning the trust mechanism

We propose the usage of a monetary system similar to the WAT-system, but combined with a trust model. Credit value is established per credit, based on the list of users on it.

Similar to the collective retaliation mechanism, every agent a has a trust value $t_{a,b}$ for every agent b (including itself). This value

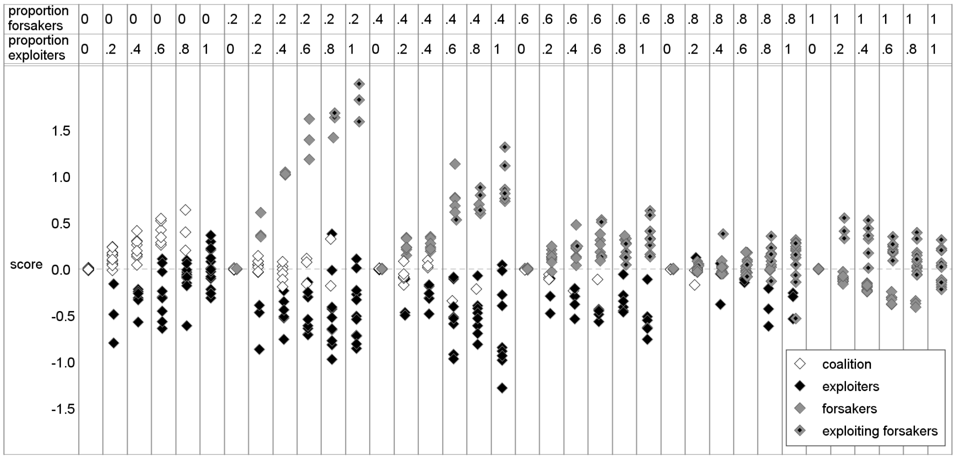

Figure 2: Results of the experiments with collective retaliation and forsaking. The first six columns on the left correspond with the experiments described in section 3.2. In subsequent series of experiments, the number of forsakers is increased, until there are only forsakers in the last six columns.

represents the effort that agent a expects agent b to provide to a in return for one undevaluated credit.

The pricing rule is dropped. Agents now simply need to ask their realistic prices. Agents should still estimate other's realistic prices and adjust their trust values accordingly. To estimate the trust value they remember past prices and estimations, and average over these. So at any moment

$$t_{a,b} = \frac{E_b}{P_b} - S$$

where E_b is the sum of agent a's estimations of the realistic prices agent b should have asked, P_b is the sum of the prices agent b did ask and S a punishment term, which should be positive when $P_b > E_b$ and zero otherwise. Agents should share with each other the received and estimated prices, so that the exploitation of one agent towards another affects its trust rating with all the other agents as well. Note that $t_{a,b}$ may be greater than 1.

An agent valuates a credit as follows. If agent a receives a credit c with a list of users $\{ b_1, b_2, ..., b_j \}$, it assesses its value as

$$v_a(c) = t_{a,b_1} * t_{a,b_2} * \ldots * t_{a,b_j}$$

So, every credit c has a value $v_a(c)$ for every agent a. Note that in the WAT-system every credit had value 1. We define for a set of credits $C = \{ c_1, c_2, \ldots, c_n \}$

$$v_a(C) = \sum_{i=1}^{n} v_a(c_i)$$

In every round, the problem owner chooses the cheapest repair candidate. However, this time this not only depends on the prices asked, but also on the value of the credits the problem owner possesses, both in the eyes of the problem owner himself as in the eyes of the ones who are being paid. For instance, if the problem owner has credits that are valued much higher by agent a than by agent b, and these agents offer the same service for the same price, it would buy the service from agent a since he needs to spend fewer credits then.

When a problem owner has received all price submissions, first the *optimal payment* for every candidate has to be determined. If we identify the agents by numbers $1, 2, \ldots, k$, given a repair candidate r with a problem owner w with $0 \leq w \leq k$, the prices $P = \{ p_1, p_2, \ldots, p_k \}$ that the agents ask for their share in r with p_w being 0, credits $C_w = \{ c_w^1, c_w^2, \ldots, c_w^n \}$ in possession of agent w, I_w the infinite set of credits agent w can issue, and valuation functions $v_1, v_2, \ldots, v_k$, the optimal payment $optpay(r, w, P, C_w) = \{ T_1, T_2, \ldots, T_k \}$ where $T_1, T_2, \ldots, T_k$ are disjunct subsets of $C_w \cup I_w$ such that $\forall x \quad v_x(T_x) \geq p_x$ and $\sum_{x=1}^{k} v_w(T_x)$ is minimal.

In other words, the problem owner pays the agents involved in the candidate in such a way that for each agent, the value of the credits it receives is equal or higher than the price it asked, and the value of all spend credits is minimal to the owner. This implies for instance that credits typically go to the agents that value them the most.

After all the optimal payments have been determined, the problem owner chooses the cheapest candidate - the one with the lowest sum of payment costs and personal utility:

$$\min_r \left[v_w\left(\bigcup optpay(r, w, P, C_w) \right) + u_w(r) \right]$$

After the winning candidate has been determined, it is enforced and the corresponding payments are made.

The main innovation of this monetary mechanism is the fact that a credit's value is determined by the reputations of the agents who have used it. So, if a credit goes through the hands of an exploiter, it loses value. As any other agent can see, the name of the exploiter is on the credit and therefore it is valued lower. As a result of this, the exploiter has trouble spending its money, since every credit it likes to spend turns out to be worth less than when he received it. More importantly, forsaking is no longer an attractive strategy. Forsakers used to make a profit by deviating from the collective retaliation rule. But now there is no such rule anymore. To forsake, they should raise their trust value of an exploiter. If they do this, they will win the deal but obtain credits that have lost worth. When spending these on non-forsakers, they will incur a loss.

This monetary system is radically different from existing systems. Its most striking property probably is the fact that it doesn't have currencies. Each credit is identified by its list of users and of these there are many possible. As a result, some well known phenomena that occur in other monetary systems will not occur. For instance, in the international currency market, some currencies are more reliable than others and therefore more in demand. Countries and companies prefer currencies with stable rates above others. In the signed money system, a user cannot just use a specific currency. Its incentive is to have a good reputation, only then will the worth of its money be stable.

5.2 Experiment

We implemented the proposed monetary system and tested it in the same scenario we used before. We incorporated the punishment term into the trust formula of a coalition agent a for an agent b as follows:

$$t_{a,b} = \frac{E_b}{P_b - s(P_b - E_b)}$$

This means that exploiters are, besides compensated for their exploitation, punished extra proportionally to the amount of exploitation. Remember that the extra punishment was necessary to make the exploiters score lower than the coalition agents. We chose $s = 1$, although other values are possible as well. A forsaker f will set its trust level for an exploiter somewhere between $t_{a,b}$ and 1. We used the trust function shown above with $s = 0$ for a forsaker, although other trust functions are possible as well. With $s = 0$, the forsaker still retaliates, but he leaves out the extra punishment.

In every round, the optimal deal has to be calculated for every candidate. Unfortunately, determining the optimal deal is a combinatorial problem which is infeasible to solve brute-force[1]. Although optimizations will be possible, we have chosen to use a simple method that quickly finds a good, but possibly suboptimal solution in linear time. The problem owner randomly selects a credit from its purse and assigns this to, from the agents that are still 'underpaid', the one that values the credit the most. This method has the important property that credits tend go from agents who value them less to agents who value them more. The fact that this payment is probably suboptimal results in a small loss to the problem owner. However, as all agents incur this loss many times, its effect can be expected to average out over the agents and not give any significant advantage to anyone.

The results of the experiments can be seen in figure 3. It can be seen that the coalition strategy dominates exploitation and forsaking in every experiment where coalition agents participate. If all agents adopt this strategy, we get the situation in the first column, with the smallest variance and therefore the most fair allocation.

Note that, unlike for instance [23], we chose to test the performance of strategies in many different populations of agent types. This is to show that the coalition strategy is dominant to others in *any* situation.

6. RELATED WORK

Multi-issuer currencies have been proposed in the peer-to-peer community before. In [27] a so-called *lightweight currency protocol* is proposed in which users issue money to pay for usage of resources. The authors don't analyse the economic aspects of such a system and, in fact, expect only a few highly coveted currencies to remain, analogous to the real world. In [17] it is shown that token- and trust-based P2P sharing mechanisms can be generalized to so-called *stamp trading* mechanisms, which are multi-issuer monetary systems.

[1]If n is the number of credits in possession of the problem owner and x the number of agents involved in the candidate in which the most agents are involved, then there are $(x + 1)^n$ possible payments the owner can make. So, for instance, if the problem owner has 100 credits and there are three agents to be paid, there are 10^{60} possible payments.

In both P2P systems mentioned above, credits are signed by their issuer only. The idea is that, the more reliable an agent is on redemption, the higher the rate of its credits will be. When users fail to keep their promises, deliver bad quality or exploit, their rate will go down and they will have problems using their own credits for payment. We have run experiments with this kind of currency and found that there are two problems with this idea. First, agents can compensate a declining rate by issuing more credits. This problem can partly be solved by setting limits on the amount an agent may issue or the rate its currency may drop to. Secondly, a malicious agent can at a certain moment stop issuing credits and only use credits of other agents. This is a serious problem because the agent could go on with its malicious behaviour without suffering from its declining rate. To the best of our knowledge, this problem had not been addressed yet. The mechanism proposed in this paper remedies this shortcoming.

In [22] the phenomenon of *quality swindling* is mentioned, which is the case when a seller deliveres less than he promised. This phenomenon is closely related to exploitation, as in both cases price paid by the buyer is more than it should be. We expect the credit valuation rule to neutralize the advantage of the swindler just as it did with exploiters.

It is often observed that human and corporate behaviour is not purely based on reciprocity but also on commitment [5, 2]. We think that the proposed mechanism is perfectly suited for cooperation based on commitment. The used trust value can serve as a commitment value. In that case, the more two agents are committed to each other, the more they find each others bids appealing and the more they will cooperate as a result.

7. CONCLUSION

We presented a collaborative decision making mechanism for the ATM plan repair problem. The mechanism enables agents to jointly decide which of a number of plan repair alternatives to enforce. A market mechanism is used to achieve efficiency and fairness. Efficiency results from the use of auctions. Fairness results from the use of money as a medium of reciprocal exchange. We proposed the use of multi-issuer monetary system, in which every user signs every credit it uses. We introduced a trust-based credit-valuation model and defined optimal payments. We showed by simulation how this mechanism neutralizes two strategies that occur in existing market mechanisms. The first strategy, exploitation, occurred when an agent misused its key position to ask a higher price than realistic. The second strategy, forsaking, occurred when an agent failed to cooperate in retaliating the exploiters, thereby making an attractive profit and eventually nullifying the effect of retaliation.

We think that the mechanism is applicable to other domains than the ATM domain as well. In any situation where agents can profit from cooperation, but need to do this efficiently and fair, and where exploitation or swindling can occur, the mechanism can be used. Examples are peer-to-peer file sharing systems and grid computing. An important property of the mechanism is that it distributes effort *fairly* over the participants. This implies that it cannot be used in situations where the utility gain as a result of cooperation needs to be split fairly, for instance in the case of a joint investment.

8. REFERENCES

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

[2] I. Back and A. Flache. The viability of cooperation based on interpersonal commitment. *Journal of Artificial Societies and Social Simulation*, 9(1), 2006.

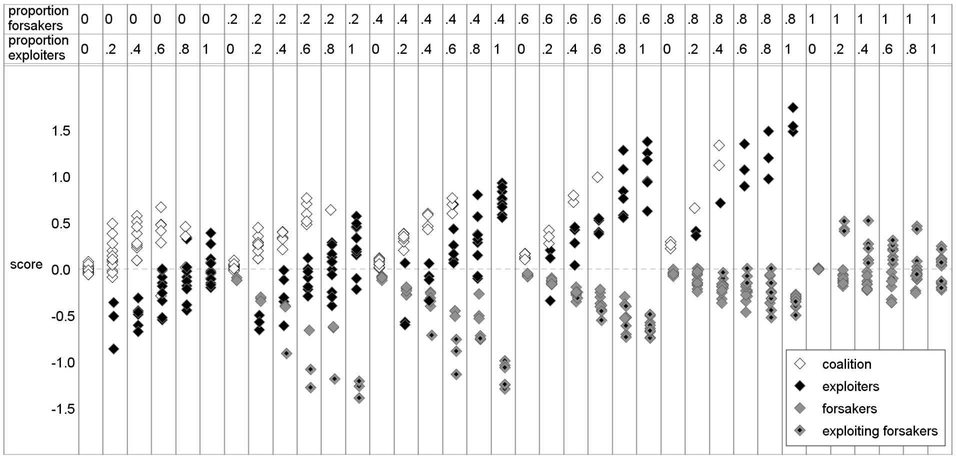

Figure 3: Results of the experiment with signed money.

[3] J. Berg, J. Dickhaut, and K. McCabe. Trust, reciprocity and social history. *Games and economic behaviour*, 10(1):122–142, July 1995.

[4] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[5] M. Emiliani and D. Stec. Aerospace parts suppliers' reaction to online reverse auctions. *Supply Chain Management*, 9(2):139–153, 2004.

[6] E. Ephrati, G. Zlotkin, and J. S. Rosenschein. A non–manipulable meeting scheduling system. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*, Seatle, WA, July 1994.

[7] S. Estivie, Y. Chevaleyre, U. Endriss, and N. Maudet. How equitable is rational negotiation? In P. Stone and G. Weiss, editors, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06)*, pages 866–873, Utrecht, The Netherlands, May 2006. ACM Press.

[8] E. M. Fich and L. J. White. Why do ceos reciprocally sit on each other's boards? *Journal of Corporate Finance*, 11:175–195, 2005.

[9] E. J. Friedman and D. C. Parkes. Pricing WiFi at starbucks: issues in online mechanism design. In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC-03)*, pages 240–241, New York, June 9–12 2003. ACM Press.

[10] J. Gimeno. Reciprocal threats in multimarket rivalry: staking out 'spheres of influence' in the u.s. airline industry. *Strategic Management Journal*, 20(2):101–128, 1999.

[11] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling. A cooperative multi-agent approach to free flight. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS '05)*, pages 1083–1090. ACM Press, 2005.

[12] R. L. Hoffman. Equitable allocation of limited resources. Technical Report 32F1103-003-R0, Metron Aviation, Nov. 2003.

[13] G. Jonker, J.-J. C. Meyer, and F. Dignum. Efficiency and fairness in air traffic control. In K. Verbeeck, A. Now, B. Manderick, and B. Kuijpers, editors, *Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC '05)*, pages 151–157, 2005.

[14] J. Kagel and A. Roth, editors. *The handbook of experimental economics*. Princeton University Press, 1995.

[15] M. Lematre, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver. Equitable allocation of earth observing satellites resources. In *Proceedings of the 5th ONERA-DLR Aerospace Symposium (ODAS'03)*, Toulouse, France, June 2003.

[16] Eurocontrol. Paradigm shift: Research agenda. Technical Report EEC Note 16/05, European Organisation for the Safety or Air Navigation, June 2005.

[17] T. Moreton and A. Twigg. Trading in trust, tokens and stamps. In *Proc. of Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, June 2003.

[18] E. Ogston and S. Vassiliadis. A peer-to-peer agent auction. In M. Gini, T. Ishida, C. Castelfranchi, and W. L. Johnson, editors, *Proceedings of the first International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 151–159, Bologna, Italy, July 2002. ACM Press.

[19] D. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, may 2001.

[20] D. C. Parkes and S. P. Singh. An MDP-based approach to online mechanism design. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS'03)*. MIT Press, 2003.

[21] Final report of RTCA task force 3: Free flight implementation. Technical report, Radio Technical Commission for Aeronautics, Washington DC, Oct. 1995.

[22] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS '02)*, pages 475–482, Bologna, Italy, 2002. ACM Press.

[23] S. Saha and S. Sen. Predicting agent strategy mix of evolving populations. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*, Utrecht, The Netherlands, July 2005. ACM Press.

[24] K. Saito. Peer-to-peer money: Free currency over the internet. In *Proceedings of the Second International Human.Society@Internet Conference (HSI 2003), Lecture Notes in Computer Science 2713*, pages 404 – 414. Springer-Verlag, Jan. 2003.

[25] T. W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 5, pages 201–258. The MIT Press, Cambridge, MA, USA, 1999.

[26] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2):179–203, 2002.

[27] D. Turner and K. Ross. A lightweight currency paradigm for the p2p resource market. In *Seventh International Conference on Electronic Commerce Research*, Dallas, Texas, June 2004.

[28] N. Vulkan and N. R. Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *International Journal of Decision Support Systems*, 28:5–19, 2000.

[29] *WATSystems homepage*. `http://www.watsystems.net/`, 2000.

[30] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behaviour*, 35:271–303, 2001.

[31] H. P. Young. *Equity: In theory and practice*. Princeton University Press, 1994.