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

Geert Jonker†, Frank Dignum, John-Jules Meyer
{geertj,dignum,jj}@cs.uu.nl
Dep. of Information and Computing Sciences
Utrecht University

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.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; J.4 [**Social and Behavioral Sciences**]: Economics; J.2 [**Physical Sciences and Engineering**]: Aerospace; K.4.4 [**Computers and Society**]: Electronic Commerce—*Cybercash, digital cash*

General Terms

Design, Economics, Experimentation

Keywords

Cooperation, Efficiency, Fairness, Exploitation, Signed Money

1. INTRODUCTION

There is a strong trend in air traffic management (ATM) research toward distributed systems [3, 2]. To reduce the workload of air traffic controllers, planning problems need to be solved locally instead of centrally and by the parties involved instead of single decision makers. This requires new distributed communication and decision techniques.

*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

†This author is a student

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

Our aim in this paper is to construct a mechanism by which airline-representing agents can jointly solve planning conflicts. Our domain is *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, by readjusting the gate and runway reservations. Plan repair must adhere to two basic principles. First, it should be *efficient*, i.e., the adjustments should be minimal. Secondly, it should be *fair*, i.e., one airline should not be the victim of the conflicts caused by another airline.

Airlines can often help each other. For instance, if one aircraft is delayed and therefore not able to leave its gate in time, the next scheduled aircraft could wait with going on-gate instead of forcing the first to leave. In current practice, it happens regularly that a delayed aircraft must leave its gate and come back later to finish its procedures. Nevertheless, as our agents are selfish they do want some guarantee that provided help will be paid back. We will introduce a monetary system that facilitates reciprocal cooperation.

2. EFFICIENCY AND FAIRNESS

We model tactical airport planning as a game with an infinite number of rounds. In each round there is a conflict and a single problem owner, which is the agent responsible for the conflict. In each round there are a number of possible plan repair schemes, called *repair candidates*, out of which one has to be elected. A candidate consists of several actions for several agents such that it solves the conflict. We use $u_a(r)$ to denote agent a's utility for repair candidate r. For a sequence of elected candidates $R = \langle r_1, r_2, \ldots, r_n \rangle$ we use $u_a(R)$ to denote $\sum_{j=1}^{n} u_a(r_j)$. Efficiency is defined as $\text{eff}(R) = \sum_{i=1}^{k} u_i(R)$ where k is the number of agents.

In current ATM plan repair, fairness is expressed in the rule that an aircraft should preferably not be involved in repairs caused by others. We assume that agents want to improve on this 'default' procedure by collaborating. This collaboration should be fair too. An agent might leave the collaboration if it feels that it is being treated unfairly. We propose the following formula to define ideal fairness:

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

where $E_{a,b}$ represents the total effort that agent a has spend to help agent b after some large number of rounds. Thus, in collaboration, an agent should give as much help as it receives.

3. MARKET MECHANISM

978-81-904262-7-5 (RPS)

In many scenario's, including the airport scenario, it is not possible to find an allocation that is both optimally fair and efficient [7]. In that case a trade-off between the two has to be found which can be done in several ways [4].

A distributed mechanism that achieves a natural trade-off between fairness and efficiency is the *market mechanism*. Consider a scenario in which problem owners 'auction' their conflicts. Agents that are involved in the repair candidates submit their costs for the actions required of them. The problem owner calculates the total cost for each candidate and buys the cheapest one. This candidate is enforced and the corresponding payments are made. To be able to buy collaboration from other agents, the problem owner needs to have enough financial means. He earns this by helping other agents with their conflicts. In that way an agent can never only receive help without giving help. Thus, the mechanism approximates fair collaboration as described in the previous section. Also, given that problem owners have enough financial means, the mechanism selects efficient candidates. When agents do run out of money, they won't be able to buy cheap candidates any more and they are punished for their unhelpfullness by having to enforce default candidates, which have a very low utility to the problem owner.

4. EXPLOITATION

In a market mechanism, sellers can ask any price they want. Usually, they compete against each other which prevents the price from becoming very high. In the plan repair scenario, this healthy competition is not so strong. Often, agents are in a position where they can safely increase their price. Consider aircraft A who would like to leave its gate five minutes later. The most obvious repair candidate involves aircraft B, scheduled next, who would have to wait for five minutes before going on the gate. Other repair candidates involve gate changes or going off-gate prematurely and are much more expensive. If aircraft B now realizes its advantage over the other sellers, it can raise its price considerably without losing the auction, resulting in an attractive financial gain. We will call this phenomenon *exploitation*. If an agent is able to exploit structurally, it gains an unfair advantage over the others. Therefore, exploitation harms fairness.

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. Each agent is problem owner 100 times. For each problem there are 15 repair candidates generated, with pseudo-random utilities chosen to reflect reality as closely as possible[1]. In every round, the problem owner opens an auction for the candidates, receives the other agents' price submissions and buys the cheapest one. There are two types of agents: coalition agents, who submit prices truthfully, and exploiters, who exploit when they have the chance. 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.

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. It can be seen that exploitation is a dominant strategy in every situation. When every agent exploits, the variance in scores is high, indicating unfair collaboration.

[1]We have explicitly modeled the fact that some agents are more often able to exploit than others. In the unlikely case that all agents can exploit equally much and are exploited equally much, there would be no problem, as all agents would experience equal advantages and disadvantages.

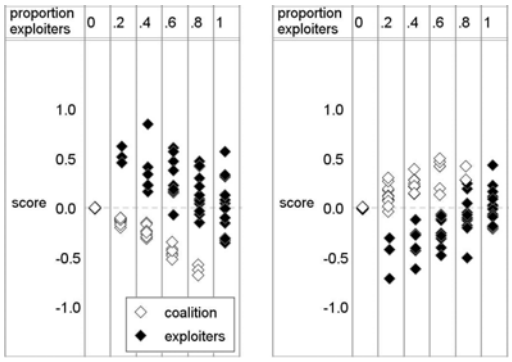

Figure 1: Experiments without (left) and with (right) the collective retaliation rule.

5. 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.

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.

Unfortunately, it is dominant to deviate from the collective retaliation rule. Suppose that an airline, which is known to be an exploiter, opens an auction. There are several airlines who can help him. As a result of collective retaliation, they all raise their price considerably. If the prices are close to each other, a single selling airline might now be tempted to lower its price to win the auction. It then makes a nice profit, since prices are higher than production costs. We will call this phenomenon *forsaking*. 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. We tested the strategy in 6×6 different distributions of exploiters, forsakers and coalition agents. A fragment of the results can be seen in figure 2, on the left. The chart shows that forsaking is a dominant strategy when there are few forsakers. This was the case in all other experiments as well. In the experiments where all agents forsaked, the variation in scores was high, indicating an unfair situation.

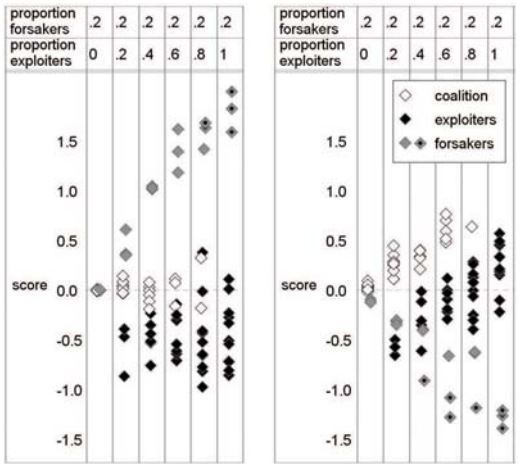

Figure 2: Experiments with ordinary money (on the left) and spender-signed money (on the right).

6. SPENDER-SIGNED MONEY

The reason that forsaking occurs is the fact that the credits that are unfairly earned can be spent again without any problems. We solve this problem by introducing a monetary system in which this 'dirty money' cannot be spent that easily any more. In this monetary system, each user that spends a credit adds its signature to that credit. Thus, a credit always bears a history of users. This kind of systems has been proposed and implemented before [6, 5, 1]. In our proposal, credit value is established per credit, based on the list of users on it.

Using this monetary system, agents need not apply the collective retaliation pricing rule any more. Agents now simply need to ask their realistic prices. Agents should still estimate other's realistic prices and adjust their trust values accordingly. 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. 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. This candidate 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.

We implemented the proposed monetary system and tested it in the same scenario we used before. To calculate the optimal deal we used an approximation algorithm with linear time complexity. The results of the experiments can be seen in figure 2 on the right. It can be seen that the coalition strategy dominates exploitation and forsaking when there are few forsakers. This was also the case in all other scenario's. If all agents adopt the coalition strategy, collaboration is fair (see figure 1).

7. CONCLUSION

We presented a coordination mechanism for the ATM plan repair problem. The mechanism enables agents to collaborate fairly and efficiently. We proposed the use of spender-signed money, with every user signing 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 can occur when ordinary money is used. The first strategy, exploitation, occurred when an agent misused its key position to raise its price. 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 fairly, and where exploitation can occur, the mechanism is applicable. Examples include peer-to-peer file sharing systems and grid computing.

8. REFERENCES

[1] D. Chaum and T. P. Pedersen. Transferred cash grows in size. In *Advances in Cryptology - EUROCRYPT '92*, volume 658 of *LNCS*, pages 390–407, Berlin, 1993. Springer-Verlag.
[2] Airport CDM applications guide, July 2003.
[3] J. M. Hoekstra. *Designing for safety: the 'free flight' air traffic management concept*. PhD thesis, 2001.
[4] G. Jonker, J.-J. C. Meyer, and F. Dignum. Efficiency and fairness in air traffic control. In *Proceedings of BNAIC '05*, pages 151–157.
[5] K. Saito. Peer-to-peer money: Free currency over the internet. In *Proceedings of the HSI '03*, volume 2713 of *LNCS*, pages 404 – 414. Springer-Verlag.
[6] *WATSystems homepage*. `http://www.watsystems.net/`, 2000.
[7] H. P. Young. *Equity: In theory and practice*. Princeton University Press, 1994.