Wirtschaftsinformatik Proceedings 2003 Wirtschaftsinformatik

September 2003

Optimizing Strategy in Agent-Based Automated Negotiation

Jörg Meyer
Albert-Ludwigs-Universität Freiburg

Torsten Eymann
Albert-Ludwigs-Universität Freiburg, eymann@iig.uni-freiburg.de

In: Uhr, Wolfgang, Esswein, Werner & Schoop, Eric (Hg.) 2003. *Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität*, 2 Bde. Heidelberg: Physica-Verlag

ISBN: 3-7908-0111-9 (Band 1)
ISBN: 3-7908-0116-X (Band 2)

Optimizing Strategy in Agent-Based Automated Negotiation

Jörg Meyer, Torsten Eymann

Albert-Ludwigs-Universität Freiburg

Abstract: Digital Business Agents (DBAs) can assist human buyers and sellers in electronic markets by strategically conducting automated negotiation to minimize transaction costs. However, the resulting information systems are complex environments, which are hard to assess analytically. The DBAs' strategies will thus need to incorporate heuristics, which adapt to ever changing environment conditions using machine learning algorithms. This article compares the performance of an evolutionary algorithm, numeric optimization methods and a hybrid, economics-based mechanism, using the electronic market setting of the multiagent system AVALANCHE *as an example.*

Keywords: Agent-Mediated Electronic Commerce, Machine Learning, Adaptive Strategies, Automated Negotiation

1 Why automated negotiation strategies have to adapt

Digital Business Agents (DBAs) [Dieb01; Eyma03] can assist human buyers and sellers in digital business processes and environments to save transaction costs. They monitor other agents and the environment continuously, e.g. by making price comparisons between different suppliers in the on- and offline world [Youl$^+$00], in order to fulfil their design goal of utility maximization for their human owner. They will be able to enter into negotiation with many potential trade partners at once, reaching an acceptable deal and setting up a contract in a matter of milliseconds [Prei98].

If transactions are conducted in the background without the need for human intervention, the resulting concept can be called **Silent Commerce** [Adam$^+$03; Sche00]. Example applications for Silent Commerce are the networked laser printer which automatically buys toner when needed [Cros00], the mobile fare payment when entering the train using PDA or mobile phone, the payment of web services by networked clients in the Grid [Arda$^+$02] or built-to-order adaptive supply chain control concepts using software agents [Livi01]. The large number of devices and software objects, the resulting exponential number of possible interactions and the possible global reach of communications makes controlling such systems by means of a centralized coordinator instance unwieldy, if not impossible. If devices and software can be personalized by the respective human user, the DBAs will finally act self-interested to maximize the utility of their human principal [RaJa99] and thus protect utility function and decision process from outside access.

Personalization requires the human principals of the software agents to define economic goals, preferences and strategies in computer processable data structures [Krau97]. In automated negotiation by software agents, the strategies of the *internal model* calculate offer prices to propose to a trade negotiations opponent, in order to achieve a maximum utility gain. The strategy decision is based on information from *sensors*, mostly price offers from other agents or institutions in the market. The price offer sent to the opponent is an *effector*, which has the intention to draw the opponent towards a favourable negotiation goal.

The strategy model itself can be based on rule-based, argumentative, game-theoretic or heuristic-adaptive approaches [Jenn$^+$01; Krau97]. The choice of strategy type depends largely on the characteristics of the problem domain. An agent-based silent commerce scenario with direct interaction constitutes a non-accessible, partly deterministic, discrete, highly dynamic and non-episodic environment [RuNo95]:

- *Not accessible:* Accessibility denotes the ability of the agent to assess the complete state of the environment by using sensory input. In the scenarios considered here, the agent frequently receives both unsolicited information and concrete responses to offers. However, it is not possible to get insights into the internal decision processes of other agents. In total, the agent’s world model is made up of historic, sporadic and infrequent information.
- *Partly deterministic:* The use of a common negotiation protocol leads to deterministic states of the negotiation under predictable conditions. However, the behaviour of the negotiation opponents in response to a particular offer is not predictable and can comprise a large, potentially infinite number of possible actions.
- *Discrete:* The possible actions of a negotiating agent are limited to choosing a price from the set of the natural numbers or terminating the negotiation by either acceptance or rejection.
- *Dynamic:* The internal model and thus the strategy of the negotiators may change both during and between negotiations using adaptive mechanisms. It is not predictable whether the response of an opponent will be equal to earlier responses when facing the same negotiation situation.
- *Non-Episodic:* Successive negotiations are linked by budget restraints and feedback propagation of success or failure of the current action decision set, even if the negotiations can be considered independent otherwise. In particular,

the outcome of a single negotiation depends only on the choice of strategy parameter set at the beginning of the negotiation process.

In such a complex, unpredictable environment with possibly thousands of acting and negotiating agents, rule-based or game-theoretic strategies alone are considered to be not realistically applicable [Krau97]. However, it is possible to devise heuristics with some economic background e.g. on reputation and cheating of opponents [Pado01], or common market and negotiation behaviour [Sack02]. However, these heuristics are geared toward the present situation of the environment. Their ability to maximize utility will decrease as the environment changes over time. This leads to the necessity to enhance heuristics with adaptation capabilities:

"In future applications in e-commerce, multi agent systems will need to be much more open-ended and dynamic [...]. In particular it is important for the negotiating agents to be able to adapt their strategies to deal with changing opponents, changing topics and concerns, and changing user profiles." [Gerd$^+$00]

But whether *heuristic-adaptive* strategies produce meaningful results depends on the specifics of both heuristics and adaptation mechanism. This article tries to show how different existing machine learning algorithms perform in a multiagent system of DBAs, using the same heuristics as a basis for adaptation.

The remainder of the article is structured as follows. Chapter 2 briefly illustrates an agent-based electronic marketplace, where agents with heuristic-adaptive strategies autonomously negotiate about goods. The following chapters shows the application of different adaptation mechanisms. Chapter 3 deals with the implementation of a genetic algorithm. In chapter 4 several numeric optimizing methods are applied. Chapter 5 presents a hybrid approach following the VID model according to Brenner [Bren96]. The paper ends with a summary of the findings.

2 Automated negotiation in AVALANCHE

The AVALANCHE multiagent system has been developed at the University of Freiburg for several years as a software project where Digital Business Agents (DBAs) act on electronic marketplaces [Eyma00]. AVALANCHE, as described here, is realized in JAVA 1.3 with the support of the agent middleware LARS 2.5 CE [Livi01]. All agents are independent JAVA threads. The system architecture consists of three basic classes: marketplaces, agents, and one experiment control object. For the experiments, all agents are initialized simultaneously from the same JAVA class, using the same initial heuristic parameter values.

The agent class defines communication abilities and negotiation protocols. Each agent communicates with every other object on the marketplace in direct, bilateral and unmediated fashion, using a unique identity. The marketplace merely serves as a passive white board, where the agent can sign in and out with name and the type of goods it demands or offers. In particular, the marketplace does not explicitly synchronize or schedule the agents' activities.

In the scenario described here, three different types of DBAs are implemented. *Producers* and *consumers* define a seller or buyer strategy, respectively. The *middlemen* in-between buy a material good from the producer and sell a product good to the consumer. The *middlemen's* production function just simulates the conversion of one piece of material to one product. The goods itself are defined as commodities, so the one-dimensional negotiation variable is the price.

This also allows to represent the utility function N of the AVALANCHE agents using price notions only. The economic goal in negotiation (neglecting e.g. production costs) is to maximize profit by increasing the price spread between selling output goods $\pi_{V,i}$ and buying input goods $\pi_{K,i}$ in a given time span $\left[t-1;t\right]$:

$$N_t = \sum_{t-1}^{t} \pi_{V,i} - \sum_{t-1}^{t} \pi_{K,i} \rightarrow MAX \tag{1}$$

The performance of the utility function is thus only dependent of the outcome of the negotiations on both sides (or on one side for the producers and consumers). The better the negotiation strategy succeeds in comparison with the opponents and the competitors, the better the agent performs overall. However, if the negotiation strategies are very similar and unchanging, no agent will be able to achieve competitive advantage, and the prices and profits will reach a common plateau for all agents (cf. [Eyma01a; Eyma03].

2.1 The negotiation protocol

All agents follow the same bilateral negotiation protocol as shown in Figure 1. The buyer agent initiates a negotiation by proposing a seller, whose address was obtained from reading the white board, sending a *propose* message containing the sender *A*'s identity, the receiver *B*'s identity and the particular offer price *x*.

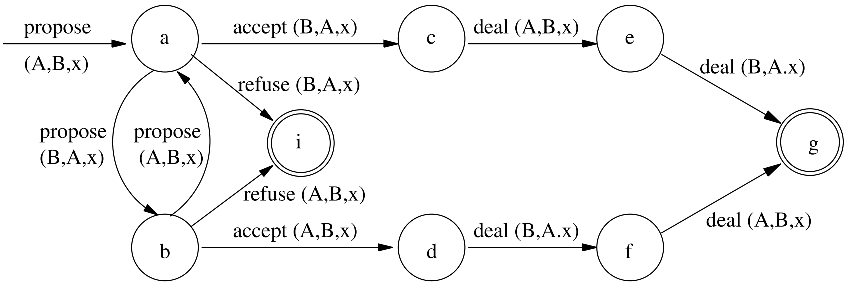

Figure 1: Course of a negotiation between two agents A and B

The receiver B has now the choice between downright accepting the price, making a counter-offer, or refusing to further negotiate at all. Whether the state transaction from state *a* to either states *b* (propose), *c* (accept), or *i* (refuse) is executed, depends on the action decision made in the agent's internal model. This decision is computed using the heuristic-adaptive strategy outlined below. The software agents negotiate with each other using a monotonic concession protocol [RoZl94], where propose and counter-propose messages with subsequent price concessions are exchanged. The negotiation continues until either a deal has been landed (state *g*) or one of the agents has unilaterally decided to refuse further negotiation (state *i*).

2.3 The heuristics of the negotiation strategy

The action decision-making of an AVALANCHE agent's negotiation strategy is controlled using 6 trivial parameters with values from a continuous value range between 0 and 1. These parameters are collectively called the *Genotype G*. The particular mechanics of the decision-making process are not required for the understanding of this article (see [Eyma01a; Eyma01b; Eyma03] for detailed descriptions).

It is sufficient to generalize by saying that every heuristic strategy, regardless of the concrete implementation, will choose with a certain probability (as seen by an outside viewer) either transition *a→b*, *a→c* or *a→i* from Figure 1. If a counter-offer is made, the concession amount is also controlled by the strategy.

Under these circumstances, the goal of the adaptation function is to maximize the total output by changing the strategy parameters (here: the *Genotype*). Formally, this goal can be described as

$$f_U : [0,1]^6 \to R \tag{2}$$

The independent vector $\in [0,1]^6$ describes the *Genotype* of the agent, which is acting in an environment *U*. The assumption is that a function *f* (of unknown shape) exists, which maps the *Genotype* to its outcome in the realm of natural numbers, given the current state of *U*. The goal of learning is thus to maximize the dependent outcome by optimizing the independent *Genotype*, which means adapting single vector elements (dimensions) until the global optimum of f_U is found.

This adaptation task, however, is bound by several restrictions of the environment.

1. No objective, central performance evaluation institution can exist, as this would require the communication of every agent's utility function so that a theoretical optimum can be computed, against which the individual performance is evaluated. For the same reasons explained in the first chapter of this article, particularly, size and reach and dynamics of the environment, such complete evaluation is not realistically possible. The agents will have to evaluate the effects of their actions on the environment using local information only.
2. The available data about the performance is historic. The agent merely knows the average profit from its previous negotiations and can make statements about the success of the current parameter configuration in relation to previous parameter configurations on the basis of its development.
3. Because the environment of the agent is highly dynamic, no statement about the success of a parameter configuration can be made on the basis of one individual negotiation. The agent must have carried out several negotiations with a parameter configuration before these can be evaluated by means of the profit generated.
4. If the other agents also implement adaptation mechanisms, the overall picture widens to a very complex co-evolution of agents, which is scientifically hard to evaluate. In this article, co-evolution has thus been ruled out. Only the middle agents implement machine learning algorithms, while the producers and consumers are static. However, all agents implement the same heuristics and start with the same initial *Genotype* set.

3 The genetic algorithm of Smith and Taylor

The first adaptation mechanism applied is mainly oriented at the evolutionary algorithm of Smith and Taylor [SmTa98] (STDEA) described in [Eyma01a; Eyma00]. A fundamental quality of the mechanism is the decentralized communication and fitness evaluation, using only locally available data.

Every agent sends one *plumage* object after a successful transaction, advertising its average income (fitness) and its genes (genotype) to all agents of the population after an evaluation phase, i.e. after it has carried out a certain number of negotiations with this genotype. If an agent receives a plumage object from another agent, it decides using a *blindness* probability whether the plumage object is evaluated, avoiding premature unification of the genotypes. Sender and recipient remain anonymous. If a certain *maturity threshold* of received plumages is exceeded, the agent replaces his old genotype with the evolved version after the completion of evaluation, selection, recombination and mutation phases as in normal genetic algorithms. Also influencing the algorithm is the *mutation rate*, which determines the frequency and the extent of explorative behaviour of the population.

The technical functionality of STDEA has been evaluated using the *De Jong Five-Function Test Bed* [Gold93]. The performance of agent populations consisting of either 1 or 25 agents searching for the minimum of the test functions shows that the algorithm is successful in all functions considered; even one agent alone takes the right direction. This evaluation has been conducted for all algorithms described in this article, in order to indicate a correct implementation of the learning function alone (without heuristics and economic decision-making).

In the AVALANCHE market scenario, STDEA proves to be a reliable optimizing procedure provided that information is able to flow between the agents. The parameter to be optimized is the profit (in Figure 2 on the y-axis) over time (x-time). Figure 2 shows the development of the agent profits on a marketplace with 15 *producer agents*, 15 *consumer* agents (both types with non-adapting strategies) and 15 *middleman* traders who use the genetic algorithm to optimize their negotiation result.

In the left half of Figure 2, all producers and consumers have the same unchanging genotype set; their profits do not change over time and are nearly zero (light grey dots). However, the middlemen’s profit (black dots) quickly and easily increases as the (initially same) genotype is optimized. In the right picture, on the other hand, all agents start with a random genotype. Over time the middlemen gradually outsmart their non-adapting opponents again and can increase profit at their expense. However, it should be noted that the exchange of information also changes the heuristics, as it leads to changed initial negotiation prices and thus influences the negotiations directly. The outcome of successive evaluations is thus not as independent as in the *De Jong testbed*.

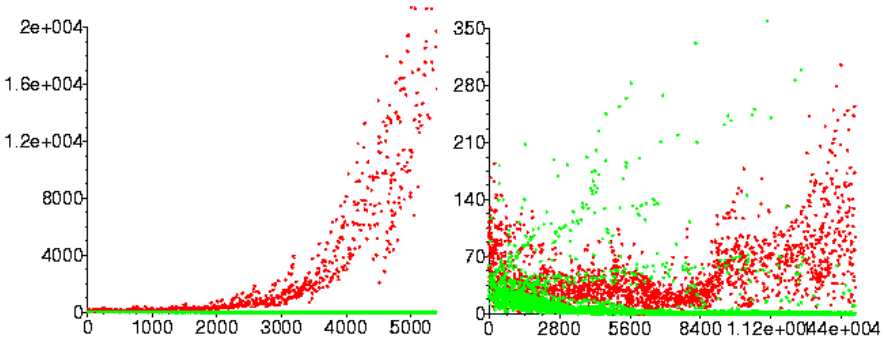

Figure 2: Performance of the genetic algorithm in AVALANCHE scenarios

In STDEA, the middlemen rely on information from other agents of the same type in the form of the plumage object. Translated to real scenarios, market participants gather performance information on others e.g. from newspapers, quarterly or annual reports, market rumours, or intelligence. Without such a “meta information flow” on the marketplace, economic agents would not be in a position to adapt their strategy accordingly and therefore incapable of action with regard to the choice of a strategy. In the next chapter, we thus implement and evaluate strategies which do not need outside information, but adapt using local feedback.

4 Numerical optimization procedures

“Unfortunately, there is no perfect optimization algorithm. This is a case where we strongly urge you to try more than one method in comparative fashion.” (Press et. al. [Pres+02])

The numeric optimizing algorithms applied here are taken in their original form from Press et al. [Pres+02] and have been slightly altered. The biggest differences are (1) that the function to be optimized can not be directly evaluated and (2) that the algorithms are defined to seek the minimum of $-f_U$ in the search space $[0,1]^6$, to search for the parameter configuration which makes the maximum profit.

The first two algorithms described here belong to the group of the “direction-set methods”, i.e. they always proceed in two steps. The first step determines the direction to climb. In the second step, a one-dimensional optimizing sub-algorithm like Brent’s algorithm or the naïve Golden Section Search [Pres+02] searches for the extremum.

4.1 Golden Section Search

The one-dimensional *Golden Section Search* is performed successively along a set of given directions that are passed through in the same sequence. When searching for the minimum of the *De Jong testbed*, this algorithm comes off worse in relation to the algorithms described in the remainder of this article, which has been expected. In the AVALANCHE scenario however, this simple algorithm equipped with the unity vectors does not cut a bad figure (see Figure 3 in comparison with Figure 2). As all agents process nearly the same market information, the variance of the development is less even without exchanging information.

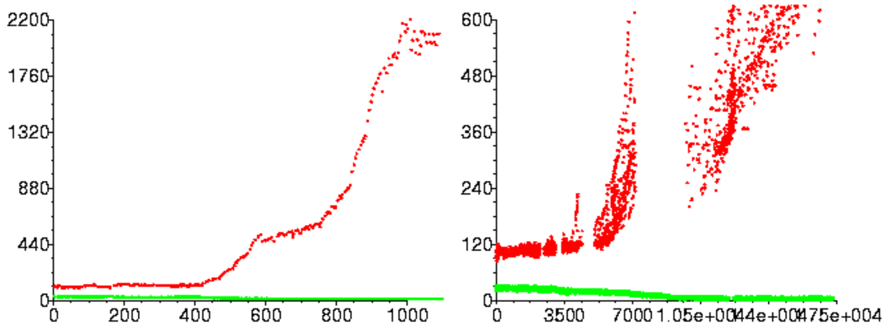

Figure 3: Performance of the naive procedure with equivalent genotypes of producers and consumers

This naive procedure is suitable to examine the significance of individual parameters for the success of the agent, where the unity vectors are regarded as direction vectors from the outset. Depending on the contour of the function, this procedure can, however, be very inefficient when searching for the extremum (see [Pres+02]). Observations of the development of the profit and the changes of the parameters attach special importance to the parameter which controls the value of the initial price offer in this particular AVALANCHE-based example. The applicability of this simple procedure to realistic heuristics thus requires cautious implementation and further research.

4.2 Multi-dimensional optimization according to Powell

The direction set method of Powell in [Pres+02] not only searches for the extremum but also for optimal directions for the one-dimensional sub-algorithm in order to accelerate the search procedure. Powell's procedure thereby derives a number of conjugated vectors from the extrema found in the respective directions. The algorithm implemented here tries to handle the problem of the linear dependence of these vectors through a heuristic process. After each optimization cycle, the optimal direction is replaced with the one which has generated the greatest progress.

As a proven and very efficient optimization procedure, Powell's procedure delivers a quick outcome in the *De Jong testbed*. In the AVALANCHE setting, this procedure also achieves better results than the genetic algorithm. Since Powell's procedure also searches for dimensional directions to take on the way to the extremum, this method offers the possibility of examining connections between the individual parameters and their significance for the choice of a strategy in small populations.

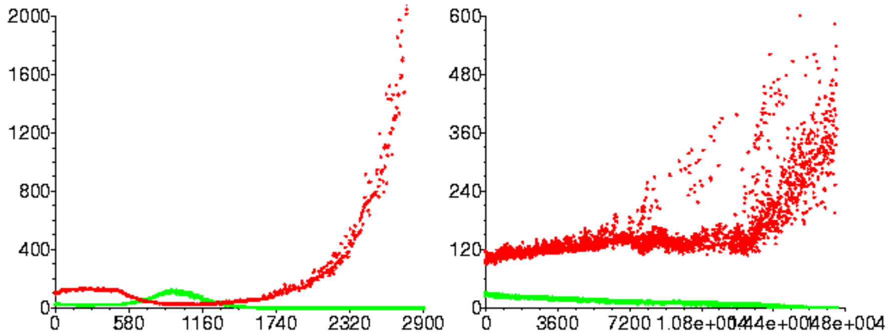

Figure 4: Performance of Powell's procedure with equivalent genotypes of producers and consumers

However, if one conveys this result of one individual agent to a population of agents, in which each agent has to carry out its own search process, the number of the required evaluations multiplies with the number of the agents until each agent has reached an acceptable result. The larger the population, the lower thus the performance, which is considered to make this mechanism impractical for real-world scenarios.

4.3 The Simplex method according to Nelder and Meat

Nelder and Meat's method [Pres+02] does not require an one-dimensional sub-algorithm. At the outset, linear independent points with their functional values, spanning an *n*-dimensional subspace, are located in an *n+1*-dimensional search space. This geometric figure is described as a simplex. The starting point of an optimizing step is the point of the simplex with the worst functional value. This point is transferred through heuristic reflections to the hyperplane, defined by the remaining *n* points, into a point with better functional value, whereby the property of the *n+1* points to form a simplex remains.

In the *De Jong testbed* the performance decreases with the growing number of agents, as in the other numeric optimizing procedures. In AVALANCHE, the results in the scenarios with unified genotypes of producers and consumers are, compared to the first two procedures, exceptionally good.

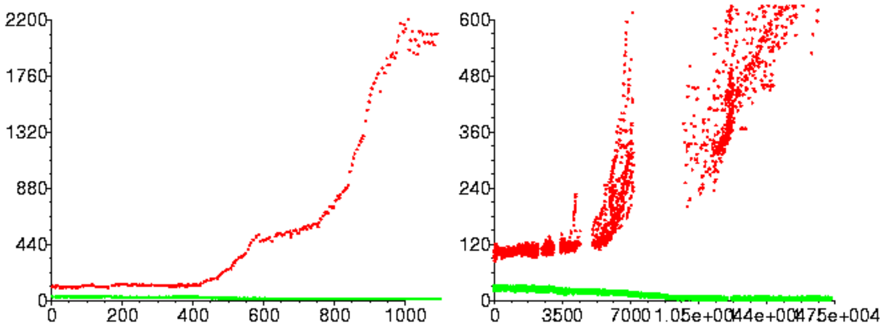

Figure 5: Performance of the Simplex method with one or 15 agents on the marketplace at a time

Summarizing for the numerical procedures, Powell's algorithm as well as the Simplex method provides better results than the GSS algorithm regarding the test functions as well as AVALANCHE. The differences to the (parallel processing) STDEA become obvious if one varies the size of the population. In the case of a single agent, the numeric algorithms take advantage of their directed search, in contrast to the random exploration of the STDEA, which performs better with an increasing number of agents.

5 A hybrid genetic algorithm on the basis of the VID model according to Brenner

In the preceding chapters, algorithms were based on two different principles. The genetic algorithm STDEA is based on random exploration combined with imitation, while the numeric algorithms are based on directed exploration. The following hybrid algorithm OVID combines both approaches. It tries to mirror human acquisition of information that is driven by imitation, random exploration, and exploration directed by cognitive processes. Technically, a parallelized version of the Simplex method is extended by the STDEA with respect to random exploration.

The VID (Variation-Imitation-Decision) model according to Brenner [Bren96; Bren02] is based on hypotheses from cognitive psychology and satisficing theory [Simo87] and combines them to a learning process for repetitive decision processes. "It assumes that decision-makers learn from their experience, are motivated to behavioural changes by unsatisfactory actions and are able to imitate successful strategies of others" [Bren02]. VID correlates (positive feedback) experience, contentment and the basic willingness to exploitative and explorative behaviour by individually (1) calculating the probability to change from one behaviour to another without any influence of other individuals, and (2) calculating the probability to imitate the strategy of another individual.

In this article, the assumptions of the VID model have been slightly altered. The limited information processing capacity of the AVALANCHE agents does not allow them to hold models of all agents' strategies. The adaptation of the *Genotype* draws from a continuous interval of possible actions, while the original VID model assumes a discrete number of actions. Thus the following questions remain: which information can be used to quantify the agent's satisfaction and experience? How can the motivation for adapting the strategy be derived from the satisfaction? How can imitations or variations be realized in a continuous action space?

Quantification of the experience. The *experience* value compares the performance of a particular agent to that of all other agents. This comparison (using publicly available data about all agent's strategies) provides the basis for imitation in the original VID model. The strategy with the highest experience value witnessed will be imitated by the comparing agent in order to increase his own performance.

Since the agents in AVALANCHE only possess information about their current parameter configuration, the procedure already known from the genetic algorithm is an alternative for direct information transfer: the agents send a *plumage* object to all agents of the population or of the market and process *plumage* objects from other agents only with a certain degree of probability. As the *experience* performance with a certain Genotype is correlated with an agent's profit, the *experience* value for an agent can be directly calculated by putting a received plumage's fitness $profit_{av}$ in relation to the perceived average fitness values of other agents $profit_{av,all}$.

$$\text{experience}_{av} = \frac{profit_{av} \text{ - } profit_{av,all}}{\left| profit_{av,all} \right|} \tag{3}$$

Quantification of the satisfaction. The *satisfaction* value is correlated with adaptation of the current own strategy. Here, the agent compares its past and current performance without looking to outside strategies.

Due to limited computational abilities, the agent can merely correlate the functional value or profit of the current parameter configuration to a previously generated average profit. This value is identified as *aspiration level*, as the satisfaction of the agent sensibly stands in proportion to the profit development. If the profits rise, the value of the satisfaction is positive, if the profits fall, it is negative. Analogous to the quantification of the experience, the satisfaction is related here to the percentage change of the profit. However, the significance of the success of an individual negotiation can not be determined; the value is thus derived by comparison to several negotiations carried out in succession.

$$satisfaction_{av} = \frac{profit_{av} - profit_{av,old}}{\left| profit_{av,old} \right|} \tag{4}$$

In the notation following [Bren96]:

$$s_i(t) = \frac{u_{av}\left(I_i(t),t\right) - u_{av}\left(I_i(t-\tau),t-\tau\right)}{u_{av}\left(I_i(t),t\right)} \tag{5}$$

where as

$$\begin{aligned} & u_{av}\left(I_i(t),t\right) = profit_{av}\left(I_i(t),t\right) \\ & = \sum_{j=0}^{n} \zeta_s^j \times (1-\zeta_s) \times profit\left(I_i(t),t\right) \\ & = \zeta_s \times u_{av}\left(I_i(t-1),t-1\right) + (1-\zeta_s) \times u\left(I_i(t),t\right) \end{aligned} \tag{6}$$

Motivation for behavioural change. If the satisfaction falls below zero, the current strategy shows inferior performance and the motivation to change the strategy rises. This can be described using a Sigmoid curve $\frac{1}{1+e^{ax+b}}$ with infimum 0 and supremum 1, where $m(0,t)$ is the basic motivation if satisfaction = 0. The motivation for the behaviour change should at least be the basic motivation in the case of negative profit.

$$m\left(s(t),t\right) = \begin{cases} \dfrac{1}{1+e^{a(t)\times s(t)+\beta}} & u_{av}\left(I_i(t),t\right) \geq 0 \\ \max\left(\dfrac{1}{1+e^{a(t)\times s(t)+\beta}}, m(0,t)\right) & u_{av}\left(I_i(t),t\right) < 0 \end{cases} \tag{7}$$

where as

$$\beta = \log\left(\frac{1}{m(0,t)} - 1\right) \tag{8}$$

Let α be calculable through the value to the point

$$x = \begin{cases} -0.5 & u_{av}\left(I_i(t),t\right) \geq 0 \\ -1 & u_{av}\left(I_i(t),t\right) < 0 \end{cases} \tag{9}$$

given through the parameter $0 < \mu_+, \mu_- < 1$ as

$$m(x,t)=\begin{cases}\mu_{+}\times\left(1-m(0,t)\right)+m(0,t) & u_{av}\left(I_i(t),t\right)\geq 0\\ \mu_{-}\times\left(1-m(0,t)\right)+m(0,t) & u_{av}\left(I_i(t),t\right)<0\end{cases}\tag{10}$$

From this follows

$$\alpha(t)=\frac{\ln\left(\frac{1}{m(x,t)}-1\right)-\beta}{x}\tag{11}$$

The value calculated in formula (11) at point *x* indicates with what probability the agent adapts his behaviour when his profit is half as high or doubly loss-making as the previously generated profit. The parameter μ_{+} and μ_{-} determine this value in relation to the basic motivation.

Partial imitation. In the process of imitation in [Bren96], a strategy regarded as successful will be copied without changes. In genetic algorithms, this relates to a crossover of the current genotype with the most successful genotype, where only the alien genotype prevails. Subsequent mutation is optional.

Directed variation. In [Bren96] the behaviour variation is an undirected exploration, which selects a strategy at random. In order to give the explorative behaviour a direction in a meaningful way, analogous to human cognitive perception, a parallelized Simplex method will be used to process information from other agents. Each choice of a strategy, defined by a certain genotype parameter configuration, is mapped to a point in the *n*-dimensional search space of the Simplex process.

The results of this optimizing VID (OVID) mechanism show a better performance overall than both genetic algorithms and numerical mechanisms alone.

In comparison to the numerical procedures described in chapter 4, the performance in the *De Jong Testbed* with a population of 25 agents is better; but it does not reach the performance of the genetic algorithm STDEA. In one particular test function f_5, however, the OVID method does not lose the direction to the minimum in contrast to the genetic algorithm, which operates at random.

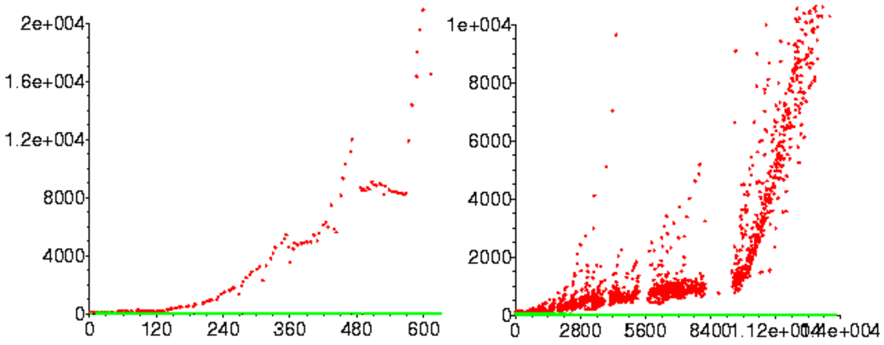

Figure 6: Performance of OVID each with one or 15 equally initialized agents on the marketplace

Figure 6 shows the price developments in the AVALANCHE setting with uniform genotypes of producers and consumers each with one and 15 agents. In the experiment in Figure 7, the information flow was disrupted and thus the learning process is slower.

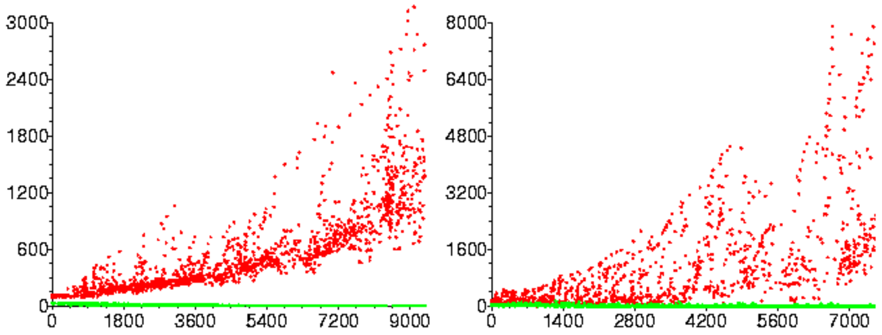

Figure 7: Performance of OVID without information flow between the agents (on the left) and with random initialization (on the right)

The OVID model presents a possibility of combining the advantages of the genetic algorithm STDEA, the numeric optimization procedure of the Simplex method and the imitating and directed exploring behaviour of human cognitive processes. This algorithm does also not depend on a constant information flow between the agents, but can meaningfully evaluate such at any time. It is thus more robust than STDEA and yields better results than pure numerical optimization approaches.

6 Conclusions and further work

This paper has shown a comparison of different optimization algorithms for profit maximization of negotiating software agents in e-commerce scenarios. These were implemented and tested in a particular multiagent system, using different numbers of agents and different parameterization of the negotiation partner. While the genetic algorithm STDEA and the numeric optimization methods already produce good results, it was possible to combine their strengths to develop the OVID procedure. This learning mechanism is considered efficient within large and also mixed populations. It can meaningfully use information from other agents; however, it does not depend on it.

The algorithms are still to be tested with regard to their behaviour (1) on marketplaces with varying population size and (2) where the negotiation partners are also equipped with a learning algorithm and therefore the optimizing procedures co-evolve. In such a scenario it is expected that the information about the success of parameter configurations lose significance. In the static scenarios described here, merely the development of the price concepts and varying starting points of the negotiations as well as stochastic irregularities caused differing results with one and the same parameter configuration.

What is also missing, of course, is the evaluation of the learning mechanisms in a real-world scenario. If the findings of this article can be transferred has yet to be shown. As an example, the business performance information published in real markets has shown to be sometimes inexact, incorrect or even completely made up. Even if this behaviour could be taken into account if reputation tracking mechanisms as in [Eyma+02] are applied, which modify the raw data prior to including it in the learning process, the question arises how this again would influence the performance of different learning mechanisms.

References

[Adam+03] Adams, T.; Ferguson, G.; Tobolski, J. F.: An Introduction to Silent Commerce: Creating New Sources of Value from Intelligent Objects. Accenture Technology Labs: Chicago, 2003.

[Arda+02] Ardaiz, O.; Freitag, F.; Navarro, L.; Eymann, T.; Reinicke, M.: CatNet - Catallactic Mechanisms for Service Control and Resource Allocation in Large Scale Application-Layer Networks. Proc. Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems, 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid. Berlin, 2002.

[Bren96] Brenner, T.: Learning in a Repeated Decision Process: A Variation-Imitation-Decision Model. Report No. #9603. Max-Planck-Institut für die Erforschung von Wirtschaftssystemen: Jena, 1996.

[Bren02] Brenner, T.: A Behavioural Learning Approach to the Dynamics of Prices. Computational Economics, Vol. 19, 2002: p. 67-94.

[Cros00] Cross, K.: Intershop around the corner. Business 2.0, May 2000: http://www.business2.com/articles/mag/0,1640,13603%7C2,FF.html, last accessed 2003-02-01.

[Dieb01] Diebold Consulting (Ed.): Digital Business Agents - Benefits and Potential of Multi-Agent Systems. Diebold Deutschland GmbH: Eschborn, 01. http://www.timelabs.de/, last accessed 2002-12-15.

[Eyma00] Eymann, T.: Avalanche - ein agentenbasierter dezentraler Koordinationsmechanismus für elektronische Märkte. Ph.D. Thesis, Albert-Ludwigs-Universität Freiburg 2000. http://www.freidok.uni-freiburg.de/volltexte/147/, last accessed 2003-02-12.

[Eyma01a] Eymann, T.: Co-Evolution of Bargaining Strategies in a Decentralized Multi-Agent System. AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems. AAAI Technical Report FS-01-03. AAAI: Falmouth, MA, 2001, p. 126-134.

[Eyma01b] Eymann, T.: Decentralized Economic Coordination in Multi-Agent Systems. In: Buhl, H.-U.; Huther, F.; Reitwiesner, A.(eds.) Information Age Economy. Proceedings WI-2001. Physica Verlag: Heidelberg, 2001, p. 575-588.

[Eyma03] Eymann, T.: Digitale Geschäftsagenten. Springer Xpert.press: Heidelberg, 2003.

[Eyma+02] Eymann, T.; Padovan, B.; Pippow, I.; Sackmann, S.: A Prototype for an Agent-based Secure Electronic Marketplace Including Reputation Tracking Mechanisms. International Journal of Electronic Commerce, Vol. 6, No. 4, 2002: p. 93-114.

[Gerd+00] Gerding, E. H.; Bragt, D. D. B.; La Poutré, J. A.: Scientific approaches and techniques for negotiation - a game theoretic and artificial intelligence perspective. Report No. SEN-R0005. Centrum voor Wiskunde en Informatica: Amsterdam, 2000. http://citeseer.nj.nec.com/333726.html, last accessed 2003-02-01.

[Gold93] Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley: Reading MA, 1993.

[Jenn+01] Jennings, N. R.; Faratin, P.; Lomuscio, A. R.; Sierra, C.; Wooldridge, M. J.: Automated negotiation: prospects, methods and challenges. International Journal of Group Decision and Negotiation, Vol. 10, No. 2, 2001.

[Krau97] Kraus, S.: Negotiation and cooperation in multi-agent environments. Artificial Intelligence, Vol. 94, 1997: p. 79-97.

[Livi01] Living Systems AG: Living Markets Technische Dokumentation. Version 1.03. Living Systems AG: Donaueschingen, 2001. http://www.living-systems.com/, last accessed 2003-02-01.

[Pado01] Padovan, B.: Ein Vertrauens- und Reputationsmodell für Multi-Agenten Systeme. Ph.D. Thesis, Albert-Ludwigs-Universität Freiburg Freiburg, 2001.

[Prei98] Preist, C.: Economic Agents for Automated Trading. HP Technical Reports HPL-98-77. Hewlett Packard Laboratories: Bristol, 1998.

[Pres$^+$02] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: Numerical Recipes in C++ - The Art of Scientific Computing. Cambridge University Press: Cambridge, MA, 2002.

[RaJa99] Rasmusson, L.; Janson, S.: Agents, self-interest and electronic markets. Knowledge Engineering Review, Vol. 14, No. 2, 99: p. 143-150.

[RoZl94] Rosenschein, J. S.; Zlotkin, G.: Rules of encounter - designing conventions for automated negotiation among computers. MIT Press: Cambridge, 1994.

[RuNo95] Russel, S. J.; Norvig, P.: Artificial Intelligence. A Modern Approach. Englewood Cliffs: Prentice-Hall, 1995.

[Sack02] Sackmann, S.: Bilaterale Preisverhandlungen von Software-Agenten - Ein Modell und System zur Analyse des marktplatzspezifischen Verhandlungsspielraums. Ph.D. Thesis, Albert-Ludwigs-Universität Freiburg, 2002.

[Sche00] Schenker, J. L.: The Trillion-Dollar Secret. TIME Magazine, Vol. 155, No. 8, 2000. http://www.time.com/time/europe/magazine/2000/228/trading.html, last accessed 2001-03-13.

[Simo87] Simon, H. A.: Satisficing. The New Palgrave Dictionary of Economics. Macmillan: London, 1987, p. 243-245.

[SmTa98] Smith, R. E.; Taylor, N.: A Framework for Evolutionary Computation in Agent-Based Systems. In: Looney, C.; Castaing, J.(eds.) Proceedings of the 1998 International Conference on Intelligent Systems. ISCA Press: 1998, http://www.ics.uwe.ac.uk/~rsmith/fecabs.pdf, last accessed 2002-12-15.

[Youl$^+$00] Youll, J.; Morris, J.; Krikorian, R.; Maes, P.: Impulse: Location-based Agent Assistance. Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000). Barcelona, 2000, http://citeseer.nj.nec.com/youll00impulse.html, last accessed 2002-12-03.