

RESEARCH
PAPER

No 14-02

DECISION SUPPORT FOR
NEGOTIATION PROTOCOL
SELECTION: A MACHINE
LEARNING APPROACH BASED
ON ARTIFICIAL NEURAL
NETWORKS

Fabian Lang

Helmut-Schmidt-Universität Hamburg
fabian.lang@hsu-hamburg.de

Andreas Fink

Helmut-Schmidt-Universität Hamburg
andreas.fink@hsu-hamburg.de

JANUARY 2014

Decision Support for Negotiation Protocol Selection: A Machine Learning Approach Based on Artificial Neural Networks

Fabian Lang and Andreas Fink

Helmut-Schmidt-Universität Hamburg

Abstract

Decision making in operational planning is increasingly affected by conflicting interests of different stakeholders such as subcontractors, customers, or strategic partners. Addressing this, automated negotiation is a well-suited mechanism to mediate between stakeholders and search for jointly beneficial agreements. However, the outcome of a negotiation is strongly dependent on the applied negotiation protocol defining the rules of encounter. Although protocol design is well discussed in literature, the question on which protocol should be selected for a given scenario is little regarded so far. Since negotiation problems and protocols are very diverse, the protocol choice itself is a challenging task. In this study, we propose a decision support system for negotiation protocol selection (DSS-NPS) that is based on a machine learning approach – an artificial neural network (ANN). Besides presenting and discussing the system, we, furthermore, evaluate the design artifact in elaborate computational experiments that take place in an intercompany machine scheduling environment. Our findings indicate that the proposed decision support system is able to improve the outcome of negotiations by finding adequate protocols dynamically on the basis of the underlying negotiation problem characteristics.

Keywords: Automated negotiation, DSS, Machine learning, Predictive analytics.

1 Introduction

Negotiation is an important coordination mechanism for problems that involve several stakeholders. The interests of these stakeholders are commonly conflicting, e.g., in resource allocation scenarios or price negotiations (distributive negotiation). Nevertheless, there exist also win-win scenarios – especially in the case of complex contracts. Complex contracts are characterized by multiple contract items that are interdependent. This leads to an enormous contract space with a multitude of local optima which makes the exploration of this space hard (Fink 2006; Klein et al. 2003). As conceding strategies become infeasible because of the complexity, the negotiation parties seek for Pareto gains jointly, e.g., in inter-company production scheduling (integrative negotiation) (Benyoucef et al. 2001; Kersten 2001; Raiffa 1982; Vetschera 2013). Since there are reoccurring operational coordination problems of both types, the automation of negotiations has been a much-noticed research field (Chalamish and Kraus 2012; Jennings et al. 2001; Kraus 2001). Computerized negotiations are a highly interdisciplinary domain. On the one hand, economists, lawyers, and social scientists are concerned with negotiation procedures, i.e., strategies, tactics, and techniques. On the other hand, computer scientists and information systems researchers deal with negotiation media systems; at this, the central objects of research from an information systems perspective are decision support systems as well as group and negotiation support systems (Bichler et al. 2003).

In automated negotiations, decisions are delegated to software agents that decide on the behalf of their principals. The absence of human negotiators makes the negotiation protocol, which defines the rules of encounter (see Rosenschein and Zlotkin 1994), a very important element of the negotiation system. Whereas protocol design has been gaining a lot of attention, the choice of protocols is little regarded so far (Marsa-Maestre et al. 2013). For the decision which protocol should be used, the decision makers are interested in the performance an existing protocol would yield. For this purpose, predictive analytics methods are appropriate as they are focused on predicting the outcomes for new out-of-sample instances such as a new negotiation scenario (Shmueli and Koppius 2011). In contrast to that, explanatory models serve the verification of causal hypotheses which can sup-

port the design of new protocols. Prediction is an important scientific challenge; nevertheless, although many information systems papers state their goals serve a prediction purpose, most works draw on explanatory methods instead of predictive analytics ones such as support vector machines or artificial neural networks (Shmueli and Koppius 2011).

The purpose of this paper is to develop a decision support system for negotiation protocol selection (DSS-NPS). Consequently, the study follows a design science approach which is a central research area of information systems research (see Hevner et al. 2004; March and Smith 1995). We contribute by showing the relevance of the protocol selection problem. As a further contribution, we propose a DSS-NPS based on machine learning. Specifically, we use an artificial neural network for making predictions on the performance of a negotiation protocol for a given scenario. Finally, the system’s efficacy is evaluated by computational experiments. Concluding, the study tries to answer the research question if an artificial neural network can improve the welfare generated by negotiations by supporting the protocol choice.

The remainder of this study is structured as follows: After this introduction (section 1), we give an introduction into automated negotiation research as well as protocol design and selection (section 2), before, in section 3, an overview on related work is presented. In section 4, we describe our proposed decision support concept and its technical design. Afterwards, the setup for the computational experiments is introduced and the results of those experiments are presented (section 5). Finally, a discussion on the study’s findings and assumptions (section 6) as well as a summary and outlook (section 7) conclude the paper.

2 Theoretical Background

In this section, we give an introduction into the general field of research. Firstly, the issue of automated negotiation and its relevance is presented. Secondly, we discuss the impact of negotiation protocol design and the corresponding protocol choice problem.

2.1 (Automated) Negotiation

A negotiation is a dialogue between two or more parties (set of agents $\mathcal{J} = \{1, \dots, j, \dots, J\}$ with $J \geq 2$) in which they intend to find an agreement on a single or several negotiation issues ($\mathcal{I} = \{1, \dots, i, \dots, I\}$). The negotiation issues are governed by a contract $c \in \mathcal{C}$ (Lang and Klein 2013). During a negotiation, the negotiators either try to find a consensus by making conceding offers (distributive) or they look for joint gains that are (weakly) Pareto dominant for all of them (integrative) (Benyoucef et al. 2001; Kersten 2001; Raiffa 1982; Vetschera 2013). The latter shows the similarities between optimisation and negotiation problems: in an optimisation problem, the objective is to find a solution $x \in \mathcal{X}$ that optimises a given function $f : \mathcal{X} \rightarrow \mathbb{R}$, whilst, in a negotiation problem, the stakeholders search for a contract c that somehow optimises their individual objectives $O_j : \mathcal{C} \rightarrow \mathbb{R}$ (Jennings et al. 2001; Lang and Klein 2013).

The increasing connectedness between organizational entities leads to an increased need for making arrangements between (semi-)autonomous parties on an operational level. For instance, in many modern production processes, a company is not able anymore to plan the production solely on its own anymore: it also has to consider, inter alia, the availability, demand, response time, or quality requirements of the subcontractors or customers, respectively (Dudek and Stadtler 2005; Stadtler 2007; Thomas and Griffin 1996). Thus, those “second parties” should be – or even have to be – integrated in the planning process. Nevertheless, every stakeholder is concerned with his or her private goals and objectives. Therefore, he or she might not want to reveal private information due to privacy reasons or because it could be used against him or her (Fujita et al. 2010a; Hurwicz 1973; Lang and Fink 2013; Sandholm 1999). Even worse, a negotiator could lie strategically by making unnecessary false announcements (cheap talk) (Farrell and Rabin 1996) or misstating his or her preferences (Gibbard 1973; Myerson 1979, 1983; Satterthwaite 1975). Hence, central planning is neither desirable nor feasible because, on the one hand, information is withheld and, on the other hand, revealed information might be false and, hence, poisoned. For multi-party problems, one not only has to take multiple objectives into account (given by the private interests), but also one has to cope with strategic considerations of the agents (Binmore and

Vulkan 1999; Faratin 2000). Regarding this, a negotiation can mediate between individual interests as a coordination mechanism which ordinarily does not require centralized information.

Typically, operational planning problems are frequently reoccurring and can suffer from a high complexity; that is why automated negotiation has been gaining a lot of interest (Lang and Fink 2012; Yang et al. 2009). In automated negotiation applications, the actions within the negotiation process are delegated to software agents that decide on the behalf of their (human) principals (Bichler et al. 2010; Braun et al. 2006). Key advantages are that software agents are capable to negotiate, e.g., millions of rounds (Vulkan 1999), or that they can be duplicated such that they can handle situations in parallel. Recent information systems research addressed the issues of modelling and adequateness of preference representation in negotiation agents (Lang et al. 2011, 2012); however, since operational planning tasks are usually associated with direct, measurable costs, this issue rarely arises.

Examples for joint operational planning by negotiations are port terminal scheduling by several ship owners (Douma et al. 2007), supply chain coordination between different stages of the value chain (Fink 2006), lot-sizing in supply chains (Homberger 2010, 2011), multi-project scheduling with shared resources (Homberger 2012), or distributed vehicle routing between different enterprises (Sandholm and Lesser 1997). Consequently, negotiation is an important method for the coordination of interests in intercompany applications – especially in the research field of supply chain management and logistics (Fink 2006; Rief and van Dinther 2010).

2.2 Protocol Design and Selection

Every negotiation needs rules that govern the types, states, events, and (inter)actions of the negotiation (Jennings et al. 2001). Those “rules of encounter” (Rosenschein and Zlotkin 1994) are called negotiation protocol. The major objective of a protocol is to lead to a socially beneficial outcome that is favourable for all participating parties while fulfilling a variety of further properties such as, among others, leading to a guaranteed success, incentive compatibility, behavioural stability, or simplicity of optimal strategies (see Jennings et al. 2001; Lang and Fink

2014; Sandholm 1999). By applying a sophisticated protocol like, e.g., a second-best price auction (Vickrey auction (Vickrey 1961)), agents' malicious strategic actions can be prevented and they are incentivized to reveal information truthfully which can be used to find good solutions. Nevertheless, it also has to be considered that agents might not want to reveal certain information and should not be forced to do so (Lang and Fink 2014; Rothkopf et al. 1990). Concluding, protocols have to meet a lot of requirements simultaneously which is not a trivial task.

The result of a negotiation is centrally determined by the applied protocol. Therefore, protocol design is one of two key problems of automated negotiation research¹ (Jennings et al. 2001). The design task is generally addressed by mechanism design theory (also known as economic engineering). In mechanism design, the challenge is to find a mechanism to a certain scenario that leads to a desired social goal (Maskin 2008; Myerson 1988). For this, the scenario has to be analysed with regard to potential strategies and equilibriums. Commonly, this is done by means of game theory. Nevertheless, a lot of scenarios are too complex to be represented analytically in proper form and, consequently, researches have to draw on heuristic approaches and simulations for developing successful protocols (Jennings et al. 2001; Kraus 1997; Pasquier et al. 2010).

There has been a lot of effort in developing successful protocols for specific scenarios. This effort results in a variety of potential protocols (Marsa-Maestre et al. 2013). However, the research on which protocol should be selected for a given scenario is very little regarded until now. Negotiation problems can differ widely from each other (Büttner 2006; Ströbel and Weinhardt 2003): there are bilateral vs. multilateral settings, single vs. multiple issues, time limited vs. unrestricted processes, or monetary vs. non-monetary criteria – to name just a few dimensions. Furthermore, negotiation protocol design approaches are also very heterogeneous: for instance, there are auction-based, argumentation-based, integrative, or distributive approaches (Bichler et al. 2003; Ströbel 2000; Vetschera 2013). Consequently, on the one hand, it is unlikely that one protocol can solve all kinds of problems, and, on the other hand, the possibility space of negotiation problems is far too large to create individually designed protocols exhaustively.

¹The other one is software agent strategy design.

Concluding, the choice of protocol as well as its configuration and parameterization is a relevant problem that can influence the outcome of a negotiation substantially. Since information is decentralized and agents have to come to an agreement, we postulate that there is a need for decision support systems that facilitate finding a suitable protocol.

3 Related Work

In the following, we present a survey on related work that also deals with selection and configuration of negotiation mechanisms. Sandholm (2003) as well as Conitzer and Sandholm (2004) present an automated mechanism design approach. At this, the design of a mechanism becomes a computational problem for the designer. The designer only takes his or her very own preferences into account during decision making regardless of the preferences of the agents. Nevertheless, in this scenario, the designer needs detailed information about the outcomes of the interactions (in game theory: payoffs of the game).

Marsa-Maestre et al. (2011) developed a set of tools, the so-called Negowiki, that is supposed to support researchers in comparing and evaluating their negotiation protocols. First of all, Negowiki provides test instances for specific problems and protocol developers are encouraged to upload their negotiation results for these. After the upload, Negowiki automatically computes different metrics concerning Pareto efficiency, social welfare, or fairness. Like the name suggests, the data is openly accessible. Nevertheless, since the project requires full knowledge about the preferences of the agents, it is rather an environment for researchers than practitioners.

Based on the Negowiki project, Marsa-Maestre et al. (2013) designed a negotiation handbook which is an infrastructure consisting of different components. In the proposed process, negotiation scenarios are uploaded according to a defined mark-up language or generated with a provided scenario generator, before an integrated solver computes the metrics of the scenario. This scenario and its metrics is stored in the Negowiki project where either researchers can upload their experiment results or a simulation testbed (GENIUS, see Lin et al. (2012)) can be used. Again, the platform is primarily designed for research purposes.

Another approach is the work of Lang and Klein (2013) which proposes an architecture that supports agents in high-level negotiations (metanegotiations) on the protocols for an actual low-level negotiations. The metanegotiation support system provides different tools for decision and negotiation support. However, up to today, the architecture is still just a proposal and has not been realized as a prototype so far.

The survey on related work shows that the problem is not much regarded yet. As the related work is either based on just theoretical concepts or first and foremost targets researchers and not practitioners, the existing solution approaches do not provide satisfying decision support for protocol selection.

4 A Decision Support System for Protocol Selection

After having demonstrated the need for facilitating protocol choice, we present the design of our decision support system for negotiation protocol selection (DSS-NPS) that is going to be evaluated in computational experiments later on (see section 5).

4.1 Concept

The core idea of the proposed DSS-NPS is to take advantage of the frequent reoccurrence of operational planning negotiations. For instance, computerized applications, such as negotiations on the scheduling of cloud computing resources, can take place thousands of times each day (An et al. 2010). As a result of this, a DSS-NPS can gather a huge amount of data which can be used for aiding the decision making. In this study, we propose a supervised learning approach by means of an artificial neural network (ANN). The ANN makes use of historic data and provides predictive analytics for the success of potential protocol candidates. The general setup is shown in Figure 1.

First of all, a negotiation takes place using an arbitrary protocol. After the negotiation, relevant information is stored in a repository. Such relevant information is certainly the evaluation of the protocol by the agents but also structural

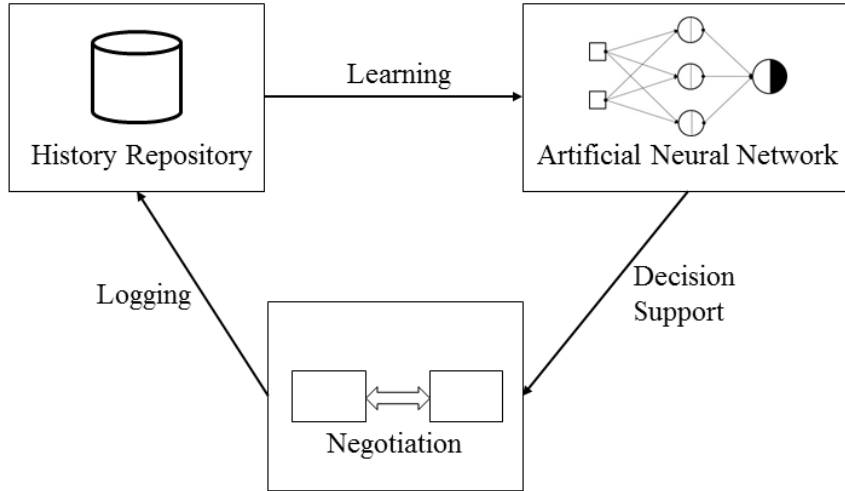


Figure 1: Setup of the decision support system for negotiation protocol selection (DSS-NPS)

information about the negotiation such as number of agents, agent types, or other problem parameters. However, the evaluation of a negotiation outcome is an intricate problem because agents might not want to share specific values such as realized costs with a third party. Although a lot of works regard third parties like a mediator as trustworthy (see, e.g., Fujita et al. 2010b; Myerson 1983), it is definitely not an axiomatic self-evidence and there are strong doubts about it (see Bosse et al. 2004; Hurwicz 2008; Klemperer 2002). Nevertheless, it may be reasonable to use proxies such as an indicator for a level of satisfaction instead of actual preference figures as the negotiation parties may be willing to reveal such information.

In the second step, there is an ANN that is supposed to learn patterns from the historic data from the repository. Since an ANN may need large amounts of training data to perform satisfactorily (see Yegnanarayana 2004, Appendix D) there is an initialization phase that consists of logging only. After sufficient data sets are gathered, the ANN learns this training data by adjusting weights between its neurons (we present more technical details in section 4.2 and a discussion on ANN in section 6).

After learning, the ANN should be capable to make a prediction for an ex

ante data set, i.e., for a negotiation that is not started yet. The prediction is based on the learned patterns and depends on the adjusted weighted connections within the neural net. The protocol selection can be supported by this prediction, as it represents a belief about the expected performance of a protocol. In other words, the protocol with the best predicted performance should be used. After the negotiation, the data is fed into the repository again and the ANN can be updated with regard to this new information.

4.2 Artificial Neural Network

In the following, we introduce the design of an ANN for the proposed DSS-NPS which we use and evaluate in computational experiments (see section 5). Later on, in section 6, we will discuss the advantages and disadvantages of the method.

4.2.1 Architecture

We use a multi-layered feedforward neural network² with two hidden layers. Feedforward networks, which are characterized by no circles in the directed graph, are a classical architecture for continuous input and output variables (Hudson and Postma 1995). The usage of two hidden layers can increase the performance of an ANN compared to a single layered setup (Kenyon and Paugam-Moisy 1997; Kůrková and Sanguineti 2013); however, adding more than two hidden layers complicates the learning process and may have unfavourable effects (Bengio and LeCun 2007)³. Figure 2 depicts the ANN’s topology and the related notation.

Let the set \mathcal{L}_l denote the l^{th} of the $L = 4$ layers. The first layer is the input layer which consists of K input units: $K - 1$ input variables plus one constant ($\mathcal{L}_1 = \{in_1, \dots, in_K\}$). The second and third layer constitute the hidden layers, also referred to as the black box component of the network. Those layers comprises M and N , respectively, artificial neurons ($\mathcal{L}_2 = \{n_1^2, \dots, n_M^2\}; \mathcal{L}_3 = \{n_1^3, \dots, n_N^3\}$). Finally, the last layer is the output layer with a single artificial neuron in our case ($\mathcal{L}_4 = \{n^4\}$).

²Not to be confused with a multi-layered perceptron.

³However, there have been recent improvements in the field of deep learning by utilising parallel computing on graphic processing units (GPUs) and computer clusters (see Le et al. 2011).

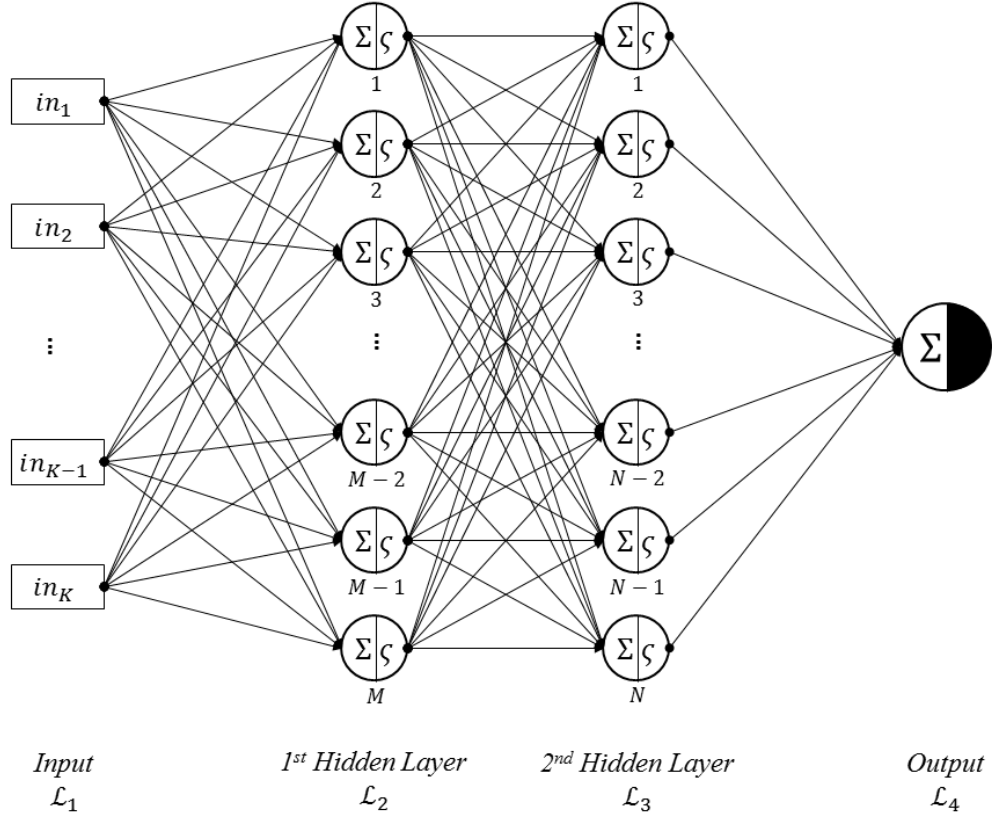


Figure 2: Topology of the neural network

Each node in the network is connected to each node of the subsequent layer, i.e., the network has $(K * M) + (M * N) + (N * 1)$ edges. Each connection is assigned with an adaptive weight w_{n_1, n_2} (where $n_1 \in \mathcal{L}_l$ and $n_2 \in \mathcal{L}_{l+1}$).

The neurons are modeled after the McCulloch-Pitt model (McCulloch and Pitts 1943); the output o_η of a neuron η is determined by a transfer function ζ : $\zeta(\sum_{\Omega=1}^{|\mathcal{L}_{l-1}|} w_{\Omega, \eta} o_\Omega + \theta_\eta)$ where $n_\eta \in \mathcal{L}_l$, and θ_η is a constant factor for neuron n_η . We use the popular sigmoid function $\zeta(x) = 1/(1+e^{-x})$ as transfer function. Its advantages are that it squashes all values to the interval $(0, 1)$ and that it is continuous and differentiable: $\frac{\partial \zeta(x)}{\partial x} = \zeta(x) * [1 - \zeta(x)]$. The latter is important for learning which we will see next.

4.2.2 Learning Algorithm: Backpropagation

For learning, we use the established backpropagation algorithm (Rumelhart et al. 1986) which is a multi-layer generalization of the Delta learning rule. The name of the algorithm refers to the propagation of errors from the output back to the input. The precedent delta rule (Widrow and Hoff 1960) is based on the derivation of the error with respect to the weight. A weight is decreased depending on its impact on the error and a predetermined learning rate α : $\Delta weight = -\alpha * \frac{\partial error}{\partial weight}$.

Therefore, the sigmoid function and its derivative are useful, since it is well-shaped and inexpensive to calculate. Generally, backpropagation is a special case of the gradient descent method for minimizing the mean squared error (Rumelhart et al. 1986). However, as an implication of this, the finding of a global optimum cannot be guaranteed in the presence of local optima.

We implemented the backpropagation algorithm as described below (for a derivation of the algorithm, see Henseler (1995) or Buscema (2013)):

For every training data set δ_τ , do the following:

1. An output $y(= o_\eta; \eta \in \mathcal{L}_4)$ is computed by feedforwarding.
2. An error term E_η for the output node is computed by comparing the output with the intended result t from the training data:

$$E_\eta = y(1 - y)(y - t) \text{ with } \eta \in \mathcal{L}_4$$

3. An error term E_η , for every node η in the two hidden layers, is computed:

$$E_\eta = o_\eta(1 - o_\eta) \sum_{\Omega=1}^{|\mathcal{L}_{l+1}|} w_{\eta,\Omega} * E_\Omega \quad \forall \eta \in \mathcal{L}_2 \cup \mathcal{L}_3$$

4. In every iteration τ , each weight is updated based on the previous data set $\delta_{\tau-1}$ according to the following formula:

$$\Delta w_{\eta,\Omega}(\delta_\tau) = -\alpha * E_\Omega * o_\eta + \beta * \Delta w_{\eta,\Omega}(\delta_{\tau-1})$$

$$w_{\eta,\Omega} \leftarrow w_{\eta,\Omega} + \Delta w_{\eta,\Omega}(\delta_\tau)$$

$$\forall \Omega \in \mathcal{L}_l; \forall \eta \in \mathcal{L}_{l-1} \text{ with } 1 < l \leq 4$$

The parameter α (> 0) represents the learning rate. The learning rate indicates the degree of the backpropagation of errors (degree of descent). As mentioned above, the backpropagation algorithm can get stuck in local optima which can be

exacerbated, but not prevented, by adding a momentum parameter β (≥ 0). The momentum incorporates the movement of the descent in the iteration before the current one: $\Delta w_{\eta, \Omega}(\tau - 1)$. This can enable leaving of a local optimum and finding better solutions (Buscema 2013; Henseler 1995).

5 Computational Experiments

In this section, we briefly describe the setup of the simulation and, afterwards, show the results of the computational experiments.

5.1 Test Instances and Simulation

Firstly, we simulated millions of negotiations with different problems and protocols. Afterwards, we trained the neural network to prepare our DSS-NPS. Finally, we evaluated the DSS-NPS with regard to two dimensions: on the one hand, the ability to predict the performance of a protocol to a given problem, and, on the other hand, the performance in negotiations in which the DSS-NPS was deployed. In total, we generated more than 10 million negotiation simulations.

The negotiation problems involve single and multiple homogenous machine scheduling scenarios. In these scenarios, every agent owns a set of jobs and wants to minimise either their individual total weighted completion time (see Abdul-Razaq et al. 1990; Cheng et al. 2013) or the individual maximum lateness (see Agnetis et al. 2007; Cheng and Sin 1990). The reduced forms (with only one agent, i.e., a central planner) of those problems are \mathcal{NP} -hard (Lenstra et al. 1977). For the former, we need a job-specific weight ($\sim \mathcal{U}[1, 10] \in \mathbb{N}$, \mathcal{U} : uniform distribution). For the latter, we assumed a due date that is more or less restricted subject to a tardiness factor ($\sim \mathcal{U}[0.2, 1.0] \in \mathbb{R}$) and a due-date range factor ($\sim \mathcal{U}[0.2, 1.0] \in \mathbb{R}$) (cf. Akturk and Ozdemir 2000; Cicirello 2003; Mönch et al. 2005). Each of the objectives is used in half of the simulations. The test instances comprise different numbers of machines ($\sim \mathcal{U}[1, 10] \in \mathbb{N}$), agents ($\sim \mathcal{U}[2, 10] \in \mathbb{N}$) as well as jobs per agent ($\sim \mathcal{U}[20, 200] \in \mathbb{N}$). Besides the job-specific owner, weight, and due date, the jobs are also characterized by a respective processing time ($\sim \mathcal{U}[1, 10] \in \mathbb{N}$). The negotiation was allotted, with equal probability, 10.0, 1.0, or 0.1 seconds of

negotiation time.

For the simulation of the negotiations, we used the *Mediated Negotiation Protocol Based on Simulated Annealing* (MNP-SA) by Lang and Fink (2014). In the basic protocol, a mediator proposes a set of contract candidates that are alterations of the current contract draft, and the negotiation parties choose the contracts that are acceptable for them, i.e., they accept or reject the proposals. If a contract is accepted by every party, it can become the new contract draft, and new contract candidates are generated deflecting from it. Nevertheless, since a lot of applications are characterized by competition for resources, there may be less overlapping responses than socially desirable. Therefore, the protocol demands that a certain quota of the proposal set has to be accepted. This quota declines over time and imitates the cooling of the Metropolis criterion of the simulated annealing meta-heuristic (see Metropolis et al. 1953; Kirkpatrick 1983; Černý 1985). By doing so, socially desirable outcomes close to the Pareto frontier can be obtained even with selfish behavior and conflicting interests (Lang and Fink 2014).

Besides with and without quotas, the protocol has optional policy building blocks that can be applied additionally to the basic protocol: Instead of having a mediator agent, the agents can propose individually desirable alterations as candidates. In the case of quotas, there can be a third response state – besides accept and reject – that reveals that a contract is only accepted due to the quota restriction. If every agent responded this way, the candidate would also be rejected. Furthermore, there can be a prenegotiation for determining the initial contract draft or a contract just needs a majority instead of unanimity to be able to become the current contract draft.

The mentioned five building blocks (quotas, agent-based proposals, three-valued logic, prenegotiation, majority rule) can be combined and yield different protocol variants. Moreover, we used shift as well as swap operations for mutating the sequences of the machine schedule and assumed two different start parameter values for the quotas building block. The usage and combination of the building blocks and the two last mentioned options led to 96 different protocol candidates after excluding infeasible protocol configurations. Finally, we trained the ANN with the negotiation parameters (e.g., number of agents and machines or tardiness factor), the protocol parameters (e.g., which building blocks were used), and a target value

(see next section). Furthermore, as a benchmark method, we applied a standard linear regression with ordinary least squares on a sample of 2.5 million data sets⁴.

5.2 Target Measuring

As argued before, the objective function cannot simply be observed and used in a centralized way as the agents might not be willing to directly share this information with a third party. Therefore, we introduce a measure that reveals little information about the actual cost functions of the negotiating parties. The agents reveal a level of satisfaction, i.e., they state how satisfied they are with the outcome. For this purpose, they need to make up their mind what outcome value for a given scenario could be expected. We simulated this by running two individual heuristic optimisation procedures for each agent – resulting in an estimate for the individual best possible and individual worst possible value. At first, the jobs of an agent are sorted by the weight (for the case of total weighted completion time) or the due date (for the case of maximum lateness) to construct an initial solution. Then, a greedy local search procedure is applied. Finally, we computed the level of satisfaction for each agent as follows: $1 - \frac{\text{outcome} - \text{best}}{\text{worst} - \text{best}}$. For the evaluation, we used averaged values over all agents. The histogram of the obtained satisfaction levels for all 10 million data sets is shown in Figure 3.

The histogram shows that the negotiation results are mostly distributed between 85% and 100%. Some negotiations even obtained better results decentrally than the heuristic estimation – without having information such as the weight values or due dates. The average value of the level of satisfaction was 85.7% with a standard deviation of 17.9%.

The clustering near to 100% may result from the huge spread between the best and worst conceivable value. Whereas, for the best value, only the jobs of one agent are considered and distributed among several machines, the jobs are lined up on one machine for the worst value and the respective agent’s jobs are the last ones in the sequence.

⁴More data sets would have exceeded the available working memory of 8 GB.

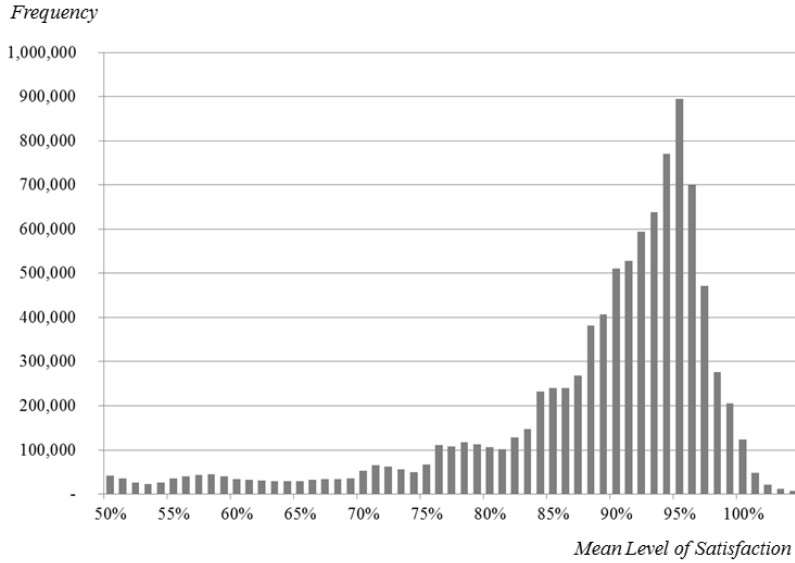


Figure 3: Histogram of the satisfaction level of the simulation data sets (outliers truncated)

5.3 Topologies

We tested 12 different network topologies for the ANN by using different parameterizations for the following scenarios: (1) a single hidden layer only, (2) the same number of neurons in both hidden layers, (3) more neurons in the second hidden layer than in the first, and (4) vice versa. We conducted the learning process for every setup five times using five million training data sets. The mean prediction errors for separate 10,000 evaluation data sets are shown in Table 1.

The experiments show that the topology of the neural network can have a significant influence on the prediction performance. First of all, there is no clear tendency regarding the total number of nodes. Adding neurons leads to better performances at first, but this effect is limited and is reversed at a given point. Usually, more neurons also lead to a greater variation in the experiments. The best results in terms of prediction and stability are achieved by the asymmetric configuration with more nodes in the first hidden layer. On the other hand, the reversed setup, more neurons in the second layer, achieves the worst results. Except for the latter, two layers are superior to one hidden layer in terms of prediction

One Hidden Layer				Symmetric			
M	100	250	500	M	50	100	250
N				N	50	100	250
Min	3.96%	3.79%	4.21%	Min	3.81%	3.39%	3.40%
μ	4.41%	4.04%	4.94%	μ	4.33%	3.77%	4.66%
σ/μ	8.87%	5.45%	13.51%	σ/μ	8.87%	7.21%	22.91%
Asymmetric (Opening)				Asymmetric (Closing)			
M	100	100	250	M	250	500	500
N	250	500	500	N	100	100	250
Min	3.92%	3.97%	4.23%	Min	3.60%	3.26%	3.36%
μ	4.75%	5.43%	6.33%	μ	3.82%	3.47%	3.91%
σ/μ	20.33%	27.39%	38.19%	σ/μ	4.42%	5.12%	14.33%
Benchmark							
Regression: 8.24%							

Table 1: Average prediction error for the five learning runs (5 million training data sets and 10,000 evaluation data sets)

errors. As mentioned before, we used a linear regression as a benchmark method. The prediction error of the regression is approx. twice as high as the average error of the ANN. Except for the configuration $M = 250, N = 500$ every learning run had better performance than the linear regression.

5.4 Training Data Size

One disadvantage of artificial neural networks is said to be the high requirements regarding the training data size. Figure 4 shows a typical history of the (moving) average prediction error subject to different sizes of training data.

With more training data, the prediction error is decreasing. The alteration rate of the descent also decreases and seems to be converging to zero. The fact that the neural network is able to make predictions comparable to the linear regression after only 50,000 training sets shows the difficulty of making predictions with common statistical methods based on the very heterogeneous negotiation data, which is characterized by a lot of noise. The latter is also indicated by the peaks in the

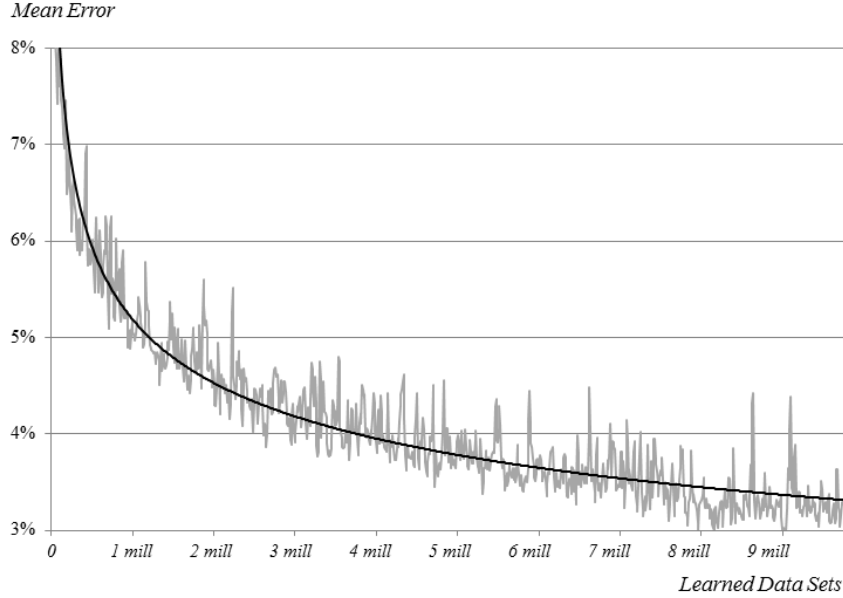


Figure 4: Moving average of prediction error (grey) and trend (black)

learning curve.

5.5 Negotiation

Eventually, we conducted 1,000 negotiation simulations in which we applied the proposed DSS-NPS with varying protocols based on an ANN prediction. As a benchmark, we used a protocol that resulted from the linear regression (P-REG) as well as the protocol that achieved the best average performance in the 10 million training data sets (P-BES). For the regression-based protocol, we configured the parameters according to the sign of the significant coefficient estimates, i.e., if a building block had a positive estimate, we applied it in the protocol; in the case of insignificant coefficient estimates, we used a random generator to decide if it is applied in the protocol.

The average level of satisfaction of the DSS-NPS was 91.16% (coefficient of variation (CV): 13.39%), whereas the P-REG leads to a negotiation result of 90.70% (CV: 14.18%) and P-BES achieved 90.75% (CV: 14.19%). To test for statistical significance, we did a Wilcoxon-Mann-Whitney test (Mann and Whitney 1947). The

satisfaction level of the DSS-NPS is statistically significantly larger than of P-REG (p-Value: 0.049%) as well as P-BES (p-Value: 0.026%). The outcomes of P-REG and P-BES, however, are not statistically distinguishable (p-Value: 40.94%).

Concluding, the DSS-NPS not only yields the best negotiation results with a very high significance but also the most stable ones.

6 Discussion

The computational results suggest that the proposed DSS-NPS is capable to provide better results than a regression based approach (P-REG) or a protocol that is successful in the training data (P-BES). In contrast to the benchmark procedures, the DSS-NPS is able to recommend a protocol dynamically with respect to the specific underlying characteristics. This flexibility and discretion is a major advantage of the proposed system. Although there is a noticeable prediction error, the ANN delivers good predictions of the performance of protocol candidates which is shown by the gap between the predictive power of the ANN and the regression. Finally, since the welfare generated by the negotiation, measured by the proxy of level of satisfaction, could be enhanced, the application of the DSS-NPS proved itself to be beneficial.

As we have addressed before, the exposure of private information about negotiation parameters is not a self-evident assumption as agents may have objections against this revelation (Lang and Fink 2013). Since sensible information might be given to competing parties, there is an issue of trust in the DSS-NPS. If the confidential information was exposed, it could be utilised during the negotiation, i.e., the now better informed counter agents can make better decisions to the disadvantage of the exposed agent (Sandholm 1999). Furthermore, the information about operational data can also be used against the agent outside of the negotiation context on a strategic level. For instance, negotiations on machine allocation can disclose information about the respective order situation of the companies; this can be utilised for competitive price changes or customer acquisitions due to increased advertisement activities. Another aspect to discuss is the possibility of manipulating the system. There may be a way to change the recommendations of the DSS-NPS in favour of oneself by stating false information. For example,

changing the revealed level of satisfaction leads to a different linking between the neurons; this could asymmetrically alter the prediction of the ANN. Furthermore, an agent could lie about his or her information in the prediction phase, e.g., by stating a false objective, which also might result in an asymmetrically change to the benefit of the lying agent (cf. Myerson 1983, 2008).

A drawback of artificial neural networks is that they require large amounts of training data sets. As shown in the computational experiments, millions of data sets are needed to obtain a satisfying predictive power. The simulations of this study addressed an important but small area of cooperative operational planning. An implementation that is supposed to create a universal DSS-NPS has to take this into account and make sure that sufficient training data is available. Besides the restricted problem space, the protocol space of this study mainly focussed on a single protocol family which incorporates a large number of potential configurations and parameterizations, though. Nevertheless, a general support system also has to deal with an unrestricted protocol space; thus, creating a universal system may be very challenging. As a consequence, the system seems to be adequate for specific purposes – such as the machine scheduling example presented here – rather than being a single universal tool that can handle all kinds of problems and protocols simultaneously.

Artificial neural networks do not need statistical assumptions or knowledge about the input data. An ANN is self-organizing and independent of the underlying problem, whereas statistical methods are often based on assumption such as, e.g., normal distribution of residuals (Shmueli and Koppius 2011). The neural network itself is a black box, i.e., the weights of the links between the neurons are opaque and can take numerically very different values with a comparable predictive power. However, an ANN possesses implicit knowledge that has to be extracted costly before it can be utilised (Buscema 2013; Kaikhah and Doddameti 2006). The bias towards explanatory models (see Shmueli and Koppius 2011) and the fact that an ANN is initially a black box could obstruct technology acceptance of ANN based systems.

7 Conclusion and Future Work

The aim of this study was to investigate the problem of selecting an adequate protocol for automated negotiation and develop a design artifact that improves the welfare generated by a negotiation by finding adequate protocols. To address this issue, we designed a decision support system for negotiation protocol selection (DSS-NPS). The system is based on a machine learning approach for supervised learning. Specifically, we implemented an artificial neural network (ANN) that is supposed to learn patterns and connections between the performance of a protocol and the characteristics of the protocol and negotiation scenario. The negotiating parties can use this prediction as protocol recommendation. The computational experiments suggest that the design is, on the one hand, able to make adequate predictions on the protocol performance and, on the other hand, capable to improve the negotiation by utilising those predictions. We showed that the negotiation outcome can be significantly improved.

This work contributes to existing research by addressing the problem of protocol selection. Furthermore, as a main contribution, we build and evaluate an artifact that can support organizations in their managing activities, namely, joint operational planning by means of negotiation. The prototype proved itself to be beneficial and, by doing so, advances the knowledge of information systems research.

Nevertheless, the positive results also underlie some limitations: First of all, the experimental environment is characterized by a restricted protocol and negotiation problem space. The implications cannot be generalized to a universal application that can cope with unrestricted spaces. The need for enormous training data suggests that the DSS-NPS is not applicable to be a unified support system that handles all classes of problems and protocols simultaneously, but rather can be applied for specific purposes – as carried out in the experiments. Another limitation is the possibility of manipulation; negotiating agents could misstate information to obtain an advantage. This is feasible if false information in the learning or prediction process leads to a one-sided benefit.

Future work will, among other things, address these limitations. Future research will address possible ways to manipulate the support system which has to

be analysed in detail. Based on the analysis, there might be countermeasures such as adequate incentives to prevent or at least limit manipulation. Furthermore, the DSS-NPS is going to be evaluated for further problems and protocols. At this, the sensitivity of adding further elements to the protocol or problem space will be investigated in detail. Finally, the findings of this study's and future evaluations will be considered for improving the design (build-and-evaluate loop).

References

- Abdul-Razaq, T., Potts, C., and Van Wassenhove, L. (1990). A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Applied Mathematics*, 26 (2-3) 235–253.
- Agnetis, A., Pacciarelli, D., and Pacifici, A. (2007). Multi-agent single machine scheduling. *Annals of Operations Research*, 150 (1), 3–15.
- Akturk, M. and Ozdemir, D. (2000). An exact approach to minimizing total weighted tardiness with release dates. *IIE transactions*, 32 (11), 1091–1101.
- An, B., Lesser, V., Irwin, D., and Zink, M. (2010). Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proceedings of Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)* (van der Hoek, W., Kaminka, G. A., Les rance, Y., Luck, M., and Sen, S., eds.), pp. 981 – 988.
- Bengio, Y. and LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large-Scale Kernel Machines* (Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., eds.), pp. 321–360, MIT Press, Cambridge, MA, USA.
- Benyoucef, M., Alj, H., V zeau, M., and Keller, R. K. (2001). Combined negotiations in e-commerce: Concepts and architecture. *Electronic Commerce Research*, 1 (3), 277–299.
- Bichler, M., Gupta, A., and Ketter, W. (2010). Designing smart markets. *Information Systems Research*, 21 (4), 688–699.
- Bichler, M., Kersten, G., and Strecker, S. (2003). Towards a structured design of electronic negotiations. *Group Decision and Negotiation*, 12 (4), 311–335.
- Binmore, K. and Vulkan, N. (1999). Applying game theory to automated negotiation. *Netnomics*, 1 (1), 1–9.

- Bosse, T., Jonker, C. M., and Treur, J. (2004). Experiments in human multi-issue negotiation: Analysis and support. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, Volume 2, pp. 671–678.
- Braun, P., Brzostowski, J., Kersten, G., Kim, B., Strecker, S., and Vahidov, R. (2006). e-Negotiation systems and software agents: Methods, models, and applications. *Intelligent Decision-making Support Systems* (Gupta, J. N. D., Forgy, G. A., and Mora, M. T., eds.), pp. 271–300, Springer, London, UK.
- Buscema, M. (2013). Supervised artificial neural networks: Backpropagation neural networks. *Intelligent Data Mining in Law Enforcement Analytics* (Buscema, M. and Tastle, W. J., eds.), pp. 119–135, Springer, Dordrecht, The Netherlands.
- Büttner, R. (2006). A classification structure for automated negotiations. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology (IEEE/WIC/ACM, eds.)*, pp. 523–530.
- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45 (1), 41–51.
- Chalamish, M. and Kraus, S. (2012). AutoMed: An automated mediator for multi-issue bilateral negotiations. *Autonomous Agents and Multi-Agent Systems*, 24 (3), 536–564.
- Cheng, T. C. E., Chung, Y.-H., Liao, S.-C., and Lee, W.-C. (2013). Two-agent single-machine scheduling with release times to minimize the total weighted completion time. *Computers & Operations Research*, 40 (1), 353–361.
- Cheng, T. C. E. and Sin, C. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47 (3), 271–292.
- Cicirello, V. (2003). Weighted tardiness scheduling with sequence-dependent setups: A benchmark library. Technical Report, Intelligent Coordination and Logistics Laboratory, Carnegie Mellon University, Pittsburgh, PA, USA,
- Conitzer, V. and Sandholm, T. (2004). Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce (ACM-EC 2004)* (ACM, ed.), pp. 132–141.
- Douma, A., Schuur, P., and Schutten, M. (2007). Barge rotation planning and quay scheduling in the port of Rotterdam. In *Proceedings of the Sixth Triennial Symposium on Transportation Analysis*, pp. 1–32.

- Dudek, G. and Stadtler, H. (2005). Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163 (3), 668–687.
- Faratin, P. (2000). Automated Service Negotiation between Autonomous Computational Agents. Dissertation, University of London, London, UK.
- Farrell, J. and Rabin, M. (1996). Cheap talk. *The Journal of Economic Perspectives*, 10 (3), 103–118.
- Fink, A. (2006). Supply chain coordination by means of automated negotiations between autonomous agents. *Multiagent based Supply Chain Management (Studies in Computational Intelligence, Vol. 28)* (Chaib-draa, B. and Müller, J., eds.), pp. 351–372, Springer, Berlin / Heidelberg, Germany.
- Fujita, K., Ito, T., and Klein, M. (2010a). Secure and efficient protocols for multiple interdependent issues negotiation. *Journal of Intelligent and Fuzzy Systems*, 21 (3), 175–185.
- Fujita, K., Ito, T., and Klein, M. (2010b). Representative based multi-round protocol based on revealed private information for multi-issue negotiations. *Multiagent and Grid Systems*, 6 (5-6), 459–476.
- Gibbard, A. (1973). Manipulation of voting schemes: A general result. *Econometrica*, 41 (4), 587–601.
- Henseler, J. (1995). Back propagation. *Artificial Neural Networks: An Introduction to ANN Theory and Practice* (Braspenning, P. J., Thuijsman, F., and Weijters, A. J. M. M., eds.), pp. 37–66, Springer, Berlin / Heidelberg, Germany.
- Hevner, A., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28 (1), 75–105.
- Homberger, J. (2010). Decentralized multi-level uncapacitated lot-sizing by automated negotiation. *4OR*, 8 (2), 155–180.
- Homberger, J. (2011). A generic coordination mechanism for lot-sizing in supply chains. *Electronic Commerce Research*, 11 (2), 123–149.
- Homberger, J. (2012). A (\cdot , \cdot)-coordination mechanism for agent-based multi-project scheduling. *OR Spectrum*, 34 (1), 107–132.
- Hudson, P. and Postma, E. (1995). Choosing and using a neural net. *Artificial Neural Networks: An Introduction to ANN Theory and Practice* (Braspenning, P. J., Thuijsman, F., and Weijters, A. J. M. M., eds.), pp. 273–287, Springer, Berlin / Heidelberg, Germany.

- Hurwicz, L. (1973). The design of mechanisms for resource allocation. *The American Economic Review*, 63 (2), 1–30.
- Hurwicz, L. (2008). But who will guard the guardians? *American Economic Review*, 98 (3), 577–585.
- Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., and Wooldridge, M. (2001). Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10 (2), 199–215.
- Kaikhah, K. and Doddameti, S. (2006). Discovering trends in large datasets using neural networks. *Applied Intelligence*, 24 (1), 51–60.
- Kenyon, G. and Paugam-Moisy, H. (1997). Multilayer neural networks: One or two hidden layers? *Advances in Neural Information Processing Systems 9* (Mozer, M., Jordan, M., and Petsche, T., eds.), pp. 148–154, MIT Press, Cambridge, MA, USA.
- Kersten, G. E. (2001). Modeling distributive and integrative negotiations. *Group Decision and Negotiation*, 10 (6), 493–514.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220 (4598), 671–680.
- Klein, M., Faratin, P., Sayama, H., and Bar-Yam, Y. (2003). Negotiating complex contracts. *Group Decision and Negotiation*, 12 (2), 111–125.
- Klemperer, P. (2002). What really matters in auction design. *The Journal of Economic Perspectives*, 16 (1), 169–189.
- Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94 (1-2), 79–97.
- Kraus, S. (2001). Automated negotiation and decision making in multiagent environments. *Multi-agent Systems and Applications* (Luck, M., Marik, V., Stepankova, O., and Trappl, R., eds.), pp. 150–172, Springer, New York, NY, USA.
- Kůrková, V. and Sanguineti, M. (2013). Can two hidden layers make a difference? *Adaptive and Natural Computing Algorithms* (Tomassini, M., Antonioni, A., Daolio, F., and Buesser, P., eds.), pp. 30–39, Springer, Berlin / Heidelberg, Germany.

- Lang, F. and Fink, A. (2012). Negotiating in dynamic environments: Time-efficient automated negotiations by means of combinatorial auctions. *Evolving Systems*, 3 (3), 189–201.
- Lang, F. and Fink, A. (2013). Information sharing in interorganizational systems – A story about falsehood, greed, and privacy. In *Proceedings of the 13th International Conference on Group Decision and Negotiation* (Martinovski, B., ed.), pp. 319–323.
- Lang, F. and Fink, A. (2014). Learning from the metaheuristics: Protocols for automated negotiations. *Group Decision and Negotiation*, to appear.
- Lang, F. and Klein, M. (2013). A metanegotiation support system: Supporting protocol search between rival agents. MIT Sloan Research Paper No. 5030-13, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Lang, F., Schryen, G., and Fink, A. (2011). Automated negotiations under uncertain preferences. *Proceedings of the International Conference on Information Systems (ICIS 2011)*.
- Lang, F., Schryen, G., and Fink, A. (2012). Eliciting, modeling, and processing uncertain human preferences for software agents in electronic negotiations: An empirical study. *Proceedings of the International Conference on Information Systems (ICIS 2012)*.
- Le, N. Q., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., and Ng, A. Y. (2011). On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (Getoor, L. and Scheffer, T., eds.), pp. 265–272.
- Lenstra, J. K., Kan, A. H. G. R., and Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343–362.
- Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., and Jonker, C. M. (2012). GENIUS: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, to appear.
- Mann, H. and Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18 (1), 50–60.
- March, S. T. and Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15 (4), 251–266.

- Marsa-Maestre, I., Klein, M., Hoz, E. de la, and Lopez-Carmona, M. A. (2011). Negowiki: A set of community tools for the consistent comparison of negotiation approaches. *Agents in Principle, Agents in Practice* (Kinny, D., Hsu, J. Y.-J., Governatori, G., and Ghose, A. K., eds.), pp. 424–435, Springer, Berlin / Heidelberg, Germany.
- Marsa-Maestre, I., Klein, M., Jonker, C. M., and Aydogan, R. (2013). From problems to protocols: Towards a negotiation handbook. *Decision Support Systems*, to appear.
- Maskin, E. S. (2008). Mechanism design: How to implement social goals. *American Economic Review*, 98 (3), 567–576.
- McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5 (4), 115–133.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*. 21 (6), 1087–1092.
- Mönch, L., Balasubramanian, H., Fowler, J. W., and Pfund, M. E. (2005). Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. *Computers & Operations Research*, 32 (11), 2731–2750.
- Myerson, R. (1979). Incentive compatibility and the bargaining problem. *Econometrica*, 47 (1), 61–73.
- Myerson, R. (1983). Mechanism design by an informed principal. *Econometrica*, 51 (6), 1767–1797.
- Myerson, R. (1988). Mechanism design. Discussion Paper, No. 796, The Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, IL, USA.
- Myerson, R. (2008). Perspectives on mechanism design in economic theory. *The American Economic Review*, 98 (3), 586–603.
- Pasquier, P., Hollands, R., Rahwan, I., Dignum, F., and Sonenberg, L. (2010). An empirical study of interest-based negotiation. *Autonomous Agents and Multi-Agent Systems*, 22 (2), 249–288.
- Raiffa, H. (1982). *The Art and Science of Negotiation*, Harvard University Press, Cambridge, MA, USA.

- Rief, D. and van Dinther, C. (2010). Negotiation for cooperation in logistics networks: An experimental study. *Group Decision and Negotiation*, 19 (3), 211–226.
- Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge, MA, USA.
- Rothkopf, M., Teisberg, T., and Kahn, E. (1990). Why are Vickrey auctions rare? *Journal of Political Economy*, 98 (1), 94–109.
- Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, 323 (6088), 533–536.
- Sandholm, T. (1999). Distributed rational decision making. *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence* (Wei, G., ed.), pp. 201–258, MIT Press, Cambridge, MA, USA.
- Sandholm, T. (2003). Automated mechanism design: A new application area for search algorithms. *Principles and Practice of Constraint Programming* (Rossi, F., ed.), pp. 19–36, Springer, Berlin / Heidelberg, Germany.
- Sandholm, T. and Lesser, V. R. (1997). Coalitions among computationally bounded agents. *Artificial Intelligence*, 94 (1-2), 99–137.
- Satterthwaite, M. A. (1975). Strategy-proofness and Arrow’s conditions. Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10 (2), 187–217.
- Shmueli, G. and Koppius, O. (2011). Predictive analytics in information systems research. *MIS Quarterly*, 35 (3), 553–572.
- Stadtler, H. (2007). A framework for collaborative planning and state-of-the-art. *OR Spectrum*, 31 (1), 5–30.
- Ströbel, M. (2000). Effects of electronic markets on negotiation processes. *Proceedings of the European Conference of Information Systems (ECIS 2000)*.
- Ströbel, M. and Weinhardt, C. (2003). The Montreal taxonomy for electronic negotiations. *Group Decision and Negotiation*, 12 (2), 143–164.
- Thomas, D. J. and Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94 (1), 1–15.
- Vetschera, R. (2013). Negotiation processes: An integrated perspective. *EURO Journal on Decision Processes*, 1 (1-2), 135–164.

- Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16 (1), 8–37.
- Vulkan, N. (1999). Economic implications of agent technology and e-commerce. *The Economic Journal*, 109 (453), 67–90.
- Widrow, B. and Hoff, M. (1960). Adaptive switching circuits. In IRE WESCON Convention Record Part 4, 96–104.
- Yang, Y., Singhal, S., and Xu, Y. (2009). Offer with choices and accept with delay: A win-win strategy model for agent based automated negotiation. Proceedings of the International Conference on Information Systems (ICIS 2009).
- Yegnanarayana, B. (2004). *Artificial Neural Networks*. Prentice-Hall of India, New Delhi, India.

IMPRINT

SERIES

HSU Institute of Computer Science Research Paper Series
ISSN 2198-3968

EDITOR

Prof. Dr. Andreas Fink

Institute of Computer Science
Helmut-Schmidt-Universität Hamburg

Holstenhofweg 85
22043 Hamburg, Germany

<http://ifi.hsu-hh.de>