

RESEARCH
PAPER

No 14-01

DESIGN OF AUTOMATED
NEGOTIATION MECHANISMS
FOR DECENTRALIZED
HETEROGENEOUS MACHINE
SCHEDULING

Fabian Lang

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

Andreas Fink

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

Tobias Brandt

Albert-Ludwigs-Universität Freiburg i. Br.
tobias.brandt@is.uni-freiburg.de

JANUARY 2014

Design of Automated Negotiation Mechanisms for Decentralized Heterogeneous Machine Scheduling

Fabian Lang^a, Andreas Fink^a, and Tobias Brandt^b

^aHelmut-Schmidt-Universität Hamburg

^bAlbert-Ludwigs-Universität Freiburg i. Br.

Abstract

The increasing coupling of planning and scheduling between different companies leads to novel challenges in devising and implementing effective decision support systems. In this paper, we describe a hard decentralized scheduling problem with heterogeneous machines and competing job sets that belong to different self-interested stakeholders (agents). These agents want to minimize their costs that consist of individual tardiness cost as well as their share of the machine operating cost. The determination of a beneficial solution, i.e., a respective contract in terms of a common schedule, is particularly difficult due to information asymmetry and self-interested behavior of the involved agents. To solve this coordination problem, we present two automated negotiation protocols with a set of optional building blocks. In the first protocol, new solutions are iteratively generated as mutations of a single provisional contract and proposed to the agents, while feasible rules with quotas restrict the acceptance decisions of the agents and, thus, the successive adaptation of the provisional contract. The second protocol is based on a population of contracts and mimics evolutionary processes. For evaluation purposes, we built a simulation testbed and conducted computational experiments. The computational study shows that the protocols can achieve high quality solutions very close to results from centralized multi-criteria procedures. Particular building block configurations yield improved outcomes, e.g., in case that the agents are also allowed to make contract proposals. Thus, the presented approach contributes to the methodology and practice of collaborative decision making.

Keywords: Decision support systems, Negotiations, Mechanism design, Complex contracts.

1 Introduction

Planning and scheduling face new challenges these days. As companies are often strongly integrated within supply networks, related decisions are not exclusively subject to a single company's own preferences but increasingly depend on the interaction with, e.g., suppliers, subcontractors, and customers (Dawande et al., 2006). Consequently, different autonomous decision makers are more and more connected with each other such that decision making becomes a collaborative task. Since the involved stakeholders have their very own preferences and objectives, planning and scheduling must consider strategic behavior (Kersten and Mallory, 1990). For example, decision makers that compete for shared machine resources may not be willing to disclose private information (Agnētis et al., 2007a) or may provide biased information such as exaggerated needs to work towards higher individual profits to the detriment of the overall allocation efficiency (Klein et al., 2003). Thus, revealed information can be incomplete or misleading and, hence, is a fruit of a poisonous tree for a central authority. As a consequence, a traditional centralized scheduling approach may not be effective in obtaining high quality solutions. Since Operational Research is generally concerned with analytical methods for decision making, the issue of collaborative decision making within a group of autonomous decision makers is highly relevant. In multi-party situations, there is usually no single fully-informed authority that is entitled to hierarchically allocate resources and determine the courses of action for the whole system. Thus, a decentralized group decision making procedure is needed which takes into account strategic behavior of self-interested parties and restricted availability of information.

In this paper, we focus on decentralized decision problems on the level of operations management (in particular scheduling problems as described below), which are characterized by regular processes with recurring decision tasks as well as the potential and need for a formalized and automated decision support procedure. Thus, it is appropriate to devise and apply an automated negotiation approach. In automated negotiations, decision making entities are represented by autonomous non-cooperative software agents that negotiate with each other. In contrast to human negotiators, software agents can easily negotiate for millions of rounds to find a mutually agreed upon solution (also called contract). Respective negotia-

tions may be regarded as decentralized search processes which aim to iteratively enhance a contract from an underlying contract space (search space). A negotiation procedure is defined by a protocol that controls the observable actions of the agents and related interactions. Such procedures work without requiring the parties to reveal full information on preferences. The rules of the protocol should work towards finding a mutually accepted contract which achieves pursued criteria such as Pareto-efficiency and social welfare (Jennings et al., 2001).

Hence, this paper introduces a novel decentralized scheduling problem that incorporates competing job sets that are connected to multiple autonomous agents, machine operating costs, particularly energy costs, as well as tardiness costs. Extending conventional scheduling assumptions, we argue that both cost functions may be nonlinear and decisions are to be made by finding a contract between self-interested agents. Energy costs have drawn the interest of managers and researchers in recent years. For instance, while they have always constituted a large part of operational costs for energy-intensive industries, rising energy prices and increasing power consumption of hardware have turned energy into a major cost driver for providers of IT infrastructure as well (Chen et al., 2005). The fact that energy consumption of IT hardware does not necessarily increase linearly with workload adds an additional computational challenge to scheduling mechanisms that include energy costs (Bodenstein et al., 2012), as do dynamic tariff schemes that compensate industrial customers for reducing their consumption during peak times (Braithwait and Hansen, 2011). We include these determinants of energy costs into the negotiation scheme introduced in this paper to provide a realistic representation of total expenditures.

The aim of this paper is twofold. Firstly, we want to advance the knowledge on generic decentralized negotiation procedures for complex contract spaces. For this purpose we describe, enhance, and analyze negotiation mechanisms with different building blocks for achieving beneficial outcomes. Secondly, we introduce and solve the considered multi-agent machine scheduling problem by means of the negotiation protocols with problem-specific operators. For this purpose, we incorporate characteristics of restricted information availability and strategic behavior of autonomous agents in a decentralized scheduling situation.

The findings from this study contribute to the methodology and practice of de-

cision support for hard decentralized scheduling problems with information asymmetry and multiple self-interested agents. For such kinds of problems, for the first time, we devise a rich framework of mainly generic negotiation mechanisms which is applied for challenging problem instances with up to 19 agents. The presented experiments show that the most effective configurations of the procedures often achieve Pareto-efficient solutions and accomplish a relatively low extra cost in consequence of the selfish behavior. The computational study provides detailed insights into the relevant determinants for a successful application of the protocol.

The remainder of this paper is structured as follows: After this introduction, we formally define the considered multi-agent scheduling problem and give illustrative examples for applications. Afterwards, we give an overview of related work and introduce the devised negotiation-based solution mechanisms. These protocols are evaluated in a computational study which is subsequently presented together with a discussion of results. Finally, we conclude the paper and present future work.

2 Problem

2.1 Problem Definition

A set of non-preemptable jobs $\mathcal{J} = \{1, \dots, j, \dots, J\}$ which originate from a set of competing agents $\mathcal{I} = \{1, \dots, i, \dots, I\}$ has to be scheduled on a set of machines $\mathcal{M} = \{1, \dots, m, \dots, M\}$. Each job j has an associated agent $a_j (\in \mathcal{I})$, a standardized processing time p_j^s , a required resource requirement per time slot r_j (in terms of a single resource type), a release time s_j , and a due date d_j ($p_j^s, r_j, s_j, d_j \geq 0$). Without loss of generality, we assume a discrete planning horizon with $\mathcal{T} = \{1, \dots, t_{max}\}$ as set of all time slots and p_j^s , s_j , and d_j as non-negative integer values. We assume that p_j^s, r_j , and s_j are publicly known to all parties, whereas d_j is just known to a_j .

The M machines are heterogeneous and have three relevant characteristics: (1) a capacity cap_m , which is the maximal resource provision by a machine m per time slot, (2) an operating speed os_m , which determines the speed of job processing, and (3) an operating cost function $E_{t,m}$, which determines the cost at time t subject to the machine's utilization $u_{t,m}$ (as described below). All machines

have to fulfill the capacity constraint which is $0 \leq u_{t,m} \leq 1, \forall t \in \mathcal{T}, \forall m \in \mathcal{M}$. The machines' parameters are public information, i.e., all parties are aware of them.

The key decision variable of the problem is the schedule π which determines the start time (σ_j) and assigned machine (μ_j) for each job:

$$\pi = \{(\sigma_1, \mu_1), \dots, (\sigma_J, \mu_J)\}. \quad (1)$$

The time $p_j(\pi)$ required for the processing of a job j on a machine $m = \mu_j$ is determined by a machine's operating speed os_m , the job's standardized processing time p_j^s , and a standardized operating speed \overline{os} : $p_j(\pi) = \lceil \frac{\overline{os}}{os_{\mu_j}} * p_j^s \rceil$ (if $os_{\mu_j} = \overline{os}$, then $p_j(\pi) = p_j^s$). The completion time f_j of a job j is determined by its start and processing time: $f_j = \sigma_j + p_j(\pi)$. As the problem includes release times, the start time must not be earlier than the release time of a job, that is $\sigma_j \geq s_j, \forall j \in \mathcal{J}$.

The objective of an agent i is the minimization of his or her total cost which consists of two components: machine operating costs and tardiness costs. The agents use the same measuring commodity (numéraire; e.g., monetary units).

The operating cost $E_{t,m}$ for a given machine m at a given time slot t is subject to the three non-negative parameters $\alpha_m^E, \beta_m^E, \gamma_m^E$, the machine utilization $u_{t,m}(\pi) = \frac{\sum_{k \in \mathcal{J}: \mu_k = m \wedge \sigma_k \leq t < f_k} r_k}{cap_m}$, and a tariff $\Gamma(t)$: $E_{t,m}(u_{t,m}) = [\alpha_m^E * (u_{t,m})^{\beta_m^E} + \gamma_m^E * \Theta(u_{t,m})] * \Gamma(t)$, with Θ as the Heaviside function: $\Theta(\bullet) = 1 : \bullet > 0; \Theta(\bullet) = 0 : \bullet \leq 0$.

The overall operating costs of the machines are given by

$$EC(\pi) = \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} E_{t,m}(u_{t,m}). \quad (2)$$

The overall operating costs EC have to be split among the set of agents \mathcal{I} , i.e., each agent i has to cover a share EC_i with $\sum_{i \in \mathcal{I}} EC_i = EC$. Different cost allocation schemes are discussed later on.

The tardiness T_j of a job j depends on the completion time f_j and the due date d_j : $T_j = \max\{f_j - d_j; 0\}$. Only if a job is tardy, there arises a tardiness cost $TC_j(T_j)$. This function $TC_j(T_j)$ can be nonlinear and is represented by $TC_j(T_j) = \alpha_j^w * (T_j)^{\beta_j^w} + \gamma_j^w * \Theta(T_j)$, with $\alpha_j^w, \beta_j^w, \gamma_j^w \geq 0$ and only known to the job's owner a_j . The total tardiness cost of an agent i is given by

$$TTC_i(\pi) = \sum_{j \in \mathcal{J}: a_j = i} TC_j(T_j). \quad (3)$$

Finally, we obtain an agent i 's overall objective, minimization of his or her individual cost C_i consisting of tardiness and operating cost:

$$\min_{\pi} C_i = TTC_i(\pi) + EC_i(\pi). \quad (4)$$

Since we are in a group decision situation, the agents decide according to their individual objectives, i.e., a central authority can neither determine the schedule nor instruct how they must decide or behave. The outcome depends on the applied negotiation protocol P . Consequently, we define that the schedule is determined by a contract c that is the outcome of a negotiation N_P subject to a protocol P :

$$\pi \leftarrow c = N_P(\mathcal{J}, \mathcal{I}, \mathcal{M}). \quad (5)$$

The task of mechanism design which facilitates the planning process is to find an effective negotiation protocol P for a given objective. The outcome of a negotiation protocol should ideally be Pareto-efficient (Jennings et al., 2001; Zhang et al., 2011), i.e., no agent can reduce his or her individual cost without increasing another agent's cost. For nonlinear search spaces, achievement of Pareto-efficiency is not guaranteed. Therefore, the basic objective is to find a negotiation protocol that minimizes the Euclidean distance $d(\bullet, \bullet)$ to the closest Pareto-efficient solution:

$$\min_P \arg \min_{\pi_P \in \mathcal{P}} d(\pi, \pi_P) \quad (6)$$

Since assessing Pareto-efficiency may be computationally infeasible in case of a considerable number of agents, we consider as a further evaluation measure the minimization of social cost (maximization of social welfare):

$$\min_P \sum_{i \in \mathcal{I}} TTC_i + EC_i. \quad (7)$$

Such a cost optimum is itself a Pareto-efficient solution. The aggregate cost represent the social welfare function (or in this case, the social loss) which is also a

usual criterion of multi-party optimization (Myerson, 1981). We expect a protocol P to deliver near Pareto-efficient solutions for a small agent count and a preferably high social welfare, i.e., low aggregate costs.

The introduced scheduling problem is complex due to the decentralized decision situation and the nonlinear objective functions in addition to the fact that the included single-agent problems are already \mathcal{NP} -hard as a simplified version:

Proposition 1 (Complexity). *The single agent version of the presented problem is \mathcal{NP} -hard.*

The considered problem contains the well-known \mathcal{NP} -hard single machine total weighted tardiness problem (SMTWTP), which was proven to be \mathcal{NP} -hard by Lenstra et al. (1977), as a special case. We obtain the SMTWTP, if we set $M = 1$; $I = 1$; $s_j = 0$, $r_j = 1$, $\beta_j^w = 1$, $\gamma_j^w = 0$, $\forall j \in J$; $cap_1 = 1$; $os_1 = \overline{os}$; $\Gamma(t) = 0$, and let the one and only agent decide about the procedure P minimizing TTC_i . Consequently, the presented problem is \mathcal{NP} -hard as well. \square

2.2 Illustrative Applications

In this section, we present two illustrative application examples from the domains of (1) production and (2) high-performance scientific computing services.

In classical production scheduling, the machines can represent, e.g., industrial robots or workshops. We argue that such machines may have nonlinear operating cost functions for two reasons: firstly, present fixed costs are associated to nonlinear average variable costs (see Varian, 2010, Ch. 21); secondly, a lot of technical procedures of machines and relationships of machine components are nonlinear (see Vergnano et al., 2012). The agents can be interpreted differently: for instance, they can represent (1) different departments within a company (e.g., the production and the product/prototype engineering department), (2) different companies using shared resources (e.g., a joint venture sharing a factory), or (3) large customers (e.g., in a supply chain). The jobs can be production orders with an announced fulfillment date. There is very little empirical work about tardiness cost functions. Nevertheless, Cook et al. (2012) showed empirically that the perceived inconvenience and propensity to switch airlines of air travelers is in-

deed nonlinearly increasing with delay time or, in other words, with fulfillment tardiness.

Our second example refers to job scheduling for high-performance computing (HPC) – e.g., in the academic domain. The computing infrastructure of research organizations and universities generally includes HPC machines which are used by several researchers or research groups. Commonly, computers in data centers are heterogeneous, since failed components are replaced with newer and better ones and expansions in capacity and performance favor state-of-the-art components (Heath et al., 2005). This heterogeneity results in varying levels of power consumption among the different types of machines for a given utilization. However, previous research has shown that this power consumption is often nonlinear and best approximated by an allometric function (Bodenstein et al., 2012). In a scientific computing scenario, the agents represent the different stakeholders that share computing resources (e.g., researchers, institutes, departments, or even universities). The individual entities are almost exclusively interested in getting their jobs done (Kraus, 2001). In this scenario, typical jobs are simulations or optimization procedures. The tardiness cost function is arguably not linear: for instance, in the critical phase before a project deadline, the cost may be exponentially increasing, because the risk of missing the deadline rises (see Kraus et al., 1995). To conclude, the management of HPC services involves multi-agent machine scheduling problems with nonlinear operating and tardiness costs as well.

3 Related Work

This section surveys recent related work that studies multi-agent scheduling approaches in the field of Operational Research.

Firstly, there are several papers that analyze the computational complexity of multi-agent scheduling: Baker and Smith (2003) proof \mathcal{NP} -hardness of two-agent single machine problems with various individual objective functions. Similarly, Leung et al. (2009) consider the complexity of different two-agent multi-machine scheduling problems which incorporate either identical or different private objective functions. Agnetis et al. (2004, 2007b) analyze the complexity of finding nondominated solutions for various problems from the multi-agent scheduling do-

main, whereas Agnetis et al. (2007a) theoretically address complexity issues of different mechanisms such as auctions or negotiations for multi-agent scheduling. Cheng et al. (2006) analyze the complexity of finding a feasible schedule given that agents are appointed with a limited total weighted number of tardy jobs and Cheng et al. (2008) study the complexity of different multi-agent single machine scheduling problems with objectives in max-form. From a game-theoretic perspective, Agnetis et al. (2009) show that finding a Nash bargaining solution for multi-agent scheduling is \mathcal{NP} -hard and present an algorithm for this problem. Briand et al. (2012) analyze the complexity of computing a Nash equilibrium for multi-agent project scheduling. Finally, Huynh Tuong et al. (2012) summarize, extend, and give a literature overview on the findings on multi-agent scheduling complexity.

Secondly, there is work studying centralized approaches for multi-agent problems disregarding strategic considerations. Balasubramanian et al. (2009) evaluate a genetic algorithm approach for finding nondominated solutions for parallel machine scheduling with two agents with different objectives (makespan and total flow time, respectively). Mor and Mosheiov (2010) introduce a polynomial-time algorithm for the minimization of the weighted earliness of an agent given that a counter agent's maximum earliness does not exceed an upper bound. Similarly, Cheng et al. (2013) discuss a branch-and-bound algorithm and a simulated annealing approach for a single machine problem minimizing one agent's total weighted flow time under the constraint that the maximum lateness of another agent is bounded. Ramacher and Mönch (2011) study a NSGA-II approach for finding nondominated solutions for two-agent single machine scheduling with mixed agent objective functions (total weighted completion time and maximum lateness).

Finally, there are a few related studies that deal with negotiation of schedules: Dudek and Stadtler (2005) consider negotiation approaches for the coordination of supply chain planning. Fink (2004, 2006) introduces a negotiation protocol which draws on quotas for the problem of supply chain scheduling. Similarly, Lang and Fink (2012b,c) present a quota-based protocol for minimizing the total weighted tardiness of several agents with a single and parallel machines. Homberger (2012) describes a negotiation mechanism for multi-project scheduling. Liu et al. (2012) consider negotiations of inter-company scheduling with shared resources.

To our best knowledge, the problem of finding beneficial procedures for multi-

machine scheduling with autonomous agents is very little covered by the literature yet. Hence, this paper contributes to research by devising and analyzing a comprehensive framework of negotiation mechanisms with different optional building blocks which is applied for a challenging decentralized multi machine scheduling problem with nonlinear tardiness and machine operating cost functions. The proposed methodology particularly takes care in considering the restrictions due to the decentralized decision situation with multiple autonomous self-interested agents and information asymmetry.

4 Negotiation Protocols

In this section, we present two negotiation protocols which are based on ideas that originate from metaheuristics. However, the protocols account for strategic issues and maintain privacy of the negotiation parties as far as possible. The protocols are improvement-based, i.e., they are designed to seek for joint gains (Vetschera, 2013). In contrast to previous work of others, our procedures do not require much information and maintain the privacy of the agents. Alongside the basic protocols, we additionally describe optional policy building blocks which may be used to extend the negotiation procedures for possibly improving the results. The effects of the different procedures and building blocks will be analyzed in the subsequent section by computational experiments.

The global elements of the negotiation protocols may be carried out by an artificial mediator software component which facilitates the interaction between the involved agents. Needless to say, this mediator component has no information about the private information of the different agents (in particular preferences). We do not require that the mediator is a trusted third party but the mediator simply carries out fully disclosed procedural elements of the negotiation protocols. For example, the mediator may be implemented by an open source program (white box software component), which is duplicated by the agents in parallel. This also allows the implementation of the overall negotiation system according to a peer-to-peer software architecture without any artificial centralized elements.

4.1 Mediated Negotiation Protocol Based on Simulated Annealing

This protocol is based on concepts from the metaheuristic *simulated annealing*. The idea of utilizing this in negotiations was initially proposed by Klein et al. (2003, 2007) and extended by Fink (2004, 2006) as well as Lang and Fink (2013). Simulated annealing involves that a provisional solution is successively adapted by introducing random changes which may be partly accepted even when they lead to deteriorations. In particular, deteriorating moves are accepted by means of a probabilistic criterion which ensures that smaller and in the search process earlier deteriorations are more likely to be accepted. The acceptance probability is controlled by a virtual temperature parameter according to the so-called Metropolis criterion (Metropolis et al., 1953; Kirkpatrick et al., 1983; Cerný, 1985).

For the considered decentralized scheduling problem with self-interested agents, the overall acceptance of a modified provisional solution (contract) within a negotiation procedure depends on a mutual agreement between the involved autonomous agents. If the agents act greedily by only accepting proposed contracts that seem beneficial for them (i.e., direct improvements), this generally leads to unsatisfactory outcomes as the negotiation usually gets quickly stuck in local optima (Fink, 2006; Lang and Fink, 2012a). Due to private preference information, agents cannot be forced to accept deteriorations by, e.g., using the Metropolis criterion. However, a negotiation protocol can include a rule that requires an agent to accept a certain number of proposals from a set. By introducing such acceptance quotas, agents may be forced to partly accept deteriorations as well and local optima can be overcome. Although there is no virtual temperature parameter, the behavior of simulated annealing processes can be mimicked by means of these quotas.

4.1.1 Pseudo Code

Algorithm 1 shows the pseudo code for the Mediated Negotiation Protocol Based on Simulated Annealing (MNP-SA). The protocol requires data about the initial acceptance quota (q ; i.e., how many contracts have to be accepted in the first round), the number of negotiation rounds (R), the number of contract candidates per round (ρ), the agent objects/interfaces for each agent (*Agents*) as well as the

building blocks configuration, which we will introduce in the next subsection.

Algorithm 1: Mediated Negotiation Protocol Based on Simulated Annealing

Data: $q, R, \rho, Agents[i \in \mathcal{I}], Building\ blocks\ configuration$
Result: c

```

1  $c^a \leftarrow initializeContract()$ 
2  $\beta_q \leftarrow (1/q)^{(1/R-1)}$ 
3 for  $t \leftarrow 0$  to  $R - 1$  do
4    $Accepted \leftarrow \emptyset$ 
5    $Proposals \leftarrow \{c^a\}$ 
6   for  $l \leftarrow 1$  to  $\rho - 1$  do
7      $c^c \leftarrow ProposeMutation(c^a)$ 
8      $Proposals \leftarrow Proposals \cup \{c^c\}$ 
9   end
10  forall the  $i \in \mathcal{I}$  do
11     $\overrightarrow{Dec}_i \leftarrow Agents[i].Decide(Proposals, q)$ 
12  end
13  forall the  $c^c \in Proposals$  do
14    if  $\sum_{i \in \mathcal{I}} Dec_i[c^c] \geq Threshold$  then
15       $Accepted \leftarrow Accepted \cup \{c^c\}$ 
16    end
17  end
18  if  $Accepted \neq \emptyset$  then
19     $c^a \leftarrow RandomlySelect(Accepted)$ 
20  end
21   $q \leftarrow q * \beta_q$ 
22 end
23 return  $c^a$ 

```

At first, an initial contract draft is generated by some random procedure. The current contract draft c^a is called active contract and becomes the eventual contract when the negotiation stops. For the quota q , an annealing factor β_q is calculated which decreases the quota q in each round (see line 21 of Algorithm 1) and is determined such that just one contract candidate has to be accepted at the end of the negotiation ($q = 1$). When those initialization steps are done, the iterative part of the algorithm starts. In the beginning of each round, a set of contract candidates is generated (by either the mediator component or by the agents themselves); this results in the set $Proposals$ which consists of the active contract and $\rho - 1$ contract proposals. Specifically, a proposal c^c shall be a mutation in the neighborhood of c^a . Besides the random generation of an initial contract draft, the mutation constitutes the only problem-specific element of the negotiation protocol. For the considered

scheduling application we employ randomly shifting, swapping (see Geiger, 2010), and batch shifting (see Brandt and Bodenstein, 2012). Afterwards, each agent $i \in \mathcal{I}$ has to decide about which contracts he or she accepts and which not (binary vector \overrightarrow{Dec}_i). Of course, this decision is subject to the current acceptance quota q . By using the individual decision vectors \overrightarrow{Dec}_i , the overall accepted contracts are determined. A contract must be individually accepted by a certain number of agents (*Threshold*) for being accepted by the group as a whole (e.g., unanimity or simple majority). Finally, one contract from the set of overall accepted contracts is randomly selected and becomes the new active contract draft (or the incumbent active contract is maintained in case that the overall acceptance set is empty). The negotiation starts over with new proposals and, eventually, after R negotiation rounds, the last active contract draft results in the final contract c .

4.1.2 Extending Building Blocks

The procedure that has been described in the previous subsection may be enhanced by additional optional building blocks.

In the basic protocol, we assumed that the initial contract is determined at random. This may have the drawback that the starting point of the negotiation process may be rather bad and some particular agent(s) may by chance be in a distinctly better initial position than others. To alleviate this problem, the agents could determine their starting contract in a **prenegotiation**. We implemented this idea as follows: a number of random contracts are generated and the agents determine the actual starting contract by means of a pairwise elimination voting.

Generally, the contract mutation (contract proposing) can be executed randomly or even partly by the agents. In case that the agents are allowed to propose contracts, their strategies must be considered once again. In our implementation of the **agent proposal** case, one agent controls half of the contract candidates in each round by selecting the individually best ones from a set of randomly mutated contracts that is generated by the mediator component. This may also be adapted by letting the agents completely determine contract proposals on their own with possible limits regarding the computation time and the distance to the active contract. In addition to the active contract, the rest of the proposals is generated

randomly by the mediator to prevent easy advantage seeking. The proposing agent changes every iteration in a round robin fashion.

Regarding the decisions, we considered just binary decisions so far: accept or reject. This can be enhanced by introducing a third state: accept due to quota. This **tertiary logic** can express that no improvement is achieved. Consequently, if all agents accept a candidate only because of the quota, the proposal would still be rejected by the negotiation protocol. On the one hand, this can prevent that all agents are worse off, but, on the other hand, it reveals more information which might be unwanted due to privacy considerations.

Finally, the acceptance threshold has to be defined. Related work proposes unanimity ($|J|$) which we use as a default threshold (see Klein et al., 2003; Fink, 2006). As a building block, we also test simple **majority** as threshold ($\lceil J/2 \rceil$).

4.2 Mediated Negotiation Protocol Based on a Genetic Algorithm

The second protocol is inspired by evolutionary algorithms (see Holland, 1975; Goldberg, 1989). The idea of utilizing evolutionary concepts for proposal generation was briefly mentioned by Tung and Lin (2005) for general multi-issue negotiations. Homberger (2012) describe the use of related concepts (a population of contracts and recombinations) for multi-project scheduling. In this work, we present a comprehensive generic negotiation protocol which employs various concepts from genetic algorithms. The general concepts of this protocol have initially been presented by Lang and Fink (2013).

4.2.1 Pseudo Code

Algorithm 2 shows the pseudo code of the Mediated Negotiation Protocol Based on a Genetic Algorithm (MNP-GA). Regarding the input data, the MNP-GA is almost identical to the MNP-SA but, additionally, incorporates a mutation rate (δ) and a number of elitist contracts (ε).

In contrast to the MNP-SA, the MNP-GA requires not a single initial contract but rather an initial population, i.e., a set of contracts. Again, in each negotiation round, the agents accept and reject contracts out of the proposals – thus, a quota

Algorithm 2: Mediated Negotiation Protocol Based on a Genetic Algorithm

Data: $q, R, \rho, \delta, \varepsilon, \text{Agents}[i \in \mathcal{I}]$, *Building blocks configuration*

Result: c

```
1  $Proposals \leftarrow \text{initializePopulation}()$ 
2  $\beta_q \leftarrow (1/q)^{(1/R-1)}$ 
3 for  $t \leftarrow 0$  to  $R - 1$  do
4   forall the  $i \in \mathcal{I}$  do
5      $\overrightarrow{Dec}_i \leftarrow \text{Agents}[i].\text{Decide}(Proposals, q)$ 
6   end
7    $c^a \leftarrow \text{selectActiveContract}(Proposals, \overrightarrow{Dec}_0, \dots, \overrightarrow{Dec}_{I-1})$ 
8    $Elitists \leftarrow \text{selectElitists}(Proposals, \varepsilon, \overrightarrow{Dec}_0, \dots, \overrightarrow{Dec}_{I-1})$ 
9    $Parents \leftarrow \text{selectParents}(Proposals)$ 
10   $Children \leftarrow \text{recombine}(Parents)$ 
11   $Children \leftarrow \text{mutate}(Children, \delta)$ 
12   $Proposals \leftarrow c^a \cup Elitists \cup Children$ 
13  if  $\text{checkDiversity}(Proposals) = \text{false}$  then
14     $Proposals \leftarrow \text{immigration}(Proposals)$ 
15  end
16   $q \leftarrow q * \beta_q$ 
17 end
18 return  $c^a$ 
```

is imaginable again. Consequently, the basic actions of the agents stay the same, no matter which protocol is applied. Based on the agents' decisions, the mediator selects an active contract like in the MNP-SA. This step is needed to prevent drifting away from beneficial solutions and to obtain a unique contract in the end. This active contract is later added to the subsequent generation. Besides the active contract, ε (≥ 0) further contracts fulfilling the threshold can be added to the following generation as well (*Elitists*). Out of the whole set of candidates, a set of parents (*Parents*) is selected by means of a binary tournament selection (see Blickle, 2000). The contracts of *Parents* are recombined by utilizing a problem-specific operator resulting in the set *Children*. We implemented position crossover, order crossover, and partially matched crossover as recombination techniques (see Syswerda, 1991; Goldberg, 1989). Afterwards, the offspring are mutated with a probability of δ and added to the next generation along with the active contract and the elitist contracts. Thereafter, as a building block, the population is optionally checked for diversity, i.e., whether the included contracts are sufficiently different. If this is not the case, random contracts are inserted to obtain a more diverse gene

pool (see Cobb and Grefenstette, 1993). Finally, the quota is lowered and the next iteration round starts until R negotiation rounds are completed and c^a finally represents the resulting contract c .

4.2.2 Extending Building Blocks

The protocol can also be extended by policy building blocks. Firstly, the application of **quotas** could be beneficial. Like in the previous section, a **prenegotiation** can be conducted; however, in this case, the binary elimination results in a set of contracts and not in a single one. Moreover, the recombination can be controlled by **agent interference**, i.e., a mediator creates a number of recombinations and lets the agents partly decide which ones become offspring contracts. As already discussed above, **elitism** is also an optional building block and occurs if $\varepsilon > 0$. Finally, as shown in the pseudo code, **immigration** is also possible given that the diversity falls below a certain threshold. Immigration means that a part of the population is dropped and replaced by randomly generated solutions (see Cobb and Grefenstette, 1993). In doing so, genetically converged contract populations can be enriched with new genotypes.

4.3 Behavioral Analysis

In our simulations, we assume that the privacy of preferences should be maintained. Based on this supposition, the behavior of the agents regarding the acceptance of contract proposals is given as follows:

Proposition 2 (Agent Behavior). *The best strategy for an agent unaware of other agents' preferences is to accept his or her best contracts out of the proposal set.*

According to the expected utility theory (von Neumann and Morgenstern, 1944), an agent chooses the contracts with the highest risk-preference adjusted expected utility. Thus, the proposals are to be weighted with a vector \vec{b}_i depicting an agent's belief that a contract is generally accepted and a utility function $U(\bullet)$ measuring the cost savings and risk preference: $\arg \max_{c^c \in \text{Proposals}} \vec{b}_i * U(C_i(c^c))$. Since an agent has no information about private preferences of the other agents, the beliefs are not distinguishable and, thus, uniform $\vec{b}_i \mapsto \bar{b}$. The decision is now reduced

to $\arg \max_{c^c \in \text{Proposals}} \bar{b} * U(C_i(c^c))$ which is equivalent to choosing the best contract or contracts, respectively: $\arg \max_{c^c \in \text{Proposals}} U(C_i(c^c))$. This corresponds to $\arg \min_{c^c \in \text{Proposals}} C_i(c^c)$. \square

Likewise, it can be shown that the agents also select and propose their best possible contracts in case that the agent proposal extension is used.

4.4 Benchmark Methods

Firstly, we use as a benchmark negotiation protocol an approach that is also based on iterative proposals that mutate an active contract. However, in this benchmark negotiation protocol (BM), proposals are solely generated by the agents and we tried to incorporate a minimal amount of rules. Firstly, an initial active contract is determined by a prenegotiation like the building block explained in section 4.1.2. Secondly, in each negotiation round, the agents propose mutations of the active contract where every agent proposes about the same number of proposals (possible remainders are distributed randomly). Thirdly, a proposal can become the active contract in case of a unanimous acceptance. Finally, the agents are not governed by further rules and can freely decide and act.

Secondly, we compare to results that are obtained by applying state-of-the-art multi-criteria metaheuristics for corresponding hypothetical centralized decision problems. We emphasize that centralized procedures neglect self-interested agents and information asymmetry.

5 Computational Experiments

In the following we present the evaluation of the proposed solution mechanisms. At first, we introduce the simulation setup and, afterwards, the computational results. Finally, we discuss the findings.

5.1 Setup

5.1.1 Problem Instances

For the simulation experiments, 90 problem instances have been created. At this, we draw on the scientific computing application (see section 2.2) and consider a

scenario where computing jobs have to be processed on computing machines.

Each problem instance involves 60 jobs per machine and there are between 5 and 20 machines. This results in 300 – 1,200 jobs in total. A job j requires the resource with an amount of $r_j \in \{2, 4, 8, 12, 24\}$ resembling processor cores. The number of cores is randomly chosen, but must not exceed the problem instance’s machine with the largest capacity ($r_j \leq \max_m(\text{cap}_m)$). The standardized processing time of the job p_j^s is a uniformly distributed integer in the interval $\mathcal{U}[1, 10]$ (standard processing speed is 2.00 GHz). In accordance with Akturk and Ozdemir (2000) and Mönch et al. (2005), the release times as well as the due dates are derived from the resource weighted sum of all processing times $TP = \sum_{j \in \mathcal{J}} p_j * r_j$, the total capacity $CAP = \sum_{m \in \mathcal{M}} \text{cap}_m$, and two parameters (ϕ, φ) . The release time is given by $s_j = \phi * \frac{TP}{CAP} * \mathcal{R}$, where $\phi \in \{0.5, 0.75\}$ and \mathcal{R} is a random number with $\mathcal{R} \sim \mathcal{U}[0, 1)$. For the calculation of the due date, we introduce a slack time which is $\psi_j = \varphi * \frac{TP}{CAP} * \mathcal{R}$. In the generated problem instances, the slack parameter φ can have the values $\varphi \in \{0.1, 0.2, 0.4\}$. The actual due date is then calculated by $d_j = s_j + p_j + \psi_j$. Both, s_j and d_j , are integers, which was generated by rounding. According to the usual practice, the tardiness weight function’s parameters $(\alpha_j^w, \beta_j^w, \gamma_j^w)$ are randomly generated (α_j^w, γ_j^w are integers, β_j^w is a decimal number): $\alpha_j^w \sim \mathcal{U}[1, 5)$, $\beta_j^w \sim \mathcal{U}[0.75, 1.25)$, $\gamma_j^w \sim \mathcal{U}[0, 25)$.

The relevant parameters of the machines are the energy consumption structure $(\alpha_m^E, \beta_m^E, \gamma_m^E)$, the machine resource capacity (r_m), the operating speed os_m , and the tariff (Γ). For the parameterization of the machines, we used empirical data of various real-world computing servers according to the Standard Performance Evaluation Corporation (2008). The report provides performance-to-power-ratios for 182 common server systems, from which the allometric power functions were derived (Bodenstein et al., 2012). Subsequently, we randomly picked $M \in \{5, 10, 20\}$ servers from different manufacturers, each containing between 2 and 48 cores.

Moreover, we include a three-stage tariff as it is often used for industrial power contracts. The tariff $\Gamma(t)$ is equal to $0.25 \forall t \in \mathcal{T}|\{t \bmod 300 < 100\}$, $0.2 \forall t \in \mathcal{T}|\{100 \leq t \bmod 300 < 200\}$, $0.15 \forall t \in \mathcal{T}|\{200 \leq t \bmod 300 < 300\}$.

For each job and machine parameterization, five problem instances were created resulting in $5 \times 2 [\phi] \times 3 [\varphi] \times 3 [M] = 90$ problem instances in total.

We choose three agents as basic setup (except for Experiment 4), since, on the

one hand, this allows testing the simple majority building block, and, on the other hand, the computational effort to obtain a Pareto frontier (discrete Pareto set) is still feasible. The agents were assigned to the jobs in turn.

Besides the individual tardiness cost, the energy cost has to be allocated among the agents. We used an allocation scheme (except for Experiment 5) where the agents' contribution depends on their relative resource needs, i.e., $EC_i = \frac{\sum_{j \in \mathcal{J}: a_j = i} r_j * p_j(\pi)}{\sum_{j \in \mathcal{J}} r_j * p_j(\pi)} * EC$.

5.1.2 Performance Measurement

For measuring the Pareto performance of a contract c , we normalized the results with the best solution C_i^{best} obtained for agent i and computed the Euclidean distance to the nearest point in the Pareto set (see Baarslag et al. (2013)):

$$Pareto(c) = \min_{c^p \in ParetoSet} \sqrt{\sum_{i \in \mathcal{I}} \left(\frac{C_i(c) - C_i(c^p)}{C_i^{best}} \right)^2} \quad (8)$$

To obtain a high-quality approximation of the Pareto set, we executed massive multi-objective criteria optimization runs using NSGA-II (see Deb et al., 2002), a multi-objective simulated annealing approach (cf. Ulungu et al., 1999), a multi-objective evolutionary algorithm approach (cf. Zitzler and Thiele, 1999) as well as multi-objective variants of iterated local search (cf. Geiger, 2007) and Late-Acceptance Hill Climbing (cf. Burke and Bykov, 2012). In these runs, we gathered a total of more than 2.4 million outcomes for 90 problem instances of which approx. 120,000 were eventually nondominated (i.e., on average more than 1,300 nondominated solutions per problem instance). Furthermore, we included nondominated results of the negotiation simulations in the Pareto set.

To measure the social cost of a solution contract c , we computed the ratio of the resulting social cost and the best found solution:

$$SC(c) = \frac{\sum_{i \in \mathcal{I}} C_i(c)}{\sum_{i \in \mathcal{I}} C_i(c^*)} \quad (9)$$

with $c^* = \arg \min_c \sum_{i \in \mathcal{I}} C_i(c)$. This concept is also known as the price of anarchy that measures to what extent the self-interest of the agents deteriorates the solu-

tions compared to the imaginary outcomes determined centrally by a benevolent dictator with full information (Roughgarden and Tardos, 2007). Similarly to the Pareto frontier, we approximately determined the social cost minimum (i.e., the social welfare maximum) using different metaheuristics – namely, Simulated Annealing (see Kirkpatrick et al., 1983), Iterated Local Search (see Lourenço et al., 2003), Late-Acceptance Hill Climbing (see Burke and Bykov, 2012), and a Memetic Algorithm (see Moscato, 1989) – as well as the results of the Pareto search and the negotiations themselves.

Using the Pareto distance and the social cost, we derived several figures for evaluation: (1) Avg – the mean figure for all problem instances, (2) CoV – coefficient of variation, i.e., standard deviation divided by the mean, (3) OnFR – on front ratio, i.e., the percentage number of nondominated solutions obtained by a protocol (used with Pareto distance only), (4) Best – the smallest social cost, (5–7) $M = \{5, 10, 20\}$ – the respective Pareto and social cost results grouped by the number of machines M (and, thereby, number of jobs: $J = M * 60$).

We tested the results for statistical significance by the Wilcoxon-Mann-Whitney test (WMW-test, Mann and Whitney, 1947), using a single-sided hypothesis. That is, we tested whether the mean of one configuration is statistically significantly smaller than the mean of another configuration (significance level: p-value $\leq 5\%$).

5.2 Experiments

In this section, we present the findings of our computational experiments. In total, we simulated more than 13,000 negotiations grouped in five experiments. The protocol and its configuration is encoded by a character string: The first character indicates the protocol, i.e., **S** for MNP-SA and **G** for MNP-GA. The remaining characters determine the used building blocks: **N** – without quotas, **Q** – with quotas, **P** – with prenegotiation, **A** – proposals partly made by agents, **3** – three-valued logic, **M** – majority rule instead of unanimity, **I** – random immigration, and **E** – elitism. For instance, SQP means that the protocol MNP-SA was used, a quota rule was included, and the initial contract was settled by a prenegotiation. Finally, BM encodes the benchmark protocol.

5.2.1 Experiment 1: Building Blocks

Number of Proposals (ρ)	20	Number of Rounds (R)	500k
Number of Agents (I)	3	Energy Cost Distribution	consumption-based

The first experiment’s purpose is to identify successful building blocks. We have tested several policies on their own as well as in combination. In this experiment, we assumed that there are twenty contract proposals per negotiation round.

Table 1 shows the computational results for the MNP-SA. As the data shows, none of configurations of the MNP-SA performs satisfactorily without the quota rule. The benchmark protocol achieves results similar to the basic MNP-SA without quotas which is not surprising because both are closely related. Furthermore, the majority extension leads to the worst outcomes and is clearly not suited in this setting. In contrast, the quota rule yields significantly better outcomes which are close to the Pareto frontier and the social cost minimum. Looking at the single building blocks, especially the three-valued logic extension leads to Pareto as well as social cost improvements. Furthermore, agent-based proposals can decrease the variation and yield more stable results. The prenegotiation seems to achieve slightly more stable results but, basically, it is comparable to the SQ configuration. From the WMW-test, we obtain a statistically significant superiority of SQ3 compared to SQ, SQA, and SQP in terms of Pareto performance and social cost. Furthermore, SQA statistically outperforms SQ in terms of social cost. Considering combinations of additional building blocks, SQA3 leads to even more stable results and SQPA3 reaches relatively good solution but cannot outperform SQ3 with statistical significance.

Similarly to table 1, table 2 shows the results for the MNP-GA. In the evolutionary protocol, quotas seem to be counterproductive. Apart from those configurations, the MNP-GA obtains significantly better outcomes than the benchmark protocol; however, the results are not as good as those of the MNP-SA. Generally, the agent-based proposal is a successful policy again. Configurations including this block show a statistical dominance against the respective configuration without agent-based proposal for both criteria (except for the Pareto performance of GNPA vs. GNP). Moreover, adding elitism does not accomplish a statistical improvement in any case. The random immigration building block leads to mixed findings. On the one hand, the agents are worse off in most cases, but, on the other

Table 1: Building Blocks of the MNP-SA

	BM	SN	SNM	SNA	SNAM	SQ	SQM	SQ3	SQA	SQA3	SNP	SQP	SQPA3
Pareto Distance													
Avg	22.0	22.1	90.8	21.0	87.7	6.5	93.2	5.2	6.2	6.2	21.3	6.7	5.3
CoV	41.6	41.4	36.7	38.2	34.4	81.1	35.8	81.5	70.2	68.0	43.7	77.4	63.0
OnFR	0.0	0.0	0.0	0.0	0.0	17.8	0.0	20.0	15.6	7.8	0.0	14.4	13.3
$M=5$	19.2	17.2	74.2	19.4	75.3	3.9	82.1	3.7	4.0	5.4	18.2	4.4	3.9
$M=10$	19.7	20.9	94.7	18.7	88.6	5.2	94.3	3.8	5.2	4.7	18.8	5.1	4.8
$M=20$	27.1	28.2	103.5	24.9	99.2	10.5	103.3	8.2	9.3	8.6	26.9	10.7	7.2
Social Cost													
Avg	111.8	112.1	141.2	111.4	140.5	104.5	142.7	104.1	104.5	104.4	111.7	104.6	104.3
CoV	3.1	3.4	9.6	3.1	9.6	2.0	9.8	1.5	1.5	1.4	3.3	1.9	1.6
Best	104.3	103.5	113.6	104.3	112.5	101.5	116.6	101.2	101.6	102.0	104.3	101.3	101.0
$M=5$	110.3	110.2	132.9	110.6	133.5	103.2	136.5	103.4	103.6	103.7	110.0	103.3	103.4
$M=10$	111.0	111.6	143.1	110.4	142.1	104.0	142.9	103.7	104.1	104.0	111.1	104.2	104.1
$M=20$	114.2	114.4	147.7	113.3	146.1	106.5	148.7	105.1	105.7	105.3	114.1	106.3	105.5

hand, GNAI performs best within the configurations of the MNP-GA. Although GNAI has no statistical dominance in terms of Pareto performance compared to GNA, it shows significantly lower social cost and reaches the Pareto frontier more often – however, at the cost of having a higher coefficient of variation.

Table 2: Building Blocks of the MNP-GA

	BM	GN	GNI	GNE	GNEI	GNA	GNAI	GNAE	GNAEI	GQ	GNP	GNPA	GQPAEI
Pareto Distance													
Avg	22.0	10.7	11.6	11.2	14.9	8.8	8.4	9.6	9.7	55.1	10.2	9.8	44.2
CoV	41.6	48.3	52.8	49.4	45.2	58.0	67.5	56.2	56.4	39.9	60.2	52.7	40.5
OnFR	0.0	3.3	2.2	3.3	0.0	10.0	13.3	5.6	6.7	0.0	5.6	4.4	0.0
$M=5$	19.2	10.5	10.6	10.7	12.9	9.2	8.0	8.7	9.1	46.3	9.0	9.4	38.8
$M=10$	19.7	9.7	10.6	10.1	13.7	7.7	6.7	8.6	8.6	54.3	9.0	8.1	43.6
$M=20$	27.1	11.8	13.6	12.8	18.0	9.4	10.5	11.6	11.5	64.6	12.6	11.9	50.3
Social Cost													
Avg	111.8	107.0	107.2	107.0	108.6	106.3	106.1	106.4	106.4	126.3	106.9	106.4	121.6
CoV	3.1	2.2	2.3	2.2	2.8	2.1	2.2	2.0	2.1	7.7	2.1	1.8	6.2
Best	104.3	102.1	101.8	101.8	102.8	101.8	101.4	101.9	101.2	107.8	101.9	101.8	107.6
$M=5$	110.3	106.6	106.4	106.6	107.4	105.7	105.5	105.9	105.7	122.7	106.3	105.8	118.2
$M=10$	111.0	106.5	107.0	106.6	108.1	106.2	106.0	106.3	106.1	125.0	106.4	106.0	121.5
$M=20$	114.2	108.0	108.3	107.9	110.4	107.0	106.9	107.2	107.5	131.0	108.0	107.3	125.0

Generally, the protocols can handle problems with 5–10 machines and 300–600 jobs relatively well. However, the large instances involving 20 machines and 1,200 jobs are harder to solve and achieve substantially worse results. Moreover, the procedures partly found outcomes that have a very small price of anarchy: the best found solutions with regard to social cost have been 1.0% (MNP-SA) and 1.2% (MNP-GA) above the centrally approximated optimum.

Exemplary negotiation histories for selected protocol configurations with two

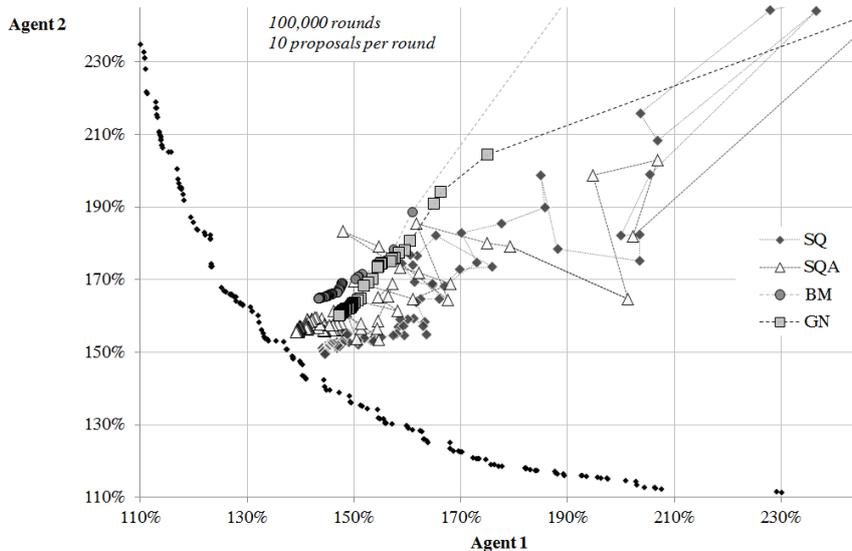


Figure 1: Exemplary negotiation histories with Pareto frontier for two agents (truncated)

agents are illustrated in figure 1. In this figure, one data point is plotted after every thousand rounds. The basic MNP-SA (SQ) shows a lot of deteriorations and moves in circles towards the Pareto frontier. The longer the negotiation continues the less deteriorations are accepted (due to the decreasing quota). Largely, the same holds true for the same protocol including agent-based proposals (SQA); however, the agent-based proposals lead to a more straightforward movement towards the frontier. The benchmark protocol (BM) makes giant improving leaps at the beginning but achieves just very small gains after a few rounds – possibly due to conflicts of interest. The basic MNP-GA (GN) moves rather directly towards the frontier as well but achieves more acceptances during the negotiation.

For the remainder of the paper, we focus on the following subset of configurations that consists of the basic protocols as well as successful strategy configurations: SQ, SQ3, SQA, SQA3, GN, and GNAI.

5.2.2 Experiment 2: Number of Proposals

Number of Proposals (ρ)	10–40	Number of Rounds (R)	250k–1M
Number of Agents (I)	3	Energy Cost Distribution	consumption-based

The second experiment tests different parameterizations of the number of proposals

and the number of rounds in comparison with the chosen parameters of Experiment 1. In this experiment, we use $\rho = 10$ proposals and $R = 1\text{M}$ rounds as well as $\rho = 40$ proposals and $R = 250\text{k}$ rounds – both yield the very same number of total proposals like Experiment 1, i.e., 10M proposals throughout the negotiation.

The findings of Experiment 2 are shown in table 3. Having more proposals per negotiation round, but less rounds in total, is not advantageous for the MNP-SA – the results become worse with this parameterization. The same comes true for the benchmark protocol. However, the performance of the MNP-GA stays unaffected compared to Experiment 1. The case of having less proposals per rounds but more rounds in total shows opposed findings: the MNP-SA as well as the benchmark protocol accomplish much better outcomes, whereas the MNP-GA suffers from minor deteriorations. Since the MNP-SA achieves significant improvements and reaches very good solutions, we use the parameterization of $\rho = 10$ and $R = 1,000,000$ for the remainder of this paper.

Table 3: Different parameterizations of ρ and T

	$\rho = 40, T = 250,000$							$\rho = 10, T = 1,000,000$						
	BM	SQ	SQ3	SQA	SQA3	GN	GNAI	BM	SQ	SQ3	SQA	SQA3	GN	GNAI
	Pareto Distance							Pareto Distance						
Avg	24.4	12.1	10.0	10.0	9.3	11.7	8.9	18.3	4.5	3.2	3.9	3.6	11.2	10.6
CoV	34.8	66.8	65.2	60.6	57.7	53.6	63.4	41.0	82.5	98.4	105.1	92.1	41.9	58.4
OnFR	0.0	3.3	3.3	3.3	3.3	3.3	13.3	0.0	21.1	34.4	32.2	31.1	0.0	4.4
$M=5$	19.8	6.0	6.0	6.8	6.1	10.5	7.5	17.5	4.3	2.5	4.1	2.6	10.2	9.9
$M=10$	23.4	9.4	8.5	8.0	7.6	10.0	7.6	16.3	3.2	2.2	2.9	3.2	10.3	9.6
$M=20$	30.1	20.9	15.4	15.1	14.3	14.6	11.5	20.9	6.0	4.8	4.7	5.0	13.3	12.4
	Social Cost							Social Cost						
Avg	113.5	107.0	106.2	106.4	105.9	107.4	106.3	110.2	103.7	103.2	103.6	103.4	107.3	107.0
CoV	3.4	3.2	2.8	2.5	2.2	2.6	2.4	2.7	1.2	1.0	1.3	1.2	2.0	2.2
Best	107.0	102.4	102.0	101.2	101.9	101.9	101.2	104.3	101.4	101.1	100.7	101.2	102.5	102.4
$M=5$	111.1	104.3	104.1	104.4	104.3	106.5	105.4	109.4	103.1	102.9	103.3	103.0	106.5	106.0
$M=10$	113.4	106.1	105.3	105.7	105.1	107.0	106.1	109.5	103.5	102.9	103.3	103.3	107.0	106.8
$M=20$	116.0	110.7	109.2	109.0	108.4	108.9	107.3	111.7	104.5	103.7	104.1	103.9	108.2	108.2

5.2.3 Experiment 3: Number of Negotiation Rounds

Number of Proposals (ρ)	10	Number of Rounds (R)	10k–5M
Number of Agents (I)	3	Energy Cost Distribution	consumption-based

In the third experiment, we analyze the impact of negotiations with more or less iterations on the performance of the protocols.

Figure 2 shows the Pareto performance subject to the number of negotiation

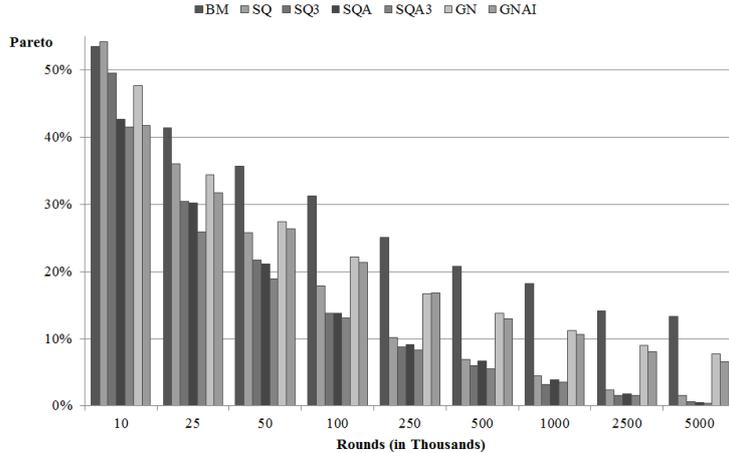


Figure 2: Pareto performance depending on negotiation rounds

rounds. Since the findings are basically the same for the social cost performance, we omitted a separate presentation of those results.

If there are very few rounds only (10k), the basic protocol version of the MNP-SA (SQ) performs comparably to the benchmark procedure. Furthermore, SQ3 and GN achieve similar outcomes as well, but are outperformed by the agent-based proposal configurations (SQA, SQA3, and GNAI) which also perform comparably among themselves. All protocols and configurations attain better results with more negotiation rounds; however, the marginal improvements seem to be decreasing. The MNP-SA benefits most from more negotiation rounds, since the performance significantly increases with more rounds. Given there are just very few rounds, the MNP-GA can mostly keep up with the MNP-SA. However, the differences between the protocols become more and more evident with longer negotiation time. Nevertheless, the MNP-GA still beats the benchmark protocol considerably. After 5M negotiation rounds, the advanced MNP-SA configurations mostly reached the Pareto frontier: SQ3, SQA, and SQA3 have a mean distance to the Pareto frontier between 0.6% – 0.4% and result in a nondominated contract in 85.6% – 88.9% of the simulations (social cost: 102.0% – 101.9%). The MNP-GA and BM come off far behind (Pareto: 7.7% – 6.6% (MNP-GA) / 13.3% (BM); on front ratio: 11.1% – 23.3% / 4.4%; social cost: 105.5% – 105.3% / 108.4%).

5.2.4 Experiment 4: Number of Agents

Number of Proposals (ρ)	10	Number of Rounds (R)	1M
Number of Agents (I)	3–19	Energy Cost Distribution	consumption-based

The fourth experiment analyzes the scalability of the protocols regarding the number of agents. Since the approximation of the Pareto frontier becomes very hard to calculate with more agents, we make use of the social cost performance here.

Figure 3 depicts the development of the social cost subject to the number of agents (for 3, 5, ..., 19 agents). Unsurprisingly, the social cost increases when there are more agents in place, although the total number of jobs to be scheduled stays the same. This results from the fact that more agents are more difficult to coordinate and there is a larger conflict of interests due to disparate preferences. Regarding the MNP-GA, GN performs better than the BM initially. With an increasing number of agents, the spread between BM and GN diminishes – BM has even lower social cost than GN for 19 agents. Nevertheless, the second tested configuration, GNAI, maintains its advantage compared to the BM. In the case of the MNP-SA, the deterioration turns out to be relatively small, and the social cost appears to go up not as vigorously as the MNP-GA. With 19 agents, each pursuing his or her own cost objective depending on his or her assigned jobs, SQA3 still attains a solution that on average just suffers from a mark-up of 12.8% on the social cost which is a relatively low price of anarchy. In the best case, the MNP-SA found a contract involving 19 parties that had a social cost of 103.1% only.

5.2.5 Experiment 5: Energy Cost Allocation

Number of Proposals (ρ)	10	Number of Rounds (R)	1M
Number of Agents (I)	3	Energy Cost Distribution	negotiated

In the final and fifth experiment, we refrained from the scheme that the energy cost are distributed based on the respective jobs' resource consumption. Instead of such a ratio, we integrated the energy cost allocation in the negotiation making it a further negotiation issue. For this purpose, we have added a payment vector to the contract determining which shares of the cost have to be covered by which agents, which is mutated by redistributing a small percentage from one randomly chosen agent to another. Table 4 shows the results. BM does not yield any satisfying outcomes and suffers from a severe decline compared to the consumption-based

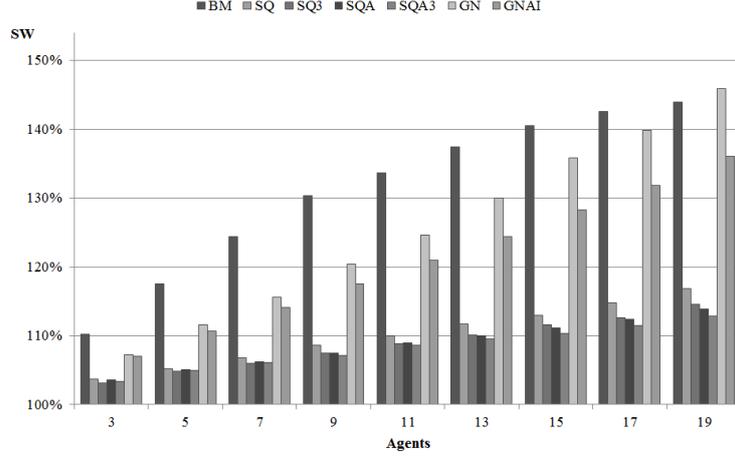


Figure 3: Impact of different number of agents on social costs

allocation. Moreover, the MNP-GA is also significantly worse off in terms Pareto and social cost performance according to the WMW-test (except for the Pareto performance of GN which was insignificant). Nevertheless, in contrast, MNP-SA is able to obtain almost exclusively Pareto optimal outcomes. 95.6% of the negotiations (i.e., 86 out of 90 problem instances) reached a nondominated solution. Considering the 30 smaller problem instances with 5 machines and 300 jobs, SQA3 missed a nondominated solution just a single time. The Pareto performance was between 2.8% and 0.6%; on the other hand, the price of anarchy (social cost) is substantially higher compared to Experiment 2 with a very high significance.

Table 4: Results with energy cost as a negotiation issue

	BM	SQ	SQ3	SQA	SQA3	GN	GNAI
Pareto Distance							
Avg	91.8	1.7	2.8	0.6	1.4	11.9	12.7
CoV	40.2	376.4	248.0	561.3	364.9	77.4	65.1
OnFR	0.0	92.2	84.4	95.6	91.1	21.1	10.0
$M=5$	74.8	0.5	0.8	0.5	0.1	8.2	10.9
$M=10$	83.2	1.5	2.9	0.4	1.7	10.2	10.7
$M=20$	117.5	3.0	4.8	1.0	2.4	17.2	16.7
Social Cost							
Avg	141.3	113.3	113.6	112.2	112.1	108.8	108.4
CoV	9.2	3.8	3.7	3.5	3.6	2.8	2.6
Best	115.7	105.6	105.6	104.7	104.6	102.5	103.0
$M=5$	130.5	109.3	109.4	108.2	108.0	107.2	106.8
$M=10$	140.1	113.2	113.8	112.3	112.3	108.3	108.1
$M=20$	153.3	117.5	117.6	116.1	115.9	110.8	110.3

Until now, being close at the Pareto frontier basically meant having a little

social cost as well. To analyze this in more detail, table 5 compares the Gini coefficients (GC) for consumption-based energy cost allocation and negotiated energy cost allocation. The GC is a measurement of inequality, i.e., with a GC of 0, every agent has the very same cost and, with a GC of 1, a single agent covers the whole cost. As the data shows, the consumption-based allocation leads to relatively homogeneous outcomes. The very low GC partly results from the fact that there is a base cost, i.e., a large part of the cost cannot be prevented. In the case of integrating the allocation in the negotiation, the MNP-GA as well as the BM still achieve comparable results. However, the MNP-SA, which also attains very good results, suffers from a huge disparity. Consequently, the improvements are just prosperous for individual agents and are borne by others.

Table 5: Gini Coefficients

	BM	SQ	SQ3	SQA	SQA3	GN	GNAI
Not Neg.	3.9%	3.3%	3.2%	3.4%	3.3%	3.5%	3.5%
Negotiated	3.4%	26.7%	24.4%	24.1%	24.4%	3.7%	3.1%

5.3 Discussion and Limitations

Outlining the findings of the previous section, the protocols are able to find high quality solutions although they do not make use of aggregated information and are subject to strategic behavior. In the presented experiments, the agents evaluate a tiny fraction of the contract space of this complex problem and provide specific acceptance and rejection signals only. Nevertheless, we found that the MNP-SA may deliver first-class solution quality, whereas the MNP-GA outperforms the benchmark protocol significantly but cannot keep up with the MNP-SA. This result complies with common experiences for traditional metaheuristics, since evolutionary algorithms that are hybridized with local search procedures mostly find better solutions than pure versions. Beyond the introduction of quotas, we identified agent-based proposal and three-valued logic as successful building blocks for the MNP-SA. The majority acceptance extension deteriorates the results significantly, whereas the prenegotiation does not seem to have a substantial impact. Nevertheless, the latter might find fairer outcomes which we did not analyze in detail. Regarding the MNP-GA, only proposals by agents could improve the out-

comes compared to the basic protocol. As a small drawback, agent-based proposals require more communication as well as computation effort on the side of the agents.

Furthermore, the experiments indicate that the number of proposals and rounds can have a substantial impact on the eventual quality of the contracts. If there is sufficient computation time, the use of MNP-SA often reaches the Pareto frontier, as shown for three agents. Moreover, a larger number of participating agents leads to an increased complexity of the problem and to a higher price of anarchy. As the last experiment shows, the integration of the energy cost allocation into the negotiation results in a very high ratio of nondominated outcomes.

Besides these beneficial findings, there are also some limitations of the computational study. First of all, a large part of the energy cost are unavoidable and there are also considerable tardiness costs. While this does not affect the assessment concerning achieving Pareto-efficiency or not, it scales the absolute figures of the outcomes and, hence, reduces the relative difference between the results (although we used the individual or global best outcomes for normalization). The second limitation addresses the results of Experiment 5, in which the MNP-SA found a great number of nondominated contracts. Although we made huge efforts for determining the Pareto frontier (more than 2.4M solutions obtained with five different multi-criteria metaheuristics; see section 5.1.2), Experiment 5 draws on slightly modified assumptions which can affect the results notably. The centralized procedures as well as the other negotiation simulations suppose consumption-based energy allocation in contrast to Experiment 5. This experiment can achieve every allocation of energy cost which is equivalent to allowing for side payments among the negotiation parties. By means of side payments, the agents can obtain solutions that are not feasible for the consumption-based energy allocation. That is why the comparability and, thus, the interpretation of the Pareto results of Experiment 5 have to be done with caution. Nonetheless, since we compare different protocols and configurations, the results still have validity, because all procedures draw on the very same assumptions within this experiment. However, the dominance of the solutions from Experiment 5 can lead to greater gaps between the discrete Pareto points and result in worse distance figures for Experiment 1 – 4.

Finally, concerning the comparison of negotiation approaches with multi-criteria procedures, a negotiation problem may be considered as a multi-criteria problem

with additional fundamental difficulties. That is, negotiation scenarios suffer from strategic interactions of autonomous self-interested agents and lack of revealed truthful information. Thus, although there is a close relation between an agent and an objective, there is no way to centrally dictate the system. A central authority can just specify reasonable and verifiable rules of the negotiation protocol. This is why decentralized decision problems (such as the considered multi-agent multi-machine scheduling problem) are different from classical multi-criteria problems. Nonetheless, one may still compare the performance of negotiation protocols, with results that are biased by individual interests, with centralized multi-criteria procedures (see section 5.1.2). We found that the obtained solutions of the negotiation procedures partly even dominate results from the centralized approaches. By using centralized approaches, we generated more than 2.4M solutions of which 119,087 were nondominated. In the negotiation simulations, we obtained 8,190 results for the two protocols if we omit Experiment 4 and 5.¹ Despite the mentioned strategic and informational issues, the negotiation results dominate 4,683 centralized results which were nondominated so far. This shows that negotiations with appropriate rules and mechanism design may keep up with centralized approaches despite self-interested agents and information asymmetry.

6 Conclusion and Future Work

This paper studies the scheduling of competing jobs with due dates on multiple heterogeneous machines by multiple agents. As a feature, the model incorporates not only tardiness cost but also machine operating cost (or energy cost). The operating cost is subject to a time-dependent tariff and the machine's utilization. Furthermore, we argue that tardiness as well as energy cost functions may be nonlinear. This already complex problem is exacerbated by selfish strategic considerations of the involved agents and their unwillingness to share relevant information. To overcome the latter, we present and evaluate two generic negotiation protocols that are partly inspired by metaheuristic concepts and are designed to achieve beneficial outcomes in terms of Pareto efficiency and social cost.

¹Experiment 4 studies more than three agents and Experiment 5 draws on different assumptions. The latter dominates 11,436 solutions.

For evaluation purposes, we present a computational study with problems that include 300–1,200 jobs, 5–20 machines, and 3–19 agents. The results of the computational experiments indicate that the proposed methods are capable to find (near) Pareto-efficient outcomes that also show a good performance in terms of social cost. Although the protocols are mainly generic, parameters such as the number of proposals or negotiation rounds have a significant impact on the eventual quality of the contracts. The data shows that the location-based protocol employing quotas is superior to the protocol that draws on evolutionary concepts. Nevertheless, the evolutionary protocol’s performance still outperforms the benchmark protocol that involves a reduced set of rules. Even when there are a large number of agents, the measured price of anarchy is rather small and the outcomes are still relatively close to the best centrally determined solutions.

Future work will be to further elaborate the protocols by developing and evaluating additional building blocks. Moreover, the validity of the findings from this study should be assessed by considering other kinds of applications. Concerning the presented kind of scheduling problem, future work may also incorporate the empirical confirmation of the hypotheses of nonlinear cost functions and analyze the performance subject to the parameters of the problem instances. Another interesting aspect is the consideration of an unbalanced job allocation among the agents or mixed objective functions for the agents. Finally, beyond efficiency considerations, fairness among the agents will be a criterion in future work.

References

- Agnetis, A., Mirchandani, P. B., Pacciarelli, D., & Pacifici, A. (2004). Scheduling problems with two competing agents. *Operations Research*, 52(2), 229–242.
- Agnetis, A., Pacciarelli, D., & Pacifici, A. (2007a). Combinatorial models for multi-agent scheduling problems. In Levner, E. (Ed.), *Multiprocessor Scheduling: Theory and Applications* (pp. 21–46). Vienna, Austria: I-Tech Education and Publishing.
- Agnetis, A., Pacciarelli, D., & Pacifici, A. (2007b). Multi-agent single machine scheduling. *Annals of Operations Research*, 150(1), 3–15.
- Agnetis, A., Pascale, G. D., & Pranzo, M. (2009). Computing the Nash solution for

- scheduling bargaining problems. *International Journal of Operations Research*, 6(1), 54–69.
- Akturk, M., & Ozdemir, D. (2000). An exact approach to minimizing total weighted tardiness with release dates. *IIE Transactions*, 32(11), 1091–1101.
- Baarslag, T., Fujita, K., Gerding, E., Hindriks, K., Ito, T., Jennings, N. R., Jonker, C. M., Kraus, S., Lin, R., Robu, V., & Williams, C. R. (2013). Evaluating practical negotiating agents: results and analysis of the 2011 international competition. *Artificial Intelligence*, 198, 73–103.
- Baker, K., & Smith, J. (2003). A multiple-criterion model for machine scheduling. *Journal of Scheduling*, 6, 7–16.
- Balasubramanian, H., Fowler, J., Keha, A., & Pfund, M. (2009). Scheduling interfering job sets on parallel machines. *European Journal of Operational Research*, 199(1), 55–67.
- Blickle, T. (2000). Tournament selection. In Bäck, T., Fogel, D.B., & Michalewicz, Z. (Eds.), *Evolutionary Computation 1* (pp. 181–186). New York, NY, USA: Taylor & Francis.
- Bodenstein, C., Schryen, G., & Neumann, D. (2012). Energy-aware workload management models for operation cost reduction in data centers. *European Journal of Operational Research*, 222(1), 157–167.
- Braithwait, S. D., & Hansen, D. G. (2011). How large commercial and industrial customers respond to dynamic pricing – the California experience. In Sioshansi, F. (Ed.), *Smart Grid: Integrating Renewable, Distributed & Efficient Energy* (pp. 289–316). Waltham, MA, USA: Academic Press.
- Brandt, T., & Bodenstein, C. (2012). Evaluating scheduling methods for energy cost reduction in a heterogeneous data center environment. In *Proceedings of 20th European Conference on Information Systems (ECIS 2012)*.
- Briand, C., Agnetis, A., & Billaut, J. (2012). The multiagent project scheduling problem: complexity of finding a minimum-makespan Nash equilibrium. In Demeulemeester, E., & Herroelen, W. (Eds.), In *Proceedings of the 13th International Conference on Project Management and Scheduling* (pp. 106–109).
- Burke, E., & Bykov, Y. (2012). The late acceptance hill-climbing heuristic. *Technical Report CSM-192*. University of Stirling, UK: Computing Science and Mathematics.
- Cerný, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41–51.

- Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., & Gautam, N. (2005). Managing server energy and operational costs in hosting centers. In Eager, D. L., Williamson, C. L., Borst, S. C., & Lui, J. C. S. (Eds.), *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (pp. 303–314).
- Cheng, T., Chung, Y.-H., Liao, S.-C., & 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., Ng, C., & Yuan, J. (2006). Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. *Theoretical Computer Science*, 362(1-3), 273–281.
- Cheng, T., Ng, C., & Yuan, J. (2008). Multi-agent scheduling on a single machine with max-form criteria. *European Journal of Operational Research*, 188(2), 603–609.
- Cobb, H., & Grefenstette, J. (1993). Genetic algorithms for tracking changing environments. In *Proceedings of the Fifth International Conference on Genetic Algorithms*.
- Cook, A., Tanner, G., & Lawes, A. (2012). The hidden cost of airline unpunctuality. *Journal of Transport Economics and Policy*, 46(2), 157–173.
- Dawande, M., Geismar, H., Hall, N., & Sriskandarajah, C. (2006). Supply chain scheduling: distribution systems. *Production and Operations Management*, 15(2), 243–261.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Dudek, G., & Stadtler, H. (2005). Negotiation-based collaborative planning between supply chains partners. *European Journal of Operational Research*, 163(3), 668–687.
- Fink, A. (2004). Supply chain coordination by means of automated negotiations. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS-37)*.
- Fink, A. (2006). Supply chain coordination by means of automated negotiations between autonomous agents. In Chaib-draa, B., & Müller, J. (Eds.), *Multiagent based Supply Chain Management* (Studies in Computational Intelligence Vol. 28, pp. 351–372). Berlin / Heidelberg, Germany: Springer.
- Geiger, M. J. (2007). On operators and search space topology in multi-objective flow shop scheduling. *European Journal of Operational Research*, 181(1), 195–206.

- Geiger, M. J. (2010). On heuristic search for the single machine total weighted tardiness problem – Some theoretical insights and their empirical verification. *European Journal of Operational Research*, 207(3), 1235–1243.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley Professional.
- Heath, T., Diniz, B., Carrera, E. V., Jr., W. M., & Bianchini, R. (2005). Energy conservation in heterogeneous server clusters. In *Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (pp. 186–195).
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press.
- Homberger, J. (2012). A (μ, λ) -coordination mechanism for agent-based multi-project scheduling. *OR Spectrum*, 34(1), 107–132.
- Huynh Tuong, N., Soukhal, A., & Billaut, J.-C. (2012). Single-machine multi-agent scheduling problems with a global objective function. *Journal of Scheduling*, 15(3), 311–321.
- Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2), 199–215.
- Kersten, G. E., & Mallory, G. R. (1990). Supporting problem representations in decisions with strategic interactions. *European Journal of Operational Research*, 46(2), 200–215.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
- Klein, M., Faratin, P., Sayama, H., & Bar-Yam, Y. (2003). Negotiating complex contracts. *Group Decision and Negotiation*, 12(2), 111–125.
- Klein, M., Faratin, P., Sayama, H., & Bar-Yam, Y. (2007). Negotiating complex contracts. *MIT Sloan School of Management Working Paper No. 4196-01*. Cambridge, MA, USA: Massachusetts Institute of Technology.
- Kraus, S. (2001). Automated negotiation and decision making in multiagent environments. In Luck, M., Marik, V., Stepankova, O., & Trappl, R. (Eds.), *Multi-agents Systems and Application* (pp. 150–172). New York, NY, USA: Springer.
- Kraus, S., Wilkenfeld, J., & Zlotkin, G. (1995). Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2), 297–345.

- Lang, F., & Fink, A. (2012a). A quota-based multi-agent negotiation protocol for complex contracts. In Filipe, J., & Fred, A. L. N. (Eds.), *ICAART 2012 – Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (Volume 2, pp. 113–118).
- Lang, F., & Fink, A. (2012b). Collaborative job processing on a single machine: a multi-agent weighted tardiness problem. In Conitzer, V., Winikoff, M., Padgham, L., & van der Hoek, W. (Eds.), *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (Volume 3, pp. 1417–1418).
- Lang, F., & Fink, A. (2012c). Collaborative single and parallel machine scheduling by autonomous agents. In Smari, W. W., & Fox, G. C. (Eds.), *Proceedings of the 2012 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 76–83).
- Lang, F., & Fink, A. (2013). Learning from the metaheuristics: protocols for automated negotiations. *Submitted to Group Decision and Negotiation (revision in review)*.
- Lenstra, J., Kan, A., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343–362.
- Leung, J. Y. T., Pinedo, M., & Wan, G. (2009). Competitive two-agent scheduling and its applications. *Operations Research*, 58(2), 458–469.
- Liu, H., Keppens, J., & Luck, M. (2012). Towards decentralised agent-based inter-company scheduling. In Frommberger, L., Schill, K., & Scholz-Reiter, B. (Eds.), *Proceedings of the ECAI 2012 Workshop on Artificial Intelligence and Logistics (AILog-2012)* (pp. 37–42).
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In Glover, F. W., & Kochenberger, G. A. (Eds.), *Handbook of Metaheuristics* (pp. 320–353). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Mann, H., & 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.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & 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., & 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.

- Mor, B., & Mosheiov, G. (2010). Scheduling problems with two competing agents to minimize minmax and minsum earliness measures. *European Journal of Operational Research*, 206(3), 540–546.
- Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program Report 826*. Pasadena, CA, USA: California Institute of Technology.
- Myerson, R. (1981). Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica*, 49(4), 883–897.
- Ramacher, R., & Mönch, L. (2011). Hybridized genetic algorithms to schedule interfering job sets on a single machine. In Kendall, G., Lei, L., & Pinedo, M. (Eds.), *Proceedings of the 5th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2011)* (pp. 185–194).
- Roughgarden, T., & Tardos, E. (2007). Introduction to the inefficiency of equilibria. In Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. V. (Eds.), *Algorithmic Game Theory* (pp. 443–458). Cambridge, UK: Cambridge University Press.
- Standard Performance Evaluation Corporation (2008). SPECpower ssj2008 benchmark.
- Syswerda, G. (1991). Schedule optimization using genetic algorithms. In Davis, L. D. (Ed.), *Handbook Of Genetic Algorithms* (pp. 332–349). New York, NY, USA: Van Nostrand Reinhold.
- Tung, H.-W., & Lin, R. J. (2005). Automated contract negotiation using a mediation service. In *Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05)* (pp. 374–377).
- Ulungu, E., Teghem, J., Fortemps, P., & Tuytens, D. (1999). MOSA method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8(4), 221–236.
- Varian, H. R. (2010). *Intermediate Microeconomics: A Modern Approach* (8th edition). New York, NY, USA: W. W. Norton & Company.
- Vergnano, A., Thorstensson, C., Lennartson, B., Falkman, P., Pellicciari, M., Leali, F., & Biller, S. (2012). Modeling and optimization of energy consumption in cooperative multi-robot systems. *IEEE Transactions on Automation Science and Engineering*, 9(2), 423–428.
- Vetschera, R. (2013). Negotiation processes: an integrated perspective. *EURO Journal on Decision Processes*, 1(1-2), 135–164.
- von Neumann, J., & Morgenstern, O. (1944). *Theory Of Games And Economic Behavior*. Princeton, New Jersey, USA: Princeton University Press.

Zhang, L., Song, H., Chen, X., & Hong, L. (2011). A simultaneous multi-issue negotiation through autonomous agents. *European Journal of Operational Research*, 210(1), 95–105.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271.

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>