Towards fully Decentralized Multi-Objective Energy Scheduling

Jörg Bremer  
Department of Computing Science  
Carl von Ossietzky University  
Oldenburg, Germany  
joerg.bremer@uni-oldenburg.de

Sebastian Lehnhoff  
Department of Computing Science  
Carl von Ossietzky University  
Oldenburg, Germany  
sebastian.lehnhoff@uni-oldenburg.de

Abstract—Future demand for managing a huge number of individually operating small and often volatile energy resources within the smart grid is preponderantly answered by involving decentralized orchestration methods for planning and scheduling. Many planning and scheduling problems are of a multi-objective nature. For the single-objective case – e.g., predictive scheduling with the goal of jointly resembling a wanted target schedule – fully decentralized algorithms with self-organizing agents exist. We extend this paradigm towards fully decentralized agent-based multi-objective scheduling for energy resources e.g., in virtual power plants for which special local constraint-handling techniques are needed. We integrate algorithmic elements from the well-known S-metric selection evolutionary multi-objective algorithm into a gossiping-based combinatorial optimization heuristic that works with agents for the single-objective case and derive a number of challenges that have to be solved for fully decentralized multi-objective optimization. We present a first solution approach based on the combinatorial optimization heuristics for agents and demonstrate viability and applicability in several simulation scenarios.

I. INTRODUCTION

The upcoming smart grid gives rise to several multi-objective control tasks. Due to the expected huge number of distributed energy resources (DER) that have to be controlled, self-organized and decentralized algorithms are seen as the most promising solution. On the other hand, surprisingly low effort has been put in developing decentralized multi-objective approaches so far. Here, we go with the example of the predictive scheduling problem [1].

To enable small and individually operated energy devices to responsibly take over control tasks, pooling of different DER is necessary in order to gain enough potential and flexibility. An established concept for such pooling is the virtual power plant (VPP) [2], [3]. Orchestration of such groups of energy units is done by different scheduling procedures that frequently involve multi-objective optimization.

Predictive scheduling [1] describes an optimization problem for day-ahead planning of energy generation in VPPs, where the goal is to select a schedule for each energy unit – from an individual search space of feasible schedules with respect to a future planning horizon – such that as a global objective the distance to a target power profile for the VPP is minimized (e.g., a product from an energy market). Actually, this constitutes a multi-objective problem. Further objectives like cost minimization, maximization of residual flexibility (for later planning periods) or environmental impact are usually to be achieved concurrently [4]. These goals are often conflicting.

So far, the problem is often reduced to the single objective case for proper solving; if applicable with a combination of different objectives to a single, weighted sum of objectives. We propose a fully decentralized multi-objective algorithm for this problem based on concepts from the combinatorial optimization heuristics for agents (COHDA) and the S-metric selection evolutionary multi-objective algorithm (SMS-EMOA). The goal is to derive a self-organization approach that results in autonomously acting agents that determine a Pareto front (or at least an approximation) without any central control.

The rest of the paper is organized as follows: We recap multi-objective optimization in general, a centralized solution to the predictive scheduling problem and the single objective approach to the decentralized solution. We define the set of challenges that have to be solved to make algorithms like COHDA multi-objective capable and present a first solution approach that integrates concepts from the SMS-EMOA. We conclude with evaluation results from a simulation study and deduce some further research questions.

II. RELATED WORK

Decentralized, multi-agent-based multi-objective optimization has so far not gained much attention; at least in the sense of jointly calculating the Pareto front of a given problem. Some approaches have been developed for tuning a multi-agent system or for internal multi-objective decisions. In [5] an example is given for the emergency response planning problem, [6] presents an elevator control. These approaches use centralized algorithms for the multi-objective part. [7] gives an example for multi-objective reinforcement learning. A decision model for objective relationships after intra-agent multi-objective solving is presented in [8].

On the other hand, some approaches have been developed for acceleration by distributing fitness evaluation in multi-objective optimization. An example for a general framework is given in [9]. But, this is not decentralized problem solving by local, agent-based decisions in a collaboration scenario, as we strive for.
A. Predictive scheduling

We here consider rather small, distributed electricity producers that are supposed to pool together with likewise distributed electricity consumers and prosumers (like batteries) in order to jointly gain more degrees of freedom in choosing load profiles. In this way, they become a controllable entity with sufficient market power. In order to manage such a pool of DER, the following distributed optimization problem has to be frequently solved: A partition of a demanded aggregate schedule has to be determined in order to fairly distribute the load among all participating DER. A schedule \( x \) is a real valued vector with each element \( x_i \) denoting the amount of energy generated or consumed during the \( i \)th time interval withing the planning horizon. Optimality usually refers to being from a specific range. In the smart grid domain, often equality as well as a box-constraint demanding all parameters equal to a reference point \( \gamma \) and can thus be described as the Lebesgue measure \( \Lambda \) of the union of non-dominated points \( M \) for a set of objective functions \( f_1, \ldots, f_n: S \rightarrow \mathbb{R} \) defined on some search space \( S \) [20]; and with \( x \prec x^* \) denoting that \( x \) dominates \( x^* \), i.e. all objective values of \( x \) are better than \( x^* \). Different algorithms have been proposed [20]; among them are evolutionary algorithms [21], [22], genetic algorithms including the famous NSGA-II [23], or swarm-based approaches [24].

Predictive scheduling imposes some special needs on constraint handling to ensure that all local schedules are within the feasible phase-space of the individual energy resources [1], [25]. For constraint-handling in multi-objective optimization two general concepts are usually applied [26]. Either a penalty [27], [28] is added to each objective function degrading constraint violating solutions or the definition of Pareto-dominance is extended to take into account constraint violation [26], [29]. Introducing a penalty term changes the objective function and as in multi-objective optimization the impact on different objectives has to be balanced, a too weak set of penalties may lead to infeasible solutions whereas a too strong impact leads to poor distributions of solutions [26]. Nevertheless, all approaches for constraint integration so far need a closed form description of constraints. Constraints are given as a set of (possibly non-linear) inequalities and equalities as well as a box-constraint demanding all parameters being from a specific range. In the smart grid domain, often no closed form descriptions of constraints are available. Such closed form description does not exist in decentralized energy resource scheduling that includes (at least in general) arbitrary unit types [30].

A first solution approach to hybridizing multi-objective optimization and decoder was given in [31], with a centralized approach based on SMS-EMOA.

B. SMS-EMOA

Using S-metric selection for evolutionary multi-objective algorithms has first been proposed by [22]. The S-metric is based on the hypervolume encapsulated by the set of non-dominated solutions and a reference point [21] and can thus be described as the Lebesgue measure \( \Lambda \) of the union of hypercubes defined by the reference point \( x_r \) and the set of non-dominated points \( m_i \) [19], [22]:

\[
S(M) = \Lambda(\bigcup\{a_i|m \in M\}).
\]
Algorithm 1 Basic algorithmic scheme of the SMS-EMOA (cf. [22]).

\[
P^{(0)} \leftarrow \text{randomPopulation}() \\
\theta \leftarrow 0 \\
\text{while } \theta < \text{max iterations do} \\
\quad \sigma \leftarrow \text{mutate} \circ \text{crossover}(P^{(\theta)}) \\
\quad \{R_1, \ldots, R_k\} \leftarrow \text{fast-nondominated-sort}(P^{(\theta)} \cup \{\sigma\}) \\
\quad p \leftarrow \arg \min_{s \in R_k} [\Delta s(s, R_k)] \\
\quad P^{(\theta+1)} \leftarrow P^{(\theta)} \setminus \{p\} \\
\quad \theta \leftarrow \theta + 1 \\
\text{end while}
\]

This metric constitutes an unary quality measure by mapping a solution set to a single value in \( \mathbb{R} \): the size of the dominated space [32]. As it is desirable to have a large \( S \)-metric value for solution sets in multi-objective optimization, [32] first used this measure in a Simulated Annealing approach and [22] developed an evolution strategy (SMS-EMOA) based on this measure. Algorithm 1 shows the basic idea of SMS-EMOA. The algorithm repeatedly evolves a population of \( \mu \) solutions. In each iteration, first a new solution is generated and added to the population. Subsequently, a selection process is started to find the worst individual in the solution which is then removed from the population. Thus, the number of individuals stays constant from a steady state perspective. Selection is done by first issuing a fast non-dominated sort after [23]. In this way, the Pareto fronts are ranked and from the front with the lowest rank the individual with the lowest contribution to the hypervolume (measured by the \( S \)-metric) is removed. This process is repeated until some stopping criterion – e.g. a number of maximum objective evaluations – is met.

In [31] the latter has already been hybridized with a decoder approach for flexibility modeling and constraint-handling in multi-objective energy management.

In general, two types of objective have to be considered. In [30] constraints have been identified on different locality levels. The same holds true for objectives in a VPP. Objectives on a global as well as on a local level have to be integrated. Objectives on a global level have to be achieved jointly. An example is given by the minimization of the deviation of the aggregated joint schedule from a given product schedule that has to be delivered as contracted. These objectives can only be achieved with joint effort. In contrast, local objectives like individual cost minimization are also to be integrated. Although evaluation can only be performed locally (individual cost), help from other to be able to choose a cheaper schedule is often necessary to achieve the goal. In the following we denote with \( f \) local and with \( F \) global objectives.

**C. COHDA**

In general, decentralized algorithms are considered advantageous in many fields of smart grid computation [33], [34]. For the case of predictive scheduling, [35] developed a decentralized algorithm for constrained combinatorial problems: The Combinatorial Optimization Heuristics for Distributed Agents (COHDA). Combined with an appropriate abstraction for individual flexibilities [10]. So far, this fully decentralized approach works with a single objective function and integrates multiple objectives only by combining different objectives into a single one as weighted sum. We extended COHDA to a full-fledged decentralized multi-objective optimization algorithm.

COHDA has been designed as a fully distributed solution to the predictive scheduling problem (as distributed constraint optimization formulation) in smart grid management [36]. Each agent in the multi-agent system is in charge of controlling exactly one distributed energy resource (generator or controllable consumer) with procurement for negotiating the energy. All energy resources are drawn together to a virtual power plant and the controlling agents form a coalition that has to control the VPP in a distributed way.

An agent in COHDA does not represent a complete solution as it is the case for instance in population-based approaches [37], [38]. Each agent represents a class within a multiple choice knapsack combinatorial problem [39]. Applied to predictive scheduling each class refers to the feasible region in the solution space of the respective energy unit. Each agent chooses schedules as solution candidate only from the set of feasible schedules that belongs to the DER controlled by this agent. This selection is done according to local constraints and to a given objective that usually reflects solely the distance (dissimilarity) of the sum of this selection and the schedules of all other agents to the given target schedule.

Each agent is connected with a rather small subset of other agents from the multi-agent system and may only communicate with agents from this limited neighborhood. The neighborhood (communication network) is defined by a small world graph [40]. As long as this graph is at least simply connected, each agent collects information from the direct neighborhood and as each received message also contains (not necessarily up-to-date) information from the transitive neighborhood, each agent may accumulate information about the choices of other agents and thus gains his own local belief of the aggregated schedule that the other agents are going to operate. With this belief each agent may choose a schedule for the own controlled energy unit in a way that the coalition is put forward best while at the same time own constraints are obeyed and own interests are pursued. Thus, we have a multi-objective optimization problem when deciding on the best schedule.

All choices for own schedules are rooted in incomplete knowledge and beliefs in what other agents do; gathered from received messages. The taken choice (together with the basis for decision-making that has been received with prior messages) is communicated to all neighboring agents and, in this way, knowledge is successively spread throughout the coalition without any central memory. Thus, COHDA is a type of gossiping algorithm [41].

Each information update results in recalculating the own best schedule contribution and spreading it to the direct neighbors. By and by all agents accumulate complete information and as soon as no agent is capable of offering a schedule
leading to a better solution, the algorithm converges and terminates. Convergence has been proved in [42].

More formally, each time an agent receives a message, three successive steps are conducted. First, during the perceive phase an agent \( a_j \) updates its own working memory \( e_j \) with the received working memory \( e_i \) from agent \( a_i \). From the foreign working memory the objective of the optimization (i.e. the target schedule) is imported (if not already known) as well as the configuration that constitutes the calculation base of neighboring agent \( a_i \). An update is conducted if the received configuration is larger or has achieved a better objective value, what is only directly possible with a single-objective. In this way, schedules that reflect the so far best choices of other agents and that are not already known in the own working memory are imported from the received memory.

During the decision phase agent \( a_j \) has to decide on the best choice for its own schedule based on the updated belief about the system state \( S_k^{(k)} \). Index \( k \) indicates the age of the system state information. The agent knows from a subset of (or from all) other agents, which schedules they are going to operate (the system state \( S_k^{(i)} \)). Thus, the schedule that fills the gap to the desired target schedule exactly can be easily identified. Due to operational constraints of the controlled DER, this optimal schedule can usually not be operated. In addition, other reasons might render some schedules largely unattractive due to high cost.

Because of this reason, each agent is equipped with a so called decoder that automatically maps the identified optimal schedule to a nearby feasible schedule that is operable by the DER and thus feasible. Based on a set of feasible schedules sampled from an appropriate simulation model for flexibility prediction [43], a decoder can e. g. be based directly on this set (by linearly searching the schedule with the smallest deviation) or be built by learning a support vector model after the approach of [10]. Both approaches have individual advantages and drawbacks regarding computational complexity, search space size and accuracy. Here, we used the support vector version for efficiency reasons.

As the whole procedure is based exclusively on local decisions, each agent decides privately which schedules are taken. Private interest and preferences can be included and all information on the flexibility of the local DER is kept private.

D. Challenges

The COHDA approach can be adapted to many different optimization problems as has been demonstrated e.g. in [3], [30], [44], [45]. Basically, solution encoding, objective evaluation and internal, local decision method have to be adapted to the problem at hand. On the other hand, adapting to the multi-objective case entails some additional challenges that have to be solved:

a) Solution representation: As the goal now is a Pareto front, each agent will have to manage a set of schedules (a set of own contributions to the joint set of schedules). In the multi-objective case, determining the best own selection will no longer work by just determining the missing difference to the target and repairing it with a decoder.

b) Solution quality assessment: Each time an agent in COHDA decides on a new contribution to a solution (remember: an agent represents just the local contribution, not a full solution) the quality of the solution with the old contribution is compared to the one resulting from the new contribution. This assessment is usually done using the objective function evaluating both candidates. In the multi-objective case an agent represents a set of contributions to a set of solutions, thus it is not possible to compare an old solution directly with a new one using the objective.

This problem can be overcome by using measures that evaluate the quality of a set of solutions with regard to the Pareto front. When using concepts from SMS-EMOA, the achieved hypercube volume can be used.

c) Incomplete solutions: During the initial setting time of COHDA, solutions are incomplete by design. COHDA has been developed for predictive scheduling. One agent starts by issuing a schedule (as local solution contribution) for the own energy resource. At this point in time, a solution consists only of this single contribution (as if it was a VPP with just a single energy resource). For the single-objective predictive scheduling case this is admissible as such a solution is always worse regarding the single objective of resembling the wanted target schedule as close as possible. After some negotiation steps, more agents join in and finally a contribution from all agents are on hand. For the multi-objective case it cannot be guaranteed that solutions with incomplete contribution are worse than complete solutions. An example may be given by minimizing cost as objective. Cost dominated by primary energy would deteriorate the objective if more energy resources joint to contribute to the solution; contradicting a minimization.

Several solutions are possible.

1) The protocol could be altered and each agent could be requested to calculate an initial contribution in order to avoid incomplete solutions. Depending on the problem at hand these initial contributions might be nonsense as they have been determined without knowledge on the others’ decisions.

2) A penalty term could be added to the objective in order to deteriorate solutions based on the number of agents that still have to join. In this case the agents would need knowledge on the number of agents that are in the group.

3) Solutions with a larger number of contributions are always considered better regardless of the evaluation result. This might not hold for all objectives and may lead to wrong convergence directions.

For some objective functions there seems be no issue at least if the number agents is low enough compared with the number of network connections in between them and if the agents join in quick enough. For the simulations conducted with the first approach proposed here, after an initial deterioration of the solution quality a convergence could be observed towards
better solutions. Improved versions should integrate one of the
described existing approaches to guarantee convergence.

d) Localization of objectives: In standard COHDA each
agent knows the global objective function. In multi-objective
COHDA all global objective functions might also be known
by all agents. On the other hand, not all objectives can be
calculated by using the schedules of other agents directly, as it
is the case in predictive scheduling. Calculating individual
costs for example requires knowledge on private cost factors
of other energy resources. Such factors are usually not known
publicly nor communicated. Thus essential information for
calculating the objectives is missing in a fully decentralized
scenario.

In [46], an extension to the decoder approach has been
proposed that is capable of annotating individual schedules
with performance indicators. In [47] an ontology has been
proposed that is capable of annotating individual schedules
with objective information. Calculating the objectives is missing in a fully decentralized
scenario.

A. Implementation

We implemented a fully decentralized multi-objective ap-
proach for predictive scheduling by extending the COHDA
algorithm. First, we had to define the solution format. A
solution to the overall problem (Eq. 2) is – in the multi-
objective case – a set of sets of schedules for the virtual power
plant; each one consisting of a schedule for the respective
energy resource in the group:

\[ X = \{ X_1, \ldots, X_n \} \quad (4) \]

with

\[ X_k = \{ x_{ij} \} \in \mathbb{R}^{m \times d} \quad (5) \]

where \( x_{ij} \) denotes the mean real power of energy resource \( i \)
during the \( j \)th time period. Thus each row of the matrix \( X_k \)
represents a schedule for the respective energy resource. This
solution set \( X \) is determined in a way that it approximates the
Pareto front. In this way, each agent holds and negotiates on
a set of schedules for the own energy resource.

Let \( \{ x_{1}^{(a_j)}, \ldots, x_{n}^{(a_j)} \} \) denote the set of local schedules (for the
own, controlled device) that is negotiated by agent \( a_j \). Let \( \kappa_j \)
be the current working memory of agent \( a_j \) (updated by
an incoming message; cf. II-C). Let

\[ X_O = \{ x_{1}^{(a_1)}, \ldots, x_{n}^{(a_1)} \}, \ldots, \{ x_{1}^{(a_{m-1})}, \ldots, x_{n}^{(a_{m-1})} \} \in \kappa_j \quad (6) \]

be the currently known schedule selections (local solution
candidates) from all the other agents \( a_1, \ldots, a_{m-1} \in A \backslash \{ a_j \}. \)

Basicallly, this is the system state belief \( \mathcal{S}(a_j)_k \) without
the agent’s own contribution from decision \( k \):

\[ X_O = \mathcal{S}(a_j)_k \{ x_{1}^{(a_j)}, \ldots, x_{n}^{(a_j)} \}. \quad (7) \]

Now the procedure (performed by agent \( a_j \)) for deciding on
the own schedule selection is as follows: The sum of schedules of
the other agents is calculated as

\[ O = \{ O_1, \ldots, O_n \} \quad (8) \]

with

\[ O_i = \sum_{x_i \in X_O} x_i. \quad (9) \]

Now a solution of the MAS (cf. Eq. 4) to the joint Problem
is represented as

\[ \left[ \begin{array}{c}
O_1^{(a_j)} \\
\vdots \\
O_n^{(a_j)}
\end{array} \right] \quad (10) \]

with \( x_{1}^{(a_j)}, \ldots, x_{n}^{(a_j)} \) being the decision variables of the local
problem of deciding on the best local schedules under the
assumption that the other agents’ schedules are operated as
communicated.

Solution candidate \( x_{1}^{(a_j)}, \ldots, x_{n}^{(a_j)} \) is initialized randomly and

evolved for some iterations towards the Pareto front. For
each evolution step one randomly chosen schedule \( x_k \in 
\{ x_{1}^{(a_j)}, \ldots, x_{n}^{(a_j)} \} \) is mutated to \( x_k' \) by adding a Gaussian delta
\[ r \in \mathcal{N}(0,1) \] to one randomly chosen element of \( x_k \). As
crossover operator, uniform crossover is applied. Please note,
as an agent can only decide on its own schedules, mutation
and crossover may not be applied to other agents’ schedules
from \( O \). Then, the agent performs a fast non-dominated sort
on \( O \cup \{ x_k' \} \)

For applying the fast non-dominance-sort as introduced in
[48], from the worst front the worst individual (solution with
the lowest hypercube contribution) is removed. For sorting
and selecting the worst individual by \( \gamma \)-metric selection, the
dominance of solutions has to be determined by using the
objective functions. For this purpose, we extend the definition of
dominance by integrating the decoder set:

\[ y < y' \iff \forall i = 1, \ldots, d : y_i < y'_i \quad (11) \]

in order to keep track of the individual (technical) constraints of
the energy resources by setting

\[ \left\{ \right. \]

\[ \left. f_1(\gamma(x_1)) + \cdots + f_{n}(\gamma(x_n)) \right. \]

\[ \left. f_2(\gamma(x_1)) + \cdots + f_{n}(\gamma(x_n)) \right. \]

\[ \vdots \]

\[ f_m(\gamma(x_1)) + \cdots + f_{n}(\gamma(x_n)) \]

\[ F_1(M) \]

\[ \vdots \]

\[ F_{\gamma}(M) \]

\[ \left. \right\} \]

as introduced in [31]. In this way, variations of the previous
solutions in the solution set are produced by applying variation
operators to the genotype. In this way the selection and cross-over operators merely have to obey an easy to integrate box constraint that ensures that each value of a solution candidate is kept within $[0, 1]$ if max power is scaled to 1 (corresponding to 100% rated power). No further constraints have to be integrated. Thus, the problem formulation can be regarded as constraint-free. Constraint-handling is introduced by using decoder functions that abstract from individual capabilities or technical constraints of energy units. The set of decoders ensures that selection is done on feasible solutions only and thus that the solution set approaches a Pareto front without any knowledge about controlled energy units.

The last step in the COHDA process requires comparing the result achieved in the previous round with the current achievement. If the new one is better, it is communicated to the neighboring agents, else it is discarded and the old is kept without communicating any achievement. In the single objective case, the achieved objective values can be compared directly. This is not possible in the multi-objective case as the result constitutes a set of solutions. Thus, we decided to compare both results using the hypervolume between a reference point and the solution set as rather usual in multi-objective algorithms. For fast calculation of exact hypervolumes we applied the WFG (walking fish group) algorithm [49].

With these settings we addressed all challenges identified in section II-D to constitute a fully decentralized, agent-based determination of the Pareto front of a joint multi-objective problem.

B. Results

For our evaluation we simulated different virtual power plants consisting of different co-generation plants. The model has already been used and evaluated in different projects, e.g. [13], [46], [50], [51]. We started with a rather small setting of four agents and 96-dimensional schedules resulting in a 384-dimensional search space which has already been evaluated to be highly multi-modal and ragged [52]. As goal, two objectives were set: $F_1$ denotes the deviation of the joint schedule from the desired target schedule ($\| \cdot \|_2$) and $f_2$ equalizes the run of the co-generation plants by minimizing peak loads:

$$f_2 = \sum_{x_i \in X} \sum_{j=1}^{d} (x_{ij} - \mu)^2,$$

(13)
with $\mu$ being the mean power over the whole planning horizon. In this way, the variance in power levels is minimized. Figure 1 shows an example of the resulting Pareto front approximations of the individual 4 agents. The evolutionary part during the decision phase has been run for 500 iterations each. A cross marks an individual for the solution set (50 schedules per agent) and a circle marks a non-dominated solution from the front. Note, that solution here denotes already a solution (joint schedule) for the predictive scheduling problem. Even if all schedules from the individual result sets of the agents are plotted jointly in a single plot, all solutions are non-dominated; see Figure 2. Thus, all solutions that are individually generated by the agents stem from the same Pareto front approximation, although there are slight differences in the solution sets (regarding distribution on the front) as can be visually seen in Figure 1. The agents approximate the same front but exhibit differences in the individual solution sets.

Some statistics on this scenario with different numbers of evolutionary iterations are given in Table I.

As the proposed approach is still a heuristic, results are as yet not that perfect with growing problem size. Figure 4 shows another example from a scenario with 50 agents approximating a 50 schedule solution set each.

Here another objective has been tested: Achieve a desired state of charge (SOC) for some of the thermal buffer stores that are attached to the co-generation plants.

$$f_3 = \sum \delta(SOC_i, soc(x_i)).$$  

(14)

In this larger scenario some of the solutions in the joint solution set are still dominated by the solutions from others, even though 5000 iterations have been conducted during decision phases. Obviously, the number of necessary iterations quickly grows with problem size: deteriorating needed negotiation time.

On the other hand, the mean hypervolume (as quality measure for a solution) converges quite early to an acceptable value. Figure 3 shows the convergences of the process that led to result 1. Here, the mean (so long achieved) hypervolume of all agents is measured at discrete points in time from the concurrently asynchronously running multi-agent system. Error bars show the inter-agents variance. At the same time the variance among all agents is determined. Depicted are only the first iterations, not the full process. Depending on the specific use case at hand, it might thus be possible to stop the negotiation at an earlier stage with a still acceptably good solution. Some more investigations will be necessary here. Figure 5 shows the situation for the second case.

### Table I

<table>
<thead>
<tr>
<th>Indicator</th>
<th>4 agents</th>
<th>50 agents</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypervolume</td>
<td>0.401 ± 0.213</td>
<td>0.533 ± 0.258</td>
</tr>
<tr>
<td>best $F_1$</td>
<td>0.103 ± 0.110</td>
<td>0.083 ± 0.006</td>
</tr>
<tr>
<td>best $F_2$</td>
<td>1.785 ± 0.213</td>
<td>1.745 ± 0.142</td>
</tr>
<tr>
<td># messages</td>
<td>2191.6 ± 2407.1</td>
<td>679.6 ± 516.6</td>
</tr>
<tr>
<td># decisions</td>
<td>1064.1 ± 1166.6</td>
<td>328.4 ± 245.3</td>
</tr>
</tbody>
</table>

Mean achieved results for 100 runs each are depicted in Table I. The achieved hypervolume in the larger scenario larger due to the better aggregated schedule ($F_1$). This observation is consistent with the single-objective case and rooted in the higher flexibility of larger VPPs. The number of exchanged messages and decisions decreases significantly with growing scenario size probably allowing for more complex decision routines of the agents in future improvements.
For many applications within the smart grid scheduling domain, multi-objective optimization problems have to be solved. As for scalability reasons decentralized (agent-based) algorithms are seen as a promising solution, multi-objective capability has to be integrated into these methods. At the same time, proper constraint-handling is indispensable for acceptance.

With the approach demonstrated here, we integrated multi-objective capabilities taken from SMS-EMOA into fully decentralized energy scheduling.

Applicability and effectiveness of the proposed approach have been demonstrated by simulations. Nevertheless, some questions remain for further research: Replacing the decoder decision by a more complex optimization based decision method entails the need for tuning additional parameters; e.g. the (probably adaptive) number of iterations during each decision procedure. Maybe a substantial acceleration could be reached with additional convergence detection methods. Designing mutation, crossover and selection is also subject to specialized improvements. Moreover, an integration of indicators into the communicated solution data format could improve privacy as local objectives (and thus their calculation details) would no longer be publicly known. Nevertheless, this first approach demonstrated the general feasibility of fully decentralized multi-objective optimization.

REFERENCES


