Privacy-Preserving Self-Organization in Distributed Energy Scheduling

Joerg Bremer[1] and Sebastian Lehnhoff[2]
[1]*Department of Computing Science, University of Oldenburg, Uhlhornsweg, Oldenburg, Germany*
[2]*R&D Division Energy, OFFIS – Institute for Information Technology, Escherweg, Oldenburg, Germany*
fi

Keywords: Privacy Preservation, Distributed Optimization, Multi-Agent Systems, Secret Sharing.

Abstract: Negotiation among agents that are controlling and orchestrating a set of distributed processes often relies on frequent data exchange to allow solution evaluation and thus convergence towards a joint solution. Solving decentralized coordination problems with coalitions of agents that exchange messages and information to build beliefs for problem solving, inevitably allows insight into other agents' operational options. Keeping local information private is thus of utmost importance for a wide user acceptance of such algorithms. We present an extension to a distributed, self-organizing algorithm for energy scheduling in virtual power plants or energetic neighborhoods that keeps all information about possible operations of participating energy resources private. For calculations during optimization the algorithm relies on secret sharing and joint multi-party computations. We evaluate the algorithm against the original non privacy-preserving standard version and present some insights for future work.

1 INTRODUCTION

Multi-agent systems are widely seen as one of the most promising solutions for optimization and asynchronous coordination in future, autonomous cyber-physical systems like the energy grid (Stark et al., 2024). Digitalized energy systems can be characterized by a high degree of complexity in monitoring and controlling a large number of distributed energy resources. The infrastructure is dynamically optimized with regard to generation, consumption, load flow, supply quality and safety, cost, comfort, and more.

Multi-agent systems are widely seen as the best tool to integrate autonomy by self-organization principles and to cope with the specific problem characteristics in the smart grid (Ramchurn et al., 2012). Whereas trust has been on the research agenda in multi-agent systems for years (Ramchurn et al., 2004), privacy has received less attention so far. Trust is a major issue when humans have to interact with a system. Many CEOs identify trust and reputation as the key driver of their action. On the other hand, privacy is just as important for a broad acceptance of distributed, agent-based algorithms (Rapp and Bremer, 2023).

An example is given by the predictive scheduling use case. In virtual power plants (VPP), a task that has often to be solved is the scheduling problem that assigns an optimal operation schedule to each energy resource. Of course, the algorithm has to take into account a set of objectives like accurate resemblance of the desired load profile, robustness of the schedule, costs, remaining flexibility for subsequent planning periods, and many more (Stark et al., 2024; Bremer and Lehnhoff, 2020a). A schedule in this context is a real-valued vector with each element denoting the amount of energy (or mean active power) generated or consumed during the respective time interval for a given future discrete planning horizon. A simple predictive scheduling algorithm just tries to assign a schedule to each energy resource such that the sum of all schedules resembles a desired aggregated schedule as close as possible. Such aggregated schedule might for example be the result of some market bidding and the VPP wants to jointly operate this schedule.

As a distributed, self-organizing solution to predictive scheduling, (Hinrichs et al., 2013) proposed the combinatorial optimization heuristic for distributed agents (COHDA). COHDA was proposed as a solution to problems that can be decomposed on an algorithmic level (Talbi, 2009) and can thus be adapted to a wide range of different problems (Stark et al., 2021; Narayanan et al., 2024; Bremer and Lehnhoff, 2017b; Bremer and Lehnhoff, 2017a). The general concept is closely related to Cooperative Coevolution (Potter and Jong, 2000). The key concept of

Bremer, J. and Lehnhoff, S.
Privacy-Preserving Self-Organization in Distributed Energy Scheduling.
DOI: 10.5220/0013094900003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 1*, pages 253-262
ISBN: 978-989-758-737-5; ISSN: 2184-433X

COHDA is an asynchronous iterative approximative best-response behavior that relies on collected information from other agents for own decision making.

In the predictive scheduling use case in VPPs, operable schedules are frequently exchanged (Hinrichs and Sonnenschein, 2017) that contain information about time and possible energy generation and/ or consumption schemes – often for a whole day with 15 minute (or smaller) time resolution. These schedules are needed by the agents to calculate the distance of the joint schedule to the wanted target schedule to evaluate solution candidates during local decision making. Each schedule contains data of the possible portion of energy that may be generated (or consumed) during a given time period for a series of multiple time intervals. With each negotiation round, a new possible operable schedule is sent to several other agents together with transient information on other agents' schedules. This is related to the widely used gossiping principle that comes into play in many distributed optimization algorithms (Poli, 1996). The sent information can be collected, aggregated, and exploited by other agents.

Simulation examples for exploitation can be found in (Dabrock, 2018), where detailed information on internal processes like heating profiles, and thus working hours, machinery load factors, or current capacity utilization were derived. Moreover, in (Bremer and Lehnhoff, 2022) individual time of use tariffs were reconstructed from aggregated, collected schedules.

Parties that are potentially interested in participating in distributed energy management – e.g. in virtual power plants (Naval and Yusta, 2021) or in energy neighborhoods (Wehkamp et al., 2020) – will likely refrain from engaging in agent-based coordination if the risk of revealing data is larger than the benefit from participating. In order to achieve a broad acceptance of such algorithms, it is essential to handle the disclosure of data sparingly or to otherwise ensure that it cannot be misused.

We extended the meanwhile widely used COHDA protocol by integrating privacy preservation through secret sharing. In this way, no schedules with private information have to be sent around anymore and all calculations for solution candidate evaluation are done jointly in a way that no party can reveal any information of a single agent's energy resource.

The rest of the paper is organized as follows. We start with an overview on existing privacy concepts in multi-agent systems and the smart grid. After describing the standard COHDA algorithm, we identify the privacy gap and develop the integration of secret sharing for keeping information private. Finally, we evaluate privacy-preserving COHDA and its performance with regard to solution quality, message volume and convergence.

2 RELATED WORK

Privacy is probably one of the oldest human concerns (Schoeman, 1984; Schermer, 2007). Nevertheless, with the advent of delegating more and more tasks to autonomous systems that act on behalf of a user ambiantly in the background, the technical perspective grows significantly in importance. For the information technological perspective several taxonomies for classification have been proposed, e.g. (Spiekermann and Cranor, 2009; Bostwick, 1976; Kang, 1997). Thus, privacy as a unitary concept with a uniform value (Such et al., 2014).

According to (Solove, 2005), privacy can be threatened by three main information-related activities: information collection, processing, and dissemination. The use case of self-organized, distributed energy scheduling covers all three and therefor requires special attention and a holistic solution that covers the complete algorithmic approach (Rapp and Bremer, 2023). Due to the algorithmic level decomposition of the solved problem (Talbi, 2009), energy generation or consumption schedules are collected from other agents, are processed several times to calculate the objective value of different solution candidates and are eventually disseminated to other agents for further decision making. This general scheme occurs (at least partly) also in other distributed solutions to the problem, e.g. in (Bremer and Lehnhoff, 2016).

When it comes to privacy prevention measures in multi-agent systems in general, several approaches have already been developed. Some good overviews can be found in (Such et al., 2014; Chandramohan et al., 2015). For algorithmic applications, some effort has been put into consensus algorithms (Wang et al., 2021; Fiore and Russo, 2019), but for distributed optimization only few examples using encryption can be found. For example, (Huo and Liu, 2021) uses the Pailler cryptosystem (Wu et al., 2016) with some inefficiency issues (Peng et al., 2004).

For the energy domain, several surveys explore solutions for different use cases: e.g. for vehicle to grid applications, communication, or metering (Han and Xiao, 2016; Kumar et al., 2019; Finster and Baumgart, 2015). These works cover the collection perspective and are not concerned with any algorithmic use of the data. As possible technologies for securing distributed optimization algorithms against private data leakage, two techniques seem to be most promising: (partial) homomorphic encryption and

multiparty computation based on secret sharing (Rapp and Bremer, 2023).

The concept of homomorphic encryption was originally introduced as privacy homomorphism by (Rivest et al., 1978). The first working fully homomorphic encryption scheme has been proposed in 2009 by (Gentry, 2009) whereas partially homomorphic schemes have been known for longer. Since then, great progress regarding performance and applicability has been made (Meftah et al., 2022; Moore et al., 2014). But, this issue is still not fully solved. In general, homomorphic encryption allows for arbitrary commutations to be conducted on the encrypted data without a need to decrypt it first. The result is then a cypher text or number in the same cryptosystem. Good overviews on fully homomorphic encryption schemes can be found in (Marcolla et al., 2022).

Partially homomorphic encryption schemes are often more efficient but support just a subset of operation, e.g. multiplication or addition only. Examples can be found in (Wu et al., 2016; Guo et al., 2022; Mikhail et al., 2014). Also simple order-preserving schemes exist (Agrawal et al., 2004). Order-preserving encryption had previously been used for predictive scheduling in energy management in the smart grid in (Bremer and Lehnhoff, 2020b). But, in order to make it work, the solved optimization problem had to be reformulated, what limits the approach to a simplified problem version. The results can thus not be generalized. This seems to be the general case because no efficient higher level operations are supported that are usually needed in evaluating solution candidates in optimization.

Homomorphic encryption is still computationally rather expensive and would slow down an optimization process significantly (Alaya et al., 2020; Moore et al., 2014) or require extensive GPU or in-memory calculations (Meftah et al., 2022; Reis et al., 2020; Al Badawi et al., 2018) that cannot be guaranteed in the field in smart grid applications. Additionally, homomorphic encryption with a public key scheme (Rothblum, 2011) rather supports client-server or peer-to-peer structures instead of agent coalitions as in our use case. The ability to decrypt the result also allows for decrypting the used data.

A different approach for keeping privacy relies on the idea of secret sharing. In a secret-sharing scheme, a dealer shares a secret with n parties P by distributing shares to the parties such that: (Beimel, 2011):

1. given a set A of subsets of the n parties (the access structure), a subset in A can reconstruct the secret from the shares (all in the subset are needed),
2. no one outside A can reveal any (partial) information of the secret.

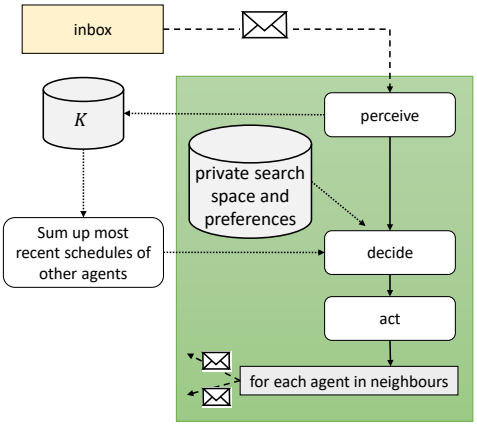


Figure 1: Flow chart for a single agent in standard COHDA. This process chain is started each time a message arrives in the in-box of the agent.

Often, A equals P; except for example in (Dawson and Donovan, 1994). Hence, all members of P together are needed to reveal the secret (Gordon and Katz, 2006). Several applications of secret sharing for distributed computing can be found for example for Byzantine agreement (Rabin, 1983), securing multiparty computations (Ben-Or et al., 2019; Chaum et al., 1988), threshold cryptography (Desmedt and Frankel, 1991), access control (Naor and Wool, 1996), or generalized oblivious transfer (Tassa, 2011). We used a scheme of secret sharing where each share constitutes a random part of a horizontally stratified schedule – a vector of random shares of active power for each time interval – and used multi-party-computation (Catalano et al., 2005) to prepare objective evaluation in our distributed self-organizing algorithm. In this way the computationally expensive part has to be conducted only once for each agent decision.

3 ALGORITHM

3.1 COHDA

An asynchronous iterative approximate best-response behavior is the core idea of COHDA. Each agent is responsible for one dimension of the algorithmic problem decomposition. The intermediate local solutions of other agents (represented by published decisions) are regarded as temporarily fixed. Thus, each agent only searches along a low-dimensional cross-section of the search space and thus has to solve merely a simplified sub-problem. Nevertheless, for evaluation of the solution, the full objective function is used after aggregation of all agent's contributions. In this way, the approach achieves an asynchronous coordi-

nate descent with the additional ability to escape local minima by parallel searching different regions of the search space; and because former decisions can be revised if newer information becomes available.

Hence, all agents coordinate themselves by updating knowledge and exchanging information about each other that supports local decision-making. For message exchange, the agents are logically drawn together by an artificial communication overlay network. Most often, a small world topology (Watts and Strogatz, 1998) is used for this purpose. Starting with an arbitrarily chosen agent and by passing it a message containing just the global objective, each agent repeatedly goes through three stages: perception, decide, and act.

`perception stage.` In this first phase, the agent prepares for local decision-making based on incoming information. Every time an agent receives a message from one of the neighboring agents that precede in the directed communication topology, it aggregates the data that is included in the message(s) into the own knowledgebase K. Each message contains two essential pieces of information: the result of the updated local decision of the sending agent and the transient information about decision updates of other agents that led to this decision. The important privacy issue is that this information contains operation schedules of all other agents (or rather of their controlled energy resources respectively) and probably also cost annotations for these schedules.

`decision stage.` In this second phase, the agent is making a local decision on a schedule for the own controlled energy resource that puts the coalition forward as best as possible. To do this, a local optimization is solved. The agent determines the gap to the optimal solution by summing up the schedules (that are the results of the most recent decisions) of all other agents and by calculating the difference to the wanted target schedule. To do this, the agent needs to know the decisions of the others. During optimization, the agent searches for the local schedule that fills the gap as best as possible. As constrained-handling technique for the individual constraints of the energy resource, often a decoder approach is used (Bremer and Sonnenschein, 2013).

`act stage.` During this last phase, the agent compares the best found solution with the previous solution by using the global objective function. If the new solution is better, the agent broadcasts a message containing its new local solution contribution together with everything it has learned from preceding agents and their current local solution contributions (the decision base) to the immediate neighbors in the communication topology. Upon these messages, the receiving agents then also go through these three stages, what in turn may lead to revised local solution contributions and thus to a further improved overall solution.

If no local solution can be found that improves the overall solution, no message is sent and the process ceases. After the system has generated a series of intermediate solution candidates, the heuristic eventually terminates in a state where all agents know an identical solution. This one is taken as the final solution of the negotiation. Properties like guaranteed convergence and local optimality have been formally proven in (Hinrichs and Sonnenschein, 2017).

3.2 Adding Privacy

In order to enable privacy preservation in COHDA, we need to get rid of sending plain schedule information to other agents. We achieve this goal by using a secret sharing approach. To do this, the local process of summing up the schedules of all the other agents was replaced by a joint summation that incorporates all agents; with each agent knowing only a random part of the information. In privacy COHDA schedules are stored locally so that only the agent that controls a device knows the schedules that have been selected as possible contributions to solution candidates. Because each agent has to store several schedules during the course of a negotiation and because also older schedules need to be kept (for comparison and because several paths are searched in parallel), we chose a hash map for local storage. Only the identifying hash code is sent to other agents instead of the plain schedule. Any identifier type would do.

In standard COHDA an agent a_j (for the set of all agents A) evaluates solution candidates (single objective case) during decision making by calculating

$$e = \delta\left(\sum_{i=1}^{|A|} \boldsymbol{x}_i, \boldsymbol{\zeta}\right), \tag{1}$$

where δ represents a distance measure (often $||\cdot||_2$) that measures the similarity between the joint schedule of all agents (that includes solution candidate x_j of agent a_j) and the target schedule ζ. This can be converted to

$$e = \delta\left(\boldsymbol{x}_j + \sum_{a_i \in A \setminus a_j} \boldsymbol{x}_i, \boldsymbol{\zeta}\right) \rightarrow_{\boldsymbol{x}_j \in \mathcal{F}_{a_j}} \min. \tag{2}$$

Thus, we need the sum of schedules of all other agents to combine with our own solution candidates x_j for

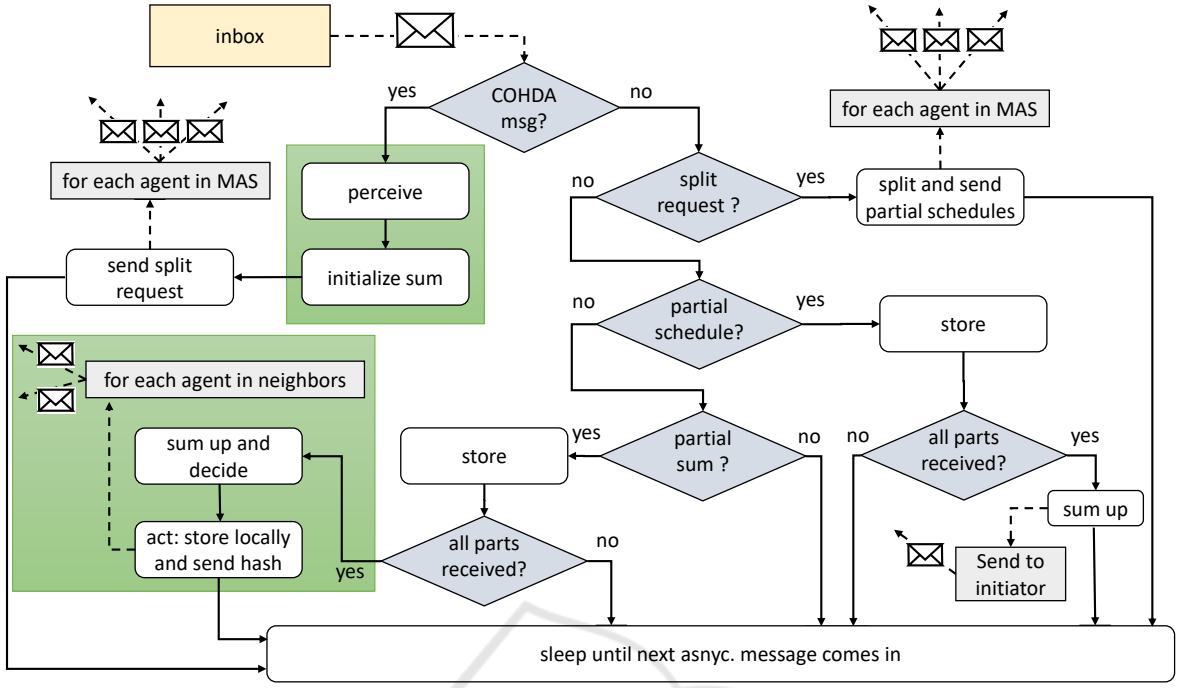


Figure 2: Flow chart for a single agent in privacy COHDA. This process chain is started each time a message arrives in the in-box of the agent. The green framed part corresponds to the standard COHDA process with just perceive-decide-act phases.

evaluation. Only the elements of x_j (from the local feasible region $\mathcal{F}_{a_j}$) are mutable for optimization; the sum of others is locally treated as fixed. In privacy COHDA, the agent does not know the other schedules. Instead, this summation has to be done jointly by a secret sharing approach. Let x_j be the schedule that is requested from agent a_j to be summed up with the other agents' schedules. Schedules are scaled to $[0,1]$ (percent of rated power). Each agent splits up its own schedule. Agent a_j splits up x_j into the set $R_{a_j} = \boldsymbol{r}_{a_j}^{(1)}, \ldots, \boldsymbol{r}_{a_j}^{(n)}$ of random pieces of the same dimension as the original schedule (stratification):

$$R_{a_j} = \{\boldsymbol{r}_{a_j}^{(\ell)} | \sum_{\ell=1}^{n} \boldsymbol{r}_{a_j}^{(\ell)} = \boldsymbol{x}_j \wedge \boldsymbol{r}_{a_j}^{1}(\ell) \sim \mathcal{N}(0,1)^d\}. \quad (3)$$

The elements $\boldsymbol{r}_{a_j}^{(1)}, \ldots, \boldsymbol{r}_{a_j}^{(n-1)}$ of set R_{a_j} are distributed to the other $n-1$ agents; one piece is kept. This is done by each agent. Then, for each agent holds: agent a_k knows $\boldsymbol{r}_{a_1}^{(k)}, \ldots, \boldsymbol{r}_{a_n}^{(k)}$ and can calculate the sum $S_k = \sum \boldsymbol{r}_{a_i}^{(k)}$ that is sent to the requesting agent (the one that wants to make a new decision). This agent finally can calculate the sum of all schedules by summing up the sums of random parts from all agents: $S = S_1 + \cdots + S_n$.

Neither the random pieces of schedules nor the intermediate sums of random parts from different agents contain any usable information. Yet, the sum of all parts eventually yields the desired sum. The random partition for schedule stratification can be chosen differently every time. Actually, it is even not necessary to reveal the identity of a schedule part owner.

As a result, the decision stage was split up into two parts and the summation process coordination was integrated by adding three new message types. Figure 2 shows the new overall process of a single agent. When a standard COHDA message is received, the standard perceive stage takes place with the only difference that the message contains only hash codes of schedules. The actual schedules and are stored privately at the respective agents.

With the hash codes the summation of schedules is initiated by sending a split request message to all agents. This message contains for each agent: (1) the individual hash code for the receiving agent (the one that is part of the joint solution), (2) the agent ID of the requesting agent (the one to which the results have to be returned), and (3) a process ID for the new summation process. The process ID is necessary due to the asynchronous nature of COHDA. While an agent is waiting for the result, new COHDA messages might come in from other agents that spawn additional summation processes.

If an agent receives a split request message, it looks up the respective schedule using the received hash code and splits it up into random parts for every agent (including itself). After splitting, one different one of the random partial schedules is sent to

each agent (including itself) together with the process ID and the ID of the initiator. Because the agent includes itself in the summation process, there is at least on random part that is not revealed even if all other agents work together against this agent. After sending, the agent sleeps again until the next message comes in.

If an agent receives a partial schedule, it stores the part together with the process ID and waits for other partial schedules with the same process ID from other agents to come in. If the agent has received a partial schedule from each agent with the same process ID, it sums up these parts. From the received partial schedules as well as from the sum, no information can be derived. If all parts are received and the (partial) sum is calculated, the sum is sent back to the initiator together with the process ID.

If an agent receives a partial sum it stores it and waits for the other partial sums. A partial sum is the sum of random parts from each agent and thus does also contain no information on a specific schedule. If an agent receives a partial sum it is the initiator of this summation process. As soon as all partial sums are received, they are added up to the complete sum of all schedules of all other agents that belong to a specific solution candidate. Now, the agent can continue with the standard COHDA procedure by determining a schedule for the own controlled energy resource based on the sum and the gap to the goal. If better than the previous solution, it stores the new solution candidate and sends out the hash code for the new schedule. In Fig. 2 the parts that correspond to standard COHDA are marked by a green background (cf. Fig. 1). The rest is an addition to the protocol to achieve privacy by secret sharing.

4 RESULTS

For our evaluation experiments, we used the following scenario: Groups of distributed co-generation plants (CHP) are supposed to jointly generate a given energy generation schedule (e. g. given from some market). Each CHP is controlled by an agent and all agents together form a coalition and negotiate the individual operation planning together by using the COHDA algorithm. As model for the CHP, we used a well know simulation model that has been used and evaluated e.g. in (Neugebauer et al., 2015). This model comprises a micro co-generation plant with 4.7 kW of rated electrical power (12.6 kW thermal power) and is bundled with a thermal buffer store. Constraints restrict power band, buffer charging, gradients, min. on and off times, and satisfaction of thermal demand. Thermal demand is determined by simulating losses of a detached house (including hot water drawing) according to given weather profiles.

Table 1: Results for four CHP generator examples with different numbers of agents, different time horizons, and different problem difficulty averaged over 100 simulation runs each. We compare the mean number of messages sent during negotiation, the mean number of decisions that the agents made to revise previous solution candidates and the final solution quality measured as symmetric mean absolute percentage error.

	standard COHDA	privacy COHDA
50 agents, $l = 8$		
sMAPE	0.46 ± 0.69	3.2 ± 0.81
decisions	4612.1 ± 2490.241	3597.8 ± 301.5
messages	109728 ± 59154.5	6992777.7 ± 501955.3
10 agents, $l = 96$		
sMAPE	0.024 ± 0.095	0.78 ± 0.29
decisions	149.81 ± 45.9	338.43 ± 79.9
messages	620.07 ± 194.307	35971.74 ± 8773.9
100 agents, $l = 96$		
sMAPE	0.12 ± 0.046	0.73 ± 0.39
decisions	13191.9 ± 3861.7	11383.45 ± 4415.5
messages	650751.8 ± 190802.5	$93115416.5 \pm 40164449.1$
10 agents, $l = 96$, unsolvable		
sMAPE	36 ± 1.9	37 ± 1.3
decisions	62.9 ± 7.58	74.55 ± 7.44
messages	259.9 ± 31.1	6906.3 ± 768.6

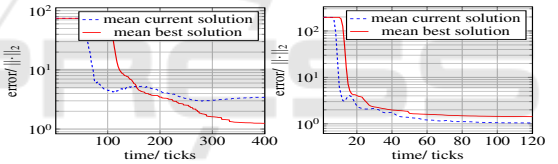


Figure 3: Mean convergence behavior of privacy COHDA (left) and standard COHDA (right).

First, we looked at the convergence of privacy COHDA. Convergence of standard COHDA has been formally proven in (Hinrichs and Sonnenschein, 2017). COHDA converges under the condition that the global objective evaluation function is monotonically decreasing and no agent sends a message if no improvement was made. We use the same evaluation function as in the standard version. On the other hand, in privacy COHDA messages can be sent also if no improvement was made because an agent may participate in some joint summation. But, such message does not trigger any new COHDA action (act phase). So, the conditions for termination still hold in privacy COHDA. Nevertheless, we analyzed the convergence behavior. From 100 optimization runs with 20 agents each, we collected at each point in time the best so far seen and the currently scrutinized solution qualities from each agent. To do this in an asynchronously acting system, we used an observer that queried all agents at global time ticks. As all of these collected

Table 2: Results for a mixed and a pure consumption scenario with 25 agents each. We compare the mean number of messages sent during negotiation, the mean number of decisions that the agents made to revise previous solution candidates and the final solution quality measured as symmetric mean absolute percentage error.

	standard COHDA	privacy COHDA
mixed scenario, 25 agents, $l = 96$		
sMAPE	0.16 ± 0.12	3.7 ± 1.34
decisions	3424.2 ± 1107.52	5259.8 ± 1518.7
messages	41215.1 ± 13310.8	3142462.6 ± 935036.1
25 consumer agents, $l = 96$		
sMAPE	0.36 ± 0.15	2.1 ± 1.45
decisions	2974.1 ± 445.9	4747.5 ± 1508.1
messages	35794.6 ± 5368.8	2557552.1 ± 921311.3

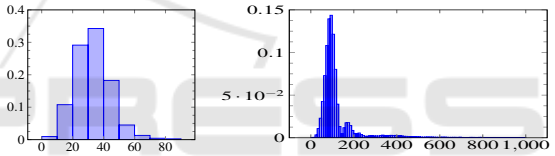
convergence series are of a different length, we averaged the solution qualities for all agents for each time tick for the first 400 ticks.

A sample result is shown in Figure 3(a). Obviously, a rather good solution appears on average rather early, but needs some time to propagate to the other agents. This propagation time is longer than in standard COHDA due to the fact that the summation takes some time until a solution candidate from other agents can be considered. In (Hinrichs and Sonnenschein, 2014) it has been demonstrated that a certain degree of additional disturbance in the system leads to a better solution quality, because more solution candidates are considered in parallel. Such disturbance is already incorporated in both COHDA version by random message delays. An additional effect due to the privacy related messages could not be observed although the decision making delay obviously leads to a larger exploration compared with standard COHDA. For comparison: Fig. 3(b) shows the mean convergence of standard COHDA for the same scenario. Compared with this result, privacy COHDA still explores larger regions in the end instead of switching to exploitation. Future work will show whether this is a potential for improvement.

Table 1 shows some results that compare the achieved final solution quality of both approaches. In order to allow comparability of the results of differently sized scenarios, all qualities are measured as symmetric mean absolute percentage error $\text{sMAPE} = \frac{1}{n} \cdot \sum \frac{|\zeta_i - x_i|}{0.5 \cdot (\zeta_i + x_i)}$. The mean absolute percentage error allows for comparing scenarios of different size (different number of agents), but in our case the maximum negative deviation is due to the nature of the scenario bounded by zero. Thus, standard MAPE is biased towards higher values. For this reason, we used the symmetric version. All results are in percent.

Table 1 shows the results for four different scenarios with different numbers of agents that all control a CHP and different problem complexities due to different schedule lengths. The mean total number of messages for all scenarios is significantly higher for the privacy case. This is immediately clear, because three new message types have been introduced. The mean total number of decisions that all agents make, stays more or less in the same magnitude. The result quality slightly degrades except for the unsolvable scenario. Here, both COHDA versions stagnate with almost the same sub-optimal solution quality. For all other scenarios, the global optimum is known to be zero. In smaller scenarios, standard COHDA seems to be advantageous in terms of solution quality. In more realistically sized larger scenarios with 100 or more agents, the degradation in solution quality for privacy COHDA is neglectable. That is because it is smaller than the inherent uncertainty in such scenarios due to forecast errors; e.g. of the thermal demand (Bremer and Lehnhoff, 2017c).

Table 2 shows two additional results for a mixed (generation and consumption) scenario comprizing 10 CHP, 10 heat pumps, 3 batteries, and 2 medium sized cool storages, as well as a pure consumption scenario with warm water boilers instead of CHPs.



(a) (b)

Figure 4: Mean convergence behavior of privacy COHDA (left) and standard COHDA (right).

In the last two experiments, we had a deeper look into the sent messages. obviously, the total number of messages grows significantly compared with standard COHDA. For standard COHDA, the main impacts on the number of messages used for negotiation are based on the used topology, the message delay, the reaction time of individual agents, and the number of agents. Research on these impacts can be found in (Oest et al., 2021; Anders et al., 2012; Hinrichs and Sonnenschein, 2014). This overall weak quadratic behavior is also present in privacy COHDA as we use the same basic principles. Additionally, we can observe two effects that are due to introducing privacy.

First, we can observe a change in the distribution of messages over time. Figure 4(a) shows the distribution of messages over time for standard COHDA. The number of messages grows for some time and ceases eventually after a single peak. The distribution over time for privacy COHDA differs in two characteristics (cf. Fig. 4(b)). First, after the main peak, the duration until ceasing is way longer. Introducing a distributed calculation of the sum of schedules

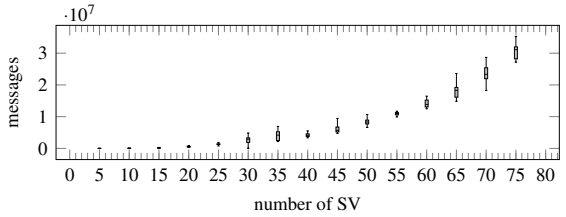


Figure 5: Relationship between the number of agents and the distribution of the amount of messages that are sent during negotiation.

obviously results in a way longer phase of propagating an already found optimal solution to all agents, what is necessary for termination (Hinrichs and Sonnenschein, 2017). Secondly, we can observe several smaller peaks after the first one and thus a damped oscillation behavior. The reason for this is so far unclear. Nevertheless, as COHDA shows an anytime behavior after some initial warm-up phase (Hinrichs and Sonnenschein, 2017), the process could be stopped earlier with still having a feasible (sub-optimal) solution in case a deadline is approaching. The conditions for the anytime property still hold for the privacy-preserving version, so we can stop even earlier – compared to the total length of the process.

The growth in the number of messages that is induced by the number of agents grow quadratically (with $R^2 \approx 0.9$) what can be seen in Fig. 5 for a CHP example with 5 to 75 agents. This is quite obvious as for each new agent one additional schedule has to be summed up and this additionally involves all present agents. On the other hand, the optimization problem that is solved with COHDA also has a profound impact on the number of exchanged messages. Some example analysis on other use cases with partly worse growth behavior can be found in (Volkova et al., 2019; Buhl et al., 2017; Bremer and Lehnhoff, 2017a; Radtke et al., 2023).

5 CONCLUSION

Multi-agent systems in the smart grid domain in which individual agents represent the interests of different (private) operators of energy resources in a distributed coordination approach for joint planning, must treat private data with care. This data should only be disclosed sparingly (ideally not at all) in order to achieve broad acceptance of such mechanisms. We presented the integration of secret sharing into distributed, multi-agent based energy scheduling as one building block for a privacy-preserving smart grid. With the presented approach, it becomes possible to achieve an optimal planning of the energy generation within a group of distributed energy resources in a distributed way without a need for disclosing information on one's own production schemes. The evaluation results are promising for up to medium sized virtual power plant or energetic neighborhood scenarios. For larger scenarios, a research path for accelerating negotiation could be sketched as a hierarchical multi-part computation of the sums in order to reduce group size and thus communication traffic. In this way, predictive scheduling with integrated secret sharing is a suitable approach for achieving privacy through data sparsity in an important class of algorithms in the future smart grid.

REFERENCES

Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. (2004). Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 563–574.

Al Badawi, A., Veeravalli, B., Mun, C. F., and Aung, K. M. M. (2018). High-performance fv somewhat homomorphic encryption on gpus: An implementation using cuda. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 70–95.

Alaya, B., Laouamer, L., and Msilini, N. (2020). Homomorphic encryption systems statement: Trends and challenges. *Computer Science Review*, 36:100235.

Anders, G., Hinrichs, C., Siefert, F., Behrmann, P., Reif, W., and Sonnenschein, M. (2012). On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty. In *Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2012)*, pages 29–38, Lyon, France. IEEE Computer Society. (Best Paper Award).

Beimel, A. (2011). Secret-sharing schemes: A survey. In *International conference on coding and cryptology*, pages 11–46. Springer.

Ben-Or, M., Goldwasser, S., and Wigderson, A. (2019). Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali*, pages 351–371.

Bostwick, G. L. (1976). A taxonomy of privacy: Repose, sanctuary, and intimate decision. *Calif. L. Rev.*, 64:1447.

Bremer, J. and Lehnhoff, S. (2016). A decentralized PSO with decoder for scheduling distributed electricity generation. In Squillero, G. and Burelli, P., editors, *Applications of Evolutionary Computation: 19th European Conference EvoApplications (1)*, volume 9597 of *Lecture Notes in Computer Science*, pages 427–442, Porto, Portugal. Springer.

Bremer, J. and Lehnhoff, S. (2017a). An agent-based approach to decentralized global optimization-adapting cohda to coordinate descent. In *International Confer-*

ence on Agents and Artificial Intelligence, volume 2, pages 129–136. SCITEPRESS.

Bremer, J. and Lehnhoff, S. (2017b). Decentralized surplus distribution estimation with weighted k-majority voting games. In *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems: International Workshops of PAAMS 2017, Porto, Portugal, June 21-23, 2017, Proceedings 15*, pages 327–339. Springer.

Bremer, J. and Lehnhoff, S. (2017c). *Enhancing Support Vector Decoders by Integrating an Uncertainty Model*, pages 114–132. Springer International Publishing, Cham.

Bremer, J. and Lehnhoff, S. (2020a). Controlled self-organization for steering local multi-objective optimization in virtual power plants. In *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness. The PAAMS Collection: International Workshops of PAAMS 2020, L'Aquila, Italy, October 7–9, 2020, Proceedings 18*, pages 314–325. Springer.

Bremer, J. and Lehnhoff, S. (2020b). Encrypted decentralized optimization for data masking in energy scheduling. In *Proceedings of the 3rd International Conference on Big Data Research*, ICBDR '19, pages 103–109, New York, NY, USA. Association for Computing Machinery.

Bremer, J. and Lehnhoff, S. (2022). Information disclosure in vpp-information disclosure by decentralized coordination in virtual power plants and district energy systems.

Bremer, J. and Sonnenschein, M. (2013). Constraint-handling for optimization with support vector surrogate models - a novel decoder approach. In *International Conference on Agents and Artificial Intelligence*, volume 2, pages 91–100. SciTePress.

Buhl, H., Dombrowski, T., Hogen, E., Kreutz, M., Palm, D., Stark, S., Stubbe, P., Warsch, S., Bremer, J., Lehnhoff, S., et al. (2017). Ein algorithmus für den wiederaufbau eines smart grid nach einem blackout.

Catalano, D., Cramer, R., Di Crescenzo, G., Darmgård, I., Pointcheval, D., Takagi, T., Cramer, R., and Damgård, I. (2005). Multiparty computation, an introduction. *Contemporary cryptology*, pages 41–87.

Chandramohan, D., Sathian, D., Rajaguru, D., Vengattaraman, T., and Dhavachelvan, P. (2015). A multi-agent approach: To preserve user information privacy for a pervasive and ubiquitous environment. *Egyptian Informatics Journal*, 16(1):151–166.

Chaum, D., Crépeau, C., and Damgard, I. (1988). Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 11–19.

Dabrock, K. (2018). Privacy in der automatisierten prädiktiven Einsatzplanung von Energieanlagen im Smart Grid. Master's thesis, University of Oldenburg, Dept. of Energy Informatics, Germany.

Dawson, E. and Donovan, D. (1994). The breadth of shamir's secret-sharing scheme. *Computers & Security*, 13(1):69–78.

Desmedt, Y. and Frankel, Y. (1991). Shared generation of authenticators and signatures. In *Annual International Cryptology Conference*, pages 457–469. Springer.

Finster, S. and Baumgart, I. (2015). Privacy-aware smart metering: A survey. *IEEE communications surveys & tutorials*, 17(2):1088–1101.

Fiore, D. and Russo, G. (2019). Resilient consensus for multi-agent systems subject to differential privacy requirements. *Automatica*, 106:18–26.

Gentry, C. (2009). *A fully homomorphic encryption scheme*. Stanford university.

Gordon, S. D. and Katz, J. (2006). Rational secret sharing, revisited. In *Security and Cryptography for Networks: 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006. Proceedings 5*, pages 229–241. Springer.

Guo, Y., Cao, Z.-F., and Dong, X.-L. (2022). Generalized goldwasser and micali's type cryptosystem. *Journal of Computer Science and Technology*, 37(2):459–467.

Han, W. and Xiao, Y. (2016). Privacy preservation for v2g networks in smart grid: A survey. *Computer Communications*, 91:17–28.

Hinrichs, C. and Sonnenschein, M. (2014). The Effects of Variation on Solving a Combinatorial Optimization Problem in Collaborative Multi-Agent Systems. In Mueller, J. P., Weyrich, M., and Bazzan, A. L., editors, *Multiagent System Technologies*, volume 8732 of *Lecture Notes in Computer Science*, pages 170–187. Springer International Publishing.

Hinrichs, C. and Sonnenschein, M. (2017). A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents. *International Journal of Bio-Inspired Computation*, 10(2):69–78.

Hinrichs, C., Sonnenschein, M., and Lehnhoff, S. (2013). Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces. In Filipe, J. and Fred, A. L. N., editors, *International Conference on Agents and Artificial Intelligence (ICAART 2013)*, volume Volume 1 – Agents, pages 25–34. SciTePress.

Huo, X. and Liu, M. (2021). Privacy-preserving distributed multi-agent cooperative optimization – paradigm design and privacy analysis. *IEEE Control Systems Letters*, 6:824–829.

Kang, J. (1997). Information privacy in cyberspace transactions. *Stan. L. Rev.*, 50:1193.

Kumar, P., Lin, Y., Bai, G., Paverd, A., Dong, J. S., and Martin, A. (2019). Smart grid metering networks: A survey on security, privacy and open research issues. *IEEE Communications Surveys & Tutorials*, 21(3):2886–2927.

Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H., and Aaraj, N. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10):1572–1609.

Meftah, S., Tan, B. H. M., Aung, K. M. M., Yuxiao, L., Jie, L., and Veeravalli, B. (2022). Towards high performance homomorphic encryption for inference tasks on cpu: An mpi approach. *Future Generation Computer Systems*, 134:13–21.

Mikhail, M., Abouelseoud, Y., and Elkobrosy, G. (2014). Extension and application of el-gamal encryption scheme. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–6. IEEE.

Moore, C., O'Neill, M., O'Sullivan, E., Doröz, Y., and Sunar, B. (2014). Practical homomorphic encryption: A survey. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2792–2795. IEEE.

Naor, M. and Wool, A. (1996). Access control and signatures via quorum secret sharing. In *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pages 157–168.

Narayanan, A., Pournaras, E., and Nardelli, P. H. (2024). Large-scale package deliveries with unmanned aerial vehicles using collective learning. *IEEE Intelligent Systems*.

Naval, N. and Yusta, J. M. (2021). Virtual power plant models and electricity markets-a review. *Renewable and Sustainable Energy Reviews*, 149:111393.

Neugebauer, J., Kramer, O., and Sonnenschein, M. (2015). Classification cascades of overlapping feature ensembles for energy time series data. In *Proceedings of the 3rd International Workshop on Data Analytics for Renewable Energy Integration (DARE'15)*. Springer.

Oest, F., Radtke, M., Blank-Babazadeh, M., Holly, S., and Lehnhoff, S. (2021). Evaluation of communication infrastructures for distributed optimization of virtual power plant schedules. *Energies*, 14(5):1226.

Peng, K., Aditya, R., Boyd, C., Dawson, E., and Lee, B. (2004). Multiplicative homomorphic e-voting. In *International Conference on Cryptology in India*, pages 61–72. Springer.

Poli, R. (1996). *Parallel distributed genetic programming*. Citeseer.

Potter, M. A. and Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29.

Rabin, M. O. (1983). Randomized byzantine generals. In *24th annual symposium on foundations of computer science (sfcs 1983)*, pages 403–409. IEEE.

Radtke, M., Stucke, C., Trauernicht, M., Montag, C., Oest, F., Frost, E., Bremer, J., and Lehnhoff, S. (2023). Integrating agent-based control for normal operation in interconnected power and communication systems simulation. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 228–233. IEEE.

Ramchurn, S. D., Huynh, D., and Jennings, N. R. (2004). Trust in multi-agent systems. *The knowledge engineering review*, 19(1):1–25.

Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. R. (2012). Putting the'smarts' into the smart grid: a grand challenge for artificial intelligence. *Communications of the ACM*, 55(4):86–97.

Rapp, B. and Bremer, J. (2023). Masking sensitive data in self-organized smart region orchestration. In *Proceedings of the 2023 8th International Conference on Information and Education Innovations*, ICIEI '23, pages 235–240, New York, NY, USA. Association for Computing Machinery.

Reis, D., Takeshita, J., Jung, T., Niemier, M., and Hu, X. S. (2020). Computing-in-memory for performance and energy-efficient homomorphic encryption. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(11):2300–2313.

Rivest, R. L., Adleman, L., Dertouzos, M. L., et al. (1978). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180.

Rothblum, R. (2011). Homomorphic encryption: From private-key to public-key. In *Theory of cryptography conference*, pages 219–234. Springer.

Schermer, B. W. (2007). *Software agents, surveillance, and the right to privacy: a legislative framework for agent-enabled surveillance*. Leiden University Press.

Schoeman, F. D. (1984). *Philosophical dimensions of privacy: An anthology*. Cambridge University Press.

Solove, D. J. (2005). A taxonomy of privacy. *U. Pa. l. Rev.*, 154:477.

Spiekermann, S. and Cranor, L. (2009). Engineering privacy. *Software Engineering, IEEE Transactions on*, 35:67–82.

Stark, S., Frost, E., and Nebel-Wenner, M. (2024). Distributed multi-objective optimization in cyber-physical energy systems. *ACM SIGENERGY Energy Informatics Review*, 4(2):7–18.

Stark, S., Volkova, A., Lehnhoff, S., and de Meer, H. (2021). Why your power system restoration does not work and what the ict system can do about it. In *Proceedings of the twelfth ACM international conference on future energy systems*, pages 269–273.

Such, J. M., Espinosa, A., and García-Fornes, A. (2014). A survey of privacy in multi-agent systems. *The Knowledge Engineering Review*, 29(3):314–344.

Talbi, E. (2009). *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley.

Tassa, T. (2011). Generalized oblivious transfer by secret sharing. *Designs, Codes and Cryptography*, 58:11–21.

Volkova, A., Stark, S., de Meer, H., Lehnhoff, S., and Bremer, J. (2019). Towards a blackout-resilient smart grid architecture. In *International ETG-Congress 2019; ETG Symposium*, pages 1–6.

Wang, Y., Lu, J., Zheng, W. X., and Shi, K. (2021). Privacy-preserving consensus for multi-agent systems via node decomposition strategy. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(8):3474–3484.

Watts, D. and Strogatz, S. (1998). Collective dynamics of 'small-world' networks. *Nature*, (393):440–442.

Wehkamp, S., Schmeling, L., Vorspel, L., Roelcke, F., and Windmeier, K.-L. (2020). District energy systems: Challenges and new tools for planning and evaluation. *Energies*, 13(11):2967.

Wu, H.-T., Cheung, Y.-m., and Huang, J. (2016). Reversible data hiding in paillier cryptosystem. *Journal of Visual Communication and Image Representation*, 40:765–771.