Douglas Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.): INFORMATIK 2015,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2015

A Fully Distributed Continuous Planning Approach for Decentralized Energy Units

Astrid Nieße[1] and Michael Sonnenschein[2]

Abstract: In this contribution, we present the development of DynaSCOPE as a fully distributed continuous planning approach for decentralized energy units organized in Dynamic Virtual Power Plants (DVPP). The work at hand elaborates on possible basic algorithms for the DVPP planning task, motivates the choice of a fully distributed algorithm and gives details and evaluation results for the presented system. It is shown, that DynaSCOPE meets the requirements for the DVPP planning task and effectively enhances product delivery in case of energy unit outages or prognosis deviations.

Keywords: distributed algorithms, Smart Grid, virtual power plants, DynaSCOPE, COHDA

1 Introduction

Distributed algorithms for the control of real-world components are often considered for application motivated by one of the following aspects: Either the complexity of the control tasks shows an abrupt rise or new requirements are given regarding functionality or performance. In both cases, a conventional extension of the control system at hand might not be applicable. In the context of the electrical energy system both aspects are true: With the still rising amount of distributed energy ressources (DER), controllable loads and electrical storage devices the complexity of the control task escalates. The flexibility retrieved with these components is needed for an active operation of the system itself as a new functional requirement. Consequently, a large body of research has emerged from the field of distributed algorithms and distributed artificial intelligence in this domain within the last years.

One of the concepts presented is the Dynamic Virtual Power Plant (DVPP) [Ni12]. DVPPs are set up as product specific aggregations of distributed energy units and thus reflect the different products that can be handled at the energy markets. DVPPs can be understood as an extension to the static VPP concept presented about 15 years ago. Within DVPPs, each energy unit is represented by a software agent. These agents form DVPPs and control their energy product delivery during operation. Products are defined as an energy amount fed into the grid within a defined time span. Operation schedules are discretized by so-called planning intervals covering 15 minutes. Functional requirements have been deduced from this task: As the energy units are prone to prognosis deviations and outages (i.e. units do not follow the defined operation schedules), a continuous planning approach is needed to

[1] OFFIS - Institute for Information Technology, Escherweg 2, D-26121 Oldenburg, Germany, niesse@offis.de
[2] University of Oldenburg, D-26111 Oldenburg, Germany, michael.sonnenschein@uni-oldenburg.de

ensure product delivery in case of these incidents. This approach should detect incidents and trigger a replanning, if the product cannot be delivered as contracted at the market. Costs for the operation of the unit have to be reflected in the planning process, as well as soft constraints of the energy unit, like avoiding frequent switching of some DER. As the DVPP concept follows a self-organizing and distributed approach, a central concept would sacrifice this paradigm. In the work presented here, DynaSCOPE as a decentral approach for replanning of energy units in DVPPs is proposed. The underlying optimization problem is solved as a distributed constraint optimization problem. DynaSCOPE has been developed following the iterative process model Smart Grid Algorithm Engineering (SGAE) [NTS13] proposed for the development of (distributed) algorithms in the Smart Grid application domain. In terms of SGAE, the results of a first iteration are presented.

The rest of this contribution is structured as follows: In section 2 we will introduce some known distributed algorithms and discuss, to what extend they might be used for the task of continuous energy planning. We will then choose an appropriate algorithm and elaborate on the extension needs. Details on implementing the product specific DVPP approach in the planning process are given in section 4, followed by a description of the incident detection and replanning process in section 5. Finally we will present some evaluation results and conclude with remarks on future work.

2 Agent-based control concepts for Smart Grid control issues

The work presented by Akkermans, Ygge and Gustavsson on 1996 has been one of the first applications of agent-based control in the electrical energy system [AYG96]. The so-called HomeBots approach revealed the expected capabilities regarding scalability, flexibility, adaptivity and broad applicability [Gu99]. Since this work, many agent-based approaches habe been developed in the disciplines of electrical engineering, control and system theory and information technology and information systems. Some projects focus on algorithms like the work presented by Rahwan [RRJ09], others on compatibility with the current energy economy and roles or automation systems [ZV12]. The understanding of what constitutes a distributed system differs a lot as well: Some projects use software agents only as a concept to realize a energy unit gateway, some realize hierarchical systems [Le10] and others present fully distributed algorithms [HLS14]. In the following, we will focus on an selection of algorithms that might be used as a basic algorithm for the continuous energy planning within DVPPs.

2.1 Choosing the Right Basic Algorithm

In table 1 a selection of distributed concepts is presented. All of them have already been evaluated for a Smart Grid control task. For each concept it is depicted, if the approach already fulfils the requirement listed on the left side of the table (✓), if an extension would be needed (E) or if no extension is feasible to fulfil the specific requirement (×). The Holonic Virtual Power Plant (Hol. VPP) presented in [Tr10] has already been implemented for the continuous planning process in static VPPs. Although products have not been reflected, an extension would be possible. The agents presented in the concept though are not capable of evaluating the quality of a new VPP schedule – this task is performed by a dedicated agent. The same aspect holds for the Autonomous Virtual Power Plants [An10]. Thus, both concepts are not applicable for the task at hand following the distributed DVPP paradigm. ALMA [Po13] is a fully distributed and highly dynamic approach implemented for a dynamic supply-demand-matching task. The introduction of products though is not possible. With COHDA [HLS14] a decentral heuristic has been proposed lately. As can be seen in table 1 all requirements for continuous energy planning can be met by extending the algorithm. In the following, we will present COHDA and discuss the extension needs for DVPP operation.

Requirement	**Hol. VPP**	**AVPP**	**ALMA**	**COHDA**
Products	E	E	×	E
Incidents	✓	(✓)	(✓)	E
Continuous planning	✓	(✓)	(✓)	E
Costs	✓	✓	✓	E
Soft constraints	✓	E	E	E
Decentral evaluation	×	×	(✓)	✓

Tab. 1: Requirements for continuous energy planning and distributed approaches. ✓: approach fulfils requirement. E: approach does not fulfil requirement, extension feasible. ×: extension not feasible.

2.2 COHDA

COHDA is a cooperative heuristic distributed at the algorithm level [HLS14]: Agents exchange information regarding their independently working algorithms to determine the optimal solution of a defined problem. The problem is to minimize the differences between an energy amount delivered by a set of energy units and a given target schedule. We follow [Ta09] in presenting the characteristics of COHDA regarding this type of algorithm. In the work presented here, COHDA should be adapted to solve a distributed constraint optimization problem. Therefore, we follow [Ch11] to describe relevant characteristics regarding this kind of problem. Implemented with agents as means for distribution, we give a short description on the agent model as well. A formal description of COHDA is given in [HLS14]; for a detailled description of COHDA see [Hi14] (German only).

- **Information exchanged:** COHDA has been developed for the day-ahead planning of clusters of energy units using agents. The agents send their working memory κ, consisting of the target schedule ζ, the known system state and the current solution candidate γ. With the target schedule ζ, each energy unit is provided an operation schedule. The known system state is used to store the current knowledge regarding the chosen operation schedules of all agents in the system.

- **Exchange criterion:** Agents exchange their working memory κ if any changes have been performed to this memory.
- **Exchange topology:** Agents communicate using a virtual overlay network, i.e. a virtual communication topology. Each agent communicates with a subset of all agents called neighborhood.
- **Information integration:** When an agent receives a message from another agent holding the working memory of this agent, he integrates all information regarding chosen operation schedules and current solution candidate γ.
- **State evaluation:** Each agent is able to evaluate a solution candidate, i.e. a cluster schedule, with respect to the target schedule ζ. All agents within the system will yield the same value for the same solution candidate.
- **Decision rule:** If an agent received new information, it is free to choose a new operation schedule. This new schedule might e.g. better fulfill the energy unit's constraints. If the new schedule leads to a better cluster schedule regarding the target schedule ζ, a new solution candidate has been found and is communicated.
- **Agent model:** COHDA is used for a planning task – once a set of possible operation schedules is determined, no monitoring or control of the energy unit is needed. Therefore, the agent model is deliberative.

3 DynaSCOPE

In the following, we will present the design concept that leads from COHDA to DynaSCOPE, a fully distributed algorithm for the **Dyna**mic **S**cheduling **C**onstraint **OP**timization for **E**nergy units. In table 2, the characteristics of COHDA are combined with the functional requirements for continuous energy planning as discussed in section 2.2. Extending COHDA to products leads to an extension need concering the most characteristics. The agent model though has not to be changed to fulfil this requirement.

Characteristic	Products	Incidents	Replanning	Costs	Constraints
Information exchanged	×			×	
Exchange criterion		×			
Exchange topology	×				
Information integration	×				
State evaluation	×		×	×	
Decision rule	×				×
Agent model		×			

Tab. 2: Extension needs by characteristics and functional requirements. ×: Extension or adaptation regarding this characteristic is needed to fulfil the respective requirement.

The guiding principle in setting up the design concept from these extension needs was to yield a fully functional system with known expected behavior in each step. By this means, a simulative evaluation can be performed for each extension, thus following the iterative

SGAE process model. To this end, the extension needs have to be restructured to a step-wise extension approach. In figure 1 the three-step design concept is shown. In design step 1, COHDA is extended to handling products and taking costs for schedules into account. Although this extension leads to a change in many characteristics (see table 2) a simple evaluation case can be defined for evaluation purposes: If only one product is put on the market, only one global DVPP will be formed. Thus, the algorithmic features can be directly compared to COHDA. The characteristics of the basic algorithm are documented in detail in [Hi14] – they serve as test base when evaluating the result of this design step. In design step 2, energy unit specific soft-constraints are introduced in the scheduling by extending the local evaluation of feasible operation schedules. A simulative approach with exemplary soft constraints can be used to show that these are factored in the planning process. The last design step leads to DynaSCOPE as fully distributed algorithm for continuous energy unit planning: Incidents are detected by the agents and a rescheduling is triggered if needed. The agents now have to reflect not only planning but control issues, as a continuous monitoring and control of the energy units is needed. The evaluation of the resulting software artefact reflects the requirements as discussed in section 1.

In the following, we will present some details of design steps 1 and 3, including problems by using a direct extension approach in step 1. Design step 2 is concerned with the integration of soft constraints and will be covered in a separate contribution.

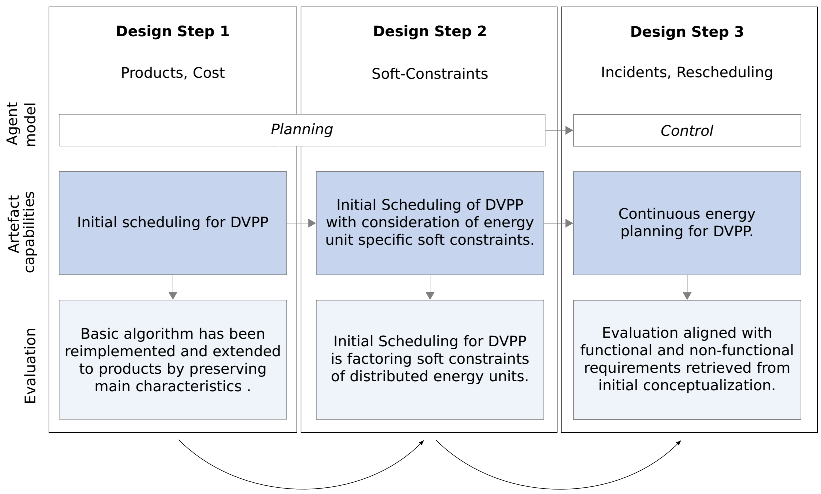

Fig. 1: Design Concept: Building DynaSCOPE from COHDA. [Ni15, translated from German]

4 Handling Products

Product specific coalitions of energy units are the main feature of DVPPs compared to static VPPs. It has to be decided how products are handled during agent information exchange, information integration, state evaluation and agent decision. Additionally it has to be decided if the agents' exchange topology should change. In the following we will first present a straightforward approach to adapting COHDA to products and the problems resulting from this leading to some basic design decisions for DynaSCOPE.

For a first extension to products, the following extensions have been made:

- Information exchanged: The agents exchange their full operation schedule and add information on the product negotiated within the respective DVPP.
- Exchange topology: An exchange topology is setup within each DVPP.
- Information integration: The agents incorporate all knowledge they receive and add product specific information.
- State evaluation: The agents evaluate the operation schedules received depending on the DVPPs they belong to.

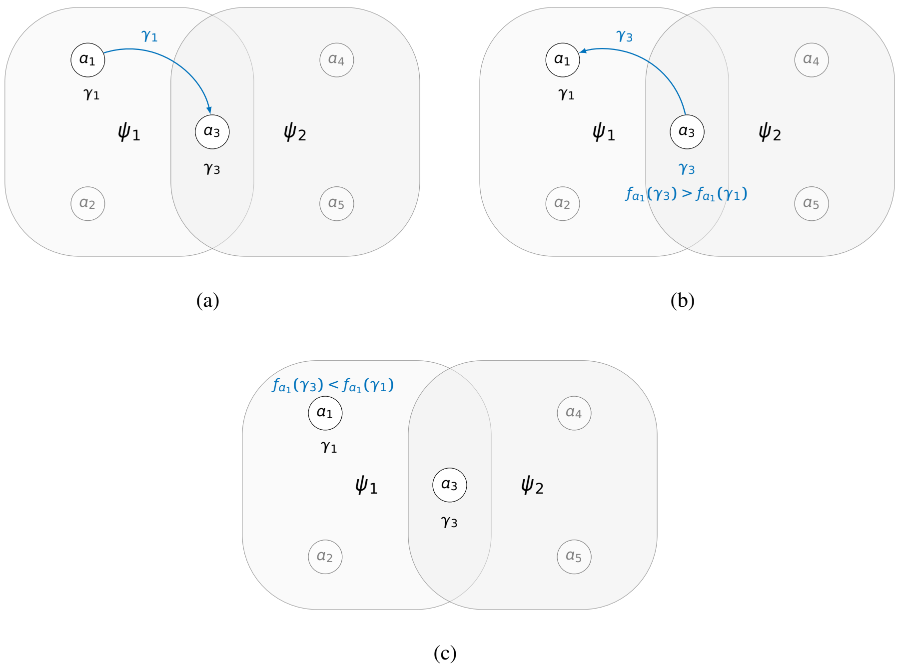

Fig. 2: Direct extension of COHDA to the product-specific DVPP concept.

In figure 2 a simple example is shown to illustrate why this straighforward extension approach is not feasible. In the example given, two DVPPs exist, with agent a_3 contributing to both DVPPs ψ_1 and ψ_2. Agents a_1 and agent a_3 both are aware of a solution candidate, i.e. a set of operation schedules for all agents within both DVPPs. In the first step (see figure 2(a)) agent a_1 sends its new solution candidate γ_1 to agent a_3. Agent a_3 now evaluates this solution candidate and compares its product performance to its own solution candidate γ_3. As agent a_3 contributes to both DVPPs, the solution candidate given by agent a_1 may perform worse in terms of product fulfillment than solution candidate a_3. Therefore, agent a_3 sends solution candidate γ_3 to agent a_1 (see figure 2(b)). Now agent a_1 evaluates solution candidate γ_3 and compares its performance regarding DVPP ψ_1 to solution candidate γ_1. In the example given, γ_1 performs better regarding ψ_1. As no new information was incorporated or generated by agent a_1 the algorithm terminates. As both agents stick to different solution candidates though, the algorithm did not converge (see figure 2(c).

Using a simple example as depicted in figure 2 it could be shown that a straighforward approach for extending COHDA to DVPPs as product specific coalitions will sacrifice convergence. Some design decisions had to be taken as a result of this:

First, if products are the basis for DVPP, operation schedules should not cover the whole planning horizon (i.e. 24 hours). Therefore, only the product specific slot of an operation schedule, the so-called product schedule) should be communicated within a DVPP. The assigment of a product schedule to all agents within a DVPP is called DVPP schedule. Beside ensuring convergence, this can be seen as an aspect of data minimization. Second, the evaluation function for evaluation the solution candidate can no longer be global on a system level. Therefore, for evaluating new solution candidates, all agents take into account only the product performance within the DVPP under negotiation. In figure 3 the main ap-

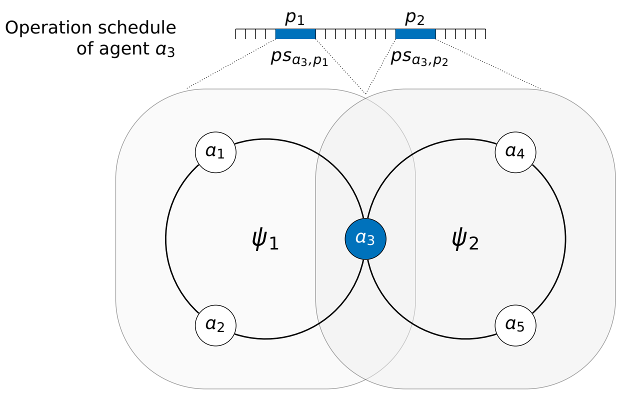

Fig. 3: Communication of product schedules as DVPP-specific excerpt of an operation schedule.

proach to reassure convergence of the algorithm is depicted using the example from figure 2: Although an energy unit has to fulfil a full operation schedule for the whole planning horizon (typically 24 hours), only the product specific excerpt is communicated within a DVPP. For the example given, agent a_3 communicated it's product schedule ps_{a_3,p_1} within DVPP ψ_1 and product schedule ps_{a_3,p_2} within DVPP ψ_2. This reveals some limitations regarding the usage of flexibility of the energy units but ensures convergence.

With this adaptations, the product specifics of DVPPs are integrated into COHDA.

5 Events and Rescheduling

As soon as agents have to detect if their respective energy units follow the negotiated operation schedule, the system has to evolve from an planning system to an agent based control system. As discussed in [BS00] this has crucial influence the agent model needed to fulfil this task. Additionally, the process model has to be adapted to interleave the processes for planning, monitoring and control in an appropriate way. Both aspects are discussed in the rest of this section.

5.1 Agent Model

Software agents in real-world environments are usually categorized in three basic types that define their interaction with the physical world: Reactive agents directly transfer a sensoric input into an action. Reaction of this type of agents can be realized very fast. Deliberative agents model the information retrieved in an internal world model that is used for planning and proactive behavior. Hybrid agents include both aspects (see e.g. [We13] for a detailled discussion on different agent types).

For the task at hand, the continuous planning in DVPP, a hybrid agent model is needed: For the (day-ahead) planning process, a detailled world model, including the energy units capabilities is needed. For monitoring and control though, direct reaction to sensoric input should be possible to ensure a fast reaction to critical states. During simulative evaluation, the differences might not be crucial. When the system is transferred to the field though, the reactive parts might be realized on different (e.g. more reliable, faster and more costly) hardware.

With the InteRRaP architecture a hybrid architecture has been presented many years ago [MP94]. InteRRaP is a vertically layered architecture that represents both knowledge and behavior in different layers. From bottom to top, the layers pass from sensors and actors in the physical world to the cooperative agents' world. In figure 4, the DynaSCOPE agent model based on this InteRRaP architecture is depicted. It shows two interfaces – the unit interface at the bottom and the cooperation interface at the top. Behavioral layers and world model layers address different abstraction layers regarding physical and agent world. In the local evaluation layer, the sensoric input is evaluated to detect incidents: Does the energy unit follow the required operation schedule? Basic unit knowledge like the current operation schedule is needed in this layer. In the second layer, the local planning layer, the agent generates optional operation schedules for the energy unit. Different realizations are possible to model the local planning knowledge, i.e. the energy unit's flexibility. For DynaSCOPE, a vector-based surrogate model using a high-dimensional representation of possible unit states has been used as proposed by Bremer et al. in [BS13] called search space for the rest of this contribution.

For the cooperative planning process, two layers have been introduced in the DynaSCOPE agent model: The global planning layer holds behavior and information for all agents in the respective DVPP needed during the cooperative search of a new cluster schedule. The cooperation layer and knowledge are needed to define, which agents communicate during cooperative search of the new DVPP schedule.

A detailed description of all information stored at the different layers is given in [Ni15]; an excerpt is given in figure 4.

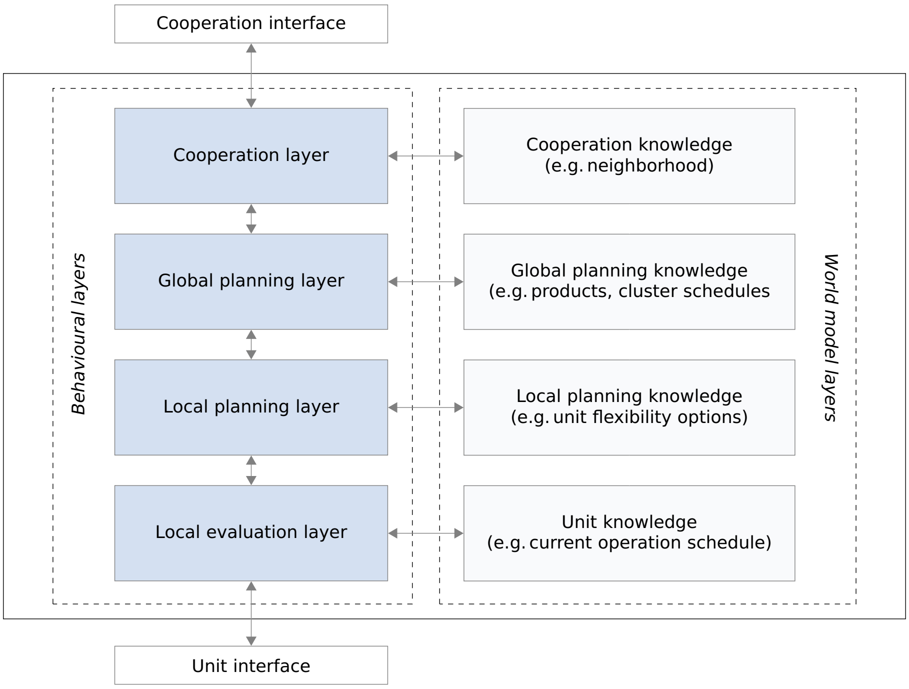

Fig. 4: Hybrid agent model for continuous energy planning

5.2 Process model

DynaSCOPE realizes the continuous energy planning within DVPPs in a fully distributed manner. The processes may be triggered by two main different events, thus comprising entry points to the DynaSCOPE agent behaviors:

- An agent detects an **incident** at its respective energy unit.
- An agent receives a **message** from an agent within the same DVPP.

In the following, we will describe the first entry point to DynaSCOPE. The second one, the process of cooperative search, is similar to the search process as defined for COHDA, although some adaptations have been made to include the incident information during information syntheses. A detailled description regarding this aspect is given in [Ni15].

In figure 5 the processes of detecting and processing an incident are shown. The figure reuses the agent model as shown in figure 4, but focusses on the behavior layers. The process starts at the local evaluation layer. The current operation schedule already has been transferred to the unit in a prior step not shown.

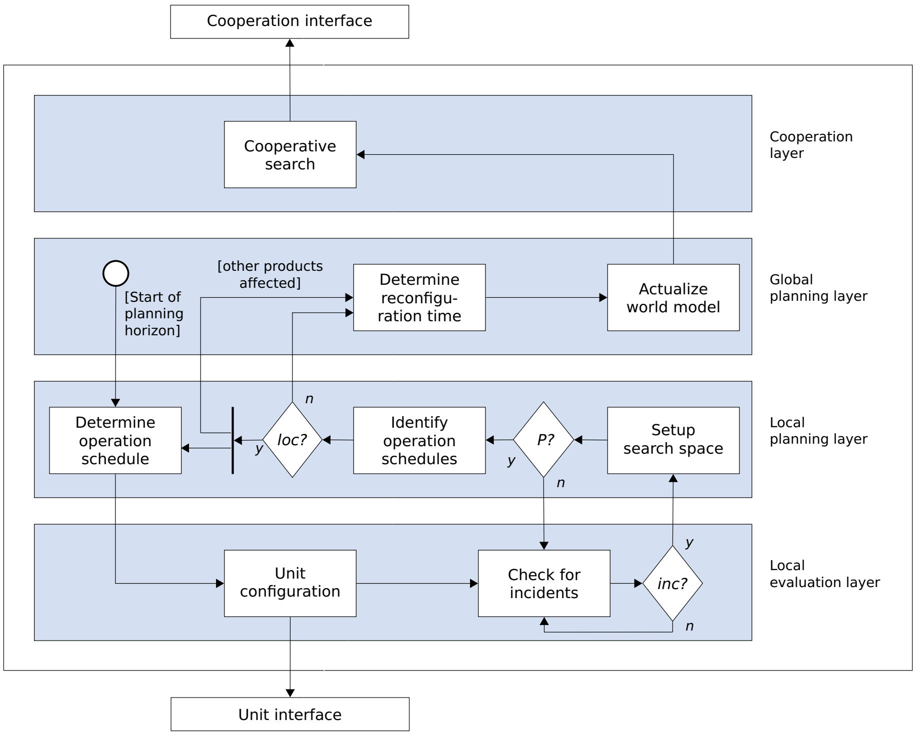

Fig. 5: DynaSCOPE processes from the perspective of an agent detecting an incident at its energy unit. *inc?*: Has an incident been detected? *P?*: Is a product affected by the incident? *loc?*: Is a local compensation possible?

1. Check for incidents: In each monitoring interval the agents check, if the operation schedule is fulfilled. If an incident has been detected, the local planning layer is activated.

2. Setup seach space: As the energy unit state changed, the flexibility model is no longer valid. A new search space is created based on the current energy units behavior.

3. Identify product affected: Not all incidents might affect a product. If no product is endangered by the incident, no further action is needed and the agent returns to the local evalaution layer.

4. Identify operation schedules: If a product is affected by the detected incident, the agent determines operation schedules using the new search space. Local soft constraints might be reflected during this step.

5. Check for local compensation: If the agent can identify a new operation schedule that compensates the incident locally, this operation schedule is passed to unit con-

figuration. If no local compensation is possible or additional DVPPs and thus products are affected, the global planning layer is called to action.

6. Determine reconfiguration time: As the agent has to start the cooperative search for a new cluster schedule, a reconfiguration time is set. The operation schedule defined by then will be used for unit configuration. The first possible reconfiguraton time is at the end of the current planning interval. With the typical time resolution of 15 minutes nearly 15 minutes can be used for the cooperative planning process.
7. Actualize world model: In the next step the agent adds the information on the detected incident and the invalid operation schedule to its world model.
8. Cooperative search: In the last step the agent sends the DVPP-specific parts of its world model to all agents within its neighborhood, thus restarting the cooperative search process. From then on, the agent will process messages received from other agents at the cooperation interface.

6 Evaluation

6.1 Experimental setup

The evaluation of DynaSCOPE has been done using a JAVA-based implementation of the algorithms. The vector-based search space model has been implemented by Jörg Bremer, University of Oldenburg. More details are presented in [Ni15]. For the results presented here, the following scenario has been chosen:

- Energy units: 20 CHPs (4.7 kW_{el}), 20 PV plants (10 kW_p), 10 heatpumps (5 kW).
- Product: 11 a.m. to 3 p.m., 150 kWh to be delivered each hour.
- Message delay: 200 ms.
- Unit flexibility model (search space) generated for a spring day.
- Incidents: CHP outages and PV prognosis deviations.

The allocation of incidents within the 4-hour product horizon is important for the evaluation of the overall system. Therefore, this aspect has been implemented in a deterministic but random based way: We choose different temporal allocations of incidents in a random way but use random seeds to guarantee reproducibility. 100 runs have been chosen for each scenario with a defined set of incidents allocated over time in such a way.

6.2 Incident detection and rescheduling quality

In figure 6 an example is given for a run with 10 CHP outages.[3] On the X axis, the solution evaluation is chronologically ordered: Each time an agents evaluates a DVPP schedule, this is logged within the system. On the Y axis, the expected product delivery performance

[3] The term outage may be misleading here: It can be understood as any type of unscheduled switching off.

is given. A value of 1.0 means perfect product delivery (i.e. 150 kWh in all 4 product hours). In the left part of the diagram, the initial planning process is shown: The agents yield an expected product delivery performace of 0.99 in the day ahead planning process. For all following values, the simulated time is given on top of the diagram for ease of understanding. Incidents are depicted using arrows.

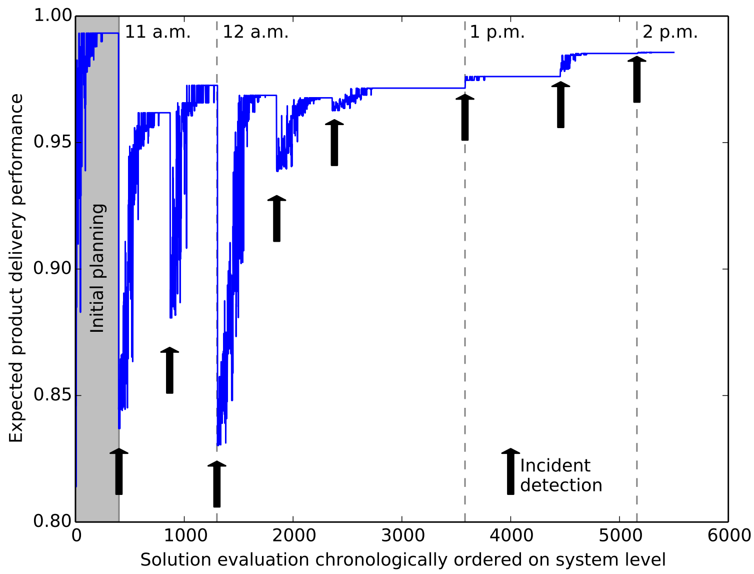

Fig. 6: Example for incident detection and expected product delivery performance.

In the first half of product delivery, each incident leads to a reduced product delivery performance first. The agents manage to enhance this value within the cooperative search for a new DVPP schedule though, thus enhancing the value. Convergence of the processes can be recognized from the plateaus right before the next incident. In the example given, the initial performance is not reached.

During the second half of product delivery, a very interesting effect is observed: with an incident, the product delivery performance does not decrease at all, but increases to a better value at once. This effect can be explained when examining the time span passed since begin of product delivery: The shorter the remaining product time span, the smaller is the expected effect of an outage.

In figure 7 the final product delivery performance is shown for all 100 runs per experimental setup and incidents configuration in a boxplot. The median is depicted with the blue line within the quartiles' box. The arithmetic mean is shown using a blue circle. The upper diagram shows the results for 0 to 10 CHP outages only, whereas the lower one shows the results of simulation runs with 4 PV prognosis deviations and 0 to 10 CHP outages combined. If no incidents are inserted, the final quality is about 0.99 for all runs (see upper left corner in figure 7). In both diagrams it can be seen, that the more incidents are inserted, the lower is the final performance. In all cases, the final product delivery performance exceeds the expected values after incident detection shown in figure 6: DynaSCOPE leads to better product delivery in all simulation runs.

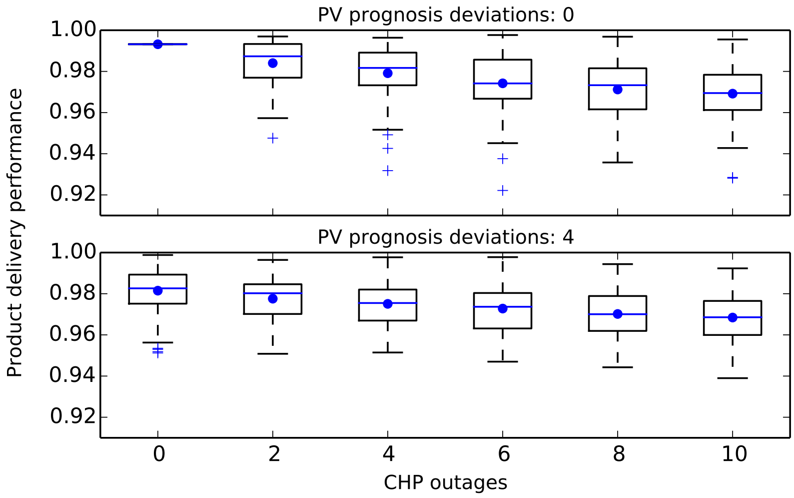

Fig. 7: Product delivery performance for a varying amount of incidents.

6.3 Communication Overhead

One aspect in evaluating the scalability of DynaSCOPE regarding the number of incidents detected in the field is the communication overhead in terms of mean number of messages sent by each agent. In figure 8 this value is set into relation with the number of incidents using boxplots. The same experimental setup was used as described in section 6.1. The maximum value can be found at about 550 with 14 incidents within the 4 hour product horizon. Furthermore, the number of messages grows in a sublinear manner. This can be explained from a DynaSCOPE detail not presented in detail in the work at hand: The more incidents are introduced in the 4-hour product horizon, the more incidents take place in the same planning interval. As the cooperative search processes are integrated by DynaSCOPE, the number of messages should raise sublinear with the number of incidents.

7 Conclusion and Future Work

In the work presented here, we introduced DynaSCOPE as a fully distributed continuous planning approach for decentralized energy units organized in DVPPs. DynaSCOPE has been developed by extending COHDA, a distributed optimization heuristic, to a distributed monitoring and control system. We evaluated DynaSCOPE using a simulative approach. It could be shown, that DynaSCOPE effectively enhances product delivery and thus allows for the delivery of an energy product with less reliable energy units prone to prognosis

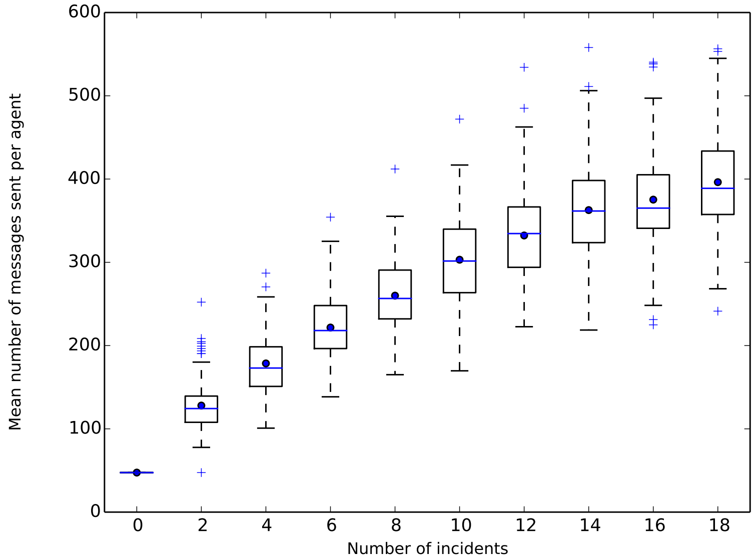

Fig. 8: Mean number of messages sent per agent by number of incidents inserted during product horizon.

deviations and outages. As a first distributed approach for this task following the DVPP concept, the performance of the system cannot be compared to other planning approaches yet. In the first iteration of Smart Grid Algorithm Engineering presented here, main findings have been retrieved, further motivating the extension of the system and narrowing the gap to an application in the field. In current work, we evolve DynaSCOPE within the open source energy unit aggregation, planning and control system Open VPP[4].

8 Acknowledgements

The Lower Saxony research network ‘Smart Nord’ acknowledges the support of the Lower Saxony Ministry of Science and Culture through the “Niedersächsisches Vorab” grant programme (grant ZN 2764).

References

[An10] Anders, Gerrit; Siefert, Florian; Steghöfer, Jan-Philipp; Seebach, Hella; Nafz, Florian; Reif, Wolfgang: Structuring and Controlling Distributed Power Sources by Autonomous Virtual Power Plants. In: Proceedings of the IEEE Power and Energy Student Summit (PESS). 2010.

[AYG96] Akkermans, Hans; Ygge, Frederik; Gustavsson, Rune: Homebots: Intelligent decentralized services for energy management. In: Fourth International Symposium on the Management of Industrial and Corporate Knowledge. Rotterdam, NL, 1996.

[4] http://openvpp.offis.de

[BS00] Bussmann, Stefan; Schild, Klaus: Self-Organizing Manufacturing Control : An Industrial Application of Agent Technology. In: Proc. of the 4th Int. Conf. on Multi-agent Systems (ICMAS'2000). Boston, MA, USA, pp. 87–94, 2000.

[BS13] Bremer, Jörg; Sonnenschein, Michael: Constraint-handling for optimization with support vector surrograte models. In: ICAART 2013. Barcelona, 2013.

[Ch11] Chapman, Archie C.; Rogers, Alex; Jennings, Nicholas R.; Leslie, David S.: A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems. The Knowledge Engineering Review, 26(04):411–444, November 2011.

[Gu99] Gustavsson, Rune: Agents with Power. Communications of the ACM, 42(3):41–47, 1999.

[Hi14] Hinrichs, Christian: Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid. PhD thesis, 2014.

[HLS14] Hinrichs, Christian; Lehnhoff, Sebastian; Sonnenschein, Michael: A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems. In: Operations Research Proceedings 2012. Springer, pp. 297–302, 2014.

[Le10] Lehnhoff, Sebastian: Dezentrales vernetztes Energiemanagement - Ein Ansatz auf Basis eines verteilten Realzeit-Multiagentensystems. Vieweg + Teubner, 2010.

[MP94] Müller, Jörg P.; Pischel, Markus: An architecture for dynamically interacting agents. Journal of Intelligent and Cooperative Information Systems, 3(1):23–45, 1994.

[Ni12] Nieße, Astrid; Lehnhoff, Sebastian; Tröschel, Martin; Uslar, Mathias; Wissing, Carsten; Appelrath, Hans-Jürgen; Sonnenschein, Michael: Market-based self-organized provision of active power and ancillary services. In: Complexity in Engineering (COMPENG), 2012. Aachen, 2012.

[Ni15] Nieße, Astrid: Verteilte kontinuierliche Einsatzplanung in Dynamischen Virtuellen Kraftwerken. PhD thesis, 2015.

[NTS13] Nieße, Astrid; Tröschel, Martin; Sonnenschein, Michael: Designing Dependable and Sustainable Smart Grids - How to Apply Algorithm Engineering to Distributed Control in Power Systems. Environmental Modelling & Software, 2013.

[Po13] Pournaras, Evangelos: Multi-level Reconfigurable Self-organization in Overlay Services (PhD thesis). TU Delft, 2013.

[RRJ09] Rahwan, Talal; Ramchurn, Sarvapali D.; Jennings, Nicholas R.: An Anytime Algorithm for Optimal Coalition Structure Generation. Journal of Artificial Intelligence Research, 34(1):521–567, 2009.

[Ta09] Talbi, El-Ghazali: Metaheuristics: From Design to Implementation. John Wiley & Sons, Inc., 2009.

[Tr10] Tröschel, Martin: Aktive Einsatzplanung in holonischen Virtuellen Kraftwerken. OlWIR, Oldenburger Verl. für Wirtschaft, Informatik und Recht, Oldenburg, 2010.

[We13] Weiss, Gerhard: Multiagent Systems. MIT Press, 2013.

[ZV12] Zhabelova, Gulnara; Vyatkin, Valeriy: Multiagent Smart Grid Automation Architecture Based on IEC 61850/61499 Intelligent Logical Nodes. IEEE Transactions on Industrial Electronics, 59:2351–2362, 2012.