

# Evaluation Guidelines for Asynchronous Distributed Heuristics in Smart Grid Applications

Christian Hinrichs<sup>1</sup>, Michael Sonnenschein<sup>2</sup>

## Abstract

In the context of Smart Grid applications, distributed control algorithms show advantageous properties over classical centralized approaches. Regarding their operation in a critical infrastructure, however, it is of utmost importance to validate the correct behavior of such approaches beforehand. In this paper, we give an overview on different aspects of evaluating Smart Grid applications, with a special focus on asynchronous distributed heuristics.

## 1. Introduction

A significant share of global CO<sub>2</sub> emissions can be explained by the combustion of fossil fuels for power production. Hence, it has become politically widely accepted in Europe, to reduce national shares of fossil fuels in power production significantly. Such a politically driven evolution of the power system faces not only economical and societal challenges, but it must also address several technological challenges of ensuring a highly reliable power supply, as described in e.g. [1]. In order to address these challenges, new concepts for power grid operation are needed. The notion of *Smart Grids* has been introduced for this purpose. The European Technology Platform for Electricity Networks of the Future defines a Smart Grid as an “electricity network that can intelligently integrate the actions of all users connected to it – generators, consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies.” [2] However, this implicates an increased computational complexity for optimizing the coordination of these individually configured, distributed actors. A significant body of research currently concentrates on this topic, see e.g. the research agenda proposed in [3].

In this context, the Smart Grid Algorithm Engineering (SGAE) process model introduces guidelines for application-oriented research and development in information and communication technology for power systems [4]. This envelops the phases Conceptualize, Design, Analyze, Implement, Experiment and Evaluate from a high-level perspective. In the contribution at hand, we focus on the “Analyze” and “Evaluate” parts in more detail. More specifically, we restrict our view to asynchronous distributed heuristics for solving optimization problems in Smart Grid applications. As the power supply system is a critical infrastructure, such approaches must be carefully evaluated in a secure environment before being implemented in the field. For gaining reliable results, however, this secure environment should reflect as many significant properties as possible of the targeted application area. Thus the objective of this contribution is to give an overview on the different aspects of evaluating asynchronous distributed heuristics for Smart Grid applications.

First of all, we will characterize the properties of asynchronous distributed approaches in section 2. From this, various evaluation criteria are derived in section 3, followed by a description of different methods for collecting and valuating these criteria in section 4. As a case study, section 5 then presents the evaluation coverage of an exemplary heuristic. Finally, section 6 concludes the paper.

---

<sup>1</sup> University of Oldenburg, 26111 Oldenburg, Germany, christian.hinrichs@uni-oldenburg.de, Department of Computing Science, Environmental Informatics

<sup>2</sup> University of Oldenburg, 26111 Oldenburg, Germany, sonnenschein@informatik.uni-oldenburg.de, Department of Computing Science, Environmental Informatics

## 2. Asynchronous Distributed Heuristics

In centrally organized systems, a single entity with global knowledge about a given objective and all involved components is in charge of calculating an appropriate solution for the objective. For example, the traditional power supply system can be seen as a centralized system. It consists of only a small number of controllable power plants. A control center acts as a central component that knows the operational constraints of the plants and stipulates the plants' reactions when deviations from the original operating plans occur. However, as already indicated in the introduction, such a control paradigm is not suitable for future Smart Grids anymore. It is widely accepted that the power supply system of the future will be characterized by a distributed architecture comprising autonomous components with individual sub-objectives, see e.g. [5–8]. In order to orchestrate those components towards global stability and reliability of the system, appropriate control mechanisms are necessary.

An example of such a problem—which we will again refer to in section 5—is the schedule optimization problem for autonomous distributed energy units (DEU) like generators, flexible loads or electrical storages: Given a target power profile, the task is to find a schedule assignment for each participating DEU over a specified planning horizon, such that the aggregation of all selected schedules directly corresponds to the predefined target power profile. This problem is commonly present in the day-ahead planning of dynamic virtual power plants [9]. Due to the inherent computational complexity of such optimization problems, heuristic approaches are being used in order to obtain a good solution for a problem as quick as possible. Moreover, such problems naturally require using *distributed* approaches. In the described schedule optimization problem, the DEU are autonomous entities. Thus the search space of the optimization problem is naturally distributed over the system, with each unit initially knowing only its own set of feasible schedules. To find an optimized schedule assignment with respect to the global goal, communication and coordination has to take place between units. Note that, while parallelization is another reason for using distributed heuristics, in order to accelerate the process or to increase solution quality [10, 11], we focus on naturally distributed problems here.

In general, a distributed heuristic for such a task defines what, when and with whom to communicate, and what to do with received information, in order to efficiently solve the problem in a distributed manner. Depending on the communication structure, the approach can further be classified as decentralized, hierarchical, distributed or fully distributed, c.f. [12]. Moreover, we may distinguish *synchronous* from *asynchronous* approaches [13]. The former are characterized by the existence of synchronization points. These define algorithmic phases, such that the coordinating actions of all components within a specific phase (e.g. calculations and communication) have to be completed before the next phase can start. Moreover, if the actions do not depend on each other within a single phase, this leads to a strong robustness against irregularities in the underlying communication system. In turn, those approaches usually compensate such irregularities with a larger run-time. On the other hand, asynchronous approaches are characterized by the absence of synchronization points. In these approaches, communication irregularities can have a severe impact on the overall progress, because they may change the order of actions that exert influence on each other. See [14] for a study regarding such effects on synchronous vs. asynchronous approaches.

Hence, additional guidelines have to be followed when choosing or designing an asynchronous distributed heuristic for a specific problem. So besides performance and efficiency in terms of e.g. solution quality, run-time or communication complexity, further criteria are necessary. These include convergence properties, robustness analyses and scalability predictions with regard to different problem-specific parameters. In the following section, we give an overview on such criteria.

### 3. Evaluation Criteria

Before presenting our taxonomy of evaluation criteria, we have to define a few terms. In compliance with the SGAE process model [4], we understand a *scenario* as a specific collection of Smart Grid components, which then constitute the actors that the heuristic under evaluation operates on. These components may be configured using a set of *parameters*. Then an *instance* of such a scenario is a parameter assignment for all components within the scenario. Finally, an *experiment* comprises one or more computational executions of a scenario instance.

With respect to their dimensionality, we classify evaluation criteria into zeroth-, first- and higher-order criteria. In this context, a zeroth-order criterion yields a basic decision, i.e. a yes-no answer, which should generally be independent of any scenario configuration. On the other hand, a first-order criterion provides a scalar quantity, which is usually the outcome of an experiment, i.e. the interpretation of experimental data from a scenario instance. Finally, higher-order criteria allow quantifying effects that occur due to interdependencies between different scenario instances and first-order criteria, yielding higher-order quantities such as vectors or matrices as output values. For this, series of experiments are necessary, in which one or more dependent scenario parameters are varied from experiment to experiment. We will now describe these types of criteria in more detail.

#### 3.1. Zeroth-Order Criteria

One of the most basic aspects to consider when dealing with heuristics that are targeted at the implementation in critical infrastructures is their *correctness* [15]. In the SGAE process model, this corresponds to the “Analyze” phase. First of all, showing correctness involves asserting that if the heuristic yields a solution, then this solution will satisfy a given specification, e.g. it is a valid solution for the given problem (partial correctness). An additional requirement is its termination, i.e. asserting that the heuristic terminates within a finite amount of time after it has been started (total correctness). In the field of distributed heuristics, this is also known as *guaranteed convergence*. Moreover, if this behavior additionally is independent of the system’s starting conditions, the heuristic is said to be *self-stabilizing* [16]. With respect to Smart Grid applications, one usually wants to show self-stabilization, as the involved autonomous components might be in arbitrary, unknown states when an optimization process is to be started. Moreover, as the occurrence of faults leads the system into arbitrary states, self-stabilization would allow such applications to recover from these faults autonomously.

#### 3.2. First-Order Criteria

The probably most evaluated criterion, however, is *performance*. The performance of a heuristic describes a quantification of its ability to achieve its goal [11]. Typically, this is measured in terms of solution quality, e.g. a fitness value that is calculated using an objective function. Here it is important to maintain a defined frame of reference, such that the measured value can be interpreted properly. For example, an adequate approach would be to determine the theoretically best and the theoretically worst solution for a given optimization problem as upper and lower bounds, and to normalize the fitness value to the interval that is spanned by these bounds. Apart from such general measurements, Smart Grid specific performance indicators play an important role to assess the performance of a heuristic in this field. Such performance indicators are yet to be defined and will be subject to future work (c.f. the “Conceptualize” phase in the SGAE process model).

Besides performance, the *efficiency* of a heuristic is of interest, which describes the resource requirements of a heuristic [11]. Regarding centralized approaches, this is usually measured in terms of run-time, e.g. the amount of “steps” an algorithm takes for a given input, and memory, e.g. the amount of storage capacity an algorithm consumes while processing its input. For distributed approaches, determining the efficiency is more complicated: Regarding run-time, we have to

distinguish the amount of time until the whole system terminates from the amount of “steps” the individual system components will take to reach this state. The former can be measured easily by means of real time, and will be an important information regarding the speed of the system in a specific hardware environment. The latter, however, is a more general measure as it determines the amount of work a system has to carry out. In this regard, a common practice is to count the number of calls to the objective function of the optimization problem, in each distributed component respectively. This way, both the individual work of the components as well as the overall effort can be determined in a hardware-independent manner. Finally, an additional evaluation criterion for distributed systems regarding the efficiency are communication expenses. As we are focusing on autonomous distributed components here, this leads to a message-passing paradigm (in contrast to a shared-memory model, in which multiple components possess a common working memory, c.f. [13]). Following, both the amount of exchanged messages as well as the size of these messages are significant factors for determining the efficiency of a heuristic.

### 3.3. Higher-Order Criteria

In this category, first-order criteria are evaluated against varying input parameters, i.e. changing scenario instances, in order to quantify correlation effects, or to perform a sensitivity analysis. In this regard, a prominent higher-order criterion is the *scalability* of an approach [17]. Here, the influence of a change in magnitude of input parameters on one or more relevant first-order criteria is determined. For example, given a centralized heuristic for calculating the schedule of energy resources for a future time horizon with respect to e.g. demand predictions, one could study the effects of the length of the considered planning horizon on the run-time of the heuristic. An example regarding distributed heuristics is the influence of the amount of autonomous components that are present in the system on communication expenses.

Another important higher-order criterion is *robustness* [17], which determines the influence of incidental disturbances from the environment on one or more first-order criteria. Such disturbances could be either “dynamic” incidents at run-time like e.g. varying message delays during the execution of a distributed system, or “static” perturbations that determine the sensitivity to changing starting conditions.

It is natural that higher-order criteria are rather difficult to analyze as they include lower-order criteria in different magnitudes. On the other hand, they are especially important when targeting critical infrastructures such as the power supply system.

## 4. Evaluation Methods

Each of the criteria introduced in the previous section can be evaluated using different methods. Here, analytical methods are distinguished from empirical methods [10].

### 4.1. Analytical Methods

In an analytical approach, evaluation criteria are quantified by mathematical calculus, i.e. inspecting the inherent design of the heuristic formally. For this, the semantic of the heuristic has to be described rigorously. An overview in this regard is given in [18, p. 27]. For example, deterministic sequential algorithms can be described using a *denotational semantic*, which primarily relies on fixed-point iterations for modelling loops and recursions. For nondeterministic or distributed algorithms, however, the *operational semantic* (also called *transition systems*, see [19]) is more suitable, as it relies on formulating transitions between configurations, or states, of a system and thus eases the modeling of interactions between distinct components. A popular example in this context is the *I/O automata* formalization [13], which explicitly models the behavior of different components of a system through a standardized interface and thus allows for

reasoning about the system's progress as a whole. Based on this, well-known proof techniques like e.g. *variant functions* or *convergence stairs* can be easily applied [16]. Another approach would be to employ automatic model checkers. Due to the numerous different semantic descriptions and methods that are available in this field, we refer to [20] for an introduction.

The above methods are particularly useful for zeroth-order evaluation criteria, e.g. for deriving convergence and termination properties. Recently, this has been adapted to first-order criteria as well. For example, in the context of self-organizing systems, [21] proposes quantitative definitions of the first-order criteria adaptivity, target orientation, homogeneity and resilience. These are based on an operational semantic in principle, which has been extended by *stochastic automata* though. This allows for modeling the system's behavior not only in extreme cases (i.e. the best and worst cases as in the evaluation of zeroth-order criteria), but also in the average case, which is crucial for quantifying first-order criteria. The deduced average case behavior, however, directly depends on the chosen distribution functions for the stochastic parts of the model. As a consequence, special care must be taken in order to properly reflect the real behavior of the modeled system when employing such a method. Hence, if adequate distribution functions for a given system cannot be derived easily, an empirical study might be more appropriate in these cases. This approach is described in the following section.

## 4.2. Empirical Methods

In contrast to formal reasoning based on a rigorous semantic description of an algorithm, empirical methods are based on actually executing the algorithm, i.e. the heuristic in the scope of this paper, within a dedicated environment. From monitoring such executions, quantitative data can be recorded, whose dissection and interpretation then leads to the valuation of first- and higher-order criteria.

This involves a number of subsequent steps: As a single execution of a heuristic usually does not yield enough information to deduce general conclusions about the behavior of the system in the average case, an adequate *experiment design* has to be defined in the first step ("Design of Experiments" in the SGAE process model [4]). Primarily, this includes tactical decisions, such as the number of repetitions of the executions, in order to level out random effects from uncertain environments or uncontrollable parameters. This will increase the confidence level of the deduced insights later on. Especially for higher-order evaluation criteria, additional strategic decisions have to be made, such as defining a strategy for the intentional variation of input parameters in order to analyze the heuristic's behavior under varying conditions. A comprehensive overview on these topics from the perspective of simulation experiments can be found in [22]. In the context of heuristics, additional care has to be taken regarding the type of scenario instance that is to be solved by a heuristic in a series of experiments [10]. While parts of this, like e.g. the magnitude of input parameters, are usually already covered in the described tactical and strategic decisions, the inherent type of an underlying problem instance might be of interest as well. In the SGAE process model, this corresponds to the "Scenario design" phase. Here, on the one hand, synthetically crafted problem instances can be used. These do not reflect the targeted application field, but are constructed in such a way that specific properties are present in the problem to solve. For example, "deceptive" problem instances [23] are useful to analyze whether a given heuristic is able to overcome local optima in the search space. This way, a deep understanding of the observed effects can be gained. On the other hand, application-specific problem instances aim at reflecting the target application of a heuristic as close as possible, such that the heuristic's behavior can be observed directly in its presumed environment.

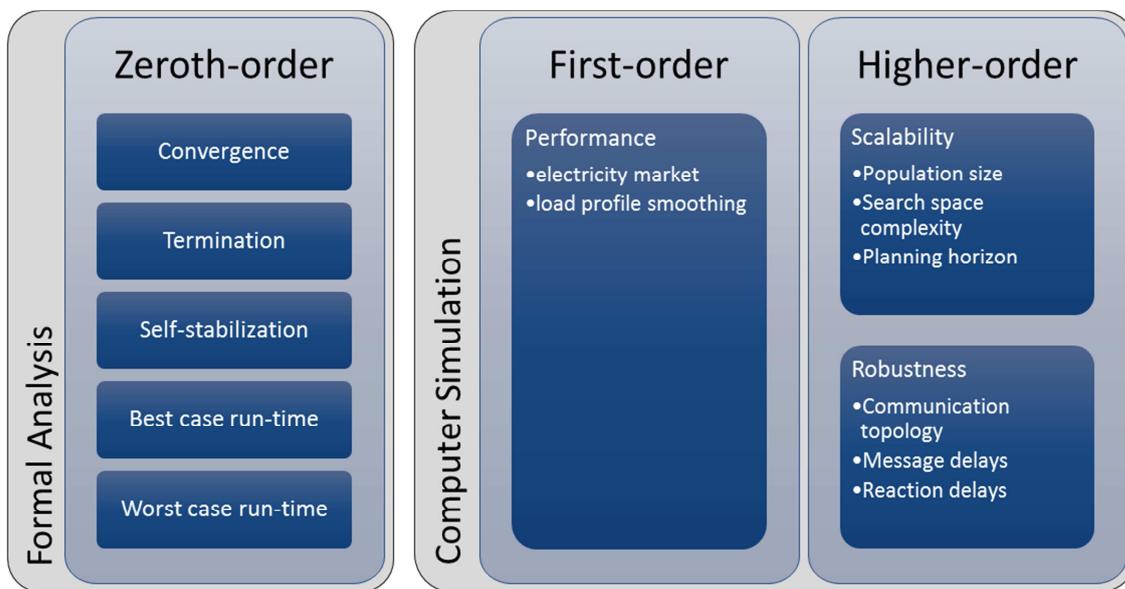
In the second step, the experiment is actually carried out. This can either be done in a physical test bed, or by means of computer simulation. Again, a physical test bed—if built properly—can provide a higher degree of realism regarding the targeted application field. But as this may be inappropriate due to pragmatic reasons like e.g. implementation costs, computer simulations are often used as a substitute for physical experiments. Moreover, computer simulations offer greater flexibility regarding the system configuration. According to J. Kleijnen, the core of a simulation is a *simulation model*, which is defined as a “dynamic model that is meant to be solved by means of experimentation.” [22] Regarding our focus on heuristic approaches for Smart Grid applications in this contribution, the simulation model for a computer simulation then comprises both the heuristic under evaluation and the environment this heuristic is executed in. Following, it is of utmost importance to build the model as realistic as needed, i.e. such that all relevant interdependencies between the (simulated) environment and the heuristic are incorporated into the model. For example, if a given distributed heuristic is said to be asynchronous based on message passing between components, possible flaws from the underlying communication technology such as message delays or buffer overflows should be anticipated. The other way around, if the outcome of a heuristic affects e.g. the power flow in an electricity grid, and the resulting effects are relevant for the evaluation, the grid must be modeled in such a way that those effects are properly accounted for. Again, [4] gives further suggestions regarding this topic. There, besides conceptual considerations, the modular Smart Grid simulation framework *mosaic* [24] is given as a tooling example in the SGAE process model. Moreover, we refer to textbooks such as [25, 26] for further reading.

Finally, in a third step, the preceding executions of the heuristic have to be analyzed with respect to the criteria of interest. Especially for higher-order criteria, specific metrics and suitable statistical methods can then be applied, in order to draw conclusions from the possibly vast amounts of recorded data. Examples for methods and metrics regarding various evaluation criteria can be found in [17, 10, 22].

## 5. Case Study

For an exemplification of the presented evaluation guidelines, we will in the following describe the evaluation process of the *Combinatorial Optimization Heuristic for Distributed Agents* (COHDA), which initially was published in [27] as a heuristic for the schedule optimization problem for autonomous DEUs that we introduced in section 2. Thus COHDA operates in a multi-agent system, where each agent represents a DEU with a private search space of feasible schedules. The goal is to select exactly one schedule for each DEU, such that a given target power profile is approximated by the sum of all selected schedules as close as possible. As the individual search spaces are to be kept private, the agents have to communicate via messages in order to coordinate towards a common solution. Basically, COHDA realizes an asynchronous iterative approximate best-response behavior, where each agent reacts to updated information from other agents, by adapting its own selected schedule with respect to the global target power profile.

The evaluation coverage for COHDA is depicted in figure 1. To prove the correctness of the heuristic in terms of convergence, termination and self-stabilization, a formal analysis has been conducted. For this, the approach has been described semantically in the I/O Automata framework [13], followed by formal reasoning using the convergence stairs method [16] (this proof will be published in a subsequent paper). As a side effect, the best case and worst case run-time could be determined in the process. Following, a simulation study has been conducted. Here, the performance of the heuristic was evaluated in two application specific scenarios: trading active power products in day-ahead electricity markets and load profile smoothing (these results are not published yet, but [9] and [28, sect. 3] provide more details on the respective tasks). Finally, the



*Figure 1: Evaluation coverage of the COHDA heuristic*

higher-order criteria scalability and robustness were evaluated for various parameters (c.f. figure 1) with respect to the first-order criteria performance (solution quality) and efficiency (run-time, computational expenses, communication expenses). Here, both synthetic and application specific scenarios have been employed, see [29, 30, 14].

## 6. Conclusion

The Smart Grid Algorithm Engineering (SGAE) process model [4] provides a foundation for the structured development of Smart Grid application algorithms. In the paper at hand, we explicate the “Analyze” and “Evaluate” phases of the model in more detail, with a specific focus on asynchronous distributed heuristics for solving optimization problems in Smart Grid applications. The main contribution of this paper is a taxonomy of evaluation criteria, followed by an overview of methods for valuating these criteria. Finally, we presented a case study regarding the evaluation of an exemplary heuristic for the schedule optimization problem for autonomous distributed energy units.

Future work in this context will be to define application specific performance indicators, such that Smart Grid application algorithms, especially distributed control approaches, can be developed and evaluated using standardized and accepted criteria from the problem domain.

## References

- [1] H.-J. Appelrath, H. Kagermann, and C. Mayer, Future Energy Grid. Springer Berlin Heidelberg, 2012.
- [2] European Technology Platform for Electricity Networks of the Future, “Strategic Deployment Document for Europe’s Electricity Networks of the Future,” 2010.
- [3] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, “Putting the ‘smarts’ into the smart grid,” Commun. ACM, vol. 55, no. 4, p. 86, Apr. 2012.
- [4] A. Nieße, M. Tröschel, and M. Sonnenschein, “Designing dependable and sustainable Smart Grids – How to apply Algorithm Engineering to distributed control in power systems,” Environ. Model. Softw. vol. 56, pp. 37–51, 2014.
- [5] F. F. Wu, K. Moslehi, and A. Bose, “Power System Control Centers: Past, Present, and Future,” Proc. IEEE, vol. 93, no. 11, pp. 1890–1908, Nov. 2005.
- [6] M. D. Ilić, “From Hierarchical to Open Access Electric Power Systems,” Proc. IEEE, vol. 95, no. 5, pp. 1060–1084, May 2007.

- [7] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1743–1752, 2007.
- [8] M. Uslar, M. Specht, C. Dänekas, J. Trefke, S. Rohjans, J. M. González, C. Rosinger, and R. Bleiker, *Standardization in Smart Grids*. Springer Berlin Heidelberg, 2013.
- [9] A. Nieße, S. Lehnhoff, M. Tröschel, M. Uslar, C. Wissing, H.-J. Appelrath, and M. Sonnenschein, "Market-based self-organized provision of active power and ancillary services: An agent-based approach for Smart Distribution Grids," in *2012 Complexity in Engineering (COMPENG)*. Proceedings, 2012.
- [10] E. Alba, *Parallel Metaheuristics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.
- [11] E.-G. Talbi, *Metaheuristics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.
- [12] M. Sonnenschein, C. Hinrichs, A. Nieße, and U. Vogel, "Supporting Renewable Power Supply through Distributed Coordination of Energy Resources," in *ICT Innovations for Sustainability. Advances in Intelligent Systems and Computing 310*, L. M. Hilty and B. Aebischer, Eds. Springer International Publishing. (In press.)
- [13] N. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.
- [14] C. Hinrichs and M. Sonnenschein, "The Effects of Variation on Solving a Combinatorial Optimization Problem in Collaborative Multi-Agent Systems," in *Multiagent System Technologies, 12th German Conference, MATES 2014, Stuttgart, Germany, September 23–25, 2014. Proceedings*. (In press.)
- [15] K. R. Apt, "Ten Years of Hoare's Logic: A Survey—Part I," *ACM Trans. Program. Lang. Syst.*, vol. 3, no. 4, pp. 431–483, 1981.
- [16] S. Dolev, *Self-Stabilization*. Cambridge, MA, USA: MIT Press, 2000.
- [17] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende, and W. R. Stewart, "Designing and reporting on computational experiments with heuristic methods," *J. Heuristics*, vol. 1, no. 1, pp. 9–32, 1995.
- [18] N. Francez, *Program Verification*. Addison-Wesley, 1992.
- [19] G. Tel, *Introduction to Distributed Algorithms*. Cambridge University Press, 1994.
- [20] M. Müller-Olm, D. A. Schmidt, and B. Steffen, "Model-Checking: A Tutorial Introduction," in *Proceedings of the 6th International Symposium on Static Analysis*, 1999, pp. 330–354.
- [21] R. Holzer and H. De Meer, "Quantitative modeling of self-organizing properties," in *Self-Organizing Systems. Lecture Notes in Computer Science*, vol. 5918, T. Spyropoulos and K. A. Hummel, Eds. Springer Berlin Heidelberg, 2009, pp. 149–161.
- [22] J. Kleijnen, *Design and Analysis of Simulation Experiments*. Springer US, 2008.
- [23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [24] S. Schütte and M. Sonnenschein, "Mosaik — Scalable Smart Grid scenario specification," in *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 2012.
- [25] A. K. Hartmann, *Practical Guide to Computer Simulations*. World Scientific Publishing Company, 2009.
- [26] F. Milano, *Power System Modelling and Scripting*. Springer Berlin Heidelberg, 2010.
- [27] C. Hinrichs, S. Lehnhoff, and M. Sonnenschein, "A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems," in *Operations Research Proceedings 2012*, 2014, pp. 297–302.
- [28] M. Sonnenschein, M. Tröschel, and O. Lünsdorf, "Smart Grids for Optimised Utilisation of Renewable Energy Supply," in *Environmental Informatics and Renewable Energies – 27th International Conference on Informatics for Environmental Protection*, 2013, pp. 178–187.
- [29] C. Hinrichs, M. Sonnenschein, and S. Lehnhoff, "Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces," in *ICAART 2013 – Proceedings of the International Conference on Agents and Artificial Intelligence*, 2013.
- [30] C. Hinrichs, J. Bremer, and M. Sonnenschein, "Distributed hybrid constraint handling in large scale virtual power plants," in *IEEE PES ISGT Europe 2013*, 2013.