IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 21, 2021, accepted August 31, 2021, date of publication September 6, 2021, date of current version September 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110607

Hybrid Optimization Toward Proactive Resilient Microgrid Scheduling

MICHAEL H. SPIEGEL[1] AND THOMAS I. STRASSER[1,2], (Senior Member, IEEE)

[1]Electric Energy Systems—Center for Energy, AIT Austrian Institute of Technology, 1210 Vienna, Austria
[2]Faculty of Mechanical and Industrial Engineering, Institute of Mechanics and Mechatronics, TU Wien, 1060 Vienna, Austria

Corresponding author: Thomas I. Strasser (thomas.strasser@ait.ac.at)

ABSTRACT Microgrids are one important lever to increase power system resilience and to tightly integrate renewable energies at the same time. Commonly, an optimization-based proactive scheduling controls assets in advance in a cost-effective way and ensures that contingencies may be successfully mitigated. However, often strong simplifications are introduced to manage the high computational complexity of scheduling, which can adversely impact fault mitigation. To consider essential phenomena such as power flow limitations and low-level control capabilities in detail, a novel hybrid scheduling approach is presented that integrates mathematical programming and arbitrary nonlinear constraint models via decision trees. A detailed case study compares the new method to an extended hybrid scheduling approach from literature. It is demonstrated that hybrid optimization can efficiently solve proactive resilient scheduling problems and that the tree-based algorithm provides a feasible solution, even in case the reference algorithm fails. Details on the convergence of both algorithms give further insights into the working principles and show that the novel method quickly finds a feasible solution that is successively improved afterwards. By the novel combination of highly-developed solvers for both mathematical programming and detailed asset models it is expected that this study further supports the operation of power systems and reduces costly reserve requirements.

INDEX TERMS Energy management system, heuristic optimization, hybrid optimization, microgrid scheduling, power system resilience, proactive resilient scheduling.

NOMENCLATURE

DEDICATED INDICES AND SETS

$\mathbb{D}_G$	Set of controllable Distributed Generators (DGs).
$\mathbb{L}_D$	Set of volatile loads and energy sources.
$\mathbb{S}_T$, $\mathbb{I}_G$	Set of storage and inverter-based plants.
$\mathbb{L}_I$, $\mathbb{B}_S$	Set of all lines and buses .
$\mathbb{T}$, t	Set and Index of time instants.
l, a, b	Index of loads, DG, and storage units, respectively.
$\mathbb{S}_C$, s	Set and index of scenarios.
$\mathbb{V}$	Set of violated constraints.

VARIABLES

$\vec{x}$	Generic vector of all model variables.
$\vec{x}^C$	Candidate solution of the scheduling problem.
$p_{a,t}^{DG}$	Active power scheduled for DG a at time t .
$o_{a,t}^{DG}$	Operational status of DG a at time t .
p_t^{NDG}	Nominal power of all scheduled DGs at time t.
$p_{b,t}^{CHG}$	Charged active power of storage b at time t.
$p_{b,t}^{DCH}$	Discharged active power of storage b at time t.
$e_{b,t}^{ST}$	Energy stored in b after time step t.
$o_{b,t}^{CHG}$	Charging mode of storage b at time t.
p_t^{BUY}	Active power bought from the main grid at time t.
p_t^{SELL}	Active power sold to the main grid at time t.
o_t^{SELL}	Main grid transfer direction at time t.
c_t^{TOT}	Total operating costs at time t.
$o_{i,t}^{STA}$	Startup indicator of asset i at time t.
p_t^{UP}	Scheduled up-spinning reserve at time t.
p^{MinUP}	Minimal up-spinning reserve.
o_t^{MinUP}	Minimal up-spinning reserve indicator at time t.

The associate editor coordinating the review of this manuscript and approving it for publication was Shafi K. Khadem.

o_i^{TR}	i-th auxiliary variable to encode tree constraints.
$P^v_{i,t,s}$, $Q^v_{i,t,s}$	Total active/reactive power of asset i at time t and in scenario s, having type $v \in \{\mathrm{DG}, \mathrm{ST}, \mathrm{PV}, \mathrm{WT}\}$.
$S_{i,t,s}$	Total apparent power of asset i at t in s.
$U_{i,t,s}$	Voltage at bus i at time t and scenario s .
$I_{i,t,s}$	Current in line i at time t and scenario s .
$\tilde{v}$	Normalized counterpart of variable or parameter v.

FUNCTIONS

$c(\vec{x})$	Costs of a schedule $\vec{x}$.
$\vec{g}^{\,1}(\vec{x})$	Set of linear constraint functions.
$\vec{g}^{\,\mathrm{n}}(\vec{x})$	Set of nonlinear constraint functions.
$\vec{g}_h^{z,\mathbb{H}}$	Heuristically defined partition of $\vec{g}^z$.
$G^z(\vec{x})$	Constraint violation level of $\vec{g}^z$.
$P(e)$	Probability of event e.
$\mathrm{dec}(\mathcal{T})$	Root decision of subtree $\mathcal{T}$.
$\mathrm{left}(\mathcal{T})$	Active subtree of tree $\mathcal{T}$, iff $\mathrm{dec}(\mathcal{T})(\vec{x}) \leq 0$.
$\mathrm{right}(\mathcal{T})$	Active subtree of tree $\mathcal{T}$, iff $\mathrm{dec}(\mathcal{T})(\vec{x}) \gtrsim 0$.

PARAMETERS AND CONSTANTS

$\bar{v}$, $\underline{v}$	Upper and lower limit of variable v.
ΔT	Duration of a single scheduling interval.
$p^{\mathrm{LD}}_{l,t}$	Expected active power demand of generalized load l at time t.
$o^{\mathrm{CHG}}_{b,-1}$	Initial charging mode of storage b.
μ^{ST}_b	Average round-trip efficiency of storage b.
$e^{\mathrm{ST}}_{b,-1}$	Initial energy in storage b.
$e^{\mathrm{ST}}_{b,\lvert\mathbb{T}\rvert}$	Energy in storage b at the end of the time horizon.
$o^{\mathrm{DG}}_{a,-1}$	Initial operational status of DG a.
c^{DG}_a	Cost of producing one unit of energy in DG a.
c^{BUY}_t	Cost of buying one unit of energy at time t.
c^{SELL}_t	Cost of selling one unit of energy at time t.
$p^{\mathrm{OP}}_{a,t}$	Forced operating point of asset i at time t.
n^{STA}_i	Number of permitted starts on asset i.
K^v_a	Droop gain or function of asset a and type v.
M	A sufficently large big-M constant.
ϵ	A small but strictly positive constant.

I. INTRODUCTION

Microgrids are considered as one solution to increase power system resilience, to tightly integrate volatile Renewable Energy Sources (RES) and to fully leverage the economic potential of Distributed Energy Resources (DERs) [1]. Although there are other definitions as well, this work follows [2] and considers microgrids as tightly controlled electrical networks that can be operated in both grid-connected and islanded mode. Due to the great flexibility that is provided by many microgrids, considerable potential is given for a scheduling algorithm to optimize the operation [3]. Resilient scheduling in emergency mode, for instance, often reduces the impact of a contingency without primarily targeting operating costs. In contrast, proactive resilient scheduling algorithms minimize the normal operation cost while no failure is observed, but at the same time, they prepare the network to gracefully degrade in case of contingencies. According to related work, resilience is considered as the ability to reduce the impact of potentially harmful events and includes both a fully robust but also a gracefully degraded operation [4].

A detailed review of proactive scheduling approaches is published in [4] noting that, although every algorithm follows an optimization-based methodology, a broad variety of solution approaches is observed. Common scheduling techniques include mathematical methods (like Mixed Integer Linear Programming (MILP) formulations) that can be solved by generic software components and heuristic approaches such as genetic algorithms [5]. As demonstrated in this work, the broad variety of approaches directly relates to the high computational complexity of microgrid scheduling that leaves room for various specialized methods including heuristics.

Since control and scheduling decisions can have a considerable impact on the safe operation of a microgrid [6], some authors included physical constraints such as voltage limits in their scheduling formulations. Due to the inherently nonconvex nature of physical power flows, mostly linearization is used to fit the MILP or heuristic optimization techniques to solve the nonlinear mixed-integer problem [4]. The former technique can suffer from linearization errors causing overapproximation or reduced confidence in the eligibility of results while the latter approach cannot fully utilize the potential of highly-developed solvers for mathematical programming problems. However, literature sparsely indicates under which circumstances one merit outweighs the other.

Most work formulates the proactive scheduling problem as one monolithic set of equations without discussing external implementations of asset models in detail [4], [7]. For instance, in [8], a linearized version of the power flow equations is directly integrated into the scheduling model. Several studies including [9] using Benders decomposition, are partitioning the models into subproblems to efficiently solve them. Although the monolithic formulation and its decomposition gives full access to details such as derivatives, engineering efforts of formulating system constraints can be considerably eased by relying on well-proven and accepted external simulation models [10], [11].

In [10], a security-constrained optimal dispatch approach is presented that uses a heuristic multi-objective optimization technique. An external power system simulator and normalized constraint violation levels are used to filter infeasible candidate solutions. In addition to static voltage and current margins, which are also reflected in related works, [10] includes transient voltage and frequency stability constraints.

However, the problem formulation is restricted to unit commitment without considering the operational status of DER.

A manual decomposition into an outer multi-objective problem that manages storage schedules and inner optimizations scheduling the other resources, both in normal and emergency operation, is presented in [12]. The nested MILP problems consider power flows by iteratively updating a power loss constant, in case the power flow does not indicate any physical constraint violation. In case a violation is encountered, the whole subproblem including its linear equations is solved by an optimal power flow solver. Despite using the external power flow optimization, less emphasis is put on the complexity of handling nonlinear constraints as most schedules are assumed to be feasible.

Also [11] described the integration of an external power system simulator to enforce voltage and current constraints in resilient scheduling. In contrast to [10], a quadratic-programming formulation with linear constraints is extended by constraints derived from a sensitivity analysis on the grid model. Both, the iterative scheme as well as the structure of added constraints can also be found in Benders decomposition as applied in [9]. However, [9] uses (integer) linear programs that cannot handle the nonlinear power flows in [11], and both approaches differ in their constraint generation.

Given the interaction of mathematical programming and the constraint enforcement heuristic, [11] successfully demonstrates the application of a hybrid optimization approach, i.e., a combination of diverse algorithmic components [13], in resilient microgrid scheduling. Still, the external constraints were only applied in a single time interval of the multi-period optimization problem. Questions regarding applicability in multi-period constraints and approximation errors remain open. Despite the detailed power system models, the effects of low-level controls such as voltage and frequency droop on the feasibility of a particular candidate schedule are hardly covered [10], [11].

A. CONTRIBUTIONS TO MICROGRID SCHEDULING

This work studies the application of external constraint models in proactive, resilient microgrid scheduling and proposes a novel hybrid optimization method to solve MILP models with external, nonlinear constraints. A common MILP basis formulation for microgrid scheduling in conjunction with external nonlinear constraints is developed. Based on the common formulation, two scheduling approaches are presented. The first one addresses the state-of-the-art by extending the sensitivity-based constraint learning technique of [11] to multi-period resilience constraints. The second one explores novel paths in hybrid scheduling by utilizing machine learning techniques to approximate the constraint surface within the MILP. To the best of our knowledge, an adapted version of the constraint synthesis technique in [14] for the first time iteratively links a stochastic local search and the global MILP scheduling problem.

A case-study is used to thoughtfully evaluate and compare both approaches on common ground. In contrast to related work, the case study demonstrates both the application of external grid models to formulate constraints and the influence of low-level controls on the feasibility of schedules. Further insights into the effects of problem formulation and decomposition are presented. Following the unavailability of a universal optimization strategy [15], several cases in which the sensitivity-based approach fails to deliver good or even any feasible solutions could be identified. At the same time it is demonstrated that due to the more powerful approximation model, the novel scheduling approach can deliver excellent results, even if the sensitivity-based one fails.

B. ORGANIZATION

This work is organized as follows. In Section II, the problem of proactive, resilient microgrid scheduling is described and the formulation of the subsequent study is developed. Section III utilizes the problem definition to define two methods of solving the scheduling task including external resilience constraints. A case study in Section IV compares the approaches and studies their performance under several problem variations. In Section V, results are discussed and finally, Section VI concludes the findings.

II. RESILIENT SCHEDULING PROBLEM FORMULATION

In this study, it is assumed that a proactive, resilient scheduling algorithm centrally computes the set points of all controllable assets based on the current operational status and the most recent forecasts. In contrast to the related work that presents a broad variety of different assets including Electric Vehicles (EVs), micro turbines, and controllable loads [9], [16], [17], this work focuses on the most generic assets in order to facilitate performance analysis and comparability. It is assumed that the microgrid hosts exactly two types of schedulable assets. The first one groups generic DERs that can be independently scheduled for each time interval. The second one comprises Electrical Energy Storages (EESs) that do have an internal state of charges that depends on previous scheduling decisions. Furthermore, it is assumed that the microgrid includes volatile RES, which are providing basic voltage control capabilities. However, the active power output and demand of volatile RES and loads, respectively, are assumed to show a stochastic behavior. For each stochastic quantity, it is assumed that appropriate deterministic forecasts are available, but that the realizations are unknown at scheduling time.

Each scheduling run optimizes the asset set points over a finite time horizon. Although some authors presented an iterative scheme that repeats a scheduling operation and only applies the most recent set points [11], this performance analysis avoids any bias due to erroneously correlated updates. Hence, it focuses on one single scheduling run without taking update mechanisms into account, but it does not prevent the application in an updating scheme. The proactive algorithm itself is executed before any contingency is encountered [18]. However, in the presence of general security policies or early warning signs, the microgrid is actively prepared to sustain

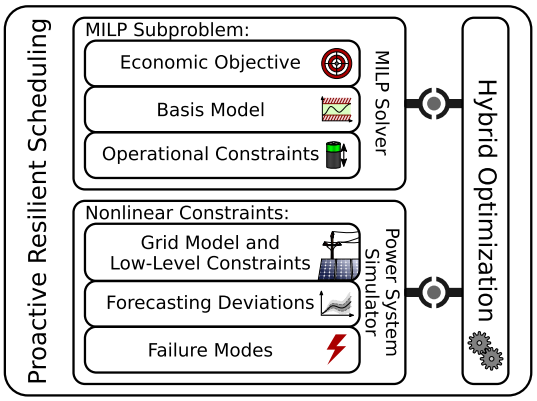

FIGURE 1. Decomposition of the proactive resilient scheduling problem into individual linear and nonlinear models.

catastrophic events and to increase power system resilience. In contrast to robustness, resilience permits a degraded operation, for instance by shedding noncritical loads, in case of contingencies. Following the definition of [18], power system resilience may even be addressed in case the scheduling algorithm manages to avoid any degraded state at all. Since scheduling is executed in the preparation phase, it is assumed that the microgrid is connected to a healthy main grid and that any countermeasures such as islanding the microgrids or parts thereof are conducted by separate emergency mechanisms. Yet, the scheduling algorithm can influence the performance of countermeasures for instance, by reserve commitment and relocation [17].

Fig. 1 gives an overview of the scheduling problem including the individual models and highlights the role of hybrid optimization techniques to solve them. To generalize the optimization procedures, the whole problem is firstly generically modeled in Section II-A and later refined by the detailed model in Sections II-B to II-D. Economic aspects are covered by a single-node MILP formulation and first resilience aspects are included by the deterministic reserve constraints in Section II-C. In more detail, a resilient operation is covered by the grid simulation scenarios in Section II-D that include detailed effects of failure modes (e.g., line outages) and mitigation techniques such as partial islanding.

A. GENERALIZED PROACTIVE SCHEDULING

In order to efficiently solve the problem of finding an optimal schedule $\vec{x}$, a decomposition into an objective function $c(\vec{x})$, as well as a set of linear constraints $\vec{g}^{\,\mathrm{l}}(\vec{x})$ and one of nonlinear constraints $\vec{g}^{\,\mathrm{n}}(\vec{x})$ is applied. The entire scheduling problem is defined as finding a schedule $\vec{x}$ that minimizes $c(\vec{x})$ such that both (1) and (2) are satisfied.

$$g_i^{\mathrm{l}}(\vec{x}) \leq 0 \quad \forall i = 1, \ldots, |\vec{g}^{\,\mathrm{l}}| \tag{1}$$

$$g_i^{\mathrm{n}}(\vec{x}) \leq 0 \quad \forall i = 1, \ldots, |\vec{g}^{\,\mathrm{n}}| \tag{2}$$

A shorthand notation $\vec{g}^{\,\mathrm{ln}}$ is used to address all constraints $\vec{g}^{\,\mathrm{ln}} = \vec{g}^{\,\mathrm{l}} \cup \vec{g}^{\,\mathrm{n}}$ at once. Note that, according to common practice, equality constraints are transformed into inequality constraints, without loss of generality [19]. The objective function $c(\vec{x})$ must be in a form such that the subproblem of minimizing $c(\vec{x})$ considering constraints (1) can be efficiently solved by mathematical programming. In particular, the study focuses on MILP subproblems.

All constraints that should be included in full detail are subsumed by the nonlinear functions $\vec{g}^{\,\mathrm{n}}(\vec{x})$. Typical representatives include voltage and current constraints considering the effects of low-level controls on the actual asset set points. Due to the decomposition, $\vec{g}^{\,\mathrm{n}}(\vec{x})$ can be computed via external power system simulations and does not have to be present in closed form. In practice, evaluating the nonlinear constraint can even involve external solvers, e.g., to compute power flows. Consequently, it is assumed that the function evaluation $\vec{g}^{\,\mathrm{n}}(\vec{x})$ is computationally intensive and should not be excessively triggered by a heuristic algorithm. Decomposition of a scheduling problem, as it is implicitly presented in related work such as [11], involves decisions on either using a simplified version or including the full complexity of the physical model. Due to the effort of evaluating and coupling $\vec{g}^{\,\mathrm{n}}(\vec{x})$, the coexistence of simplified versions in $\vec{g}^{\,\mathrm{l}}(\vec{x})$ as well as detailed counterparts in $\vec{g}^{\,\mathrm{n}}(\vec{x})$ can further improve the scheduling performance.

B. MILP BASIS FORMULATION

To facilitate a detailed analysis of the scheduling algorithm, a simple MILP formulation based on a deterministic power-balance model and insights listed in review [20] is developed. In practice, stochastic or robust formulations such as [12] and [8] may be included to account for the inherent uncertainties in microgrid scheduling. However, a detailed analysis of such models is well beyond the scope of this work. The state of each Distributed Generator (DG) unit $a \in \mathbb{D}_{\mathrm{G}}$ at time $t \in \mathbb{T}$ is modeled by two variables, the operation status $o_{a,t}^{\mathrm{DG}} \in \mathbb{B}$ and the scheduled power output $p_{a,t}^{\mathrm{DG}} \in \mathbb{R}$. The operating range of each machine is directly constrained by (3).

$$\underline{p}_a^{\mathrm{DG}} \cdot o_{a,t}^{\mathrm{DG}} \leq p_{a,t}^{\mathrm{DG}} \leq \bar{p}_a^{\mathrm{DG}} \cdot o_{a,t}^{\mathrm{DG}} \quad \forall a \in \mathbb{D}_{\mathrm{G}}, t \in \mathbb{T} \tag{3}$$

Similarly, each EES unit $b \in \mathbb{S}_{\mathrm{T}}$ is modeled by an operating mode $o_{b,t}^{\mathrm{CHG}} \in \mathbb{B}$ indicating whether b is allowed to charge, as well as the charging and discharging power $p_{b,t}^{\mathrm{CHG}}, p_{b,t}^{\mathrm{DCH}} \in \mathbb{R}$, respectively. Additionally, the usable energy that is stored in b after the scheduling interval t, i.e., right before $t+1$, is modeled by $e_{b,t}^{\mathrm{ST}}$. A constant round-trip efficiency μ_b^{ST} is applied while charging and models all internal losses. Equations (4) to (6) model the corresponding constraints.

$$0 \leq p_{b,t}^{\mathrm{CHG}} \leq \bar{p}_b^{\mathrm{CHG}} \cdot o_{b,t}^{\mathrm{CHG}} \quad \forall b \in \mathbb{S}_{\mathrm{T}}, t \in \mathbb{T} \tag{4}$$

$$0 \leq p_{b,t}^{\mathrm{DCH}} \leq \bar{p}_b^{\mathrm{DCH}} \cdot \left(1 - o_{b,t}^{\mathrm{CHG}}\right) \quad \forall b \in \mathbb{S}_{\mathrm{T}}, t \in \mathbb{T} \tag{5}$$

$$e_{b,t}^{\mathrm{ST}} = e_{b,t-1}^{\mathrm{ST}} + \left(p_{b,t}^{\mathrm{CHG}} \cdot \mu_b^{\mathrm{ST}} - p_{b,t}^{\mathrm{DCH}}\right) \cdot \Delta T \quad \forall b \in \mathbb{S}_{\mathrm{T}}, t \in \mathbb{T} \tag{6}$$

Although the approach does not exclude multiple Points of Common Coupling (PCCs), it is assumed that all PCCs access

a single market and that power flows in the simplified MILP formulation are bound by total transfer capabilities. To model different prices for buying and selling energy, the power transfer from or to the main grid is split into two variables, $p_t^{\mathrm{BUY}} \in \mathbb{R}$ and $p_t^{\mathrm{SELL}} \in \mathbb{R}$, respectively. The directional indicator variable $o_t^{\mathrm{SELL}} \in \mathbb{B}$, as well as constraints (7) and (8) ensure mutual exclusiveness.

$$0 \leq p_t^{\mathrm{BUY}} \leq \bar{p}^{\mathrm{BUY}} \cdot \left(1 - o_t^{\mathrm{SELL}}\right) \quad \forall t \in \mathbb{T} \tag{7}$$

$$0 \leq p_t^{\mathrm{SELL}} \leq \bar{p}^{\mathrm{SELL}} \cdot o_t^{\mathrm{SELL}} \quad \forall t \in \mathbb{T} \tag{8}$$

The economic evaluation of a schedule follows a deterministic single-node power balance (9) scheme without considering losses.

$$\sum_{a \in \mathbb{DG}} p_{a,t}^{\mathrm{DG}} - \sum_{l \in \mathbb{LD}} p_{l,t}^{\mathrm{LD}} + p_t^{\mathrm{BUY}} - p_t^{\mathrm{SELL}} + \sum_{b \in \mathbb{ST}} \left(p_{b,t}^{\mathrm{DCH}} - p_{b,t}^{\mathrm{CHG}}\right) = 0 \quad \forall t \in \mathbb{T} \tag{9}$$

Variable day-ahead market prices c_t^{BUY} and c_t^{SELL} that are known at scheduling time, as well as the average production costs c_a^{DG} of each DG a determine the operation expenses c_t^{TOT} at each time instant t. Equations (10) and (11) define the cost function $c(\vec{x})$ of a schedule $\vec{x}$. Constraints (3) to (9) describe the base set of linear constraints $\vec{g}^{\,\mathrm{l}}(\vec{x})$.

$$c_t^{\mathrm{TOT}} = c_t^{\mathrm{BUY}} \cdot p_t^{\mathrm{BUY}} - c_t^{\mathrm{SELL}} \cdot p_t^{\mathrm{SELL}} + \sum_{a \in \mathbb{DG}} c_a^{\mathrm{DG}} \cdot p_{a,t}^{\mathrm{DG}} \quad \forall t \in \mathbb{T} \tag{10}$$

$$c(\vec{x}) = \sum_{t \in \mathbb{T}} c_t^{\mathrm{TOT}} \tag{11}$$

C. EXTENDED MILP FORMULATION

Starting from the scheduling basis formulation that focuses on the most basic model, a set of optional constraints is developed to study the impact of model complexity on the performance of heuristic approaches. The first set of constraints limits the operating region of specific assets to incorporate needs that do not directly follow from technical asset limits. Practical applications of these additional operating limits $\bar{p}_{a,t}^{\mathrm{OP}}$ and $\underline{p}_{a,t}^{\mathrm{OP}}$ include thermal demand of a Combined Heat and Power (CHP) plant and local reserve policies. Although the operation constraints are defined on a subset $\mathbb{X}$ of assets and time, $\mathbb{X} \subseteq \mathbb{D}_{\mathrm{G}} \times \mathbb{T}$, (12) does not model any interdependence between assets a and time instants t.

$$\underline{p}_{a,t}^{\mathrm{OP}} \leq p_{a,t}^{\mathrm{DG}} \leq \bar{p}_{a,t}^{\mathrm{OP}} \quad \forall a, t \in \mathbb{X} \subseteq \mathbb{D}_{\mathrm{G}} \times \mathbb{T} \tag{12}$$

A set of dynamic constraints that link variables among instants of time is introduced by restricting the number of startup and charging operations to avoid excessive wear out. Similarly, a minimum number of startups may force an asset into operation. For both, DG and storage units, the auxiliary variable $o_{i,t}^{\mathrm{STA}} \in \mathbb{B}$ indicates whether asset i activated its operation mode at time instant t. Given the indicator constraints (13) and (14), the number of startup operations can be restricted by (15).

$$o_{a,t}^{\mathrm{DG}} - o_{a,t-1}^{\mathrm{DG}} \leq o_{a,t}^{\mathrm{STA}} \leq \frac{1}{2}\left(1 + o_{a,t}^{\mathrm{DG}} - o_{a,t-1}^{\mathrm{DG}}\right) \quad \forall a \in \mathbb{D}_{\mathrm{G}}, t \in \mathbb{T} \tag{13}$$

$$o_{b,t}^{\mathrm{CHG}} - o_{b,t-1}^{\mathrm{CHG}} \leq o_{b,t}^{\mathrm{STA}} \leq \frac{1}{2}\left(1 + o_{b,t}^{\mathrm{CHG}} - o_{b,t-1}^{\mathrm{CHG}}\right) \quad \forall b \in \mathbb{S}_{\mathrm{T}}, t \in \mathbb{T} \tag{14}$$

$$\underline{n}_i^{\mathrm{STA}} \leq \sum_{t \in \mathbb{T}} o_{i,t}^{\mathrm{STA}} \leq \bar{n}_i^{\mathrm{STA}} \quad \forall i \in \mathbb{D}_{\mathrm{G}} \cup \mathbb{S}_{\mathrm{T}} \tag{15}$$

A linear reserve model is introduced to ensure that critical loads $\mathbb{L} \subset \mathbb{L}_{\mathrm{D}}$ can be supplied in case of main grid failures. To simplify the discussion, it is assumed that storage units are grid following devices that are only used for energy arbitrage. The reserve model itself consists of three metrics, the nominal power of all DGs in the primary reserve, $p_t^{\mathrm{NDG}} \in \mathbb{R}$, the up-spinning reserve $p_t^{\mathrm{UP}} \in \mathbb{R}$ at time t, and the minimum up-spinning reserve in the entire scheduling horizon $p^{\mathrm{MinUP}} \in \mathbb{R}$. The modeled metrics can be used to manually enforce sufficiency or to link external models as shown in Section III. To study the latter use-case, a verbose formulation that does not use a relaxed lower-bound of p^{MinUP} was chosen. Constraints (16) to (18) consequently model the basic primary reserve requirements.

$$p_t^{\mathrm{NDG}} = \sum_{a \in \mathbb{DG}} \bar{p}_a^{\mathrm{DG}} \cdot o_{a,t}^{\mathrm{DG}} \quad \forall t \in \mathbb{T} \tag{16}$$

$$p_t^{\mathrm{UP}} = p_t^{\mathrm{NDG}} - \sum_{l \in \mathbb{LD}} p_{l,t}^{\mathrm{LD}} + \sum_{b \in \mathbb{ST}} \left(p_{b,t}^{\mathrm{DCH}} - p_{b,t}^{\mathrm{CHG}}\right) \quad \forall t \in \mathbb{T} \tag{17}$$

$$\underline{p}_t^{\mathrm{UP}} \leq p_t^{\mathrm{UP}} \leq \bar{p}_t^{\mathrm{UP}} \quad \forall t \in \mathbb{T} \tag{18}$$

To model the minimum spinning reserve on the scheduling horizon, a set of binary auxiliary variables $o_t^{\mathrm{MinUP}} \in \mathbb{B}$ are indicating whether the minimum reserve is reached at time t and constraints (19) to (21) are introduced.

$$p^{\mathrm{MinUP}} \leq p_t^{\mathrm{UP}} \quad \forall t \in \mathbb{T} \tag{19}$$

$$p_t^{\mathrm{UP}} - p^{\mathrm{MinUP}} \leq M \cdot \left(1 - o_t^{\mathrm{MinUP}}\right) \quad \forall t \in \mathbb{T} \tag{20}$$

$$\sum_{t \in \mathbb{T}} o_t^{\mathrm{MinUP}} = 1 \tag{21}$$

Constant M needs to be chosen such that it exceeds any left-hand-side value of (20). Note that the nonlinear constraints may imply the MILP reserve formulation and therefore (13) to (21) are also used to study the effects of redundancies.

D. GRID MODEL AND LOW-LEVEL CONTROLS

In contrast to other publications such as [8] that presented a linearized form of the static power flow equations, it is assumed that the grid model is covered by the nonlinear constraint set $\vec{g}^{\,\mathrm{n}}$ in detail. As long as the constraint function is computable, the algorithms do not require any specific model structure and may include balanced and unbalanced

steady-state models to assess asset loading and voltage limits, as well as transient models to ensure a stable operation in case of failures [10]. However, to support a detailed performance analysis, grid constraints based on a series of static power flows are derived. Due to inherent uncertainties, it is assumed that the grid constraints extend the deterministic formulation of the linear subproblem by considering forecasting deviations and failure conditions via a set of scenarios $\mathbb{S}_C$. Methods to find a few representative scenarios such as [21], are, however, beyond the scope of this work and manually selected boundary-cases are added.

For each scenario $s \in \mathbb{S}_C$ and each time step t, the balanced AC power flow equations are solved by a Newton-Raphson algorithm as described in [22]. In addition to the scheduled active power $p^{DG}_{a,t}$ of each asset a, it is assumed that each active unit a on island c participates in frequency control and provides steady-state balancing power according to its droop gain K^f_a and the frequency deviation $\Delta f_{c,t,s}$. Note that $\Delta f_{c,t,s}$ is set such that the active power on each island c is balanced [22]. The reactive power $Q^{DG}_{a,t,s}$ of each generator is determined based on a piece-wise linear droop curve $K^u_a(U_{a,t,s})$, where $U_{a,t,s}$ is the voltage at the bus connecting asset a [23]. Equations (22) and (23) summarize the injected power for each DG.

$$P^{DG}_{a,t,s} = p^{DG}_{a,t} + o^{DG}_{a,t} \cdot K^f_a \cdot \Delta f_{c,t,s} \quad \forall a \in \mathbb{D}_G \tag{22}$$

$$Q^{DG}_{a,t,s} = o^{DG}_{a,t} \cdot K^u_a(U_{a,t,s}) \quad \forall a \in \mathbb{D}_G \tag{23}$$

In contrast to DGs that participate in both primary frequency and voltage control, it is assumed that all inverter-based DERs $\mathbb{I}_G$ as given in (24) limit voltage control if the nominal apparent power $\bar{S}^{IG}_l$ is exceeded.

$$Q^{IG}_{l,t,s} = \begin{cases} K^u_l(U_{l,t,s}) & S_{l,t,s} \le \bar{S}^{IG}_l \\ \bar{Q}^{IG}_{l,t,s} & \text{otherwise} \end{cases} \quad \forall l \in \mathbb{I}_G \tag{24}$$

Given the results of the AC power flow, including the currents in each line i, $I_{i,t,s}$, the apparent power $S_{a,t,s}$ of all assets a, and the voltage magnitude $U_{j,t,s}$ of all buses j, the set of constraints $\vec{g}^{\,n}$ can be summarized as (25).

$$\vec{g}^{\,n} = \begin{pmatrix} I_{i,t,s} - \bar{I}_i & i \in \mathbb{L}_I, t \in \mathbb{T}, s \in \mathbb{S}_C \\ S_{a,t,s} - \bar{S}_a & a \in \mathbb{I}_G \cup \mathbb{D}_G, t \in \mathbb{T}, s \in \mathbb{S}_C \\ U_{j,t,s} - \bar{U}_j & j \in \mathbb{B}_S, t \in \mathbb{T}, s \in \mathbb{S}_C \\ \underline{U}_j - U_{j,t,s} & j \in \mathbb{B}_S, t \in \mathbb{T}, s \in \mathbb{S}_C \end{pmatrix} \tag{25}$$

To increase the expressiveness of the constraint model and to guide a heuristic procedure, the constraints z can be divided into several partitions, $\vec{g}^{\,z,\mathbb{H}}_h$, $h \in \mathbb{H} \cup \{\emptyset\}, z \in \{l, n, ln\}$ of $\vec{g}^{\,z}$, to express the heuristic dependence on values of $\mathbb{H}$. For instance, $\vec{g}^{\,n,\mathbb{T}}_t$ groups all nonlinear constraints that strongly depend on the state at time instant t or do not show any such heuristically defined dependency ($t = \emptyset$). Without loss of generality, the external constraints do not expose internal model variables such as voltage levels and phase angles, that need to be solved by the optimization procedure. Instead, nested solvers can be used to efficiently determine the solution of the constraint model.

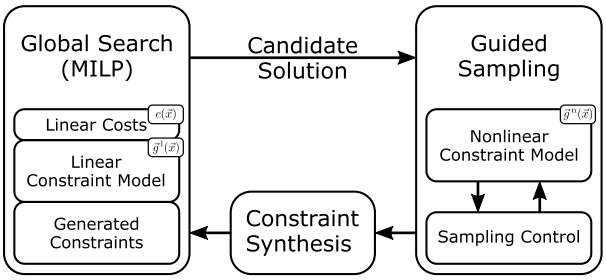

FIGURE 2. Component interaction of the hybrid optimization scheme.

III. SCHEDULING SOLVING METHODS

A first intuition on the complexity of solving microgrid scheduling is given in Appendix A, showing that the problem is at least weakly NP-hard, by providing a polonomial time reduction from the Knapsack problem to scheduling. Although several algorithms are available that solve practical instances of Knapsack [24], the reduction demonstrates that in particular the integer states of DG units raise the computational complexity. Additionally, the nonlinear constraints $\vec{g}^{\,n}(\vec{x})$ may encode arbitrary decision problems that further rise the computational complexity. Hence, complexity must be considered to keep practical instances computationally tractable, for example by using heuristics that approximate an exact solution and by efficiently using highly-developed tools such as MILP and power-flow solvers.

The main idea of the problem decomposition presented in Section II-A is to separate those models that can be efficiently handled by existing MILP solvers, i.e., $c(\vec{x})$ and $\vec{g}^{\,l}(\vec{x})$, from those that need to be linearized first. Instead of requiring a differentiable closed form representation of $\vec{g}^{\,n}$, the heuristic approach samples the nonlinear constraint function near the linear optimum and iteratively extends $\vec{g}^{\,l}$ by a local approximation. Due to the complexity of evaluating $\vec{g}^{\,n}$, samples must be drawn efficiently to generate the MILP constraints. Fig. 2 illustrates the heuristic optimization scheme.

One may note that the MILP scheduling formulation contains several variables such as $p^{CHG}_{b,t}$, $p^{DCH}_{b,t}$, and $o^{CHG}_{b,t}$ that show a strong interdependence. To reduce the number of variables when sampling, a normalized representation is introduced. Each schedule is thereby represented by the DG status $o^{DG}_{a,t}$, as well as the normalized power output $\tilde{p}^{DG}_{a,t}$, and storage level $\tilde{e}^{ST}_{b,t}$ as defined by (26) and (27), respectively.

$$\tilde{p}^{DG}_{a,t} = \frac{p^{DG}_{a,t} - \underline{p}^{DG}_a}{\bar{p}^{DG}_a - \underline{p}^{DG}_a} \quad \forall a \in \mathbb{D}_G, t \in \mathbb{T} \tag{26}$$

$$\tilde{e}^{ST}_{b,t} = \frac{e^{ST}_{b,t} - \underline{e}^{ST}_b}{\bar{e}^{ST}_b - \underline{e}^{ST}_a} \quad \forall b \in \mathbb{S}_T, t \in \mathbb{T} \tag{27}$$

To simulate a schedule and to synthesize constraints, the original MILP variables are obtained by clipping invalid values (e.g., $\tilde{p}^{DG}_{a,t} \gtrsim 0$ in case $o^{DG}_{a,t} = 0$) to the operation ranges implied by (3), (4), and (5). Note that the repair heuristic and the normalization step eliminate some, but not

all infeasible configurations, for instance in case the extended reserve requirements (16) to (18) are enabled.

A. SENSITIVITY CONSTRAINT SYNTHESIS

The sensitivity-based method repeatedly extends the MILP model by local approximations of the nonlinear constraints until either the candidate solution of the MILP model does not show constraint violations anymore or the MILP model becomes infeasible. If the latter applies, no schedule can be generated. Since the constraint synthesis approach presented in [11] addresses a single time interval only, nonlinear constraints $\vec{g}^{\,\mathrm{n}}(\vec{x})$ that cover multiple intervals cannot be directly integrated into the online optimization of the multi-stage scheduling method. Therefore, the input vector is extended to the entire scheduling horizon and the sensitivity-based synthesis is directly integrated into the scheduling problem at once. Furthermore, the approach in [11] is limited to voltage and current limits. This work generalizes the methodology to arbitrary constraints.

In case the candidate solution $\vec{x}^{\,\mathrm{C}}$ of the linear subproblem turns out to be infeasible and $\mathbb{V}$ is the set of violated nonlinear constraints, the set of MILP constraints (28) is added.

$$g_i^{\mathrm{n}}(\vec{x}^{\,\mathrm{C}}) + \frac{\partial g_i^{\mathrm{n}}}{\partial \vec{x}}(\vec{x}^{\,\mathrm{C}}) \cdot (\vec{x} - \vec{x}^{\,\mathrm{C}}) + \epsilon \leq 0 \quad \forall i \in \mathbb{V} \tag{28}$$

To feature convergence even if $\vec{g}^{\,\mathrm{n}}(\vec{x})$ or its partial derivatives are affected by numerical inaccuracies, a strictly positive constant ϵ is introduced that strengthens the permitted region. According to [11], $\frac{\partial \vec{g}^{\,\mathrm{n}}}{\partial \vec{x}}$ is numerically approximated in case the Jacobean is not directly available. For each scheduling variable $\tilde{x}_i$, the sampling control block introduces a small perturbation ρ on that variable and samples $\vec{g}^{\,\mathrm{n}}(\tilde{x}_1, \ldots, \tilde{x}_{i-1}, \tilde{x}_i + \rho, \tilde{x}_{i+1}, \ldots, \tilde{x}_{|\vec{g}^{\,\mathrm{n}}|})$ anew. For discrete variables, a state change is enforced.

To contain the number of samples and to reduce numerical errors, two sampling heuristics are introduced. The first one skips the perturbation of $\tilde{p}_{a,t}^{\mathrm{DG}}$ in case the DG is not operational, i.e., $o_{b,t}^{\mathrm{CHG}} = 0$ and no effect is expected. The second heuristic uses the partitioning $\vec{g}_t^{\,\mathrm{n},\mathbb{T}}$ with respect to scheduling time t to skip those variables that most likely do not influence the outcome of failing constraints. Scheduling variables at a time instant t are considered if and only if $\vec{g}_t^{\,\mathrm{n},\mathbb{T}} \cap \left\{ g_i^{\mathrm{n}} | i \in \mathbb{V} \right\} \neq \emptyset$ or $\vec{g}_\emptyset^{\,\mathrm{n},\mathbb{T}} \cap \left\{ g_i^{\mathrm{n}} | i \in \mathbb{V} \right\} \neq \emptyset$.

After sampling the neighborhood of $\vec{x}^{\,\mathrm{C}}$, the constraint synthesis routine approximates (28) via the observed changes in the output metric. Note that following [11], constraints from previous iterations are never revoked and that the local approximation of $\vec{g}^{\,\mathrm{n}}(\vec{x})$ is not restricted to any particular neighborhood. Hence, considerable overapproximation may be observed, in case the adjusted schedule largely deviates from the candidate solution.

B. TREE-BASED CONSTRAINT APPROXIMATION

An alternative model to approximate $\vec{g}^{\,\mathrm{n}}(\vec{x})$ as MILP problem is to encode the decisions implied by (2) as decision trees. Instead of enforcing local approximations (28) globally, a divide-and-conquer approach is implemented that recursively splits the set of schedules [25]. The tree structure which can model even nonconvex sets is then transformed into a MILP representation by adding new binary variables [14]. The tree-based method uses two mechanisms to find feasible solutions of the scheduling problem. First, a global MILP search that includes an approximation of $\vec{g}^{\,\mathrm{n}}(\vec{x})$ is used to find initial solutions. Secondly, a stochastic local search is used to sample $\vec{g}^{\,\mathrm{n}}(\vec{x})$ near the candidate solution and improve it even further. Considering all known samples, a new approximation of the nonlinear constraint function is generated in each global iteration and replaces previous approximations in the MILP. Due to the replacement, subsequent global iterations can further improve the operating costs. In the following evaluations, two global termination criteria, a static number of maximum iterations and a threshold on the improvement rate are applied.

A single decision tree $\mathcal{T}_i$ consists of a series of splits that recursively divide the solution space into feasible and infeasible regions [14]. Each binary split on the subtree $\mathcal{T}_j$ is based on a decision $\mathrm{dec}(\mathcal{T}_j)$ that involves a subset of the scheduling variables $\vec{x}$. In order to transform the tree into MILP form, each split must follow the linear form $\mathrm{dec}(\mathcal{T}_j)(\vec{x}) = \vec{w}^{\mathrm{T}} \cdot \vec{x} + w_0 \leq 0$, where $\vec{w}$ and w_0 are constant weights. To simplify the training procedure that generates $\mathcal{T}_i$, the linear form can be further restricted to decisions that only involve a single scheduling variable at once.

Each individual nonlinear objective g_i^{n} may be directly approximated by a single decision tree $\mathcal{T}_i$. Similar to each constraint that is added in (28), each tree introduces an overhead in terms of additional constraints and possibly some auxiliary variables. To reduce the size of the MILP problem and the amount of redundant constraints, the heuristic partitioning scheme $\mathbb{H}$ is used to consolidate nonlinear constraints that are likely to share dependencies. For each partition $\vec{g}_h^{\,\mathrm{n},\mathbb{H}}$, $h \in \mathbb{H}$, a decision tree $\mathcal{T}_h$ is grown that approximates the conjunction of each constraint in $\vec{g}_h^{\,\mathrm{n},\mathbb{H}}$. Clearly, the conjunction of all classification results in the forest $\mathcal{T}_h, h \in \mathbb{H}$ approximates the entire set of nonlinear constraints (2).

Similar to approximation of $\frac{\partial \vec{g}^{\,\mathrm{n}}}{\partial \vec{x}}$, a sampling-based approach is proposed to grow the decision trees $\mathcal{T}_i$ without requiring insights into $\vec{g}^{\,\mathrm{n}}(\vec{x})$. The sampling control logic now directly addresses the decision boundary of $\vec{g}^{\,\mathrm{n}}(\vec{x})$ near the candidate solution $\vec{x}^{\,\mathrm{C}}$ instead of approximating the Jacobean at $\vec{x}^{\,\mathrm{C}}$ and deducing the decision boundary in a subsequent step. Given all samples, a classification tree algorithm such as C4.5 and CART is deployed to fit the corresponding trees [25]. Note that in case the hybrid optimization algorithm includes linear, multivariate splits, the corresponding learning method must support that model as well. To ease analysis and reduce the complexity of generated splits, this work uses a CART-based algorithm as implemented in [26] that does not include multivariate splits. The training algorithm itself recursively divides the set of samples into two partitions such that the impurity considering feasible and

infeasible members is minimized according to the Gini index [25].

The accuracy, size and complexity of generated trees largely depends on the input features they are trained on. For instance, a constraint on the total up-spinning reserve, p_t^{UP} may either directly access that variable or approximate it via all status variables $o_{a,t}^{\mathrm{DG}}$, in case p_t^{UP} is not exposed to the decision tree. To boost the expressiveness and interpretability of generated trees, additional heuristics beyond the basic state variables $o_{a,t}^{\mathrm{DG}}$, $p_{a,t}^{\mathrm{DG}}$, and $e_{b,t}^{\mathrm{ST}}$ can be included in the feature set. For each sample $\tilde{x}$, all features that are exposed to the decision tree algorithm need to be calculated. Likewise, the MILP problem must model corresponding variables in order to automatically transform the decision tree. In this study, the linking variables $o_{a,t}^{\mathrm{DG}}$, $p_{a,t}^{\mathrm{DG}}$, $e_{b,t}^{\mathrm{ST}}$, $p_{b,t}^{\mathrm{CHG}}$, $p_{b,t}^{\mathrm{DCH}}$, p_t^{NDG}, and p^{MinUP} have been manually selected. An automated feature selection process that can improve the approximation even further is considered to be out of scope.

Several classification tree algorithms include pruning steps that reduce the number of nodes in favor of a less complex and more general decision tree [25]. While for classical machine-learning use cases, robustness against outliers and overapproximation plays an important role, hybrid optimization requires a constraint model that reliably excludes infeasible regions. Since it is assumed that the function $\vec{g}^{\,\mathrm{n}}(\vec{x})$ itself, except for some small numeric errors, is deterministic, no outliers are expected in the training set. Moreover, misclassification can prolong convergence in case an infeasible candidate solution is not excluded. Hence, the algorithm traits accuracy for simplified trees by excluding any pruning step that would introduce misclassified training samples.

C. SAMPLING STRATEGY

Arising from the need of drawing samples from $\vec{g}^{\,\mathrm{n}}(\vec{x})$ efficiently while determining the feasible region near the linear candidate solution $\vec{x}^{\,\mathrm{C}}$, a randomized local search strategy is introduced. Starting from $\vec{x}^{\,\mathrm{C}}$, samples are generated towards the next local optimum. In case $\vec{x}^{\,\mathrm{C}}$ is already feasible, the local search may further refine the optimum and the local approximation of $\vec{g}^{\,\mathrm{n}}(\vec{x})$. Otherwise, the search procedure first needs to find samples in the feasible region in order to subsequently approximate the boundary. Although it is, in principle, sufficient to approximate the feasibility of all nonlinear constraints well, without considering the linear ones, the region of interest largely depends on the linear subproblem. Hence, the full problem (1) and (2) is considered for local search and all linear constraints that are not already implied by the normalized representation $\tilde{x}$ are added to the constraint set for local search as well.

According to the separation technique described in [19], the total constraint violation level $G^{\mathrm{ln}}(\vec{x})$ as defined by (29) is given priority on comparing candidate solutions.

$$G^{z}(\vec{x}) = \sum_{i=1}^{|\vec{g}^{\,z}|} \min(g_i^{z}(\vec{x}), 0), \quad z \in \{\mathrm{n}, \mathrm{l}, \mathrm{ln}\} \tag{29}$$

In case no precedence on the violation level is observed, the objective value is taken into account. Local search iteratively samples from a neighborhood that contains all schedules deviating by at most n variables from the currently best local schedule $\tilde{x}^{\mathrm{L}}$. In case a better solution is sampled, $\tilde{x}^{\mathrm{L}}$ is updated accordingly. To further guide local search, the probability $P(\mathrm{alter}\,\tilde{x}_{i,t})$ of altering a single variable $\tilde{x}_{i,t}$ at time instant t is chosen as (30) proportionally to the total violation level, or the operating cost at that time.

$$P(\mathrm{alter}\ \tilde{x}_{i,t}) \propto \begin{cases} c_t^{\mathrm{TOT}}(\tilde{x}) & G^{\mathrm{ln}}(\tilde{x}) = 0 \\ G_t^{\mathrm{ln},\mathbb{T}}(\tilde{x}) + G_{\emptyset}^{\mathrm{ln},\mathbb{T}}(\tilde{x}) & \text{otherwise} \end{cases} \tag{30}$$

The deviation of each selected continuous variable will be sampled from a centered normal distribution $\mathcal{N}(0, \sigma)$ and clipped to the boundaries [0, 1] of each normalized variable. Starting from a large neighborhood, both, the number of altered variables n and the standard deviation σ are systematically decreased to support convergence. As soon as the moving average number of improvements drops below a given threshold, the next set of neighborhood parameters is applied or sampling is stopped.

D. TREE CONSTRAINT SYNTHESIS

Given the decision tree $\mathcal{T}$, the algorithm [14] is extended to generate a set of, up to an arbitrarily small tolerance ϵ, equivalent MILP constraints. A subtree $\mathcal{T}_j$ is considered feasible in case it contains a path to a feasible leaf node. For each decision $\mathrm{dec}(\mathcal{T}_j)$ that leads to both feasible subtrees, an unbounded binary variable o_k^{TR} is introduced that encodes the outcome in subsequent decisions. Specifically, $o_k^{\mathrm{TR}} = 1$, if $\mathcal{T}_j$ is active, $\mathrm{dec}(\mathcal{T}_j)(\vec{x}) \leq 0$, and the left branch is taken. Given a single tree, the set of previously added auxiliary variables encodes the active path within that tree and determines whether a constraint is considered. In case only one subtree is feasible, given the path to that constraint is active, it must always be satisfied and no auxiliary variable is added. Inactive constraints are masked by introducing a large constant M that exceeds any feasible value of $|\mathrm{dec}(\mathcal{T}_j)(\vec{x})|$.

Algorithm 1 defines the tree constraint synthesis in detail. Initially, the procedure is called on the entire tree given an empty masking term $v = 0$. It recursively adds linear constraints until all feasible subtrees are enumerated. In order to generate MILP constraints, [14] relaxes the strict inequality $\mathrm{dec}(\mathcal{T})(\vec{x}) \gtrdot 0$ to a soft one. However, such a transformation can include values that the decision tree $\mathcal{T}$ classifies as infeasible and convergence of the hybrid optimization algorithm can be adversely affected. To support convergence, this work introduces an ϵ offset to model strict inequalities instead of relaxing the decision. In addition, the application of Algorithm 1 in a closed-loop optimization setup instead of an open-loop constraint learning task is demonstrated.

IV. CASE STUDY ON SCHEDULING ALGORITHMS

From a design point of view, both methods for solving the nonlinear scheduling problem have their own merits.

Algorithm 1 Synthesis Procedure Transforming Tree $\mathcal{T}$ Into the Set of Constraints $\vec{g}^{\,t}$ (based on [14])

function Syn(feasible tree $\mathcal{T}$, masking term v)
  $\vec{g}^{\,t} \leftarrow \emptyset$
  if left($\mathcal{T}$) and right($\mathcal{T}$) are feasible **then**
    Introduce new variable $o_k^{\mathrm{TR}} \in \mathbb{B}$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \{\mathrm{dec}(\mathcal{T}) \leq M \cdot (1 - o_k^{\mathrm{TR}}) + v\}$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \{\mathrm{dec}(\mathcal{T}) \geq \epsilon - M \cdot o_k^{\mathrm{TR}} - v\}$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \mathrm{Syn}(\mathrm{left}(\mathcal{T}), v + M \cdot (1 - o_k^{\mathrm{TR}}))$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \mathrm{Syn}(\mathrm{right}(\mathcal{T}), v + M \cdot o_k^{\mathrm{TR}})$
  else if left($\mathcal{T}$) is feasible **then**
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \{\mathrm{dec}(\mathcal{T}) \leq v\}$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \mathrm{Syn}(\mathrm{left}(\mathcal{T}), v)$
  else if right($\mathcal{T}$) is feasible **then**
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \{\mathrm{dec}(\mathcal{T}) \geq \epsilon - v\}$
    $\vec{g}^{\,t} \leftarrow \vec{g}^{\,t} \cup \mathrm{Syn}(\mathrm{right}(\mathcal{T}), v)$
  end if
  return $\vec{g}^{\,t}$
end function

Sensitivity-based optimization features a simple implementation and tree-constraint synthesis, in theory, overcomes the limitations of a global plane approximation. A case study was designed to analyze the performance of both algorithms on a common ground, to verify the theoretical expectations and to give insights into preferred use-cases.

A. TEST SYSTEM AND CONTROLS

Due to the widespread application in scheduling [4], [27], [28] and the challenging network design that leads to frequent voltage violations, the whole study is based on a modified version of the well-established Baran testfeeder [29]. Since the original network does not include any generation, the feeder was extended by DG, Photovoltaics (PV), Wind Turbine (WT), and storage plants following the configuration in [27]. However, maximum and minimum power of all DGs were scaled by one half to avoid generator tripping in islanded low-load scenarios. The power limits of $\pm 1\,MW$, the location, and the constant average efficiency of $\mu_b^{\mathrm{ST}} = 0.9$ of all storage units b were kept, but to study charging cycle limitations in more detail, the storage size was reduced to 0.9 MWh usable capacity. In contrast to [27] that does not include details on volatile RES, this study considers the voltage control capabilities of all generation units and therefore additionally assumes that PV and WT plants are limited to an apparent power $\bar{S}_l^{\mathrm{DER}}$ of 0.1 MVA and 0.2 MVA, respectively. Table 1 and Fig. 3 summarize the system configuration of all DGs and the network topology, respectively.

To demonstrate the integration of low-level controls in proactive scheduling, the model of [27] and [29] was amended by voltage and frequency droop control. Based on [30], for each volatile DER and storage unit, Q-of-U droop control was enabled scaling the maximal and minimal reactive power between bus voltages of 0.92 p.u. and 1.08 p.u.

TABLE 1. Location and rating of controllable DGs.

DER a	Bus	$\bar{p}_a^{\mathrm{DG}}$ (MW)	$\underline{p}_a^{\mathrm{DG}}$ (MW)	$\bar{Q}_a^{\mathrm{DG}}$ (Mvar)
DG1	8	2.1	0.5	2.1
DG2	13	1.69	0.5	1.69
DG3	16	1.69	0.5	1.69
DG4	25	2.36	0.5	2.36

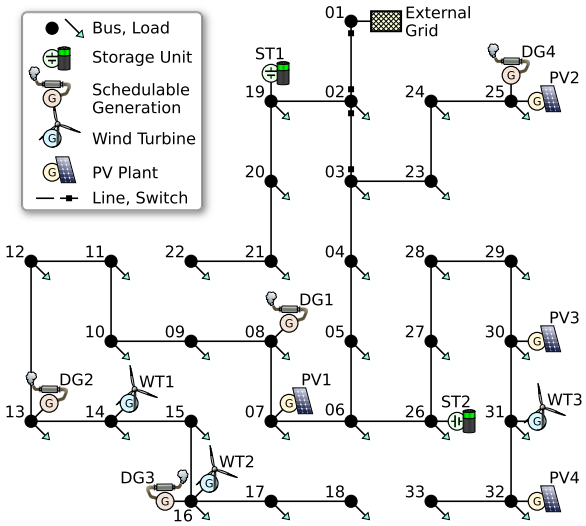

FIGURE 3. Network topology used to assess the algorithms.

considering a dead band of 0.96 p.u. and 1.05 p.u. For the static reactive power limits used to define the droop curve, it is assumed that no active power is generated or drawn from the grid, but total apparent power limitations $\bar{S}_l^{\mathrm{DER}}$ are considered as well.

All scheduled DG units also participate in voltage control via a droop curve that scales the reactive power between 0.92 p.u. and 1.08 p.u. bus voltage as well. In any islanded subgrid, all scheduled DG units also participate in grid forming and active power control via a primary frequency bias that is scaled according to the operating range $\bar{p}_a^{\mathrm{DG}} - \underline{p}_a^{\mathrm{DG}}$. The whole nonlinear network model, including the low-level controls is solved by a series of load flow calculations as implemented by the quasi-dynamic simulation of DIgSILENT PowerFactory. The feasibility of each candidate solution is rated by any equipment overload and any violation of the target voltage band of 0.95 to 1.05 p.u. For each single constraint, the level of constraint violation is calculated based on the distance to the feasible region normalized by a nominal operating condition of 100% loading and 1 p.u. bus voltage.

B. INPUT PROFILES AND FAILURE MODES

Both scheduling algorithms operate on forecasts of volatile loads and generation. Fig. 4 plots the hourly load and RES forecasts. Shading in the graphics illustrates the share of individual assets on the total power profiles. To challenge the algorithms under test by increasing grid imbalance, a day with

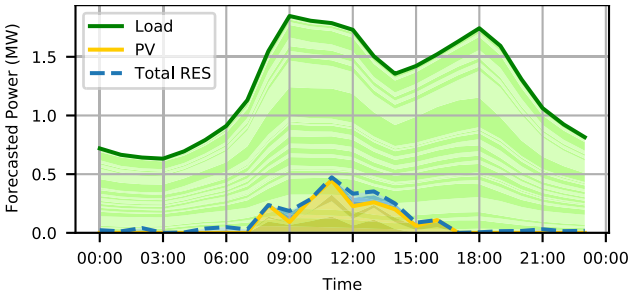

FIGURE 4. Forecasted power of all loads as well as PV, and WT generation.

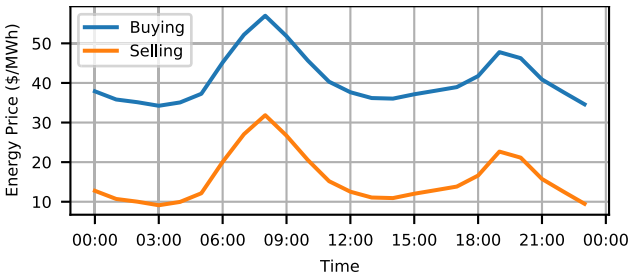

FIGURE 5. Dynamic electricity prices including grid transfer fees.

low RES generation was modeled. The load profiles were selected from [31] and scaled according to the nominal load. Dynamic prices of buying and selling electricity that are given in Fig. 5 are based on data from [32] and the assumption of constant grid transfer fees. For each DG unit, a price of \$90 $\frac{1}{\text{MWh}}$ is assumed that is considerably above the grid tariffs. To reduce the chance of over fitting each algorithm, parameter tuning was conducted on input profiles different from the presented validation data.

The linear partition $\vec{g}^{\,\text{l}}(\vec{x})$ of the scheduling model directly operates on the deterministic forecasts, but for the nonlinear constraints $\vec{g}^{\,\text{n}}(\vec{x})$, a set of deviation and failure scenarios must be available. To support further analysis, two worst-case deviations were manually defined. The first one models a power shortage and increases each load by 10% while decreasing the DER generation by the same amount. Similarly, the second scenario includes a power surplus by increasing and decreasing DER generation and loads, respectively, by 10% as well. In addition to islanding from the main grid as demonstrated in [27], this study assumes that the line connecting buses 02 and 03 is highly exposed and that the grid can be partly islanded in case that line trips. For each of the deviation cases and each of the islanding options one worst-case scenario of a complete loss of connection is added to the uninterrupted scenarios.

To assess varying complexities of the scheduling problem, some study cases impose some linear operation constraints (12) to (21) that set the minimum operating power of DG3, $\underline{p}^{\text{OP}}_{\text{DG3},t}$ to $\underline{p}^{\text{OP}}_{\text{DG3},t} = 0.5\text{MW}$ for $t \in [8{:}00 - 17{:}00]$. Additionally, the maximum number of charging cycles, $\bar{n}^{\text{STA}}_b$ for each storage $b \in \mathbb{S}_\text{T}$ is set to $\bar{n}^{\text{STA}}_b = 1$ per day and the minimum spinning reserve $\underline{p}^{\text{UP}}_t$ was set to $\underline{p}^{\text{UP}}_t = 0, t \in \mathbb{T}$.

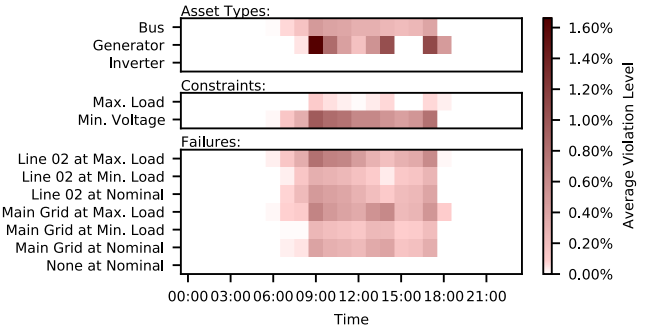

FIGURE 6. Average normalized violation levels according to different partitioning schemes.

C. SIMPLIFIED ECONOMIC SCHEDULING

The first case studies the performance of purely linear scheduling that includes both, the basic reserve constraints (16) to (18) and the operational constraints (12) to (15). In the experiment, the second storage unit, ST2, was removed to further simplify the problem and to ease the comparison with other simplified cases. Without considering the nonlinear constraints $\vec{g}^{\,\text{n}}(\vec{x})$, the model returns a lower bound for the resilient operation costs of \$1716.28. Clearly, economic scheduling, despite the reserve constraints, fails to deliver a feasible solution w.r.t. the nonlinear constraints $\vec{g}^{\,\text{n}}$. Fig. 6 shows the average violation level according to several partitioning schemes of the nonlinear constraints. Partitions that do not show any violation, are excluded from the graphics.

One may note that although most violations occur in the presence of some failures, also nominal operation as considered by linear scheduling shows some violations that can impact the operation. For both failure modes, a considerable amount of violations is observed. Even for main grid outages that are covered by the minimum spinning reserve constraint of the linear model, considerable DG overload due to the impact of voltage regulation and frequent undervoltage situations are encountered. Induced by the limited infeed in the study period, no scenario shows any overvoltage conditions. Most violations are detected at buses (undervoltage) and DG units (overload). The few violations of inverter-based generation units (WT and PV) are directly caused by saturation of voltage control. DG units show overload due to both, voltage and frequency control demands.

D. HYBRID SCHEDULING OF A SIMPLIFIED MICROGRID

In the most reduced study case of hybrid scheduling, all optional constraints are disabled and the second storage unit, ST2, is removed from the microgrid. As a consequence, a solution returned by the MILP solver does not necessarily imply the basic resilience constraints (16) to (21). Any spinning reserve requirement has to be learned from the grid model. However, since the grid model includes asset loading, every feasible solution with respect to the nonlinear constraints $\vec{g}^{\,\text{n}}$ is also feasible regarding $\vec{g}^{\,\text{l}}(\vec{x})$, the linear ones. As for all study cases, a fixed number of 20 global

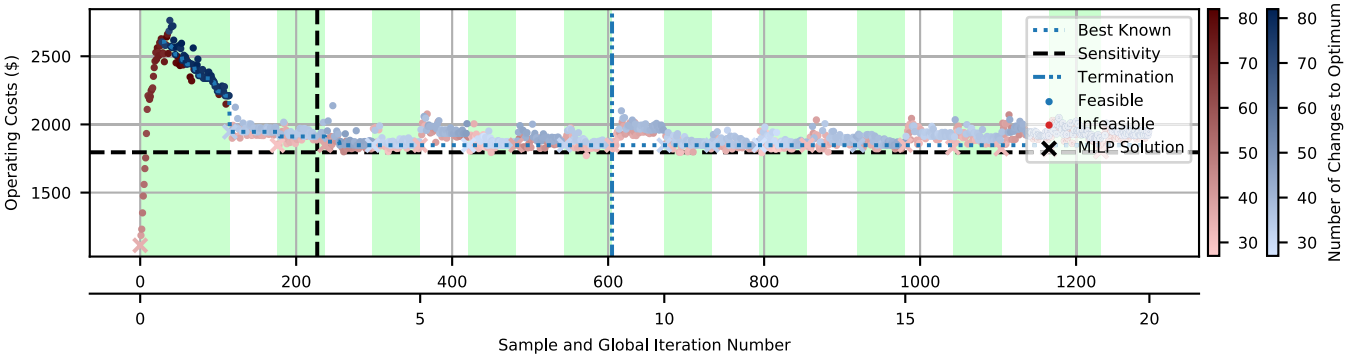

FIGURE 7. Convergence of the tree-based algorithm on the simplified test system without operational constraints.

TABLE 2. Local neighborhood configurations i including the number of altered variables n_i and the standard deviation σ_i.

Nr.	n_i	σ_i	Nr.	n_i	σ_i
0	8	0.3	1	4	0.2
2	3	0.2	3	2	0.1
4	1	0.075	5	1	0.05

iterations to demonstrate long-term minima and a dynamic termination criterion that ends global search if the last five iterations do not yield any improvement were installed. Furthermore, a neighborhood scheme of six configurations with descending sizes was chosen. The manually tuned number of changed variables n_i and the standard deviation σ_i of each neighborhood i are listed in Table 2. As soon as the improvement rate of the last ten samples drops below 20%, the next configuration is chosen or local search is ended.

The convergence of one tree-based optimization run as well as the scheduling run of the deterministic sensitivity-based approximation are illustrated in Fig. 7. Within the first iteration, local search finds several feasible solutions, but the local optimum having costs of $2210.31 shows a considerable distance to the best known solution of 76 changes in the vector of normalized scheduling variables $\tilde{x}$ and 23% higher costs. Subsequent runs of the extended linear model reduce the gap to the optimum and due to approximation of $\vec{g}^{\,n}(\vec{x})$ show both feasible and infeasible results. Although the tree-based method delivers the first feasible solution within fewer samples, even the long-term operation showed slightly higher operation costs of 2.9% compared to the optimal sensitivity-based results on that particular run.

In contrast to the deterministic sensitivity-based approach, the tree-based method includes a stochastic local search. To account for stochastic effects both in the execution time and the local search, for each case, the optimization runs are executed 32 times and the results of all experiments are listed in Table 3. For tree-based scheduling, the final result as well as the processing time highly depend on the termination criterion and an adequate tradeoff needs to be found. Both, a constant number of 20 global iterations as illustrated in Fig. 7 as well as the dynamic termination criterion that may terminate earlier are used for the tree-based configurations. Timing and accuracy of the sensitivity-based approach, likewise, can be influenced by the ϵ offset heuristically set to $\epsilon = 0.01\%$. Similar to the exemplary run illustrated in Fig. 7, the tree-based approach shows average long-term minimal costs of $1841.04 or an increase of 2.5% compared to the sensitivity-based solution of $1795.62. However, all optimization runs of the tree-based method find a first feasible solution below 41 samples and therefore before the 223 samples required by the reference.

To estimate the computational effort of executing the algorithms, the timing of all experiments was recorded and the detailed results using the rate-based termination criterion are listed in Table 4. All experiments were conducted on a virtual Windows 10 (build 18363) machine having four assigned Intel Xeon CPU E5-2690 v4 cores clocked at 2.60 GHz and 32 GB memory. The algorithms themselves were implemented in Python 3.7 accessing PowerFactory 2021 SP2. The MILP model was solved by the open source solver CBC 2.10 accessed via the Pyomo library version 6.0 [33]. To reduce mutual influence, no more than two experiments were executed in parallel.

One can see that the average sensitivity run of 22.05 minutes wall clock time is considerably faster than the average tree-based run taking 85.83 minutes. Given that most of the time is consumed by the grid simulation, the observations from Fig. 7 that the sensitivity-based scheduling converges considerably faster can be further supported. In contrast to the grid simulation that accounts for 92% and 90%, respectively of the total wall clock time, MILP constraint synthesis including and solving the MILP models in total only takes 6% of the time for tree-based and 5% for sensitivity-based scheduling, respectively. Likewise, neither the simulation setup nor the remaining actions of both algorithms have a significant effect on the computational effort. Similar effects can also be seen from the process time statistics that also accounts for parallel actions as conducted by the grid simulation.

E. SCHEDULING WITH OPERATIONAL CONSTRAINTS

To demonstrate the behavior of both algorithms in case of rising model complexity, as in Section IV-C, storage unit ST2 is

TABLE 3. Optimization results of all cases covering the tree-based and the sensitivity-based method.

Case	Variable: / Metric: / Algorithm	Costs - Const. Rounds ($) Avg.	Std.	Min.	Max.	Costs - Termination ($) Avg.	Std.	Min.	Max.	Samples to First Feasible Avg.	Std.	Min.	Max.
Most Simplified	Tree-Based	1841.04	14.52	1795.37	1850.96	1854.23	16.20	1821.09	1894.73	25.0	4.9	17	41
	Sensitivity-Based	1795.62	0.00	1795.62	1795.62	1795.62	0.00	1795.62	1795.62	223.0	0.0	223	223
Operational Constraints	Tree-Based	1981.28	0.00	1981.28	1981.28	1983.23	7.73	1981.28	2016.92	44.0	29.6	13	111
	Sensitivity-Based	2128.95	0.00	2128.95	2128.95	2128.95	0.00	2128.95	2128.95	114.0	0.0	114	114
Complete	Tree-Based	1973.48	6.74	1950.09	1994.85	1984.61	17.12	1972.52	2029.84	37.1	23.0	15	132
	Sensitivity-Based	-	-	-	-	-	-	-	-	-	-	-	-

TABLE 4. Computation time until the termination criterion is met.

Case	Timer: / Algorithm: / Metric: / Operation	Wall Clock Time (min) Tree-Based Avg.	Std.	Sensitivity-Based Avg.	Std.	Process Time (CPU min) Tree-Based Avg.	Std.	Sensitivity-Based Avg.	Std.
Most Simplified	Setup	0.75	0.08	0.56	0.04	0.75	0.02	0.47	0.02
	Grid Simulation	80.37	22.75	18.76	0.79	137.21	38.85	29.08	2.36
	MILP Solution	0.47	0.43	1.01	0.04	0.43	0.42	0.99	0.03
	MILP Synthesis	3.06	1.51	0.94	0.06	3.06	1.52	1.54	0.11
	Residual Steps	1.19	0.37	0.77	0.02	1.22	0.35	0.90	0.02
	Total	85.83	24.94	22.05	0.91	142.68	41.03	32.98	2.46
Operational Constraints	Setup	0.75	0.06	0.46	0.02	0.75	0.02	0.38	0.01
	Grid Simulation	59.73	10.64	10.02	0.06	101.54	18.23	16.71	0.12
	MILP Solution	1.11	1.94	1.38	0.01	1.07	1.94	1.36	0.01
	MILP Synthesis	1.78	0.56	0.26	0.01	1.79	0.57	0.43	0.02
	Residual Steps	0.88	0.18	0.33	0.00	0.91	0.16	0.46	0.00
	Total	64.24	11.89	12.44	0.07	106.05	19.42	19.35	0.12
Complete	Setup	1.21	0.84	0.36	0.05	0.80	0.02	0.35	0.01
	Grid Simulation	72.23	23.23	7.49	0.03	120.61	39.43	12.59	0.11
	MILP Solution	1.96	2.04	0.16	0.00	1.92	2.04	0.15	0.00
	MILP Synthesis	2.69	1.61	0.23	0.01	2.69	1.61	0.40	0.02
	Residual Steps	0.67	0.90	0.11	0.00	1.11	0.36	0.16	0.00
	Total	78.76	26.31	8.35	0.07	127.13	42.48	13.65	0.12

still excluded, but all operational constraints are enabled. Due to the cycle restrictions, considerable interdependence among scheduling variables is added and the nonlinear constraints do not imply the linear ones anymore. Additionally, the minimum generation requirements of DG3 directly affect the sampling procedures and further increase the chance of selecting a linearly infeasible solution.

Fig. 8 illustrates the convergence of one scheduling run using tree-approximation and adds the bounds achieved by the sensitivity-based method. One can note that tree-constraint approximation outperforms the sensitivity-based method both in terms of final operating costs and the number of samples until a first feasible solution is found. As listed in Table 3, sensitivity-based scheduling returns a solution that shows 7.5% higher costs than the average (and simultaneously best known) tree-based solution after 20 iterations of $1981.28. Similarly, all tree-based optimization runs delivered an initial solution within the first 111 samples, having an average of 44.0 samples while the sensitivity-based approach required 114 samples to obtain the first feasible schedule and 120 samples to finish its computation.

Compared to sensitivity-based scheduling that terminates within 12.44 minutes wall clock time, the average tree-based run requires a larger number of 644 samples until the convergence criterion is met and therefore shows an increased computational effort of 64.24 minutes. However, using the same number of samples that are needed to finalize the sensitivity-based algorithm, the average cost of $2126.27 for the tree-based method only marginally differs from the final sensitivity-based result. The increased effort could therefore be used to improve the final solution. Giving the distance to the optimum and the operating costs drawn in Fig. 8, it can be seen that due to the intended overapproximation of the decision tree and the randomization of local search, new regions of the solution space are explored, even after a feasible MILP solution was found. Although that first feasible and best known solution was found in the fourth MILP run, subsequent search shows distances of up to 27 changed variables to the optimum.

F. SCHEDULING OF THE COMPLETE MICROGRID

The last study case includes the full set of assets and operational constraints as described in Section IV-A and IV-B. Despite the moderate size of the problem covering six controllable assets, the state-of-the-art reference algorithm, sensitivity-based constraint approximation, failed to deliver any feasible solution at all. After adding the constraint plane approximation at the initial solution, the solver reported the

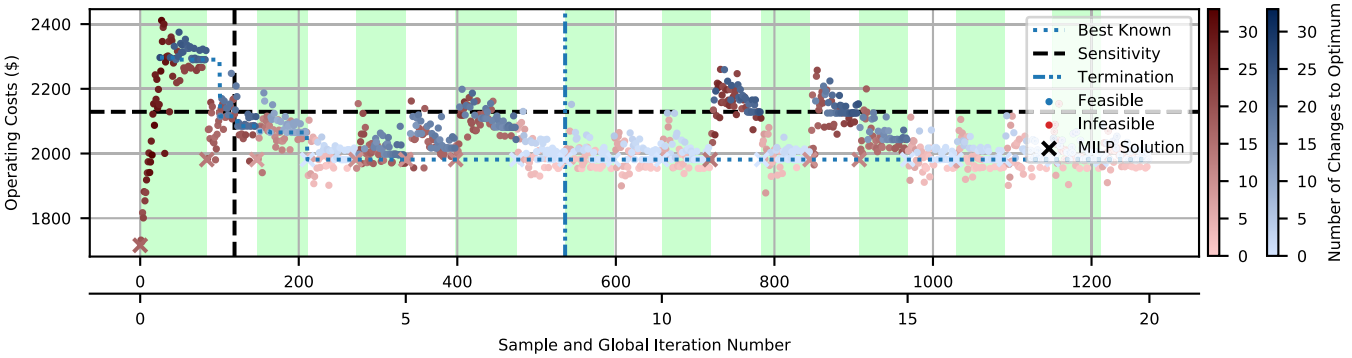

FIGURE 8. Convergence of the tree-based algorithm on the simplified test system with operational constraints.

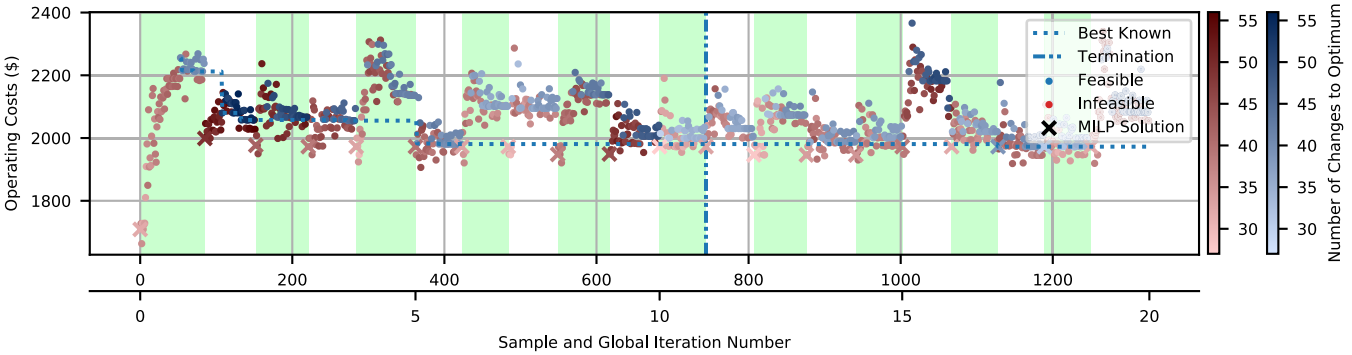

FIGURE 9. Convergence of the tree-based algorithm on the complete test system.

entire problem to be infeasible and no feasible solution could be scheduled. Even without the sample reduction heuristic that was introduced to render several problem instances computationally tractable, no feasible solution was found.

In contrast to the sensitivity constraint synthesis, the tree-constraint method finds a feasible solution within the first local search and 37.1 samples in average. The execution time of 8.35 minutes on average that is needed to determine the infeasibility of the sensitivity-based model, is considerable lower than the execution times in the other sensitivity-based cases requiring 22.05 and 12.44 minutes, respectively. For the tree-based method, an average execution time of 78.76 minutes is observed. Fig. 9 illustrates convergence of the latter method. Although the majority of 18 MILP solutions are infeasible, the approximation model was also able to successfully restrict the MILP model towards a feasible solution. In the entire scheduling run, every MILP problem could be solved to optimum. Similarly, all 32 repetitions successfully returned a feasible solution and an average operating cost of $1975.12 was achieved.

V. DISCUSSION OF CASE STUDY RESULTS

By implementing and testing two hybrid scheduling approaches, an improved sensitivity-based approach from literature and a novel tree-based method, this work successfully demonstrates the feasibility of hybrid optimization that couples nonlinear constraints and MILP. Table 5 summarizes the main features of both algorithms. In contrast to several state-of-the-art approaches, no manual linearization is needed [4], [8] and both, the MILP subproblem and the grid model could be solved by off-the-shelf solvers.

TABLE 5. Comparison of tree-based and sensitivity-based hybrid scheduling.

Aspect	Hybrid Method	
	Tree	Sensitivity
Integration of off-the-shelve solvers	✓	✓
Efficient use of grid simulation runs	✓	✓
Consideration of low-level controls	✓	✓
Suitable problem complexities	High	Low
Implementation effort	Medium	Low
Feasible intermediate solutions	✓	–

The study on hybrid optimization shows a good performance of the sensitivity-based approximation of inherently nonlinear grid constraints for some simplified cases. However, it also demonstrates that in other cases, it completely fails to generate feasible schedules. In contrast, the novel tree-constraint method that uses a more complex approximation structure in the MILP problem, quickly provides feasible solutions in all experiments and outperforms the sensitivity-based approach in all but the most simplified configurations.

The original sensitivity-based algorithm as presented in [11] is integrated into one stage of a multi-stage energy management system and originally does not include any scheduling decisions that involve multiple instants of time. In particular, no multi-period operational constraints such as the startup restrictions (15) are involved in the presented MILP model. Despite significant differences in the experimental setup, this study supports the reported effectiveness of sensitivity-based constraint approximation [11] for some cases and in addition to the original work, clearly demonstrates the limits in case of more complex setups.

In contrast to the sensitivity-based method that samples the nonlinear constraints for the sake of approximation only, the tree-based method features a sampling strategy that both quickly finds a feasible local optimum and provides the training data to extend the linear model. As a result, all experiments show that tree-based scheduling requires fewer samples to provide an initial feasible solution. Early solutions specifically enable use cases that quickly require a feasible schedule but tolerate later updates towards better solutions.

Over all sensitivity and tree-based experiments, 88% and 96%, respectively, of the process time is spent on the grid simulation performed once per sample. With the share being that high, the number of samples has a substantial impact on the total processing time. In the tree-based algorithm, the number of samples that need to be drawn strongly depends on the targeted accuracy that can be balanced by the termination criterion. However, the study case in Section IV-E demonstrates that tree-based scheduling only draws a few hundreds samples more to get significantly improved results and performs equally well, on the same number of samples. Still, the additional samples considerably prolong the execution time and need to be weighted in the accuracy tradeoff.

Due to the local search procedure, the tree-based method does not require the MILP subproblem to generate a feasible solution at all, as long as the local search finds suitable solutions and the generated tree properly restarts the heuristic search procedure. However, all cases showed that at least some valid solutions are generated by the extended MILP model and several times including iteration number 17 in Fig. 9, the MILP solution even improved the global optimum. In average over all tree-based experiments having a constant number of 20 global iterations, 7.4% of the MILP runs improved the global optimum.

It was demonstrated that several proactive scheduling problems can also be solved by purely heuristic approaches that do not include mathematical programming [4], [7]. Although the experimental setup shows significant differences (a smaller test system without considering discrete DG states but dynamic grid constraints are used), comparison to [10] indicates a considerably reduced number of evaluated samples by using hybrid optimization techniques. In contrast to purely heuristic approaches, the presented hybrid optimization techniques show a reduction in the number of samples by one order of magnitude. However, a detailed comparison on common ground is beyond the scope of this work.

Although the test system was specifically designed to challenge the algorithms under test and to trigger physical constraints, all study cases demonstrate that scheduling decisions can have adverse effect on the grid, if the assets are operated close to their limits. Given the modeled conditions, no algorithm delivered an initial MILP solution that satisfied all grid constraints which aligns well with results from related studies that highlight the need of detailed grid constraints [6], [34]. The first case on purely economic scheduling provided an in-depth analysis of the encountered constraint violations and showed that even without considering any contingencies, low-level controls can induce some overload conditions that are not predicted by the simplified scheduling formulation. However, the large majority of grid constraint violations is related to the feasibility of fault mitigation techniques.

VI. CONCLUSION AND OUTLOOK

Motivated by the high computational complexity of proactive scheduling as well as the need of efficiently considering low-level controls and nonlinear grid constraints, this work presents two hybrid approaches that successfully combine mathematical programming and heuristic optimization. A case study demonstrates that the novel optimization method based on decision trees can solve the scheduling problem, even in case a sensitivity-based method extended from literature fails to deliver results at all. However, the study also identifies a simplified case in which the sensitivity-based approach returns slightly better results and therefore gives indication which method may be best suited for a problem at hand. Detailed insights into the convergence of both algorithms show that the tree-based approach quickly delivers first feasible solutions and that the sensitivity-based method can suffer from considerable overapproximation of infeasible states.

The hybrid optimization techniques enable the usage of external grid models that cannot be included in classical mathematical programming and a first comparison to purely heuristic approaches indicates a considerably improved performance of hybrid scheduling. Similar to most related work [4], the study gives qualitative answers concerning the performance of presented algorithms. However, large-scale evaluation that covers a broad variety of grid configurations and operating conditions is needed to quantify the performance on a common ground and give final precedence over the studied algorithms. To ease analysis, the study deploys only manually defined worst-case heuristics and few fault mitigation techniques. Future work includes a refined stochastic or robust model to better quantify reserve requirements, the assessment of transient phenomena and additional fault mitigation techniques such as rerouting or load shedding to reduce local reserve needs.

In the presented experiments, decision trees are restricted to axis-parallel splits, i.e., each decision involves a single coupling variable only. Future work may also assess the effect of arbitrary linear splits [14], study the parametrization of local search in more detail and may include more

advanced termination criteria that further reduce the number of samples. Other techniques such as generalized Benders decomposition [35] may also be used in proactive scheduling. Despite the focus on microgrids future work includes the application of hybrid scheduling in other contexts such as active distribution systems. By presenting the algorithms and studying various details on solving scheduling tasks with nonlinear constraints, this work also contributes to a broader discussion on hybrid optimization methods.

.

APPENDIX A COMPLEXITY OF SCHEDULING

To give a first intuition on the computational complexity of microgrid scheduling, a polynomial-time reduction from the Knapsack problem, a weakly NP-hard problem [24], to scheduling is provided. The Knapsack problem $\mathcal{P}^{\mathrm{KN}}$ is defined as (31), given the positive integers v_i, w_i, and W.

$$\begin{aligned} &\max_{\vec{x}\in\mathbb{B}^n} \sum_{i=0}^{n} v_i x_i \\ &\text{s.t.} \sum_{i=0}^{n} w_i x_i \leq W \end{aligned} \tag{31}$$

The scheduling problem $\mathcal{P}^{\mathrm{SCH}}$ is now defined as finding $o_{a,t}^{\mathrm{DG}}, o_{b,t}^{\mathrm{CHG}}, o_t^{\mathrm{SELL}} \in \mathbb{B}$ and $p_{a,t}^{\mathrm{DG}}, p_{b,t}^{\mathrm{CHG}}, p_{b,t}^{\mathrm{DCH}}, p_t^{\mathrm{BUY}}, p_t^{\mathrm{SELL}} \in \mathbb{R}$ that minimize $c(\cdot)$ s.t. (3) to (11) are satisfied. Hence, a relaxed version with an empty set of nonlinear constraints is studied. Let $\mathcal{I}^{\mathrm{KN}} = (\vec{v}, \vec{w}, W)$ be an arbitrary instance of $\mathcal{P}^{\mathrm{KP}}$, then the mapping to $\mathcal{P}^{\mathrm{SCH}}$ is defined as (32) to (40).

$$\mathbb{T} = \{0\} \tag{32}$$

$$\mathbb{D}_{\mathrm{G}} = \{1, \ldots n\} \tag{33}$$

$$\mathbb{L}_{\mathrm{D}} = \mathbb{S}_{\mathrm{T}} = \emptyset \tag{34}$$

$$\underline{p}_i^{\mathrm{DG}} = w_i \quad \forall i \in \mathbb{D}_{\mathrm{G}} \tag{35}$$

$$\bar{p}_i^{\mathrm{DG}} = w_i \quad \forall i \in \mathbb{D}_{\mathrm{G}} \tag{36}$$

$$\bar{p}^{\,\mathrm{SELL}} = W \tag{37}$$

$$\bar{p}^{\,\mathrm{BUY}} = 0 \tag{38}$$

$$c_0^{\mathrm{SELL}} = \max_{i\in\mathbb{D}_{\mathrm{G}}} \frac{v_i}{w_i} + \epsilon \tag{39}$$

$$c_i^{\mathrm{DG}} = c^{\mathrm{SELL}} - \frac{v_i}{w_i} \quad \forall i \in \mathbb{D}_{\mathrm{G}} \tag{40}$$

Clearly, (32) to (40) can be computed in polynomial time. From (3), (35), and (36), it follows that

$$p_{i,0}^{\mathrm{DG}} = w_i \cdot o_{i,0}^{\mathrm{DG}} \quad \forall i \in \mathbb{D}_{\mathrm{G}}. \tag{41}$$

From the original power balance (9), as well as the mapping (32) to (34), and (38), the balance simplifies to

$$\sum_{i\in\mathbb{D}_{\mathrm{G}}} p_{i,0}^{\mathrm{DG}} = p_0^{\mathrm{SELL}}. \tag{42}$$

Consequently, the power transfer constraint (8) transforms to constraint (43) with the first inequality trivially fulfilled.

$$0 \leq \sum_{i\in\mathbb{D}_{\mathrm{G}}} w_i \cdot o_{i,0}^{\mathrm{DG}} \leq W \tag{43}$$

Similarly, it can be concluded from (7) and (38), that the only valid solution of $p_0^{\mathrm{BUY}} = 0$. Note that the transfer mode o_t^{SELL} can therefore be freely set to $o_t^{\mathrm{SELL}} = 1$, without loss of generality.

Given the cost definition of $\mathcal{P}^{SCH}$, (11), the defined mapping, as well as (41) and (42), the objective function simplifies to (44).

$$\begin{aligned} c(\vec{x}) &= c_0^{\mathrm{TOT}} \\ &= -c_0^{\mathrm{SELL}} \cdot p_0^{\mathrm{SELL}} + \sum_{i\in\mathbb{D}_{\mathrm{G}}} c_i^{\mathrm{DG}} \cdot p_{i,0}^{\mathrm{DG}} \\ &= \sum_{i\in\mathbb{D}_{\mathrm{G}}} \left(c_i^{\mathrm{DG}} - c_0^{\mathrm{SELL}}\right) \cdot p_{i,0}^{\mathrm{DG}} \\ &= \sum_{i\in\mathbb{D}_{\mathrm{G}}} -v_i \cdot o_{i,0}^{\mathrm{DG}} \end{aligned} \tag{44}$$

One can see that for any valid solution of $\mathcal{P}^{\mathrm{KN}}$, x_i, a schedule $o_{i,0}^{\mathrm{DG}} = x_i$, $o_0^{\mathrm{SELL}} = 1$, $p_0^{\mathrm{BUY}} = 0$, and $p_0^{\mathrm{SELL}}, p_{i,0}^{\mathrm{DG}}$ according to (42) and (41), respectively, that satisfies, constraints (3) to (9) can be found. At the same time, (43) ensures that each valid schedule of the mapped scheduling instance is mapped to a valid instance of Knapsack in polynomial time. On using the relation (44), it can be seen that any solution that maximizes the Knapsack gains minimizes the scheduling costs and vice versa.

From the polynomial time reduction from $\mathcal{P}^{\mathrm{KN}}$ to $\mathcal{P}^{\mathrm{SCH}}$ and the fact that $\mathcal{P}^{\mathrm{KN}}$ is weakly NP-hard [24], it can be concluded that $\mathcal{P}^{\mathrm{SCH}}$ is at least weakly NP-hard, as well.

ABBREVIATIONS

CHP	Combined Heat and Power
DER	Distributed Energy Resource
DG	Distributed Generator
EES	Electrical Energy Storage
EV	Electric Vehicle
MILP	Mixed Integer Linear Programming
PCC	Point of Common Coupling
PV	Photovoltaics
RES	Renewable Energy Sources
WT	Wind Turbine

REFERENCES

[1] C. Marnay *et al.*, "Microgrids 1—Engineering, economics, & experience—Capabilities, benefits, business opportunities, and examples," CIGRÉ, Paris, France, Tech. Rep. WG C6.22, Oct. 2015.

[2] N. Hatziargyriou, Ed., *Microgrids: Architectures and Control*. London, U.K.: Wiley, Mar. 2014.

[3] A. Hussain, B. Van-Hai, and H.-M. Kim, "Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience," *Appl. Energy*, vol. 240, pp. 56–72, Apr. 2019.

[4] M. Spiegel, E. Veith, and T. Strasser, "The spectrum of proactive, resilient multi-microgrid scheduling: A systematic literature review," *Energies*, vol. 13, no. 17, p. 4543, Sep. 2020.

[5] S. M. Nosratabadi, R. Hooshmand, and E. Gholipour, "A comprehensive review on microgrid and virtual power plant concepts employed for distributed energy resources scheduling in power systems," *Renew. Sustain. Energy Rev.*, vol. 67, pp. 341–363, Jan. 2017.

[6] Farag, M. M. A. Abdelaziz, and E. F. El-Saadany, "Voltage and reactive power impacts on successful operation of islanded microgrids," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1716–1727, May 2013.

[7] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*, vol. 3, no. 1, pp. 890–925, Jun. 2015.
[8] Z. Liang, Q. Alsafasfeh, and W. Su, "Proactive resilient scheduling for networked microgrids with extreme events," *IEEE Access*, vol. 7, pp. 112639–112652, 2019.
[9] A. Khodaei, "Provisional microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1107–1115, May 2015.
[10] V. Sarfi and H. Livani, "An economic-reliability security-constrained optimal dispatch for microgrids," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6777–6786, Nov. 2018.
[11] F. Yang, X. Feng, and Z. Li, "Advanced microgrid energy management system for future sustainable and resilient power grid," *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 7251–7260, Nov. 2019.
[12] H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, "A stochastic multi-objective framework for optimal scheduling of energy storage systems in microgrids," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 117–127, Jan. 2017.
[13] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *J. Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, Mar. 2011.
[14] P. Kudła and T. P. Pawlak, "One-class synthesis of constraints for mixed-integer linear programming with C4.5 decision trees," *Appl. Soft Comput.*, vol. 68, pp. 1–12, Jul. 2018.
[15] Y. C. Ho and D. L. Pepyne, "Simple explanation of the no-free-lunch theorem and its implications," *J. Optim. Theory Appl.*, vol. 115, no. 3, pp. 549–570, Dec. 2002.
[16] A. Gholami, T. Shekari, F. Aminifar, and M. Shahidehpour, "Microgrid scheduling with uncertainty: The quest for resilience," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2849–2858, Nov. 2016.
[17] F. S. Gazijahani, S. N. Ravadanegh, and J. Salehi, "Stochastic multi-objective model for optimal energy exchange optimization of networked microgrids with presence of renewable generation under risk-based strategies," *ISA Trans.*, vol. 73, pp. 100–111, Feb. 2018.
[18] A. Gholami, T. Shekari, M. H. Amirioun, F. Aminifar, M. H. Amini, and A. Sargolzaei, "Toward a consensus on the definition and taxonomy of power system resilience," *IEEE Access*, vol. 6, pp. 32035–32053, 2018.
[19] E. Mezura-Montes and C. A. Coello-Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, Dec. 2011.
[20] A. A. Khan, M. Naeem, M. Iqbal, S. Qaisar, and A. Anpalagan, "A compendium of optimization objectives, constraints, tools and algorithms for energy management in microgrids," *Renew. Sustain. Energy Rev.*, vol. 58, pp. 1664–1683, May 2016.
[21] N. Growe-Kuska, H. Heitsch, and W. Romisch, "Scenario reduction and scenario tree construction for power management problems," in *Proc. IEEE Bologna Power Tech Conf.*, vol. 3, Jun. 2003, pp. 1–7.
[22] M. Raza, *Load Flow Calculation and its Application*. Cham, Switzerland: Springer, 2014, ch. 1, pp. 1–25.
[23] Energie-Control Austria für die Regulierung der Elektrizitäts-und Erdgaswirtschaft, "TOR erzeuger: Anschluss und parallelbetrieb von stromerzeugungsanlagen des typs B," Energie-Control Austria für die Regulierung der Elektrizitäts- und Erdgaswirtschaft, Vienna, Austria, Tech. Rep., Dec. 2019. [Online]. Available: https://tinyurl.com/e-control-tor
[24] D. Pisinger, "Where are the hard knapsack problems?" *Comput. Oper. Res.*, vol. 32, no. 9, pp. 2271–2284, Sep. 2005.
[25] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, Jan. 2011.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[27] A. Gholami, T. Shekari, and S. Grijalva, "Proactive management of microgrids for resiliency enhancement: An adaptive robust approach," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 470–480, Jan. 2019.
[28] A. Kavousi-Fard and A. Khodaei, "Efficient integration of plug-in electric vehicles via reconfigurable microgrids," *Energy*, vol. 111, pp. 653–663, Sep. 2016.
[29] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.
[30] Energie-Control Austria für die Regulierung der Elektrizitäts-und Erdgaswirtschaft, "TOR erzeuger: Anschluss und parallelbetrieb von stromerzeugungsanlagen des typs a und von kleinsterzeugungsanlagen," Energie-Control Austria für die Regulierung der Elektrizitäts- und Erdgaswirtschaft, Vienna, Austria, Tech. Rep., Dec. 2019. [Online]. Available: https://tinyurl.com/e-control-tor-1
[31] H. Meier, C. Fünfgeld, T. Adam, and B. Schieferdecker, "Repräsentative VDEW-lastprofile," Frankfurt, Germany, Tech. Rep. VDEW-Materialien M-28/99, Oct. 1999. [Online]. Available: https://www.bdew.de/media/documents/1999_Repraesentative-VDEW-Lastprofile.pdf
[32] *Day-Ahead Auction*, EPEX SPOT SE, Paris, France, Oct. 2019.
[33] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo-Optimization Modeling in Python*, vol. 67. Cham, Switzerland: Springer, 2017.
[34] S. J. Kimble, D. T. Vedullapalli, and E. B. Makram, "Optimal partitioning of distribution networks for micro-grid operation," *J. Power Energy Eng.*, vol. 5, no. 9, pp. 104–120, Sep. 2017.
[35] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The Benders decomposition algorithm: A literature review," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 801–817, Jun. 2017.

MICHAEL H. SPIEGEL received the B.Sc. and Dipl.-Ing. (master's) degrees (Hons.) in computer engineering from Technische Universität Wien (TU Wien), Vienna, Austria, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Faculty of Mechanical and Industrial Engineering, with a focus on the efficient and resilient control of microgrids and multi-microgrids.

In parallel to his bachelor's and master's studies, he held a various positions at AIT, including two employments as a Research Fellow, where he studied the integration of event-based control and continuous physical models based on open standards. Additionally, for six terms, he worked at TU Wien supporting various courses held at the Faculty of Informatics. He currently works as a Doctoral Fellow at the AIT Austrian Institute of Technology. His research interests include power system optimization, multi-domain modeling and simulation of cyber-physical systems, and future sustainable power systems, in general.

THOMAS I. STRASSER (Senior Member, IEEE) received the master's, Ph.D., and venia docendi (habilitation) degrees in automation from Technische Universität Wien (TU Wien), Vienna, Austria, in 2001, 2003, and 2017, respectively.

For several years, he has been a Senior Scientist with the Center for Energy, AIT Austrian Institute of Technology. Before joining AIT, he spent more than six years as a Senior Researcher investigating advanced and reconfigurable automation and control systems at PROFACTOR Research. He is active as a Senior Lecturer (Privatdozent) at TU Wien. He has been the Project Manager of several national and European research projects. His main research interests include power utility/smart grid automation and corresponding engineering and validation approaches.

Dr. Strasser is a member of IEC and IEEE standardization working groups and as a Senior Member of IEEE, he involved in several activities of IES (AdCom Member-at-Large 2018–2020 and TC Cluster Delegate Energy 2020–2021), SMCS (BoG Member-at-Large 2018–2020 and VP Systems Science and Engineering 2021–2022), SysCo (AdCom Member-at-Large 2021–2022), and PES. He serves as the Austrian Representative for the CIGRE Study Committee C6. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and further IEEE, Springer, Hindawi, and MPDI journals.