Proactive guidance for dynamic and cooperative resource allocation under uncertainties

Gerrit Anders, Florian Siefert, Michael Mair, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Anders, Gerrit, Florian Siefert, Michael Mair, and Wolfgang Reif. 2014. "Proactive guidance for dynamic and cooperative resource allocation under uncertainties." In *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems, 8-12 September 2014, London, UK*, edited by Moez Draief, Salima Hassas, Ada Diaconescu, Nagarajan Kandasamy, and Mirko Viroli, 21–30. Piscataway, NJ: IEEE. https://doi.org/10.1109/saso.2014.14.

Proactive Guidance for Dynamic and Cooperative Resource Allocation under Uncertainties*

Gerrit Anders, Florian Siefert, Michael Mair, and Wolfgang Reif
Institute for Software & Systems Engineering, Augsburg University, Germany
E-Mail: {anders, siefert, reif}@informatik.uni-augsburg.de, michael.m.mair@t-online.de

***Abstract*—In many technical systems, such as smart grids, the central issue is to enable multiple devices to solve a resource allocation problem in a cooperative manner. If the devices' ability to change their contribution is subject to inertia, the problem has to be solved proactively. This means that the allocation of resources is scheduled beforehand, based on predictions of the future demand. Because of the scheduling problem's complexity, schedules should be created rather sporadically for a coarse-grained time pattern. However, because the resource allocation problem has to be solved for all time steps and the demand and provision of resources is uncertain, devices have to reactively adapt their contributions according to the current circumstances.**

In this paper, we present a mechanism that allows the participants to incorporate the information of proactively created schedules in their reactive decisions in order to steer the system in a stable and efficient way. In particular, the decisions are guided by schedules that already include information about possible uncertainties. While this combination avoids inertia-based problems, it significantly reduces the computational costs of searching for high quality solutions. Throughout the paper, the problem of maintaining the balance between energy production and consumption in decentralized autonomous power management systems serves to illustrate our algorithm and results.

Keywords—Resource Allocation Problems; Uncertainty; Multi-Agent Systems; Distributed Problem Solving; Smart Grids

I. The Dynamic Resource Allocation Problem

In a resource allocation problem, a number of system components have to provide a certain amount of resources in order to satisfy a specific demand. Solving a resource allocation problem is at the heart of various technical systems. Many of these, such as supply systems, are mission-critical, i.e., their failure can have massive consequences for people, industries, and public services. In such cases, the system's stability and availability is of utmost importance. In gas pipeline and water supply systems, for instance, the challenge is to maintain the system's pressure at a certain level. Regarding district heating systems, in addition, the network temperature has to be kept between specific bounds. Similarly, the main task in power management systems (PMSs) is to maintain the balance between energy production and consumption at all times.

In this paper, we address a **d**ynamic one-good **r**esource **a**llocation **p**roblem (DRAP) without externalities [1] that has to be solved by a set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$. The macro-level goal is to find an allocation so that, despite uncertainties, in each time step t, the sum $C_{\mathcal{A}}[t]$ of the individual agents' contributions matches a demand $D_{env}[t]$ as accurately as possible. The demand is imposed by the environment *env*. Because it usually exceeds the limited resources of a single agent, they have to solve the DRAP in a cooperative manner. Further, the costs of providing resources should be kept low.

Each agent a's contribution $C_a[t]$ is subject to a minimum contribution of $C_a^{\min}$ and maximum contribution of $C_a^{\max}$. Further, we assume that an agent can change its contribution by at most ΔC_a from one time step to another. The agents' behavior is thus subject to inertia – a property which can be found in many systems that control physical devices, such as power generators or energy storage devices. Because of the minimum and maximum contribution thresholds, the actual values $\Delta C_a^-[t]$, $\Delta C_a^+[t]$ by which a can decrease or increase its contribution from a time step t to $t+1$ is subject to its contribution $C_a[t]$ in time step t:

$$\begin{aligned} \Delta C_a^-[t] &= \min\left(\Delta C_a, C_a[t] - C_a^{\min}\right) \\ \Delta C_a^+[t] &= \min\left(\Delta C_a, C_a^{\max} - C_a[t]\right) \end{aligned} \tag{1}$$

On this basis, an agent's minimum $C_a^{\min}[t]$ and maximum contribution $C_a^{\max}[t]$ for a time step $t+1$ is thus defined follows:

$$\begin{aligned} C_a^{\min}[t+1] &= C_a[t] - \Delta C_a^-[t] \\ C_a^{\max}[t+1] &= C_a[t] + \Delta C_a^+[t] \end{aligned} \tag{2}$$

All agents have to adhere to these constraints when deciding about their contributions. Since the agents can represent physical devices like generators, we refer to a constraint model of agent a as *control model* $\mathcal{M}_a$ in the following.

Because of the property of inertia, the agents' contribution might not change quickly enough to reactively adapt to the demand in all situations. Consequently, their contribution has to be *proactively* fixed in the form of *schedules* for N discrete future time steps (e.g., by centralized [2], [3], market-based [4], or decentralized [5] algorithm). A schedule prescribes an agent's contribution for all time steps contained in the so-called *scheduling window* $\mathcal{W} = \{t_{\text{now}} + i \cdot \Delta\pi \mid i \in \{0, 1, \ldots, N\}\}$. With regard to the current time step t_{now}, the scheduling window starts in time step $t_{\text{now}} + \Delta\pi$. Based on the schedule's resolution $\Delta\pi$ and $N \in \mathbb{N}^+$, the *scheduling horizon* $H = N \cdot \Delta\pi$ specifies the regarded time frame.[1] When creating schedules, the current state of the system and predictions of the future demand have to be taken into account. Schedules must be recalculated every $F^{-1} \le H$ time steps, where F^{-1} is a multiple of $\Delta\pi$. Scheduling introduces two central challenges:

1) The DRAP has to be solved in spite of uncertainties. These are introduced by the environment in the form

*This research is partly sponsored by the research unit *OC-Trust* (FOR 1085) of the German Research Foundation.

[1] We define $\Delta\pi$ as a multiple of the difference Δt between two successive time steps t and $t+1$.

of inaccurate demand predictions as well as by the system participants themselves.

2) Solving the scheduling problem is NP-hard [6]; both with regard to the number of agents $|\mathcal{A}|$ involved and time steps N schedules are created for in advance, resulting in a complexity of $O\left(2^{|\mathcal{A}|\cdot N}\right)$.

In fact, all aforementioned systems solve an instance of the DRAP in which the demand has to be satisfied despite fluctuations and unexpected events. In power management systems (PMSs), e.g., because of changing weather conditions and stochastic consumer behavior, the so-called residual load[2], i.e., the fraction of the load that has to be fulfilled by dispatchable power plants and consumers, is subject to fluctuations. Dispatchable power plants might further not be able to comply with their schedules in the light of technical difficulties.

Since we assume that demand predictions become more accurate as a future point in time approaches, one could decide to recalculate schedules in every time step by setting $F^{-1} = 1$. However, as the scheduling problem is NP-hard, the scheduling frequency should be kept low. While each agent $a \in \mathcal{A}$ must be provided with a schedule and the scheduling horizon must be well-chosen with regard to the system's properties, the resolution $\Delta\pi$ of schedules can be adjusted in order to reduce computational costs. To deal with uncertainties at runtime, it is thus crucial to allow agents to deviate from their schedules.

In [7], we investigated the influence of *inter-agent variation* on the system's behavior when solving the DRAP in a fully reactive and distributed manner, i.e., without incorporating schedules. Inter-agent variation was found in the agents' *sensitivity*, defining when and how strong to react to deviations between demand and contribution. Our investigations showed that the distribution of sensitivity is crucial to the agents' ability to solve the DRAP in a specific situation (i.e., system state). Too little variation leads to oscillations, whereas too much variation to slow convergence. While we assumed that the distribution of sensitivities was predefined and a static property of the system, we found out that an adequate amount of variation allows to significantly reduce communication overhead when solving the DRAP.

In this paper, we present an approach that solves the DRAP by combining proactive resource allocation, i.e., scheduling, and reactive decisions to compensate for uncertainties in a scalable self-organizing hierarchical system structure at runtime. Because we assume that the schedules allow the system to operate optimally with regard to a specific situation, the agents try to self-adapt in a way that allows them to compensate for deviations locally, i.e., right where they occur, while adhering to their schedules as accurately as possible. Moreover, we base our approach on a scheduling principle that 1) anticipates *aleatoric*, i.e., intrinsic random and irreducible, uncertainties by creating schedules for different possible developments of the future demand and 2) reduces *epistemic*, i.e., systematic, uncertainties by putting more effort into the optimization of likely scenarios. Schedules thus serve as a blueprint for how to reactively deal with deviations from predicted and expected behavior at runtime. In other words, they guide the agents' reactive decisions. In addition to schedules, the agents' control models constitute a dynamic and situation-specific source of inter-agent variation of sensitivity. To the best of our knowledge, our approach to solve the DRAP is the first that combines a proactive and a reactive component by an explicit representation of uncertainties in the schedules.

Throughout the paper, the problem of maintaining the balance between production and consumption in decentralized autonomous PMSs serves to illustrate our algorithm and results. We introduce this case study in Section II and outline the calculation of robust schedules in Section III. Subsequently, we present our solution to the DRAP on the basis of robust schedules in Section IV. In Section V, we demonstrate that our approach efficiently and effectively solves the DRAP by exploiting the schedules' structure as well as the agents' control models as a source of sensitivity. Related work is discussed in Section VI, before we conclude the paper and give an outlook on future work in Section VII.

II. Running Example: Maintaining the Balance of Production and Consumption in Smart Grids

The wide-spread installation of weather-dependent power plants as well as the advent of new consumer types like electric vehicles put a lot of strain on the power grid. Additionally, small dispatchable power plants owned by individuals or cooperatives feed in power without external control. To save expenses, gain more flexibility, and deal with uncertainties, future *autonomous* PMSs have to take advantage of the full potential of dispatchable prosumers[3] by incorporating them into the scheduling scheme. Further, uncertainties have to be anticipated when creating schedules and compensated for locally in order to prevent their propagation through the system.

The problem of balancing production and consumption is an instance of our DRAP: a dispatchable power plant's output corresponds to a contribution and the residual load to the demand.[4] In the synchronous grid of Continental Europe, schedules are typically created with a resolution of $\Delta\pi = 15$ minutes, whereas imbalances have to be detected and compensated for within seconds to ensure the grid's stable operation [8]. The balance is maintained by a series of three successive interdependent control actions. While primary control underlies a globally standardized mechanism, the nowadays semi-automatic and centralized secondary control may be realized differently in each control area. This gives room for new approaches like the one presented in this paper.

In [9], we presented the concept of *Autonomous Virtual Power Plants* (AVPPs) as an approach to meet the challenges of future PMSs. AVPPs represent self-organizing groups of two or more power plants of various types (see Figure 1). Each AVPP has to satisfy a fraction of the overall demand. To accomplish this task, each AVPP autonomously and periodically calculates schedules for subordinate[5] dispatchable power plants (see Section III). Further, each AVPP's dispatchable power plants have to reactively compensate for local output or load fluctuations to avoid the affection of other parts of the system.

Because AVPPs autonomously adapt their structure to changing internal or environmental conditions, they can live up

[2]The residual load is defined as the difference between the overall non-dispatchable consumption and the output of non-dispatchable power plants.

[3]We use the term "prosumer" to refer to producers as well as consumers.

[4]The constraints of power plants are usually a superset of those in Section I.

[5]"Subordinate" plants are those for which an AVPP is *directly* responsible.

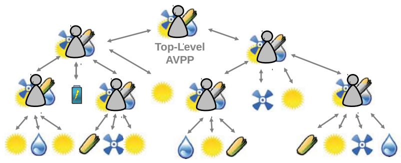

Fig. 1. Power plants self-organize into a hierarchical structure of AVPPs, thereby decreasing the complexity of control and scheduling.

to the responsibility of maintaining a structure that enables the system to hold the balance between energy supply and demand. In particular, if an AVPP cannot repeatedly satisfy the assigned fraction of the overall demand or compensate for output or load fluctuations locally, it triggers a reorganization. The goal is to constitute an anticlustering, i.e., a structure of homogeneous AVPPs that feature a heterogeneous composition: 1) By establishing a mixture of reliable and unreliable power plants, uncertainties originating from producers are distributed among AVPPs. That way, the chance of fluctuations is reduced and the system's robustness increases. 2) By balancing the AVPPs' degrees of freedom (e.g., by mixing different generator types), their ability to locally react to fluctuations is promoted.

Due to the scheduling problem's complexity, scalability becomes a major concern in future PMSs. We therefore proposed a scalable, hierarchical structure in which AVPPs act as intermediaries. Compared to a non-hierarchical approach, the number of subordinate power plants each AVPP controls is reduced (note that AVPPs are dispatchable power plants as well). This results in shorter scheduling times for each AVPP.

While we focus on power generation in this paper, both dispatchable and intermittent power consumers can easily be integrated into AVPPs as well as the presented algorithms.

III. Capturing Uncertainties in Robust Schedules

In our hierarchical system, AVPPs autonomously create schedules in a regionalized and top-down manner. We write $a \in \mathcal{A}$ for a dispatchable power plant in general and $\lambda \in \mathcal{I} \subset \mathcal{A}$ for an AVPP. The *top-level AVPP* Λ represents the tree's root. Similarly, $\mathcal{A}_\lambda$ denotes the subordinate dispatchable power plants and $\mathcal{I}_\lambda \subseteq \mathcal{A}_\lambda$ the subordinate AVPPs of some AVPP λ. When calculating schedules, AVPPs have to rely on residual load predictions for each time step t in the scheduling window $\mathcal{W}$. However, there are different sources of uncertainty that affect the quality of these predictions. In particular, aleatoric uncertainties originate from inaccurate prosumption[6] predictions created by intermittent power plants or consumers. To ensure the system's stable operation, these uncertainties have to be anticipated and incorporated into the AVPPs' decision making process when creating schedules.

In [10], we formalized the optimization problem of creating robust schedules in a self-organizing hierarchy of AVPPs on the basis of so-called *trust-based scenarios* (TBSs) [11]. TBSs are an approach to approximate the stochastic process that underlies and steers the behavior of a single or a group of agents, such as intermittent power plants or consumers. TBSs result from experiences agents gained in the course of interactions with others. In our case, AVPPs gain experiences that capture the accuracy of prosumption predictions of length N made by one or more intermittent power plants or consumers.

[6]We use the term "prosumption" to refer to production or consumption.

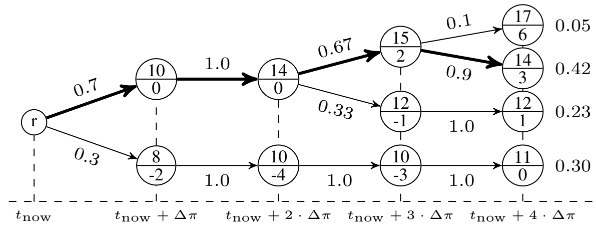

Fig. 2. A figure of two TBSTs: A TBST representing different deviation scenarios (lower part of the nodes) and the resulting TBST representing possible developments of the local residual load (upper part of the nodes) of an AVPP for a scheduling window $\mathcal{W} = \{t_{\text{now}} + \Delta\pi, \ldots, t_{\text{now}} + 4 \cdot \Delta\pi\}$. Except for the root r, every other node represents an expected local residual load (upper part of the node) in the corresponding future time step, which is calculated by adding the deviation (lower part of the node) of a deviation scenario to a local residual load prediction $(10, 14, 13, 11)$ (all values in MW). The values at the TBST's edges indicate the conditional probabilities that the local residual load changes from one value to another. Consequently, we can define the probability $p(n)$ that a node n occurs as follows: $p(r) = 1$, $n \neq r \rightarrow p(n) = p(n|f(n)) \cdot p(f(n))$, where $f(n)$ returns the father of a node n. Each path from r to a leaf is a TBS whose probability is equal to that of the associated leaf. The highlighted path indicates the *anticipated TBS* s_α. It results from, starting at the root, taking the most probable alternative at every node. Note that s_α is not necessarily the most likely TBS.

The accuracy is measured as the deviation between actual and predicted prosumption. Each TBS thus represents a sequence of possible deviations from a prosumption prediction of a single or a group of observed prosumers for a specific future time frame. Regarding the behavior of a group instead of single individuals allows to capture stochastic dependencies (as is often the case with weather predictions for adjacent regions or consumer attitude). Depending on how often a specific sequence of behavior was observed, AVPPs assign a probability of occurrence to each TBS. In conjunction with a prediction, an AVPP uses these deviation scenarios to obtain another set of TBSs that represent possible developments of the predictors' future prosumption. This is done by adding each deviation scenario to the prediction. A set of TBSs can be combined to a *trust-based scenario tree* (TBST), which is annotated with conditional probabilities that the prosumption develops in a specific direction. An example of a TBST representing possible developments of the local residual load and how it is derived from a deviation scenario tree and a local residual load prediction is depicted in Figure 2.

The TBST for an AVPP's local residual load is based on experiences with its subordinate intermittent power plants and consumers. Note that such a TBST intentionally does not capture uncertainties resulting from subordinate AVPPs. This is not necessary because an AVPP that has to propagate at least a part of its local uncertainties to its superordinate AVPP hints at an improper system structure. Instead, a reorganization has to be triggered that re-establishes a suitable distribution of uncertainties and degrees of freedom among AVPPs.

TBSTs are updated every time schedules are calculated. When creating schedules, an AVPP stipulates the output of its dispatchable power plants for every node in its TBST, resulting in schedules that are trees themselves. The AVPPs' primary objective is to satisfy its part of the residual load as accurately as possible, followed by the goal to provide energy at a low price. The *scheduled output* $C_a^{\mathbf{S},s_i}[t]$ of a dispatchable power plant a for a specific node depends on decisions made for the father and child nodes (note that several scenarios might share a path from a TBST's root to a node within the tree). The node is identified by the scenario s_i and time step t. TBST-based schedules allow AVPPs to anticipate aleatoric and reduce epistemic uncertainties by putting more effort into the optimization of likely nodes. Power plants can use TBST-based schedules to react to changing situations and even switch between different TBSs at runtime (see Section IV-B). In Section IV, we show that these characteristics allow power plants to effectively and efficiently solve the DRAP despite aleatoric uncertainties.

The top-level AVPP Λ initiates the calculation of schedules by determining its *local residual load **p**rediction* $D^{\mathbf{P}}_{\Lambda}[t]$, which is the sum of the predicted consumption and the *anticipated **l**ocal residual load* $D^{\mathbf{L},s_\alpha}_{\lambda}[t]$ of subordinate AVPPs $\lambda \in \mathcal{I}_\Lambda$, minus the predicted output of subordinate intermittent power plants, such as wind turbines. As we assume that the consumption is only part of the local residual load at the top-level, the local residual load prediction $D^{\mathbf{P}}_{\lambda}[t]$ of an AVPP $\lambda \neq \Lambda$ is recursively defined as the difference between the sum of the anticipated local residual load of subordinate AVPPs and the predicted output of subordinate intermittent plants. In contrast to the macro level, the local residual load at the meso level is thus most likely negative. The anticipated local residual load $D^{\mathbf{L},s_\alpha}_{\lambda}[t]$ refers to the *anticipated scenario* s_α which reflects the most likely development of λ's ***l**ocal residual load* $D^{\mathbf{L}}_{\lambda}[t]$. λ determines s_α on the basis of its local residual load prediction $D^{\mathbf{P}}_{\lambda}[t]$ and its TBST as explained in Figure 2. By informing its superordinate AVPP about $D^{\mathbf{L},s_\alpha}_{\lambda}[t]$ instead $D^{\mathbf{P}}_{\lambda}[t]$, λ takes on the responsibility to compensate for deviations from this expectation by reactively increasing or decreasing the output of subordinate dispatchable power plants.

Once a schedule was assigned to an AVPP λ, it combines its scheduled output $C^{\mathbf{S},s_\alpha}_{\lambda}[t]$ for the anticipated scenario s_α with the deviation scenario tree of its local residual load. This is achieved analogously to the example described in Figure 2 (note that the local residual load prediction is replaced by $C^{\mathbf{S},s_\alpha}_{\lambda}[t]$). Then it redistributes the residual load of the resulting TBST by calculating schedules for its subordinates. Consequently, a base load power plant's scheduled output might be the same for different residual load scenarios, whereas a peaking power plant's schedule might prescribe output adjustments according to the differences in the scenarios. In the following, we assume that a scheduled output $C^{\mathbf{S},s_i}_{a}[t]$ for a time step $t \notin \mathcal{W}$ is the interpolation of the scheduled outputs $C^{\mathbf{S},s_i}_{a}[t_1]$ and $C^{\mathbf{S},s_i}_{a}[t_2]$ of two adjacent time steps $t_1 < t < t_2$ with $t_1, t_2 \in \mathcal{W}$. For a detailed explanation and formalization of the scheduling problem, we refer the interested reader to [10].

IV. A Proactively Guided and Cooperative Solution to the Resource Allocation Problem

Having introduced the creation and the properties of robust schedules in the previous section, we show how they are used to proactively guide the solution to the DRAP. Please note that the presented approach is independent of the scheduling mechanism. As stated in Section II, the power plants' task is to hold the balance between their output $C_{\mathcal{A}}[t]$ and the residual load $D_{env}[t]$ at all times. Consequently, each power plant has to decide about an output $C_a[t_{\text{next}}]$ for the next time step t_{next} such that, in total, $C_{\mathcal{A}}[t_{\text{next}}]$ matches $D_{env}[t_{\text{next}}]$ in t_{next} as accurately as possible. Because $D_{env}[t_{\text{next}}]$ is not known beforehand, the power plants have to determine and adjust their output according to a *residual load **t**arget* $D^{\mathbf{T}}_{env}[t_{\text{next}}]$. This basically means that they have to compensate for the deviation between their scheduled output for t_{next} and $D^{\mathbf{T}}_{env}[t_{\text{next}}]$.

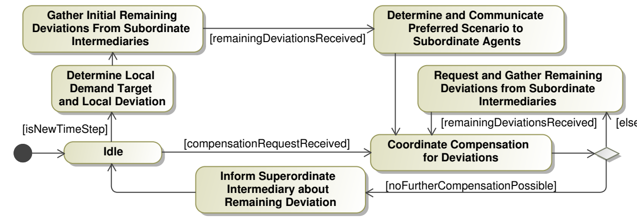

Fig. 3. The coordination of the compensation for deviations from an intermediary's perspective.

A. Hierarchical Decomposition of the DRAP

Since we regard a system of systems, we can decompose the system's task into equivalents for each individual subsystem: With respect to a subsystem $\mathcal{A}_\lambda$ represented by an AVPP λ, the objective of λ's subordinate power plants $a \in \mathcal{A}_\lambda$ is to determine $C_a[t_{\text{next}}]$ such that the subsystem's output $C_\lambda[t_{\text{next}}]$ corresponds to its residual load target $D^{\mathbf{T}}_{\lambda}[t_{\text{next}}]$. As is the case with the creation of schedules (see Section III), each AVPP is thus responsible for satisfying its fraction of the overall residual load. The self-organized distribution of sources of uncertainty and degrees of freedom among AVPPs (see Section II) promotes their ability to achieve this goal and thus prevents the propagation of uncertainties to higher levels.

However, because of the power plants' limited output and inertia and depending on the current situation, a subsystem might (temporarily) not be able to provide the necessary output on its own. In such a case, the ***r**emaining deviation* $\Delta^{\mathbf{R}}_{\lambda}$ between the residual load target and the subsystem's output has to be compensated for by "external" power plants $a \notin \mathcal{A}_\lambda$. The remaining deviation is therefore communicated to λ's superordinate AVPP λ'. Because λ is part of λ''s subsystem, λ', in turn, has to compensate for $\Delta^{\mathbf{R}}_{\lambda}$. In case of the top-level AVPP Λ, $\Delta^{\mathbf{R}}_{\Lambda}$ should only be nonzero if the whole system is not able to meet the overall demand target $D^{\mathbf{T}}_{env}[t_{\text{next}}]$.

B. Bottom-Up Compensation for Deviations

Based on the hierarchical system structure, we propose an iterative bottom-up solution to the DRAP. We opted for a bottom-up procedure since it promotes the *local* compensation for uncertainties. While each power plant autonomously decides how much to contribute in the next time step (see Section IV-C), in each subsystem, the process of adjusting the subordinates' outputs on the basis of their schedules is coordinated by the corresponding AVPP. Figure 3 depicts the AVPPs' responsibilities in the context of compensating for deviations.

Regarding a new time step t_{now}, an AVPP λ first determines its residual load target $D^{\mathbf{T}}_{\lambda}[t_{\text{next}}]$. For the sake of simplicity, we assume that AVPPs use their current *residual load to satisfy* $D_\lambda[t_{\text{now}}]$ as target $D^{\mathbf{T}}_{\lambda}[t_{\text{next}}]$ in this paper. Under the assumption that the period between two successive time steps is short enough so that the residual load does not change significantly from one time step to another, residual load and output can be equalized. Please note that this method can be replaced by an arbitrary prediction mechanism without any side effects on our algorithm.

As stated in Section I, we want to avoid the propagation of uncertainties to higher levels. This means that AVPPs have to cancel out the difference between the *actual* local residual load $D^{\mathbf{L}}_{\lambda}[t_{\mathrm{now}}]$ and the anticipated local residual load $D^{\mathbf{L},s_\alpha}_{\lambda}[t_{\mathrm{now}}]$ they sent to their superordinate AVPP in the course of the calculation of schedules (recall that we defined in Section III that the local residual load originates from subordinate intermittent power plants, consumers, and AVPPs). The *current* residual load $D_{\lambda}[t_{\mathrm{now}}]$ an AVPP has to satisfy is thus defined as the sum of the scheduled output $C^{\mathbf{S},s_\alpha}_{\lambda}[t_{\mathrm{now}}]$ for the anticipated scenario s_α and the above-mentioned difference:

$$D_{\lambda}[t_{\mathrm{now}}] = C^{\mathbf{S},s_\alpha}_{\lambda}[t_{\mathrm{now}}] + \left(D^{\mathbf{L}}_{\lambda}[t_{\mathrm{now}}] - D^{\mathbf{L},s_\alpha}_{\lambda}[t_{\mathrm{now}}]\right) \quad (3)$$

For instance, if the AVPP communicated an anticipated local residual load of $D^{\mathbf{L},s_\alpha}_{\lambda}[t_{\mathrm{now}}] = -5$ MW (in this case, an output that ought to be provided by subordinate intermittent power plants), the scheduled output $C^{\mathbf{S},s_\alpha}_{\lambda}[t_{\mathrm{now}}]$ for s_α is 20 MW, and the actual local residual load $D^{\mathbf{L}}_{\lambda}[t_{\mathrm{now}}]$ is -2 MW (i.e., the intermittent power plants' actual output), the current residual load this AVPP has to satisfy is $D_{\lambda}[t_{\mathrm{now}}] = 23$ MW.

Since we set $D^{\mathbf{T}}_{\lambda}[t_{\mathrm{next}}] = D_{\lambda}[t_{\mathrm{now}}]$, subordinate power plants have to cushion the *local deviation* $\Delta^{\mathbf{L}}_{\lambda}$, which we define as the mismatch between $D^{\mathbf{T}}_{\lambda}[t_{\mathrm{next}}]$ and the AVPPs' *schedule-compliant output* $C^{\mathbf{C}}_{\lambda}[t_{\mathrm{next}}] = \sum_{a\in\mathcal{A}_\lambda} C^{\mathbf{C}}_{a}[t_{\mathrm{next}}]$ for the *next* time step. The schedule-compliant output of a physical power plant $a \in \mathcal{A}$ is the output that is closest to its scheduled output $C^{\mathbf{S},s_\rho}_{a}[t_{\mathrm{next}}]$ for a specific scenario s_ρ, while being consistent with a's control model. Because a power plant's actual and scheduled output are likely to differ as a result of the compensation for uncertainties, its current output $C_a[t_{\mathrm{now}}]$ does not necessarily equal its scheduled output $C^{\mathbf{S},s_\rho}_{a}[t_{\mathrm{now}}]$. It might thus not be able to reach $C^{\mathbf{S},s_\rho}_{a}[t_{\mathrm{next}}]$ in the next time step. The same applies to the schedule-compliant output of its superordinate AVPP. Note that AVPPs assume in this phase that subordinate AVPPs λ' stick to their scheduled output for the anticipated scenario s_α, i.e., $C^{\mathbf{C}}_{\lambda'}[t_{\mathrm{next}}] = C^{\mathbf{S},s_\alpha}_{\lambda'}[t_{\mathrm{next}}]$ (a justification for this assumption follows).

Nevertheless, we want the power plants to stick to their schedules as close as possible because of the following reasons: Recall that uncertainties in the residual load's development of each individual subsystem are reflected by multiple different residual load scenarios. Since these scenarios form the basis for the creation of robust schedules, we can assume that the schedules give a suitable indication of how power plants should adjust their output in different situations in order to cooperatively satisfy a specific residual load with regard to the scheduling problem's objectives (see Section V). Further, a deviation between the actual residual load and a residual load scenario covered by the schedule might only be of temporary nature. In such a situation, the power plants benefit from being able to quickly return to their scheduled contribution. The ability to return to a schedule can be expressed as the time a power plant would need to change its production to the scheduled output, i.e., the ratio between the schedule deviation $|C^{\mathbf{S},s_\rho}_{a}[t_{\mathrm{next}}] - C^{\mathbf{C}}_{\lambda}[t_{\mathrm{next}}]|$ and the power plant's inertia ΔC_a. *Fairness* is regarded in the way that we want an AVPP's subordinate power plants to be equally capable of returning to their scheduled output (i.e., the variance should be small).

Before AVPPs start to coordinate the compensation for deviations, they have to identify and inform their subordinate power plants about the scenario they should be able to return to. We call this scenario the *preferred scenario* s_ρ. It is defined as the scenario that comes closest to the actual local residual load. Consequently, instead of having to stick to a predetermined scenario until TBSTs and schedules are recalculated, AVPPs can take advantage of robust schedules and switch between different scenarios that are most relevant for the current situation. An AVPP determines the preferred scenario on the basis of the actual local residual load $D^{\mathbf{L}}_{\lambda}[t]$ in the last n_{s_ρ} time steps. By laying stress on newer measurements, AVPPs adapt to new situations in a short time.

As we regard a bottom-up approach, an AVPP λ waits until all its subordinate AVPPs $\lambda' \in \mathcal{I}_\lambda$ have solved their subsystems' DRAP before it triggers the compensation for the local mismatch $\Delta^{\mathbf{L}}_{\lambda}$. Once a subordinate AVPP λ' has solved the DRAP, it informs λ about its remaining deviation $\Delta^{\mathbf{R}}_{\lambda'}$. As soon as all remaining deviations are available, λ initiates the compensation for $\Delta^{\mathbf{L}}_{\lambda}$ and the *subordinate deviation* $\Delta^{\downarrow}_{\lambda} = \sum_{\lambda'\in\mathcal{I}_\lambda} \Delta^{\mathbf{R}}_{\lambda'}$. In Section IV-C, this procedure is explained in more detail. The compensation is completed if the subordinate power plants' outputs could be adjusted such that the deviation is dissolved (i.e., $\Delta^{\mathbf{R}}_{\lambda} = 0$) or no further adjustments that improve the compensation by more than a given threshold are possible. Note that if all subordinate AVPPs can deal with their subsystems' uncertainties on their own, λ is not affected by the uncertainties they have to cope with and $\Delta^{\downarrow}_{\lambda} = 0$. From λ's perspective, these AVPPs appear to be balanced. This is why λ may assume that the schedule-compliant output $C^{\mathbf{C}}_{\lambda'}[t_{\mathrm{next}}]$ of a subordinate AVPP λ' equals its scheduled output $C^{\mathbf{S},s_\alpha}_{\lambda'}[t_{\mathrm{next}}]$ for the anticipated scenario s_α. As a result, in such a situation, λ only has to deal with uncertainties originating from its local environment or the superordinate AVPP.

In the course of a compensation coordinated by λ, subordinate AVPPs can commit themselves to adjust their output. A subordinate AVPP $\lambda' \in \mathcal{I}_\lambda$ regards its output adjustment as a deviation $\Delta^{\uparrow}_{\lambda'}$ it stands in for λ and that has to be redistributed among its own subordinate power plants. Consequently, whenever a compensation is completed, λ asks all subordinate AVPPs that have adjusted their output to compensate for $\Delta^{\uparrow}_{\lambda'}$. Since subordinate AVPPs represent independent subsystems, compensations can be performed in parallel, thereby enabling fast reactions to changing environmental conditions. If no such subordinate AVPP exists, λ informs its superordinate AVPP about the remaining deviation $\Delta^{\mathbf{R}}_{\lambda}$.

Summarizing, the DRAP each AVPP λ and its subordinate power plants $\mathcal{A}_\lambda$ have to solve can be formalized as follows:[7]

$$\begin{aligned}
\underset{C_a[t_{\mathrm{next}}],\ a\in\mathcal{A}_\lambda}{\text{minimize}} \quad & \left|\Delta_\lambda - \sum_{a\in\mathcal{A}_\lambda}\left(C_a[t_{\mathrm{next}}] - C^{\mathbf{C}}_{a}[t_{\mathrm{next}}]\right)\right| \quad (4)\\
\text{with} \quad & \Delta_\lambda = \Delta^{\mathbf{L}}_{\lambda} + \Delta^{\downarrow}_{\lambda} + \Delta^{\uparrow}_{\lambda'}\\
\text{subject to} \quad & \forall a \in \mathcal{A}_\lambda : \mathcal{M}_a(C_a[t_{\mathrm{now}}], C_a[t_{\mathrm{next}}])
\end{aligned}$$

Starting with the schedule-compliant output $C^{\mathbf{C}}_{a}[t_{\mathrm{next}}]$, subordinate power plants $a \in \mathcal{A}_\lambda$ try to compensate for the sum Δ_λ of all relevant deviations as accurately as possible by adjusting

[7] Note that deviations caused by non-compliance with schedules (e.g., due to technical difficulties) can be compensated for by adding them to Δ_λ.

their output $C_a[t_{\text{next}}]$. The function $\mathcal{M}_a(C_a[t_{\text{now}}], C_a[t_{\text{next}}])$ checks if a's output adheres to its control model $\mathcal{M}_a$. In the following section, we explain in which way power plants adjust their output $C_a[t_{\text{next}}]$ on the basis of a given deviation, their robust schedule, and the preferred scenario.

C. Proactively Guided Iterative Compensation for Deviations

As indicated in Section I, we want the power plants to make use of their schedules and control models when deciding about output adjustments. More precisely, schedules and control models should define a power plant's individual and situation-specific sensitivity to deviations. Schedules and control models thus serve as a source of inter-agent variation that can reduce the number of iterations needed to solve the DRAP [7].

For this reason, we assume that, in each time step, every power plant has a schedule with a scheduling window $H \geq 2 \cdot \Delta\pi$. Recalling that $\Delta\pi$ is the schedules' resolution and schedules are recalculated every $F^{-1} \leq H$ time steps, this constraint ensures that the power plants' reactive decisions can be guided by their schedules. Further, each power plant must have knowledge of its own control model. For the sake of simplicity, we assume that the scheduling mechanism only assigns residual loads to AVPPs they are able to redistribute.

With regard to the procedure described in Section IV-B, power plants compensate for deviations by adjusting their output for the next time step in an iterative process. In each iteration, the corresponding AVPP informs its subordinate power plants about the remaining deviation $\Delta_\lambda^{\mathbf{R}}$, which then concurrently and independently from each other decide about an output adjustment, i.e., they decide about a ***n****ew output* $C_a^{\mathbf{N}}$ (see Equation 4). This output is restricted to the power plant's physical constraints (i.e., $C_a[t_{\text{next}}] = \max\{C_a^{\min}[t_{\text{next}}], \min\{C_a^{\max}[t_{\text{next}}], C_a^{\mathbf{N}}\}\}$). Afterwards, the power plants inform their superordinate AVPP λ about $C_a[t_{\text{next}}]$ which, in turn, updates the remaining deviation according to the output adjustments and starts a new iteration. The way the power plants compensate for deviations and which information has which impact on a power plant's output adjustment depends on the iteration as well as whether it is the first compensation in this time step. Figure 4 summarizes the different possibilities to react to a deviation.

1) Dealing with Overreactions: In general, if the power plants' total possible output adjustment exceeds the remaining deviation $\Delta_\lambda^{\mathbf{R}}$, there is the chance of an overreaction. In this context, we define the $need = 1 + \frac{\Delta_\lambda^{\mathbf{R}}}{\Delta_\lambda^{\mathbf{R},prev} - \Delta_\lambda^{\mathbf{R}}}$ as the percentage of the last total output adjustment $\Delta c_{\mathcal{A}_\lambda}^{prev} = \Delta_\lambda^{\mathbf{R}} - \Delta_\lambda^{\mathbf{R},prev}$ that would have been necessary to cancel out the remaining deviation $\Delta_\lambda^{\mathbf{R},prev}$ of the previous iteration (i.e., $\Delta_\lambda^{\mathbf{R},prev} = \Delta c_{\mathcal{A}_\lambda}^{prev} \cdot need$). Note that the denominator equals $-\Delta c_{\mathcal{A}_\lambda}^{prev}$. For instance, if we have $\Delta_\lambda^{\mathbf{R},prev} = -20$ MW for the previous and $\Delta_\lambda^{\mathbf{R}} = 10$ MW for the current iteration, $need = \frac{2}{3}$.

Power plants notice the presence of an overreaction if the sign of the remaining deviation changes between two successive iterations. In such a situation, we have $0 < need < 1$. A power plant a reacts to such a situation by adjusting its output by $\Delta c_a = (need - 1) \cdot \Delta c_a^{prev}$, where Δc_a^{prev} is a's last output adjustment. If all power plants' resulting outputs abide by their control models (i.e., if $\forall a \in \mathcal{A}_\lambda : C_a[t_{\text{next}}] = C_a^{\mathbf{N}}$), $\Delta_\lambda^{\mathbf{R}}$ will

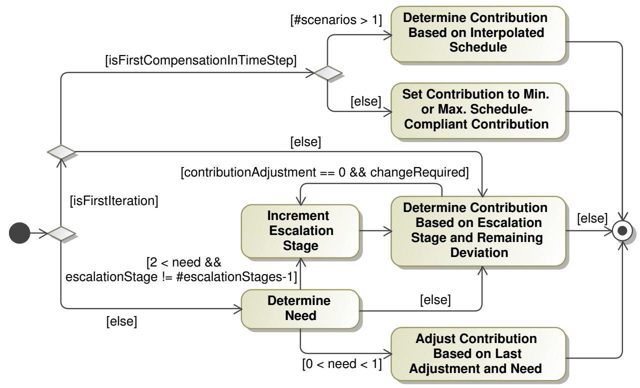

Fig. 4. An agent's basic procedure when participating in a compensation for deviations. The decision about the future contribution is based on the target demand (if available), the deviation that has to be compensated for, the scheduled contribution for different demand scenarios, and the agent's control model. Except for the first iteration and the first compensation in a specific time step, all adjustments are based on the remaining deviation. In the last escalation stage, adjustments are not necessarily schedule-compliant and only limited by the agent's control model.

be absorbed. Note that the remaining deviation $\Delta_\lambda^{\mathbf{R}}$ is the only information that is shared in the subsystem.

2) Interpolation-Based Adjustment of the Contribution: In the first compensation of a time step, AVPPs compensate for the local deviation $\Delta_\lambda^{\mathbf{L}}$ and the initial subordinate deviation $\Delta_\lambda^{\downarrow}$. In the first iteration of the first compensation, subordinate power plants decide about their output for t_{next} on the basis of a residual load $D_\lambda = D_\lambda^{\mathbf{T}}[t_{\text{next}}] + \Delta_\lambda^{\downarrow}$ if their schedule contains more than one scenario. In such a case, a power plant simply sets $C_a^{\mathbf{N}}$ to the ***i****nterpolated scheduled output* $C_a^{\mathbf{I}}$ of the two residual load scenarios $D_\lambda^{s_1}[t_{\text{next}}]$ and $D_\lambda^{s_2}[t_{\text{next}}]$ that are "nearest" to D_λ. If D_λ is between two scenarios, we define $D_\lambda^{s_1}[t_{\text{next}}]$ and $D_\lambda^{s_2}[t_{\text{next}}]$ as the nearest residual load scenario above and below D_λ, respectively. When adjusting the output, the gradient G_a, which is the ratio between the scenario-based change in contribution and the absolute difference between the residual load scenarios, serves as a source of sensitivity:

$$G_a = \frac{C_a^{\mathbf{S},s_1}[t_{\text{next}}] - C_a^{\mathbf{S},s_2}[t_{\text{next}}]}{\left|D_\lambda^{s_1}[t_{\text{next}}] - D_\lambda^{s_2}[t_{\text{next}}]\right|} \tag{5}$$

Based on G_a, a power plant determines its interpolated scheduled output $C_a^{\mathbf{I}}[t_{\text{next}}]$ (if D_λ is not between two scenarios, we define that $D_\lambda^{s_1}$ is closer to D_λ than $D_\lambda^{s_2}$):

$$C_a^{\mathbf{I}} = C_a^{\mathbf{S},s_2}[t_{\text{next}}] + G_a \cdot |D_\lambda - D_\lambda^{s_2}[t_{\text{next}}]| \tag{6}$$

Setting $C_a^{\mathbf{N}}$ to $C_a^{\mathbf{I}}$, it is possible that $C_a^{\mathbf{N}}$ does not comply with a's control model. However, if the interpolated output of all power plants in the subsystem comply with their constraints, the deviation completely dissolves after the first iteration. In any case, the schedules are a good indicator of how much output to produce in a specific situation. For example, a base load power plant whose output should not change significantly with the residual load in different scenarios, most likely should also provide a very similar output in other situations (i.e., its sensitivity is low). On the other hand, a peaking power plant might have to scale its output with the residual load (i.e., its sensitivity is high). Hence, sensitivities are predefined by the scheduling mechanism, which allows the system to make reactive decisions with respect to the scheduling problem's objectives. Note that this might lead to situations in which some power plants increase and others decrease their output. While this was probably the scheduling mechanism's intention, in all other compensations and iterations, we make certain that power plants adjust their output in the same direction to ensure progress in the compensation for deviations.

3) Schedule-Compliant Contribution Adjustments: If the residual load is above or below all scenarios, the interpolated output $C_a^{\mathbf{I}}$ is additionally restricted to the minimum or maximum value that is considered as schedule-compliant (see Section IV-B) with regard to the preferred output $C_a^{\mathbf{S},s_\rho}[t_{\text{next}}+1]$ in time step $t_{\text{next}}+1$. This ensures that power plants only disregard their schedules if necessary. Further, if only one scenario is available in the first iteration and compensation, power plants set $C_a^{\mathbf{N}}$ to their min. or max. schedule-compliant output. While this measure is likely to cause an overreaction that is damped in the subsequent iteration (see Section IV-C1), it is fair since the power plants' ability to return to their preferred output is balanced. In these situations, sensitivity is defined by a power plant's control model with respect to its preferred output.

4) Escalation of Sensitivity: In all following iterations or compensations within a time step, power plants adapt their output in an escalating manner. The idea is to use a sequence of escalation stages to gradually increase the power plants' sensitivity with the number of iterations. By this means, we encourage the solution of the DRAP while allowing the power plants to adhere to schedules as accurately as possible.

The escalation stages range from schedule-based reactions, over the full exploitation of the schedule-compliant output, to exclusively control-model-based decisions. While we present only the basic idea of some of these stages in this paper, each stage specifies a factor that is used to scale the output adjustment linearly to the given deviation. In the first escalation stage, for example, we use the average gradient (see Section IV-C2) over all adjacent scenarios in the schedule. If $need > 2$ (see Section IV-C1) less than half of the last iteration's deviation was compensated. In such a situation, a power plant switches to the next escalation stage to increase its sensitivity. In case the deviation has not changed from one iteration to another, each power plant switches to the next higher escalation stage that leads to an output change (note that this depends on the individual power plant). To promote schedule compliance, except for the last escalation stage, it is ensured that the output is schedule-compliant (see Section IV-C3). Finally, in the last escalation stage, power plants adjust their output as much as possible (i.e., they set the output to $C_a^{\min}[t_{\text{next}}]$ or $C_a^{\max}[t_{\text{next}}]$). While schedule compliance is not ensured anymore in this stage, such a reaction is very likely to cause an overreaction that is compensated for in the subsequent iteration on the basis of the $need$, yielding fair adjustments with regard to the power plants' inertia. However, before such drastic measures are taken, power plants stay in a waiting stage as long as the compensation makes any progress (i.e., if $changeRequired == false$ in Figure 4). This ensures that all power plants in a subsystem have tried to adapt their output as much as possible without infringing schedule compliance.

V. Evaluation

In the context of our case study, we evaluated our approach in four different settings that mainly differ in the information power plants use to determine their sensitivities and output adjustments. For each evaluation scenario, we performed 200 simulation runs. All presented results are average values.

A. Test Bed

We base our evaluation on a hierarchical structure of AVPPs, consisting of 173 dispatchable and 350 intermittent power plants of different types (hydro, biofuel, gas power plants as well as solar plants and wind generators). We used a hierarchy that was established by the power plants in a self-organized manner in preliminary tests. It contains 20 AVPPs and has a height of 5. While each power plant is modeled as an individual agent, we only use a single agent to represent all consumers. Weather-dependent power plants and the consumer constitute the environment *env*. Physical power plant data (such as production boundaries), load curves, and weather data influencing the output of weather-dependent power plants are based on real world data.[8] Each dispatchable power plant's inertia is defined within typical boundaries. The production costs, i.e., the average costs of providing a contribution, range from $6.50 \frac{\text{EUR cent}}{\text{kWh}}$ to $17.50 \frac{\text{EUR cent}}{\text{kWh}}$. The evaluation is implemented in a sequential, round-based execution model in which each round corresponds to a specific time step $t \in \mathcal{T}$. Power plants have to satisfy a prescribed residual load over a period of half a day, corresponding to $|\mathcal{T}| = 720$ discrete time steps, each representing 1 minute. Every $F^{-1} = 15$ minutes, AVPPs create schedules for $H = 1$ hour with a resolution of $\Delta\pi = 15$ minutes (i.e., $N = 4$) on the basis of residual load predictions. The mean residual load is 1.43 GW.

The power plant models and the scheduling problem described in Section III are formulated for and solved with the standard mathematical programming software IBM ILOG CPLEX[9]. When creating schedules, the utmost goal is to satisfy the residual load, followed by the goal to provide energy at a low price. It is thus more expensive to not produce a requested amount of energy than to produce it. Every time schedules are created, the intermittent power plants as well as the consumer predict their future output and consumption for the next $H = 1$ hour. The AVPPs then use these predictions to create TBSTs for the local residual load and schedule subordinate dispatchable power plants accordingly. For the generation of TBSTs, we use parameters that proved to be beneficial in preliminary tests: TBSTs are generated on the basis of the 10 latest experiences, allowing AVPPs to adapt to changing behavior. While the top-level AVPP Λ is able to measure the quality of the residual load predictions with an accuracy of 3.0%, all other AVPPs can perceive erroneous predictions with an accuracy of 20.0% of the mean local residual load of the last 10 experiences. Λ uses a higher resolution because its local residual load is greater as it includes the consumption.

For evaluation, the intermittent power plants of each AVPP introduce uncertainty by predicting their actual future output modified by a random prediction error (uncertainties caused by dispatchable power plants are not regarded). The mean prediction error was 3.57% of the actual local residual load with a standard deviation of 5.25%. To reflect time-dependent

[8] See EnergyMap (2012, http://www.energymap.info/), LEW (2012, http://www.lew-verteilnetz.de/), and LfL (2010, http://www.lfl.bayern.de/).

[9] IBM ILOG CPLEX Optimizer, Version 12.4, 2011: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

	GradualSensitivity	*AllEscStages*	*LastEscStage*	*BadPrefScenario*
Mean Max. Sequential Compensation Runtime [ms]	7.47 (9.88)	8.02 (10.16)	6.33 (9.66)	7.62 (10.24)
Mean Aggr. Compensation Runtime per AVPP [ms]	1.30 (2.40)	1.40 (2.49)	1.16 (2.23)	1.34 (2.49)
Mean #Compensations	2.09 (1.00)	2.08 (1.01)	3.13 (1.03)	2.14 (1.01)
Mean #Iterations	3.49 (2.23)	3.86 (2.49)	2.02 (0.20)	3.56 (2.25)
Mean #Switches of the Preferred Scenario	1.56 (0.68)	1.56 (0.69)	1.57 (0.70)	1.95 (0.71)
Mean Local Deviation [%]	76.76 (27.82)	77.62 (27.46)	66.33 (27.00)	77.51 (26.93)
Mean Subordinate Deviation [%]	0.48 (5.06)	0.47 (5.07)	0.50 (5.24)	0.44 (4.69)
Mean Superordinate Deviation [%]	22.76 (27.49)	21.91 (27.10)	33.17 (26.81)	22.06 (26.65)
Mean Schedule Compliant Output [%]	88.32 (17.9)	88.74 (17.50)	92.04 (11.53)	86.03 (20.11)
Mean Used Percentage of Max. Power Change [%]	7.30 (8.21)	6.75 (7.71)	4.49 (2.81)	8.77 (9.71)

TABLE I. EVALUATION RESULTS FOR DIFFERENT EVALUATION SCENARIOS. ALL VALUES ARE AVERAGES CALCULATED OVER 200 SIMULATION RUNS FOR EACH EVALUATION SCENARIO. VALUES IN PARENTHESES DENOTE STANDARD DEVIATIONS.

behavior, the error for a time step depends on the prediction error of previous time steps. The accuracy of predictions of intermittent power plants within the same AVPP is coupled as is the case with weather predictions for different locations. Power plants compensate for these uncertainties by solving the DRAP as described in Section IV in every time step.

B. Results

Table I summarizes our evaluation results for different configurations. Independently of a specific evaluation scenario, we observe that the average compensation runtime of about 1.30 ms per AVPP as well as the maximum sequential compensation runtime of about 7.47 ms (i.e., the maximum of the sums of the compensation times in each branch originating from the root of the AVPP structure) is only a fraction of the average scheduling runtime of 351.05 ms (with a standard deviation of $\sigma = 345.06$ ms) per AVPP or the max. seq. scheduling runtime of 2394.54 ms ($\sigma = 929.04$ ms). On average, AVPPs created schedules for 3.70 scenarios ($\sigma = 2.17$). This illustrates the benefit of solving the DRAP in a reactive manner on the basis of several scenarios in terms of computational costs as discussed in Section I. Furthermore, the power plants managed to reactively dissolve all deviations in all evaluation scenarios.

In *LastEscStage*, power plants compensate for deviations only by using the last escalation stage (see Section IV-C4). Hence, their reactions are not influenced by their schedules but only driven by their control models. Since this is likely to cause an overreaction that is damped in a subsequent iteration, the power plants need only 2.02 iterations on average to dissolve a deviation. Despite these overreactions, our procedure ensures that fairness with respect to the power plants' inertia is respected. This is reflected in the standard deviation of 2.81% of the percentage of $C_a^{\min}[t_{\text{next}}]$ or $C_a^{\max}[t_{\text{next}}]$ power plants make use of when compensating for deviations ($\mu = 4.49\%$). Additionally, the low standard deviation of 11.53% of schedule-compliant outputs shows that the procedure is also fair in terms of the power plants' ability to return to their preferred scheduled output ($\mu = 92.04\%$). However, as AVPPs usually can change their output faster than physical power plants, this single source of sensitivity leads to a propagation of deviations to subordinate AVPPs. Regarding a single compensation, only 0.50% of the deviations are propagated to the superordinate AVPP on average, but 33.17% of a deviation stems from the superordinate AVPP. This yields to a relatively high average number of 3.13 compensations per AVPP and time step. To prevent this top-down propagation of uncertainties, the scheduling problem's objective function, in fact, included a term that favored solutions in which AVPPs should provide similar outputs for different residual load scenarios (i.e., AVPPs should act as base load power plants). Our evaluation shows that this objective is ignored when power plants react without incorporating their schedules.

For this reason, we enabled the power plants to use all escalation stages presented in Section IV-C4, including those that incorporate schedule-based decisions (see *AllEscStages*). Because the power plants' sensitivity is increased step by step in this setting, they now need 3.86 iterations per compensation on average. On the other hand, we notice that now only 21.91% of a deviation stems from the superordinate AVPP. This behavior shows that, when solving the DRAP in a reactive manner, TBST-based schedules and schedule-based decisions allow the agents to respect those objectives of the scheduling problem that can be deduced from created schedules. Moreover, the mean number of compensations per AVPP and time step drops to 2.08. Although the number of iterations is greater than in *LastEscStage*, communication is significantly decreased in *AllEscStages*. That is because, in *LastEscStage*, the number of compensations of an AVPP increases with its depth in the hierarchy due to the propagation of deviations. Further, the number of power plants increases from top to bottom. Since we ran our experiments on a single machine, our runtime measurements do not include communication overhead. As the average compensation runtime per AVPP is in the range of milliseconds, this decrease in communication would improve the performance of a real distributed system (assuming that messaging ranges in milliseconds).

When we allow power plants to interpolate their output based on different scenarios (see *GradualSensitivity*), we observe that, compared to *AllEscStages*, the average number of iterations is reduced (3.49 instead of 3.86), resulting in a shorter runtime per AVPP of 1.30 ms and max. seq. runtime of 7.47 ms. Compared to *LastEscStage*, the schedule-based decisions of *GradualSensitivity* and *AllEscStages* do not yield the same degree of fairness. However, their decisions are proactively guided, allowing them to respect the scheduling problem's objectives in reactive decisions.

In *BadPrefScenario*, we forced the AVPPs to choose the most inappropriate scenario as preferred scenario. Although the agents need slightly more iterations, they still compensate for the deviations. This points out the mechanism's resilience with respect to the selection of the preferred scenario. In all other evaluation scenarios, AVPPs switched the preferred scenario about 1.56 times between two schedule calculations on average, indicating the importance of TBST-based schedules and being able to choose the best-suited scenario at runtime.

Figure 5 shows an example of how a hydro power plant's actual and scheduled output for the anticipated scenario devel-

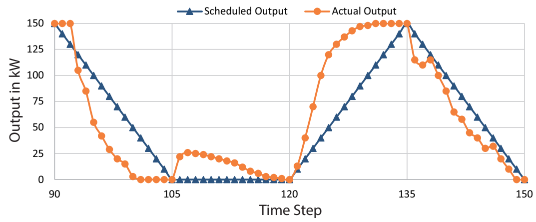

Fig. 5. The scheduled output for the anticipated scenario as well as the actual output of a hydro power plant over a time frame of 60 minutes.

ops over a time frame of 60 minutes. The scheduled output was stipulated for time steps in $\{90, 105, \ldots, 150\}$. Although uncertainties necessitate deviations between the actual and the scheduled output, the characteristic of the reactively determined actual output resembles the development of the scheduled output. This illustrates our idea of schedule compliance.

VI. Related Work

In the body of literature, there are several approaches that solve resource allocation problems similar to the DRAP presented in this paper. However, to the best of our knowledge, our approach is unique in the way we combine a proactive and reactive principle to solve the DRAP: Agents explicitly incorporate uncertainties in the form of scenarios in their proactive decisions and exploit this information when making reactive decisions afterwards. As shown in Section V, this procedure allows the agents to solve the DRAP in a very short time and to guide their reactive decisions such that they adhere to the objectives considered in the scheduling problem.

Campbell et al. investigate the role and benefit of inter-agent variation in self-organizing systems in [12] (they refer to "sensitivity" as "error"). Similar to our DRAP, their agents try to maintain a task's value at a predefined target value by making a contribution when a certain deviation is detected. Despite a changing environment, they are able to compute how much inter-agent variation is necessary so that the system reaches the target value without any communication between the agents. This is possible because 1) the task's value decreases by a constant value from one time step to another, allowing agents to anticipate their behavior, and 2) each agent contributes with the same fixed value. In our DRAP, heterogenous agents are situated in a randomly changing environment. Therefore, we cannot apply the formulas given in [12]. Instead, the agents determine sensitivities by creating schedules based on anticipated uncertainties at runtime. In the end, the agents' reactions to deviations are guided by their schedules and control models.

In the context of balancing production and consumption in the power grid, [13] presents a mechanism that is based on a great quantity of devices equipped with thermal storage capacity (e.g., refrigerators). To solve the DRAP, the control system allows temperatures that are linearly dependent on the deviation between production and consumption. This allows, e.g., to defer cooling phases. [13] assumes that the devices are in different thermal states with respect to their temperature and thus have different sensitivities when reacting to a given deviation. That way, the DRAP is solved in an exclusively reactive manner, on the basis of local knowledge, and without any communication. In Section I, we gave reasons why this is not always possible. In contrast to our approach, optimization objectives of higher system levels cannot be considered in [13].

[14], [15] present another approach to the balancing problem by grouping small generators and consumers into pools. Within each pool, agents of smaller dynamic groups immediately cancel out minor deviations by sequentially adjusting their output. In case of significant deviations that cannot be dissolved by a dynamic group, the whole pool tries to compensate for the remaining deviation by recalculating schedules in a centralized manner. In our approach, agents do not have to recalculate schedules and decide themselves about their contribution adjustments. Further, our approach improves fairness with regard to the agents' possibility to comply with their schedules, which is not the case in [14], [15].

Nieße et al. present an approach to the balancing problem in power grids in [16]. In contrast to our approach, they assume a set of different schedule combinations that would re-establish the balance as given. While they leave the calculation of these schedules for future work, their main focus is to choose a combination that causes a minimal impact on the power grid. For this purpose, they represent the power grid's structure by a weighted graph. With regard to our solution, locality is defined on the basis of a self-organizing hierarchical system structure in which agents autonomously solve the DRAP.

In [5], Hinrichs et al. present a self-organizing heuristic, called COHDA_2, that solves a combinatorial optimization problem in a decentralized manner. The algorithm is based on a neighborhood structure in which agents iteratively negotiate about their configuration in order to achieve a global as well as individual objectives. The former can be thought of as the objectives of our DRAP. In COHDA_2, every time an agent changes its configuration, it informs its neighbors about its new configuration as well as the configuration of its neighbors and the neighbors' neighbors etc. it received in a prior time step. Although each agent thus only has current information about its direct neighborhood, it builds a complete representation of the configuration of the other agents in the system over time. To decide on suitable own configurations, the agents take this locally perceived global configuration into account. As it is very likely that this configuration differs from the actual configuration of the other agents, the agents keep track of and inform their neighbors about the best configuration they have achieved so far. While the quality of COHDA_2's solutions highly depends on the topology of the overlay network defining the agents' neighborhood, with regard to our approach, agents cooperate in a self-organizing hierarchical system structure. Further, COHDA_2 does not deal with uncertainties and thus has to recalculate schedules in every time step.

In the domain of PMSs, there are various market-based approaches that try to solve a problem similar to our DRAP (see, e.g., [17], [18], [19]). PowerMatcher [17] assumes a hierarchical system structure in which the root balances supply and load by determining an equilibrium price, based on aggregated load, supply, and price predictions, to establish a market equilibrium. The auctioneer is thus a central component of the system. DEZENT [18] is a bottom-up market-based mechanism in which agents negotiate and conclude contracts within fixed price frames that are tightened from one iteration to another. PowerMatcher and DEZENT rather solve the scheduling problem for a single time step than react to deviations. Stigspace [19] is a coordination mechanism that uses a blackboard, called stigspace, as the medium of communication between distributed energy resources in order to create schedules for multiple time steps in an iterative process. Initially, the stigspace is used to announce the load that has to be fulfilled by the distributed energy resources. These in turn revise their schedules (i.e., their contribution for the regarded time frame) in order to minimize the remaining load. The new schedule is then posted to the stigspace where the remaining load is updated accordingly. This process is repeated until the load is sufficiently satisfied. In Stigspace, agents randomly and independently from each other adjust their schedules. Obviously, this might impair the algorithm's convergence in some situations. With regard to our solution, agents do not disclose their behavioral model either, but the schedule-based decisions indirectly couple the agents' decisions and thus allow them to respect optimization objectives of higher system levels.

VII. Conclusion and Future Work

In this paper, we showed that the combination of proactive and reactive resource allocation allows to effectively and efficiently solve a dynamic resource allocation problem in uncertain environments. Because we assume that the agents' behavior is subject to inertia, their future contribution has to be stipulated in the form of schedules in advance. However, due to the combination with a reactive approach, the frequency of schedule creation can be kept low. This is advantageous since the solution of the scheduling problem is NP-hard with respect to the number of agents and time steps schedules are calculated for. Since we calculate schedules for different possible developments of the demand, they give the agents a clue of how to adjust their contribution in different situations. In particular, together with the agents' control model, they act as a dynamic and situation-specific source of sensitivity to deviations between demand and contribution. As our evaluation showed, the agents' schedule-based reactive decisions respect those objectives of the scheduling problem that can be deduced from created schedules. To the best of our knowledge, this way to explicitly incorporate uncertainties in the proactive component and use the resulting structure for autonomous reactive decisions is a novelty.

In future work, we want to investigate various feedback loops we identified in the course of this work in more detail. On the one hand, feedback from the reactive to the proactive component could be used to schedule reserves that might become necessary to compensate for deviations. Further, if it is observed that existing schedules are not feasible to cope with the current situation, an on-demand recalculation of schedules could be triggered. In the end, this shift from rigid to flexible scheduling intervals could reduce computational costs. On the other hand, we want to examine the feedback loop between the reactive component and the self-organized generation of the hierarchical system structure, e.g., if a subsystem repeatedly had difficulties to locally deal with uncertainties. With regard to our case study, we will integrate a power grid topology into the self-organizing hierarchy of AVPPs and look into solutions of the DRAP that comply with network constraints in order to minimize power losses and avoid congestions.

References

[1] T. Van Zandt, "Hierarchical computation of the resource allocation problem," *European Economic Review*, vol. 39, no. 3-4, pp. 700–708, April 1995.

[2] J. Heo, K. Lee, and R. Garduno-Ramirez, "Multiobjective Control of Power Plants Using Particle Swarm Optimization Techniques," *Energy Conversion, IEEE Transactions on*, vol. 21, no. 2, pp. 552–561, 2006.

[3] A. Zafra-Cabeza, M. Ridao, I. Alvarado, and E. Camacho, "Applying Risk Management to Combined Heat and Power Plants," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 938–945, 2008.

[4] G. Anders, J.-P. Steghöfer, F. Siefert, and W. Reif, "A Trust- and Cooperation-Based Solution of a Dynamic Resource Allocation Problem," in *Proc. of the 7th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. Philadelphia, PA: IEEE Computer Society, Washington, D.C., September 2013.

[5] C. Hinrichs, M. Sonnenschein, and S. Lehnhoff, "Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces," in *International Conference on Agents and Artificial Intelligence (ICAART 2013)*, vol. Volume 1 – Agents. SciTePress, 2013, pp. 25–34.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, "A Unified Approach to Approximating Resource Allocation and Scheduling," *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1069–1090, 2001.

[7] G. Anders, C. Hinrichs, F. Siefert, P. Behrmann, W. Reif, and M. Sonnenschein, "On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty," in *Proc. of the 6th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE Computer Society, Washington, D.C., 2012, pp. 29–38.

[8] UCTE, "UCTE Operation Handbook – Policy 1: Load-Frequency Control and Performance," Union for the Co-ordination of Transmission of Electricity, Tech. Rep. UCTE OH P1, 2009.

[9] J.-P. Steghöfer, G. Anders, F. Siefert, and W. Reif, "A System of Systems Approach to the Evolutionary Transformation of Power Management Systems," in *Proc. of INFORMATIK 2013 – Workshop on "Smart Grids"*, ser. Lecture Notes in Informatics, vol. P-220. Bonner Köllen Verlag, 2013.

[10] G. Anders, A. Schiendorfer, J.-P. Steghöfer, and W. Reif, "Robust Scheduling in a Self-Organizing Hierarchy of Autonomous Virtual Power Plants," in *Proc. of the 2nd International Workshop on "Self-optimisation in Organic and Autonomic Computing Systems" (SAOS14) in conjunction with ARCS 2014*, vol. 2, February 2014.

[11] G. Anders, F. Siefert, J.-P. Steghöfer, and W. Reif, "Trust-Based Scenarios – Predicting Future Agent Behavior in Open Self-Organizing Systems," in *Proc. of the 7th International Workshop on Self-Organizing Systems (IWSOS 2013)*, May 2013.

[12] A. Campbell, C. Riggs, and A. Wu, "On the Impact of Variation on Self-Organizing Systems," in *Proc. of the 5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, Oct. 2011, pp. 119 –128.

[13] J. Short, D. Infield, and L. Freris, "Stabilization of Grid Frequency Through Dynamic Demand Control," *IEEE Trans. on Power Systems*, vol. 22, no. 3, pp. 1284 –1293, 2007.

[14] A. Kamper and A. Eßer, "Strategies for Decentralised Balancing Power," in *Biologically-inspired Optimisation Methods – Parallel Algorithms, Systems and Applications*, ser. Studies in Computational Intelligence, A. Lewis, S. Mostaghim, and M. Randall, Eds. Springer, 2009, vol. 210, pp. 261–289.

[15] A. Kamper and H. Schmeck, "Adaptives verteiltes Lastmanagement in Bilanzkreisen," *Informatik-Spektrum*, vol. 35, pp. 102–111, 2012.

[16] A. Nieße and M. Sonnenschein, "Using Grid Related Cluster Schedule Resemblance for Energy Rescheduling," *Proc. of the 2nd International Conference on Smart Grids and Green IT Systems*, May 2013.

[17] J. K. Kok, C. J. Warmer, and I. G. Kamphuis, "PowerMatcher: Multiagent Control in the Electricity Infrastructure," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '05. New York, NY, USA: ACM, 2005, pp. 75–82.

[18] H. Wedde, "DEZENT – A Cyber-Physical Approach for Providing Affordable Regenerative Electric Energy in the Near Future," in *2012 38th EUROMICRO Conf. on Software Engineering and Advanced Applications (SEAA)*, 2012, pp. 241–249.

[19] J. Li, G. Poulton, and G. James, "Coordination of Distributed Energy Resource Agents," *Applied Artificial Intelligence*, vol. 24, no. 5, pp. 351–380, 2010.