Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1517–1525
DOI: 10.15439/2016F76
ACSIS, Vol. 8. ISSN 2300-5963

Local Soft Constraints in Distributed Energy Scheduling

Astrid Nieße, Michael Sonnenschein
R&D Division Energy
OFFIS – Institute for Information Technology
26121 Oldenburg, Germany
{niesse, sonnenschein}@offis.de

Christian Hinrichs, Jörg Bremer
Environmental Informatics
University of Oldenburg
26129 Oldenburg, Germany
{christian.hinrichs, joerg.bremer}@uni-oldenburg.de

***Abstract*—In this contribution we present an approach on how to include local soft constraints in the fully distributed algorithm COHDA for the task of energy units scheduling in virtual power plants (VPP). We show how a flexibility representation based on surrogate models is extended and trained using soft constraints like avoiding frequent cold starts of combined heat and power plants. During the task of energy scheduling, the agents representing these machines include indicators in their choice for a new operation schedule. Using an example VPP we show that our approach enables the agents to reflect local soft constraints without sacrificing the global result quality.**

I. Introduction

In DECENTRALIZED energy systems, small combined heat and power (CHP) plants, electrical storages and renewable energy units are aggregated both for an integration of these energy units into the energy markets and for the provision of ancillary services for a stable grid operation. Both applications are widely known as virtual power plants (VPP) and expected to be one of the core concepts for distributed energy systems [1]. One of the core challenges during operation of such a VPP arises from the complexity of the scheduling task due to the large amount of (small) energy units in the distribution grid [2]. To this end, multiple scalable scheduling algorithms have been proposed for distribution grid energy unit scheduling for VPP, with many of them using software agents technology and distributed algorithms [3], [4]. During scheduling, both global constraints (i. e. concerning the VPP as a whole) and local constraints (i. e. restricted to a single energy unit) have to be handled in an appropriate way. Both types of constraints may be either hard or soft constraints (cf. Table I). Local hard constraints set defined limits to the operational flexibility of an energy unit, thus defining feasible operation schedules. For the example of a CHP installation including thermal storage, the thermal capacity of the storage sets a hard constraint to the CHP's operation in combination with the current thermal load. Local soft constraints comprise technical or economical preferences, e. g. preferred operation times or the avoidance of technically unfavorable frequent cold starts. Global hard constraints can be market driven, like obligations from existing contracts. The reflection of global soft constraints is out of scope of the work presented here.

This work was partly supported by the Lower Saxony Ministry of Science and Culture through the "Niedersächsisches Vorab" grant program (grant ZN 2764 'Smart Nord'

TABLE I
Constraint types and examples for a VPP comprising CHP including thermal storage.
HC: hard constraints. SC: soft constraints.

	local: unit level	**global: VPP level**
HC	operational limits of thermal storage	energy amount contracted at the market
SC	avoidance of cold starts	*(out of scope of this work)*

VPP scheduling is the task of identifying distinct operation schedules for all components of a VPP that are within the flexibility range of the respective components. Thus, the process can be split in two parts: First, the flexibility of the components has to be assessed and modelled, and then a scheduling solution is identified within an optimization process. The overall process thus constitutes a multi-objective optimization problem, with the different types of constraints either being reflected during modelling or during optimization.

In this contribution, we extend a scalable VPP scheduling approach from previous work that takes into account local and global hard constraints with the ability to additionally reflect local soft constraints of individual energy units, thus solving the given multi-objective optimization problem. Individual preferences (i. e. preferences for single energy units and their operation) do not have to be disclosed within the VPP during the scheduling process.

The rest of this contribution is structured as follows: In section II we present relevant approaches for the tasks of modelling and optimization in distributed energy scheduling and identify basic algorithms used for the work presented here. We then elaborate on the chosen approach to model feasible operation schedules using a support vector data description (SVDD) based method and extend this model to include local soft constraints (section III). In section IV the COHDA heuristic is presented. In former work, this distributed algorithm has been used to generate VPP schedules satisfying local and global hard constraints [5]. We show how the modelled local soft constraints can be included in the optimization process using an apriori-multi-objective optimization approach. Results from a case study evaluating the presented general multi-objective optimization approach are presented in section IV. We summarize and discuss open issues in section V.

II. Choosing the basic algorithms

In recent years, a large body of research has emerged in the field of distributed energy scheduling. For the tasks of constraint modelling and optimized scheduling relevant for the contribution at hand, in the following some basic approaches are presented, along with a discussion on the chosen algorithms.

Flexibility modelling can be understood as the task of modelling constraints. Apart from global VPP constraints, constraints often appear within single energy components; affecting the local decision making. Since these constraints are not of a distributed nature, they can be solved locally using central approaches. A widely used approach is the introduction of a penalty into the objective function that devalues a solution that violates some constraint [6]. In this way, the problem is transferred into an unconstrained one by treating fulfillment of constraints as additional objective. Alternatively, some combinatorial optimization problems allow for an easy repair of infeasible solutions. In this case, it has been shown that repairing infeasible solutions often outperforms other approaches [7]. Another popular method treats constraints or aggregations of constraints as separate objectives, also leading to a transformation into a (unconstrained) multi-objective problem [8]. A hierarchical approach that combines both hard and soft constraints in an explicit model formulation and weighted objective functions has been introduced in [9]. For optimization approaches in smart grid scenarios however, black-box models capable of abstracting from the intrinsic model have proven useful [10], [11]. They do not need to be known at compile time. A powerful, yet flexible way of constraint-handling is the use of a decoder that gives a search algorithm hints on where to look for schedules satisfying local hard constraints (*feasible schedules*) [11], [12]. This approach has been chosen in the work presented here. In section III-A an introduction to the decoder approach is given.

The chosen flexibility representation is the foundation for scheduling algorithms. The work presented by Akkermans, Ygge and Gustavsson in 1996 has been one of the first applications of distributed agent-based control in the electrical energy system [13]. The so-called HomeBots approach was motivated by an expected need for scalability, flexibility, adaptivity and broad applicability for future distributed energy systems [14]. Since this work, many distributed agent-based approaches have been developed in the disciplines of electrical engineering, control and system theory, information technology and information systems. The understanding of what constitutes a distributed system differs a lot, from software agents as gateways to the energy units and hierarchical systems [15] up to fully distributed algorithms [16]. Prior to the work presented here, a requirement based analysis has been done to identify appropriate algorithms for the task of distributed

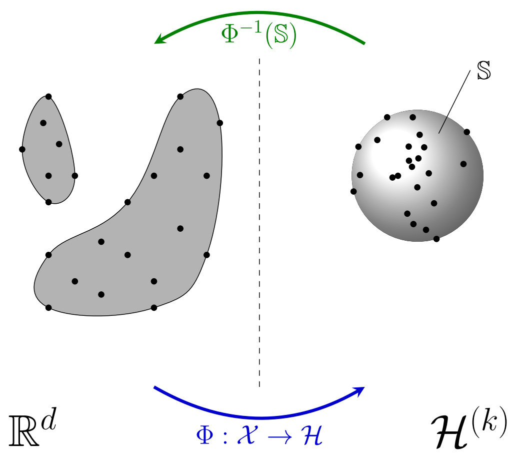

Fig. 1. General support vector model scheme for individual search spaces.

energy scheduling [17]. In the following a short overview on those algorithms already evaluated for energy scheduling tasks is presented.

The Holonic Virtual Power Plant (Hol. VPP) presented in [18] was introduced for the reactive rescheduling process in VPPs. In this concept, a dedicated agent performs the task of evaluating a VPP schedule regarding global constraints. The agents presented in the concept are not capable of evaluating the quality of a new VPP schedule but decide on their contributions based on local constraints. The same approach has been chosen for Autonomous Virtual Power Plants [19]. ALMA [20] is a fully distributed and highly dynamic approach implemented for a dynamic supply-demand-matching task. The scheduling task at hand is not solved using a set of feasible schedules but based on a different modeling approach: The energy units communicate comfort levels, thus allowing to operate them flexibly within the defined levels. With COHDA, a fully distributed heuristic has been presented for energy scheduling [16]. The operational limits are modeled using the concept of a set of feasible schedules. Although possible, soft constraints have not yet been integrated in the process.

As COHDA satisfies the requirements regarding our motivating use case, the approach has been chosen as basic algorithm in the work presented here. For the full analysis, cf. [17]. In section IV-A we will introduce COHDA before discussing the concept of soft constraint integration.

III. Flexibility and Local Constraint Modelling

A. Decoder for Local Constraint Handling

In this section we briefly recap the technique for local hard constraint handling used in this work. For handling individual local constraints from different types of energy units, we use a decoder technique. An in-depth discussion of the technique can for example be found in [11], [12].

In general, a decoder is a technique that gives algorithms hints on where to look for feasible solutions and thus allows for a targeted search. It imposes a relationship between a decoder solution and a feasible solution and gives instructions on how to construct a feasible solution [12]. A simple version without a need for machine learning techniques to deduce a meta-model, a response surface or similar uses directly a given set $\mathcal{X}$ of feasible schedules derived from a simulation model [16]. This approach has the limitation of supporting only discrete combinatorial problems. In [21] a homomorphous mapping between an n-dimensional hyper cube and the feasible region has been proposed in order to transform the problem into a topological equivalent one that is easier to handle. This approach has the problem of introducing additional parameters that have to be tuned and adapted to the problem instance at hand. In order to be able to derive a decoder automatically from any given energy unit model, [22] developed an approach based on a support vector model [23].

Fig. 2 shows the idea of using a so called support vector decoder. The basic idea is to start with a set of feasible example schedules derived from a simulation model of the respective energy unit and use this sample as a stencil for the region (the sub-space in the space of all schedules) that contains just feasible schedules.

We regard a schedule of an energy unit as a vector $\boldsymbol{s} = (s_0, \ldots, s_d) \in \mathcal{S} \subset \mathbb{R}^d$ with each element s_i denoting mean power generated (or consumed) during the ith time interval. As has been shown in [24], it is advantageous from a machine learning point of view to use scaled schedules for learning the feasible region. Thus, we construct the training set $\mathcal{X}$ by a normalization with

$$\begin{aligned} &\mathcal{N} : \mathcal{S} \to \mathcal{X} \subset [0,1]^d \\ &\boldsymbol{s} \mapsto \boldsymbol{x} = \mathcal{N}(\boldsymbol{s}), \text{with } x_i = \frac{s_i - p_{min}}{p_{max} - p_{min}}; \end{aligned} \tag{1}$$

p_{min} and p_{max} denoting minimum and maximum power respectively. The scaled sample $\mathcal{X}$ is then used as a training set for a support vector data description (SVDD) approach [25] that derives a geometrical description of the sub-space that contains the given data (in our case: the set of feasible schedules). Given a set of data samples, the inherent structure of the scope of action of the respective energy unit can be derived as follows: After mapping the data to a high dimensional feature space by means of an appropriate kernel, the smallest enclosing ball in this feature space is determined. When mapping back this ball to data space, it forms a set of contours enclosing the given data sample.

This task is achieved by determining a mapping

$$\Phi : \mathcal{X} \subset \mathbb{R}^d \to \mathcal{H};\ x \mapsto \Phi(x) \tag{2}$$

such that all data from a sample $\mathcal{X}$ from the feasible region $\mathcal{F}$ is mapped to a minimal hypersphere in some high-dimensional space $\mathcal{H}$. The minimal sphere with radius R and center a in $\mathcal{H}$ that encloses $\{\Phi(\boldsymbol{x}_i)\}_N$ can be derived from minimizing $\|\Phi(\boldsymbol{x}_i) - a\|^2 \leq R^2 + \xi_i$ with $\|\cdot\|$ as the Euclidean norm and slack variables $\xi_i \geq 0$ for soft constraints (here for getting a smoother ball).

After introducing Lagrangian multipliers and further relaxing to the Wolfe dual form, the well-known Mercer's theorem (cf. e.g. [26]) may be used for calculating dot products in $\mathcal{H}$ by means of a kernel in data space: $\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In order to gain a more smooth adaption, it is known to be advantageous to use a Gaussian kernel: $k_G(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}$ [27]. Putting it all together, the equation that has to be maximized in order to determine the desired sphere is:

$$W(\beta) = \sum_i k(\boldsymbol{x}_i, \boldsymbol{x}_i)\beta_i - \sum_{i,j} \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{3}$$

With $k = k_G$ one gets two main outcomes from the training procedure: the center $a = \sum_i \beta_i \Phi(\boldsymbol{x}_i)$ of the sphere in terms of an expansion into $\mathcal{H}$ and a function $R : \mathbb{R}^d \to \mathbb{R}$ that allows to determine the distance of the image of an arbitrary point from $a \in \mathcal{H}$, calculated in $\mathbb{R}^d$ by:

$$R^2(\boldsymbol{x}) = 1 - 2\sum_i \beta_i k_G(\boldsymbol{x}_i, \boldsymbol{x}) + \sum_{i,j} \beta_i \beta_j k_G(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4}$$

Because all support vectors show the characteristics of being mapped onto the surface of the sphere, the sphere radius $R_{\mathcal{S}}$ can be easily determined by the distance of an arbitrary support vector to the center a. Thus the feasible region can now be modeled as $\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^d | R(\boldsymbol{x}) \leq R_{\mathcal{S}}\} \approx \mathcal{X}$.

The comparably small set of support vectors together with a reduced version of vector $\boldsymbol{\beta}$ that contains non zero weight values (denoted $\boldsymbol{w}$) for the support vectors is sufficient for building the model. The model might then be used as a black-box that abstracts from any explicitly given form of constraints and allows for an easy and efficient decision on whether a given solution is feasible or not. In this way, the model allows for an easy check whether a given schedule is operable or not by using decision function (4).

So far, this surrogate model is just capable of checking feasibility when already given a schedule. In this way, the surrogate may tell feasible and infeasible schedules apart on behalf of the specific simulation model of the energy unit and thus already allows for an abstraction from any model specific implementation. On the other hand, it is not yet a sufficient constraint-handling technique as it still needs externally (e. g. by any optimization algorithm) generated schedules which can merely be checked. But, due to the tiny share of the search space that is actually feasible, it is quite unlikely that a feasible schedule is generated by an algorithm just by chance [28].

Hence, a way is needed to guide an algorithm where to look for feasible schedules. To achieve such systematic search for a good and still feasible solution, a decoder can be derived automatically from the support vector surrogate. The set of feasible schedules is represented as pre-image of a high-dimensional ball $\mathbb{S}$. Fig. 1 shows the geometric situation. This representation has some advantageous properties. Although the pre-image might be some arbitrary shaped non-continuous blob in $\mathbb{R}^d$, the high-dimensional representation is still a ball and thus geometrically easier to handle.

The relation is as follows: If a schedule is feasible, i.e. can be operated by the unit without violating any technical constraint, it lies inside the feasible region (grey area on the left hand side in Fig. 2). Thus, the schedule is inside the pre-image (that represents the feasible region) of the ball and thus

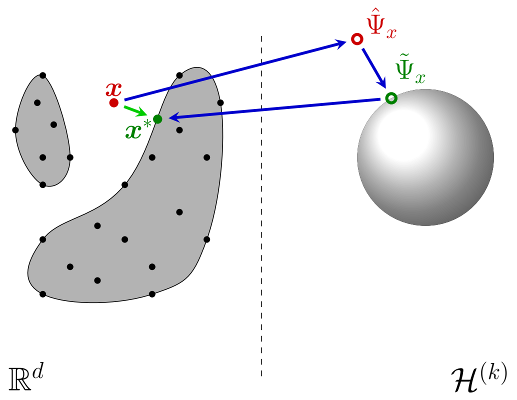

Fig. 2. General support vector decoder scheme for solution repair and constraint handling.

its image in the high-dimensional representation lies inside the ball. An infeasible schedule (e. g. $\boldsymbol{x}$ in Fig. 2) lies outside the feasible region and thus its image $\hat{\Psi}_x$ (generated by the empirical kernel mapping $\hat{\Psi}_x$) lies outside the ball. But we know some relations: the center of the ball, the distance of the image from the center and the radius of the ball. Hence, we can move the image of an infeasible schedule along the difference vector towards the center until it touches the ball. Finally, we calculate the pre-image of the moved image $\tilde{\Psi}_x$ (mover by translation function Γ_a) and get a schedule at the boundary of the feasible region: a repaired schedule $\boldsymbol{x}^*$ that is now feasible. We do not need a mathematical description of the original feasible region or of the constraints to do this. More sophisticated variants of transformation are e. g. given in [22].

Formally, we want to derive a mapping function (the so called decoder Θ)

$$\begin{aligned} \Theta : [0,1]^d &\to \mathcal{F}_{[0,1]} \subseteq [0,1]^d \\ x &\mapsto \Theta(x) \end{aligned} \tag{5}$$

that transforms any given (maybe in-feasible) schedule into a feasible one. This decoder mapping Θ is derived automatically from the trained SVDD repesentation of the search space using three steps:

$$\begin{array}{ccc} \boldsymbol{x} \in [0,1]^d & \xrightarrow{\hat{\Phi}_\ell} & \hat{\Psi}_x \in \mathcal{H}^{(\ell)} \\ \Theta \downarrow & & \downarrow \Gamma_a \\ \boldsymbol{x}^* \in \mathcal{F}_{[0,1]} \subseteq [0,1]^d & \xleftarrow[\Phi_\ell^{\sim 1}]{} & \tilde{\Psi}_x \in \mathcal{H}^{(\ell)} \end{array} \tag{6}$$

Applying such decoder to some internal solution representation $\boldsymbol{x} \in [0,1]^d$ transforms the solution to some feasible solution $\Theta(\boldsymbol{x}) \in \mathcal{F}_{[0,1]}$.

Thus, we are able to transform any global scheduling problem into a formulation that is unconstrained regarding local hard constraints. Apart from finding a combination of schedules whose sum resembles a given target power profile best, further objectives are usually integrated due to the many-objective nature of energy scheduling.

This procedure of training a decoder has to be done only once prior to the scheduling process. During scheduling the all decoders are merely used for generating feasible schedules. For our experiments, training was usually done within milliseconds. For an in-depth discussion of computational issues of the decoder we refer to [29].

For distributed problem solving, the decoder can serve as a substitute for an often (particularly with regard to a fully automated generation) hardly derivable mathematical model of feasibility. Like in well studied industrial approaches for model predictive control [30] only a simulation model as a source for learning the model is needed. Due to the abstraction from the underlying simulation model (or real unit), no information on which operations are possible and no information on limiting restrictions, cost considerations or soft constraints are needed at runtime. Thus, the ad-hoc integration of arbitrary (even of at compile-time unknown units) becomes easily possible and hence eases the implementation of many control algorithms for the smart grid.

B. Modeling local soft constraints

In general, the satisfaction of a local soft constraint by a defined operation schedule s is modeled as a DER specific function $\mathcal{I}_i$ that assigns a value between 0 and 1 to the respective operation schedule, with i denoting the respective DER within the set of DER regarded in the scheduling task, i. e. the current VPP setup.

$$\begin{aligned} \mathcal{I}_i : S &\to [0,1] \\ \mathcal{I}_i(s) &= \sigma, \forall s \in S \end{aligned} \tag{7}$$

In the following, we omit the subscript i for reasons of brevity. As the soft constraint evaluation is a component-specific task it is performed locally in all upcoming definitions.

We now define an extended search space S' that integrates the soft constraints into the search space S:

$$S' : \{ (\ s,\ \sigma^{(1)}, ..., \sigma^{(m)}\ ) \mid s\ \in\ S \} \tag{8}$$

With this definition, the extended search space S' is the set of all tuples of feasible operation schedules s and their respective soft constraint values, with m being the number of modelled soft constraints.

But how to model this extended search space? For the decoder concept presented in Section III-A, a possible integration of indicators has been shown: Data vectors containing the mean power levels for the respective time intervals are extended by one element per indicator to mixed feature vectors. This approach has been proposed for environmental performance indicators [23], but in general, arbitrary indicators can be added as long as a functional relationship exists between the power part and the indicator. In this way, we can build a modified sample $\boldsymbol{x}'$ as

$$\boldsymbol{x}' = (x_1, \dots, x_d, \mathcal{I}^{(1)}_{[0,1]}(\boldsymbol{x}), \dots, \mathcal{I}^{(m)}_{[0,1]}(\boldsymbol{x})),\ \boldsymbol{x}' \in [0,1]^{d+m}, \tag{9}$$

with the first d elements denoting real power and m trailing elements denoting indicator values. Whereas $\mathcal{I}$ takes a schedule, $\mathcal{I}_{[0,1]}$ maps an already scaled training vector $\boldsymbol{x}$ instead. This sample is fed into exactly the same support vector training process to build the model. The decoder is derived in exactly the same way. The decoder mapping Θ then likewise maps feature vectors $\boldsymbol{x}' \in \mathcal{X}'$

$$\begin{aligned} [0,1]^{d+m} &\rightarrow \mathcal{F}_{[0,1]} \times [\mathcal{I}_{[0,1]}]^m \\ \Theta(\boldsymbol{x}') &\mapsto (\boldsymbol{x}, \mathcal{I}^{(1)}_{[0,1]}(\boldsymbol{x}), \ldots, \mathcal{I}^{(m)}_{[0,1]}(\boldsymbol{x})). \end{aligned} \tag{10}$$

If $\boldsymbol{x}'$ is given with arbitrary values then $\Theta(\boldsymbol{x}')$ contains a feasible active power schedule in the first d elements as well as m elements evaluating this schedule correctly (slight inaccuracies are possible) with regard to the secondary optimization objectives.

Using this concept of the decoder approach including indicator reflection, we can set up an extended search space from a set of samples (i. e. normalized schedules) as shown in equation 9. For this purpose, we have to add the indicator value $\mathcal{I}$ for the chosen soft constraint to the sample schedule prior to the support vector training phase. In the example chosen here, we want to reduce the amount of cold starts within an operation schedule to minimize motor deterioration. We can infer the number of cold starts directly from the schedules within a preprocessing step. We omit the precise definition of cold starts for reasons of brevity, but usually it is a defined change of switching an engine off and on within a given time span. The indicator value thus matches the soft constraint value as given in equation 7. We now have to define the DER specific soft constraint function as an indicator for the amount of cold starts $\mathcal{I}^{cs}$ precisely to feed it into the SVDD training:

$$\mathcal{I}^{cs}(\mathsf{s}) = \left(1 - \frac{cs_{\mathsf{s}}}{cs_{max}}\right)^2 \tag{11}$$

with cs_{s} as the amount of cold starts in the given schedule s and cs_{max} as maximum amount of cold starts in the given schedule set. Using the squared value, a rising amount of cold starts is punished disproportionally high. For schedule sets without cold starts, the value is undefined – these cannot be used to distinguish schedules using this characteristic.[1]

It can be seen that there is a functional relationship between the schedule and the indicator, thus allowing to use the modified sample definition as given in equation 9 and using equation 1 to map schedule s to its normalized sample x. For each sample x in the given sample set we append the indicator as defined in equation 11 and use it for SVDD training.

As a result of these steps, we yield the extended search space S' and can now integrate this in the scheduling process.

IV. Distributed Energy Scheduling

A. Introducing COHDA

The Combinatorial Optimization Heuristic for Distributed Agents (COHDA, originally introduced in [16]) can be used to solve scheduling problems in VPPs. In the present contribution, we consider predictive scheduling: the goal is to select a schedule for each energy unit—from a given search space of feasible schedules with respect to a future planning horizon—such that a global objective function (e. g. a target power profile for the VPP) is optimized. This target profile is understood as global hard constraint within the scheduling process for the rest of this contribution. We will recap the approach briefly, based on the description in [5].

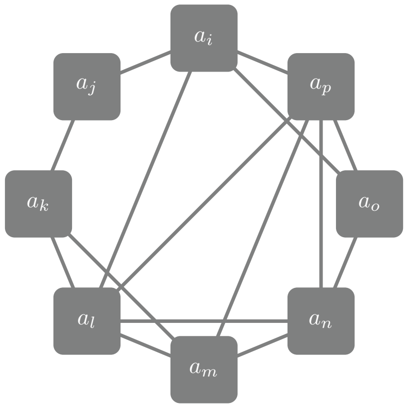

Fig. 3. Exemplary communication topology in the form of a small world topology for a system comprising eight agents.

The key concept of COHDA is an asynchronous iterative approximate best-response behavior, where each agent—representing a decentralized energy unit—reacts to updated information from other agents by adapting its own selected schedule with respect to the global objective. All agents $a_i \in \mathsf{A}$ initially only know their own respective set of schedules S_i, so from an algorithmic point of view, the difficulty of the problem is given by the distributed nature of the system in contrast to the task of finding a common allocation of schedules for a global target power profile.

Thus, the agents coordinate by updating and exchanging information about each other. But, in order to preserve privacy, the amount of information that is exchanged is restricted. In particular, the set of feasible schedules S_i is not communicated as a whole by an agent a_i. Instead, the agents try to publish as little information as possible. How these possibly conflicting goals are handled, and how the system is able to converge to sound and satisfying solutions, will be explained in the following.

First of all, the agents are placed in an artificial communication topology (e. g. a *small world* topology, see Fig. 3), such that each agent is connected to a non-empty subset of other agents. To compensate for the resulting non-global view on the system, each agent a_i collects two distinct sets of information: on the one hand the believed current configuration γ_i of the system (that is, the most up to date information a_i has about currently selected schedules of all agents), and on the other hand the best known combination γ_i^* of schedules with respect to the global objective function it has encountered so far.

Beginning with an arbitrarily chosen agent by passing it a

[1]Please note that in the implementation presented here, $\mathcal{I}$ is defined identically for all DER, without limiting the applicability of the presented approach to DER specific soft constraint functions $\mathcal{I}_i$.

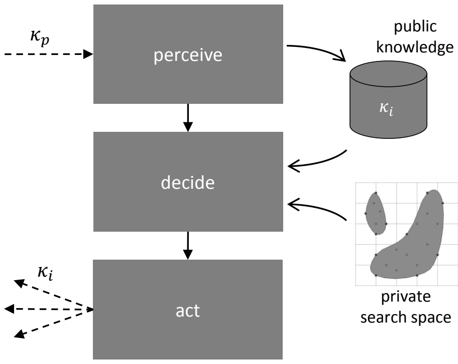

Fig. 4. The *perceive–decide–act* behavioral pattern in COHDA from the point of view of an agent a_i.

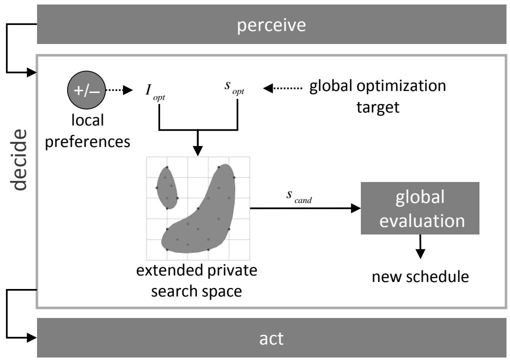

Fig. 5. The *decide* behavioral pattern of COHDA including the reflection of soft constraints.

message containing only the global objective (i. e. the target power profile), each agent repeatedly executes the three steps *perceive*, *decide*, *act* (cf. [5]) as visualized in Fig. 4:

1) **`perceive:`** When an agent a_i receives a message κ_p from one of its neighbors (say, a_p), it imports the contents of this message into its own memory.
2) **`decide:`** The agent then searches S_i for the best schedule regarding the updated system state γ_i and the global objective function, thus respecting the global hard constraints. The reflection of local hard and soft constraints depends on the chosen approach to model the energy unit's flexibility and will be discussed in a later part of this contribution. If a schedule can be found that satisfies both the global and the local objectives, a new schedule selection is created. For the following comparison, only the global objective function must be taken into account: If the resulting modified system state γ_i yields a better rating than the current solution candidate γ_i^*, a new solution candidate is created based on γ_i. Otherwise the old solution candidate still reflects the best schedule combination regarding the global objective the agent is aware of, so the just created schedule selection is discarded and the agent reverts to its schedule selection stored in γ_i^*.
3) **`act:`** If γ_i or γ_i^* has been modified in one of the previous steps, the agent finally broadcasts these to its neighbors in the communication topology.

Following this behavior, only small subsets of the sets of feasible schedules S_i are communicated by the agents. During this process, for each agent a_i, its observed system configuration γ_i as well as solution candidate γ_i^* are empty at the beginning, will be filled successively with the ongoing message exchange and will some time later represent valid solutions for the given optimization problem. After producing some intermediate solutions, the heuristic eventually terminates in a state where for all agents γ_i as well as γ_i^* are identical, and no more messages are produced by the agents. At this point, γ^* (which is the same for all agents then, so the index can be dropped) is the final solution of the heuristic and contains exactly one schedule selection for each agent.

B. Reflecting soft constraints during scheduling

In Fig. 4 an overview on the heuristic COHDA has been given from the perspective of a single agent representing a DER within a VPP. The algorithm is designed originally to optimize for a global objective only: In a two-stepped procedure, first a new schedule is chosen from the agent's a_i local search space S_i. After that, the resulting global result quality is evaluated. However, each agent must be permitted to decide itself which schedule it contributes. This way, technically, economically or ecologically rooted local soft constraints can be taken into account as secondary optimization goals. Moreover, in order to preserve privacy and autonomy of the participating entities, these individual secondary objectives must be treated as private to the corresponding agent, i. e. similar to the set of feasible schedules S_i, the local objectives are not part of the communicated information.

To integrate such local soft constraints into the decision process without compromising the convergence of the distributed algorithm, we modified the first step by replacing the search space S by the extended search space S' as defined in equation 8. During the *decide* phase, the agent thus has surplus information regarding the indicator (see Fig. 5). With this modelling approach, an additional constraint is integrated in the search space. Using the decoder approach as presented in section III-A, the agent can now try to identify a candidate schedule s that enhances the performance regarding the global objective (e. g. energy amount) and additionally enhance the performance regarding local quality as defined by the local soft constraint function $\mathcal{I}_i(s)$ (cf. equation 7). To enable this multi-objective optimization, the *decide* phase of COHDA is extended (cf. Fig. 5): In the first step, the needed schedule s_{opt} to best reach the global optimization target is calculated. The optimal indicator value $\mathcal{I}_{opt}$ (i. e. the best possible soft constraint performance) is added. For multiple soft constraints, additional indicator values would be added. In the next step this combination of needed schedule and indicator(s) is passed to the decoder for a targeted search. While without soft constraint modelling only the schedule best fitting the global target would be identified, the decoder now returns a schedule depending on both the needed schedule s_{opt} and the optimal local soft constraint performance $\mathcal{I}_{opt}$. Using the decoder concept with an extended search space modelling approach thus leads to an apriori-multi-objective optimization with an implicit weighting of the different constraints. The schedule identified by the decoder is passed as candidate schedule s_{cand} to the evaluation of the global result quality, i. e. the global hard constraint.

In the example chosen here, the indicator gives information regarding the amount of cold starts contained in the schedule. The extended search space S' therefore is built using the indicator $\mathcal{I}^{cs}$ as defined in equation 11. The agent now can choose a candidate schedule that might enhance the performance regarding the global objective and minimize the number of cold starts simultaneously.

In summary, the outlined modelling and decision process yields a reasonable hierarchy of constraint handling in the domain of distributed energy scheduling (cf. Table I in section I): Using the decoder concept as depicted in section III-A, local hard constraints are modelled in such a way that only feasible schedules are returned by the decoder. Local soft constraints are used for SVDD training (see section III-B and guide the decoder targeted search (see section IV-B. Therefore, schedules satisfying the soft constraints are returned preferentially by the decoder. In the last step, the global evaluation is performed, reflecting global hard constraints. With this concept, local hard constraints are prioritized over global objectives, while local soft constraints are being taken into account with least importance.

V. Results

To evaluate the presented approach for the integration of soft constraints in distributed energy scheduling, two hypotheses have been chosen:

H1 The integration of local soft constraints in the distributed scheduling enhances the performance of the chosen schedules regarding the modeled soft constraint, i. e. the local quality.

H2 The integration of local soft constraints does not reduce the quality of the chosen solution regarding the global objective, i. e. producing the target power profile.

In the following, we will first introduce the evaluation setup, then discuss the evaluation results using these hypothesis. In all experiments, regarding the local soft constraints, we go with the example of reducing the amount of cold starts within an operation schedule to minimize motor deterioration, as introduced in section III-B.

A. Evaluation Setup

To evaluate the effect of an integration of local soft constraints in the distributed scheduling process, a setup is needed where agents have the choice to either reflect or ignore the performance indicator regarding the amount of cold starts $\mathcal{I}^{cs}(s)$. As the global objective is to fulfill a defined energy profile, enough flexibility within the aggregation of agents is needed to fulfill this objective either using schedules with high or low performance values regarding cold starts. In the experimental setup we therefore choose a mixed set of agents representing small CHP plants ($4.7\ \mathrm{kW_{el}}$): For 15 CHP plants, the conventional search space is used, without adding the indicator value. For additional 15 CHP plants, the search space is extended, thus allowing these agents to reflect the indicator during scheduling. For the support vector training phase, we need a set of schedules that can be distinguished regarding cold starts: On a winter day, CHP plants are expected to run nearly the whole day. Therefore, schedules of a CHP for a winter day are not suitable for the evaluation task. The opposite holds for a summer day. We chose a spring day and generated schedules from a CHP simulation for this task. The largest number of cold starts (cs_{max}) within this set has been 9. Thus, a schedule with 9 cold starts within one day might have been chosen without reflecting this constraint for each CHP.

For this aggregation of 30 CHP plants, different target profiles have been defined manually in such a way that fulfilling the profile is possible with more than 90 % accuracy, thus covering a range of possible target profiles. Each target setting has been simulated 100 times with different random seeds for generating the training set for reasons of statistical soundness.

B. Local Quality (Hypothesis H1)

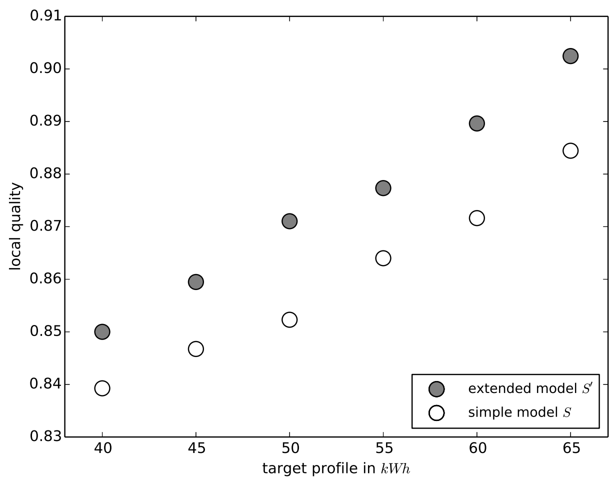

Fig. 6. Simulation results for a VPP with 30 CHPs and different target profiles. Mean values from 100 simulation runs are shown.

In Fig. 6 the results for the simulative experiments are shown regarding local quality: The horizontal axis denotes the different target profiles. On the vertical axis the mean local quality is shown. With only one soft constraint given in our scenario, we define σ as the local quality of a schedule under evaluation (cf. equation 7). The filled circles depict the mean value of 15 CHP plants over 100 simulations for the agents reflecting the amount of cold starts in their search space, whereas the unfilled circles show the mean local quality of those 15 CHP plants over 100 simulations for the agents that do not consider this local soft constraint.

It can be seen that the local quality is higher for all target profiles, if soft constraints are integrated in the scheduling process. Additionally a trend can be seen: The higher the energy amount of the target profile, the better is the local quality. This can be explained as follows: If more electrical energy has to be produced, the agents choose schedules with longer runtimes. Thus, less cold starts are expected.

For one of the simulation setups (target profile 50 kWh), the raw data are depicted in Fig. 7. It can be seen that using the extended search space model (S'), the number of schedules with 2 cold starts is reduced, whereas the number of schedules without cold starts is increased. There is a slight rise in the number of schedules with 1 and 3 cold starts. This rise is considered to be non-significant compared to the effect regarding the reduction of schedules without cold starts. In general, a shift to schedules with less cold starts can be observed. The results are similar for the other simulation setups but not displayed here.

With the given results, we consider hypothesis H1 strengthened: The integration of local soft constraints in the distributed scheduling enhances the performance regarding the modeled soft constraint.

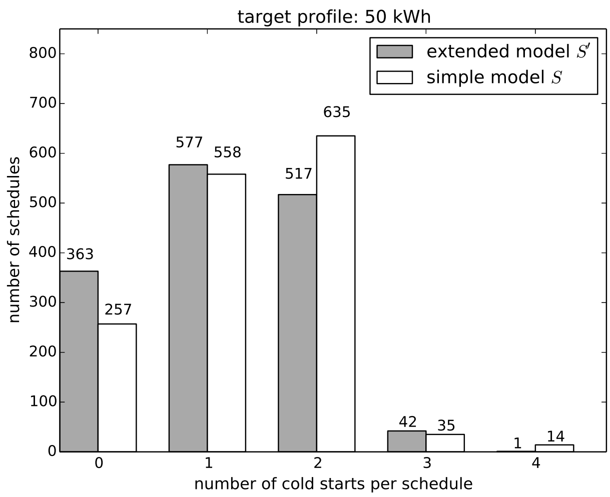

Fig. 7. Detailed simulation results for a VPP with 30 CHPs and target profile 50 kWh. Regarding 100 simulation runs, the distribution of schedules with respect to the amount of cold starts per schedule is shown.

C. Global Quality (Hypothesis H2)

We now focus on the effects of the integration of soft constraints in the scheduling process on the global quality. In Table II the normalized global quality regarding the target profile fulfillment is summarized using the mean values over 100 simulation runs for different target profiles. In general, the energy unit aggregations perform best for a profile with either 50 or 55 kWh, although a high quality could be reached for all simulation settings. This depends on the chosen aggregation of energy units and their feasible schedules: Within the defined range from 40 to 65 kWh, enough operational flexibility is given to adapt to a defined target profile. We now compare the global quality within one target profile simulation setting with and without the reflection of soft constraints: It can be seen that for all profiles, the values differ only very slightly. This effect can be understood from the chosen concept of soft constraint integration: If an agent manages to identify an operation schedule that will outperform the current target fulfillment, this would be chosen although this schedule might decrease local quality. With the chosen concept of guiding the search within the schedule search space along the soft constraint performance though, the heuristic COHDA tends to find those local optima that not only increase global quality but additionally show better local quality.

With the given results, we consider hypothesis H2 strengthened: The integration of local soft constraints does not reduce the quality of the chosen solution regarding the target profile delivery significantly for the chosen setting.

TABLE II
COMPARISON OF TARGET PROFILE FULFILLMENT WITH AND WITHOUT REFLECTION OF SOFT CONSTRAINTS. MEAN VALUES FROM 100 SIMULATIONS RUNS PER TARGET PROFILE ARE GIVEN.

40kWh	**45kWh**	**50kWh**	**55kWh**	**60kWh**	**65kWh**
with reflection of soft constraints					
0.9862	0.9886	0.9921	0.9939	0.9874	0.9590
without reflection of soft constraints					
0.9850	0.9907	0.9931	0.9937	0.9830	0.9605

VI. CONCLUSION AND OUTLOOK

In this contribution we presented an approach on how to include local soft constraints in the fully distributed algorithm COHDA for the task of energy units scheduling in virtual power plants (VPP). For this task, we extended a flexibility representation based on SVDD using indicator values and used these indicators to guide the search for a new schedule. Using the example of preventing frequent cold starts for CHP plants, we could show that the presented approach enables the agents to reflect the modeled local soft constraint without sacrificing the global result quality. As the information on local soft constraints is not communicated within the system and considered only locally, the presented approach reveals benefits regarding privacy without sacrificing global result quality.

With these results, further work should be done on the integration of extended search spaces in the presented distributed scheduling heuristic COHDA. With the extension of the search space using indicators, an implicit weighting is given by the length of the schedules and the number of indicators. Additional evaluation effort is needed to yield an appropriate weighting depending on the specific soft constraint. A straightforward approach would be to adapt the weights of an indicator by multiplying its value during the SVDD training phase, thus yielding an explicit weighting. The formalization given in the contribution at hand is compatible with this concept.

Additionally, the boundaries of effectiveness for energy unit aggregations with less flexibility should be evaluated, especially compared to other multi-objective optimization concepts: With the presented approach, soft constraints guide the search in the search space. Therefore it has to be evaluated, if for some types of DER the global quality would be reduced to an unaccepted extent. Straight-forward extensions of the presented approach like time-dependent soft constraint relaxation could overcome such problems.

REFERENCES

[1] H.-J. Appelrath, H. Kagermann, and C. Mayer, Eds., *Future Energy Grid. (acatech STUDY).* acatech, Munich, 2012.

[2] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part 1: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems*, vol. 22, pp. 1743–1752, 2007.

[3] ——, "Multi-Agent Systems for Power Engineering Applications—Part 2: Technologies, Standards, and Tools for Building Multi-Agent Systems," *IEEE Transactions on Power Systems*, vol. 22, pp. 1753–1759, 2007.

[4] P. Vrba, V. Mařík, P. Siano, P. Leitão, G. Zhabelova, V. Vyatkin, and T. Strasser, "A Review of Agent and Service-Oriented Concepts Applied to Intelligent Energy Systems," *IEEE Transactions of Industrial Informatics*, vol. 10, no. 3, pp. 1890–1903, 2014.

[5] A. Nieße, S. Beer, J. Bremer, C. Hinrichs, O. Lünsdorf, and M. Sonnenschein, "Conjoint Dynamic Aggregation and Scheduling Methods for Dynamic Virtual Power Plants," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 2. IEEE, 2014. doi: 10.15439/2014F76. ISBN 978-83-60810-58-3 pp. 1505–1514. [Online]. Available: http://dx.doi.org/10.15439/2014F76

[6] A. Smith and D. Coit, *Handbook of Evolutionary Computation*. Department of Industrial Engineering, University of Pittsburgh, USA: Oxford University Press and IOP Publishing, 1997, ch. Penalty Functions, p. Section C5.2.

[7] G. E. Liepins and M. D. Vose, "Representational issues in genetic optimization," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 2, 1990.

[8] O. Kramer, "A review of constraint-handling techniques for evolution strategies," *Appl. Comp. Intell. Soft Comput.*, vol. 2010, pp. 1–19, 01 2010. doi: http://dx.doi.org/10.1155/2010/185063

[9] A. Schiendorfer, J.-P. Steghöfer, and W. Reif, "Synthesised Constraint Models for Distributed Energy Management," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 2. IEEE, 2014. doi: 10.15439/2014F49. ISBN 978-83-60810-58-3 pp. 1529–1538. [Online]. Available: http://dx.doi.org/10.15439/2014F49

[10] F. Gieseke and O. Kramer, "Towards non-linear constraint estimation for expensive optimization," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, A. Esparcia-Alcázar, Ed. Springer Berlin Heidelberg, 2013, vol. 7835, pp. 459–468. ISBN 978-3-642-37191-2

[11] J. Bremer and M. Sonnenschein, "Model-based integration of constrained search spaces into distributed planning of active power provision." *Comput. Sci. Inf. Syst.*, vol. 10, no. 4, pp. 1823–1854, 2013.

[12] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, Jan. 2002. doi: 10.1016/S0045-7825(01)00323-1

[13] H. Akkermans, F. Ygge, and R. Gustavsson, "Homebots: Intelligent decentralized services for energy management," in *Fourth International Symposium on the Management of Industrial and Corporate Knowledge ISMICK'96*, Rotterdam, NL, Oktober 1996.

[14] R. Gustavsson, "Agents with Power," *Communications of the ACM*, vol. 42, no. 3, pp. 41–47, 1999.

[15] S. Lehnhoff, *Dezentrales vernetztes Energiemanagement - Ein Ansatz auf Basis eines verteilten Realzeit-Multiagentensystems.* Vieweg + Teubner, 2010.

[16] C. Hinrichs, S. Lehnhoff, and M. Sonnenschein, "A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems," in *Operations Research Proceedings 2012*. Springer, 2014. doi: 10.1007/978-3-319-00795-3_43. ISBN 978-3-319-00795-3 pp. 297–302.

[17] A. Nieße and M. Sonnenschein, "A Fully Distributed Continuous Planning Approach for Decentralized Energy Units," in *45. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Informatik, Energie und Umwelt*. Cottbus: Gesellschaft für Informatik, Köllen Druck+Verlag, 2015.

[18] M. Tröschel, *Aktive Einsatzplanung in holonischen Virtuellen Kraftwerken*. Oldenburg: OlWIR, Oldenburger Verl. für Wirtschaft, Informatik und Recht, 2010. ISBN 978-3-939704-55-3

[19] G. Anders, F. Siefert, J.-P. Steghöfer, H. Seebach, F. Nafz, and W. Reif, "Structuring and Controlling Distributed Power Sources by Autonomous Virtual Power Plants," in *Proceedings of the IEEE Power and Energy Student Summit (PESS)*, 2010.

[20] E. Pournaras, "Multi-level Reconfigurable Self-organization in Overlay Services," Ph.D. dissertation, TU Delft. ISBN 9789461860989 2013.

[21] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evol. Comput.*, vol. 7, pp. 19–44, 03 1999. doi: http://dx.doi.org/10.1162/evco.1999.7.1.19

[22] J. Bremer and M. Sonnenschein, "Constraint-handling for optimization with support vector surrogate models – a novel decoder approach," in *ICAART 2013 – Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, J. Filipe and A. Fred, Eds., vol. 2. Barcelona, Spain: SciTePress, 2013. doi: 10.5220/0004241100910100. ISBN 978-989-8565-38-9 pp. 91–105.

[23] J. Bremer, B. Rapp, and M. Sonnenschein, "Including environmental performance indicators into kernel based search space representations," in *Information Technologies in Environmental Engineering*, ser. Environmental Science and Engineering, P. Golinska, M. Fertsch, and J. Marx-Gómez, Eds. Springer Berlin Heidelberg, 2011, vol. 3, pp. 275–288. ISBN 978-3-642-19535-8. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19536-5_22

[24] P. Juszczak, D. Tax, and R. P. W. Duin, "Feature scaling in support vector data description," in *Proc. ASCI 2002, 8th Annual Conf. of the Advanced School for Computing and Imaging*, E. Deprettere, A. Belloum, J. Heijnsdijk, and F. van der Stappen, Eds., 2002, pp. 95–102.

[25] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004. doi: http://dx.doi.org/10.1023/B:MACH.0000008084.60811.49

[26] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10(5), pp. 1000–1017, 1999.

[27] A. Ben-Hur, H. T. Siegelmann, D. Horn, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, vol. 2, pp. 125–137, 2001.

[28] J. Bremer and M. Sonnenschein, "Sampling the search space of energy resources for self-organized, agent-based planning of active power provision," in *27th International Conference on Environmental Informatics for Environmental Protection, Sustainable Development and Risk Management, EnviroInfo 2013, Hamburg, Germany, September 2-4, 2013. Proceedings*, ser. Berichte aus der Umweltinformatik, B. Page, A. G. Fleischer, J. Göbel, and V. Wohlgemuth, Eds. Shaker, 2013. ISBN 978-3-8440-1676-5 pp. 214–222.

[29] J. Bremer, "Constraint-Handling mit Supportvektor-Dekodern in der verteilten Optimierung," Ph.D. dissertation, 2015. [Online]. Available: http://oops.uni-oldenburg.de/2336/

[30] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 1st ed., ser. Communications and Control Engineering. Springer, 2011. [Online]. Available: http://www.springer.com/978-0-85729-500-2