Unfolding Ensemble Training Sets for Improved Support Vector Decoders in Energy Management

Joerg Bremer and Sebastian Lehnhoff
Department of Computing Science, University of Oldenburg, Uhlhornsweg, Oldenburg, Germany
R&D Division Energy, OFFIS – Institute for Information Technology, Escherweg, Oldenburg, Germany

Keywords: Flexibility Modeling, Folded Distributions, Simulated Annealing, Predictive Scheduling.

Abstract: Smart grid control demands delegation of liabilities to distributed, rather small energy resources in contrast to todays large control power units. Distributed energy scheduling constitutes a complex task for optimization algorithms regarding the underlying high-dimensional, multimodal and nonlinear problem structure. Additionally, the necessity for abstraction from individual capabilities is given while integrating energy units into a general optimization model. For predictive scheduling with high penetration of renewable energy resources, agent-based approaches using classifier-based decoders for modeling individual flexibilities have shown good performance. On the other hand, such decoder-based methods are currently designed for single entities and not able to cope with ensembles of energy resources. Combining training sets randomly sampled from individually modeled energy units, results in folded distributions with unfavorable properties for training a decoder. Nevertheless, this happens to be a quite frequent use case, e. g. when a hotel, a small business, a school or similar with an ensemble of co-generation, heat pump, solar power, and controllable consumers wants to take part in decentralized predictive scheduling. We use a Simulated Annealing approach to correct the unsuitable distribution of instances in the aggregated ensemble training set prior to deriving a flexibility model. Feasibility is ensured by integrating individual flexibility models of the respective energy units as boundary penalty while the mutation drives instances from the training set through the feasible region of the energy ensemble. Applicability is demonstrated by several simulations using established models for energy unit simulation.

1 INTRODUCTION

Across Europe, especially in Germany where a financial security of guaranteed feed-in prices is given since 1991, the share of distributed energy resources (DER) is rapidly growing. Following the goal defined by the European Commission (European Parliament & Council, 2009), a concept for integration into electricity markets is needed (Abarrategui et al., 2009; Nieße et al., 2012) leading in turn to a need for grouping small energy resources due to their rather low potential and flexibility and for predictive planning. A well-known concept for aggregating DER to a jointly controllable entity is known as virtual power plant (VPP). Apart from controlling distributed electricity generation, e. g. combined heat and power (CHP), photovoltaic or wind power, controllable consumption like shiftable loads, heat pumps or air conditioning might also be included for planning active power schedules. Battery storages are discussed to complement such groups of DER.

The general optimization problem to be solved for scheduling in a VPP is known as predictive scheduling (day-ahead based on predicted conditions) as approach for the unit commitment problem (Padhy, 2004). Under given constraints, energy unit's operation modes have to be chosen for each unit such that the joint operation meets some desired load profile for a given planning horizon.

In order to choose an appropriate schedule of operation modes for each participating DER, the algorithm must know for each DER, which schedules are actually operable and which are not. Depending on the type of DER, different constraints restrict possible operations. The information about individual local feasibility of schedules has to be modeled appropriately in (distributed) optimization scenarios, in order to allow unit independent algorithm development. For this purpose, meta-models of constrained spaces of operable schedules have been shown indispensable as a means for independently modeling constraints and feasible regions of flexibility. Each energy unit has its

Bremer, J. and Lehnhoff, S.
Unfolding Ensemble Training Sets for Improved Support Vector Decoders in Energy Management.
DOI: 10.5220/0006543503220329
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 322-329
ISBN: 978-989-758-275-2

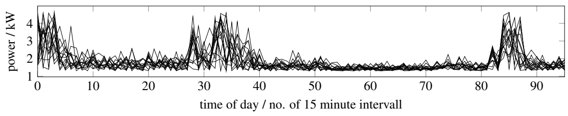


Figure 1: Example for a training set of schedules for a co-generation plant. A state-of-charge of 50% at night and an increased thermal demand for showering in the morning and dish washing in the evening result in higher flexibilities during these periods.

own individual flexibility – i. e. the set of schedules that might be operated without violating any technical operational constraint – based on the capabilities of the unit, operation conditions (weather, etc.), cost restrictions and so forth. Modeling flexibility independently from specific energy units demands a means for meta-modeling that allows model independent access to feasibility information. (Bremer et al., 2011) introduced a support vector based model that captures individual feasible regions from training sets of operable example schedules. Figure 1 shows an example training set for a co-generation plant. An extend use case for systematic solution repair with these models has been introduced in (Bremer and Sonnenschein, 2013a). Agent-based approaches can derive a so called support vector decoder automatically from the surrogate model and use it as a means for systematically generating feasible solutions without domain knowledge on the (possible, situational) operations of the controlled energy resource. In general, the idea works in two successive stages – a decoder training phase and the actual planning phase. First a training set of feasible schedules is generated for each energy unit using a situationally parametrized simulation model of the energy unit. Then, the flexibility model is derived from the training set. During the succeeding load planning phase, these decoders may be used by an optimization algorithm that determines the optimal partition of a given active power target schedule into schedules for each single unit. The decoder automatically generates feasible solutions. With this approach as abstraction layer, the solver does not need any domain knowledge about the energy units, their individual constraints, or possible operation.

An example for a recently developed agent approach for fully decentralized predictive scheduling is given by the combinatorial heuristics for distributed agents (COHDA). In COHDA (Hinrichs, 2014) each agent locally decides on feasible schedules for the represented unit with a decoder, but, as soon as an agent has to represent a local ensemble of energy units instead of a single device, a problem arises because usually only flexibility models of single units are available and a concept for statistically sound aggregating a set of flexibility models is missing so far (Bremer and Lehnhoff, 2017). Generating a single decoder for handling all constraints and feasible operations of the whole ensemble is hardly possible due to statistical problems when combining training sets from individually sampled flexibility models. Due to the folded densities only a very small portion from the interior of the feasible region (the dense region) is captured by the machine learning process. But, a combined training set is needed. The same holds true for any other centralized or agent-based orchestration.

In this paper, an approach is presented that introduces an intermediate density optimization step into the training process. Schedules from individual training sets modeling individual flexibilities are aggregated to a joint training set. The skewed, aggregated training set is then unfolded by Simulated Annealing. The feasibility of the joint training set is maintained by using the individual feasibility models.

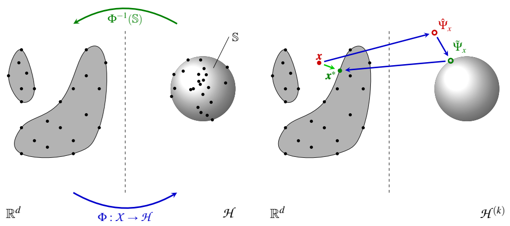
2 PREDICTIVE SCHEDULING AND FLEXIBILITY MODELING

Virtual power plants are a means for aggregating and controlling DER (Awerbuch and Preston, 1997). In scenarios with independently operated units, self-organizing algorithms are required also for coordination. In general, distributed control schemes based on multi-agent systems are considered advantageous for large-scale problems as expected in future smart grids due to the large number of distributed energy resources that take over control tasks from large-scale central power plants (Nieße et al., 2012). Some recent implementations are (Hinrichs et al., 2013; Ramchurn et al., 2011; Kamphuis et al., 2007).

One of the crucial challenges in operating a VPP arises from the complexity of the scheduling task due to the large amount of (small) energy units in the distribution grid (McArthur et al., 2007). In the following, we consider predictive scheduling, where the goal is to select exactly one schedule $\boldsymbol{x}_i$ for each controlled energy unit U_i from a search space $\mathcal{F}_i$ of feasible schedules specific to the possible operations and technical constraints of unit U_i and with respect to a future planning horizon, such that a global objective function (e. g. resembling a target power profile) is optimized by the sum of individual contributions. A basic formulation of the scheduling problem is given by

$$\delta\left(\sum_{i=1}^{m} \boldsymbol{x}_i, \boldsymbol{\zeta}\right) \rightarrow \min; \text{ s. t. } \boldsymbol{x}_i \in \mathcal{F}_i \; \forall U_i \in \mathcal{U}. \qquad (1)$$

In equation (1) δ denotes an (in general) arbitrary distance measure for evaluating the difference



(a) (b)

Figure 2: General support vector decoder scheme for solution repair and constraint handling (Bremer and Lehnhoff, 2017).

between the aggregated schedule of the group and the desired target schedule $\boldsymbol{\zeta}$. W. l. o. g. we assume that the Euclidean distance is used.

To each energy unit U_i exactly one schedule $\boldsymbol{x}_i$ has to be assigned. The desired target schedule is given by $\boldsymbol{\zeta}$. Solving this problem without unit independent constraint handling leads to specific implementations that are not suitable for handling changes in VPP composition or unit setup and thus leads to enlarged integration cost for new units.

Flexibility modeling can be understood as the task of modeling constraints for energy units. Apart from global VPP constraints, constraints often appear within single energy components; affecting the local decision making. Popular methods treat constraints or aggregations of constraints as separate objectives or penalties, leading to a transformation into a (unconstrained) many-objective problem (Kramer, 2010; Smith and Coit, 1997).

For optimization approaches in smart grid scenarios, black-box models capable of abstracting from the intrinsic model have proved useful (Pinto et al., 2017; Gieseke and Kramer, 2013; Schiendorfer et al., 2014; Bremer and Sonnenschein, 2013a). The units 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 (Bremer and Sonnenschein, 2013b; Coello Coello, 2002).

Thus, a decoder allows for a targeted search by. It imposing a relationship between a decoder solution and a feasible solution (Coello Coello, 2002).

A schedule of an energy unit can be seen as a real valued vector $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathcal{F}_i \subset \mathbb{R}^d$ with each element x_j denoting mean electrical power during the jth time interval. $\mathcal{F}_i$ denotes the specific feasible subset of schedules that may be operated by energy unit U_i without violating any technical constraints.

Fig. 2 shows the idea of a support vector decoder starting with a set of feasible example schedules derived from a simulation model of the respective energy unit and using it as a stencil for the region that contains just feasible schedules.

A training set $\mathcal{X}$ containing only valid schedules, can e. g. be derived after a sampling approach from (Bremer and Sonnenschein, 2013c). From such a training set, a support vector data description (SVDD) can derive a geometrical description of the sub-space that contains the given data (Tax and Duin, 2004); in our case: the set of feasible schedules. As a prerequisite, the samples from the training set have to be distributed appropriately across the feasible region. Given a set of data samples, the enclosing envelope (a model of the feasible region and thus of the flexibility) can be derived as follows: After mapping the data to a high dimensional feature space, the smallest enclosing ball in this feature space is determined. When mapping back the ball to data space, it forms a set of contours enclosing the given data sample.

This task is achieved by determining a mapping $\Phi : \mathcal{X} \subset \mathcal{F} \subset \mathbb{R}^d \to \mathcal{H};\ x \mapsto \Phi(x)$ such that all data from a training set $\mathcal{X}$ is mapped to a minimal hypersphere in $\mathcal{H}$. The minimal sphere with radius R and center a in $\mathcal{H}$ that encloses $\{\Phi(\boldsymbol{x}_i)\}_N$ can be derived from minimizing $\|\Phi(\boldsymbol{x}_i) - a\|^2 \leqslant R^2 + \xi_i$ with slack variables $\xi_i \geqslant 0$ for a smoother ball.

After some relaxations one gets two main outcomes: the center $a = \sum_i \beta_i \Phi(\boldsymbol{x}_i)$ (with $\boldsymbol{\beta}$ weighting the impact of different schedules) of the minimal sphere in terms of an expansion into $\mathcal{H}$ and a function that allows to determine the distance of the image of an arbitrary point from $a \in \mathcal{H}$, calculated in $\mathbb{R}^d$ is derived: $R^2(\boldsymbol{x}) = 1 - 2\sum_i \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}) + \sum_{i,j} \beta_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, with a kernel k that substitutes dot products in Hilbert space. Because all support vectors are mapped onto the surface of the sphere, the sphere radius $R_{\mathbb{S}}$ can be easily determined by the distance of an arbitrary support vector to the center. Thus the feasible region can now be modeled by a flexibility model $\mathcal{M}^{\mathcal{F}}$ as

$$\mathcal{M}^{\mathcal{F}} = \{\boldsymbol{x} \in \mathbb{R}^d | R(\boldsymbol{x}) \leqslant R_{\mathbb{S}}\} \approx \mathcal{X}. \qquad (2)$$

The model can be used as a black-box that abstracts from any explicitly given form of constraints and allows for a decision on whether a given solution is feasible or not. At the same time, decoders serve as an abstraction layer. Learned from a training set of feasible example schedules, a decoder hides all unit specific details. In this way, no domain specific knowledge on possible operation, constraints or cost of incorporated energy units have to be implemented or integrated into the algorithm.

For our experiments, we used a decoder as described in (Bremer and Sonnenschein, 2013a). Here, a decoder γ is given as mapping function for schedules $\boldsymbol{x}$ $\gamma : \mathbb{R}^d \to \mathbb{R}^d;\ \gamma(\boldsymbol{x}) \mapsto \boldsymbol{x}^*$. With $\boldsymbol{x}^*$ having the following properties (Sonnenschein et al., 2014):

- $\boldsymbol{x}^*$ is operable by the respective energy unit without violating any constraint,
- the distance $\|\boldsymbol{x}-\boldsymbol{x}^*\|$ is small and small depends on the problem at hand and often denotes the smallest distance of $\boldsymbol{x}$ to the feasible region.

The right hand side of Figure 2(b) shows how such a decoder can be derived from model (2). If a schedule is feasible it is inside the feasible region (grey area on the left in Fig. 2(b)). Thus, the schedule is inside the pre-image (modeling the feasible region) of the ball and thus its high-dimensional image lies inside the ball. An infeasible schedule (e. g. $\boldsymbol{x}$ in Fig. 2(b)) lies outside the feasible region and thus its image $\hat{\Psi}_x$ lies outside the ball. But, some relations are known: the center, the distance of the image from the center and the radius of the ball. Hence, the image of an infeasible schedule can be moved along the difference vector towards the center until it touches the ball. Then, the pre-image of the moved image $\tilde{\Psi}_x$ represents a repaired schedule $\boldsymbol{x}^*$ at the boundary of the feasible region. No mathematical description of the original feasible region or of the constraints is needed to do this. More sophisticated variants of transformation are e. g. given in (Bremer and Sonnenschein, 2013a).

With such decoder concept for constraint handling one can now reformulate the optimization problem as

$$\delta\left(\sum_{i=1}^{m}\gamma_i(\boldsymbol{x}_i),\boldsymbol{\zeta}\right)\rightarrow \min, \tag{3}$$

where γ_i is the decoder function of unit i that produces feasible, schedules from $\boldsymbol{x}\in[0,p_{max}]^d$ (with rated power p_{max}) resulting in schedules that are operable by that unit. Please note, that with this constraint free formulation, many standard algorithms for optimization can be easily adapted and no domain specific implementation (regarding the energy units and their operation schedules) has to be integrated. Equation (3) is used as a surrogate objective to find the solution to the constrained optimization problem equation (1).

So far, this approach has been proven to work fine if each entity in a virtual power plant is modeled as a single controlled entity. On the other hand, many scenarios exist where also ensembles of energy units should be integrated. In (Bremer and Lehnhoff, 2017) the problem has been circumvented by integrating a second level optimization for orchestrating an ensemble internally and representing it by a single coordinating agent. This approach entails additional optimization effort into the overall coordinating process. Thus, a single flexibility model would be desirable as an abstraction layer for ensembles of energy units.

3 SAMPLING FROM ENSEMBLES OF ENERGY UNITS

Sometimes the technical equipment of a single unit in a VPP consists of more than just a single generator (or prosumer or controllable load). Nevertheless, the owner as operator is still represented by a single controlling agent when embedded into a decentralized agent-based control scheme inside a virtual power plant. In this case that agent has to handle the ensemble of energy units as a single unit (in a sense as a single sub VPP) and negotiate to the other agents with the aggregated flexibility. Nevertheless, there is usually no joint model of the whole ensemble, and thus the agent has to use an individual model of each unit and thus a set of individual decoders for deciding on an aggregated schedule for the ensemble.

If an agent covers a set of energy units instead of a single unit, a decoder for the joint feasible region of the group of units has to be used. A model of the operation of the ensemble of units is often not available. Using the training sets of individual energy units and randomly combining them (adding up exactly one from each training set) to joint schedules in order to gain a training set for the joint behavior is not targeted. The problem is that all source trainings sets are independent random samples and thus the resulting training set exhibits a density (of operable power levels) that results from folding the source distributions. Figure 3 shows an example. Uniformly distributed values for levels of power as in the case of an co-generation plant with sufficient buffer capacity fold up – in case of ensembles with more than one CHP – to an multi-modal Irvin-Hall-distribution (Hall, 1927). This distribution has some similarities to a sharp normal distribution and the more samples (energy units in the ensemble) are folded the more leptokurtic the pdf gets. This leads to a sample with a very high density in the middle of the feasible region. At the outskirts the sample is extremely sparse. Thus, instances from the outer parts are neglected as outliers from the support vector approach that generates the surrogate model and the decoder.

For this reason, a decoder trained from such a training sample reproduces only a very small, inner portion of the feasible region. In this way, most of the flexibility that an ensemble could bring in into a virtual power plant control is neglected. This can also be seen in Figure 4. The rather small grey boxes represent the data (power levels for different time intervals) that actually should spread over the area denoted by the outer whiskers. Only the small inner part is going to be learned by a model.

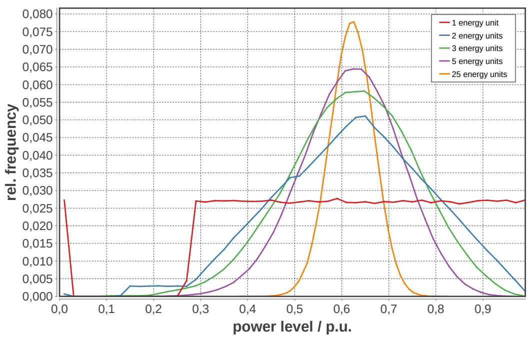


Figure 3: Probability density of different numbers of folded distributions of operable power levels for co-generation plants.

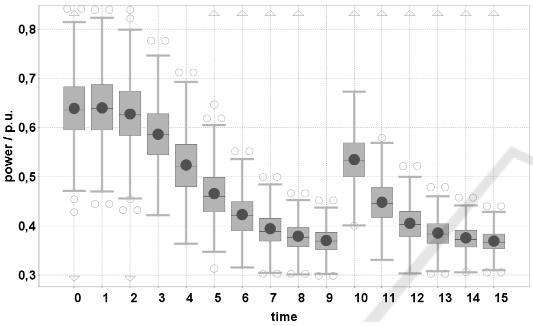


Figure 4: Probability distribution of power levels at different time intervals of an ensemble of 10 micro CHP units. The training set exhibits a concentration in the inner part of the whole flexibility (grey boxes denoting 3/4 of the samples) making it highly imbalanced.

4 OPTIMIZING THE SAMPLE DISTRIBUTION

To overcome the problem of folded distributions in ensemble training sets, we introduce a post processing step that improves a training set of feasible schedules by trying to equally distribute the schedules from a randomly generated training set across the whole feasible region that reflects the joint flexibility of the ensemble of energy units.

To achieve this task, we use a Simulated Annealing approach (Kirkpatrick et al., 1983).

For optimization we consider the following model. For each energy unit U_i a training set $\mathcal{X}_j = \{\boldsymbol{x}_i\}_m \subseteq \mathcal{F}_i \subset \mathbb{R}^d$ of feasible schedules is given. Now, we can define a set of matrices $\mathcal{X}^* = \{\boldsymbol{X}_1, \dots, \boldsymbol{X}_n\}$, with

$$\boldsymbol{X}_i = \begin{pmatrix} \boldsymbol{x}_{r_1} \in \mathcal{X}_1 \subseteq \mathcal{F}_i \\ \vdots \\ \boldsymbol{x}_{r_n} \in \mathcal{X}_n \subseteq \mathcal{F}_n \end{pmatrix}, \tag{4}$$

with uniformly distributed random indices $r_1, \dots, r_n \sim \mathcal{U}(1, m)$. Then, row j in matrix $\boldsymbol{X}_i$ represents a randomly chosen sample of a feasible schedule from energy unit U_j. The joint training set that reflects the flexibility of the whole group of energy units can now be defined as $\mathcal{X}^\sigma = \{S(\boldsymbol{X}_1), \dots, S(\boldsymbol{X}_m)\}$, with $S(\boldsymbol{X}_j) = \sum_{i=1}^{n} \boldsymbol{x}_i \in \boldsymbol{X}_j$ ($\boldsymbol{x}_i$: ith row of matrix $\boldsymbol{X}_j$). In $\mathcal{X}^\sigma$ each element represents the sum of randomly chosen elements (schedules); one from each energy unit in the group. In this way, $\mathcal{X}^\sigma$ represents the aggregated flexibility of the group. For deriving a machine learning model, this training set is hardly suitable because of the folded densities due to summing up over different random series. In the next step, we want to improve this training set by correcting the unfavorable densities. To do this, we first define

$$h^\sigma(\mathcal{X}^\sigma) = \sum_{i=1}^{d-1} h\left(\begin{matrix} \{x\}_{i,j} \\ \{x\}_{(i+1),j} \end{matrix} \right). \tag{5}$$

Function $h^\sigma(\mathcal{X}^\sigma)$ denotes a concave hull (Duckham et al., 2008) around the training set and as the calculation of high-dimensional concave hulls quickly grows intractable, we approximate by summing up over a set of 2-dimensional concave hulls around neighboring cross-sections through the d-dimensional schedules in the training set. Maximizing the area of the concave hull $A(h^\sigma)$ ensures that the flexibility is captured also at the outskirts of the feasible region. In order to spread samples equally across this maximized area, the second indicator comes into play. Let $x_{j,1} \leqslant x_{j,2} \leqslant \dots \leqslant x_{j,m}$ be the sorted values of the jth elements of $\mathcal{X}^\sigma$ and let $\boldsymbol{x}^\sigma = x_{i,2} - x_{i,1}, x_{i,3} - x_{i,2}, \dots, x_{i,n} - x_{i,m-1}$ be the series of successive differences. Now we define the variance

$$\sigma_\delta^2 = \sum_{i=1}^{m-1} P_i \left(x_i^\delta - \frac{1}{n} \sum_{j \geqslant 1}^{n-1} x_j^\delta \right) \tag{6}$$

to measure the spread of differences of the vectors in the training set. Minimizing this spread ensures equalizing the spread across the feasible region.

With these two indicators we can now define our objective: minimize E:

$$E(\mathcal{X}^\sigma) w \cdot \sigma_\delta^2 + \frac{1-w}{A(h^\sigma)} \to \min \tag{7}$$

as a weighted mixture of both criteria, which is to be optimized with respect to the following constraint.Let $\boldsymbol{x}_j = S(X_j) \in \mathcal{X}^\sigma$ be an instance from the ensemble training set. We define feasibility over $S(X_j)$: $\boldsymbol{x}_j$ is feasible (cf. Eq. (2)) iff

$$(X_j)_1 \in \mathcal{F}_1 \wedge (X_j)_2 \in \mathcal{F}_2 \wedge \dots \wedge (X_j)_n \in \mathcal{F}_n, \tag{8}$$

for all units $U_1, \ldots, U_n$ in the ensemble $\mathcal{U}$ of all units.

In Eq. (8) feasibility of an aggregated schedule $\boldsymbol{x}_j$ is checked by probing whether all schedules (rows) from the respective component matrix that make up the aggregated schedule are feasible for the respective energy unit. This can be easily tested with the help of the respective unit specific flexibility models $\{\mathcal{M}\}$ as described in Eq. (2).

Simulated Annealing (Kirkpatrick et al., 1983) is an established Markov Chain Monte Carlo Methods (MCMC) for non-linear optimization. It mimics a physical cooling process. In general, MCMC methods are an effective tool for statistical sampling applied to optimization problems (Li et al., 2009).

Algorithm 1 shows the general process for optimizing the unfavorable densities due to the folded distributions in aggregated ensemble training sets. First, for each energy unit in the group, a specific training set is sampled from an appropriately parameterized simulation model and a flexibility model $\mathcal{M}^{\mathcal{F}}{}_i$ is trained for units i. Each of these flexibility models is able to decide for the respective energy unit whether a given schedule is operable or not. Next, the algorithm is initialized with temperature ϑ, weights w for the objective and a cooling rate. While no stopping criterion is met the following loop is executed. The training set is mutated to generate a new offspring training set. The mutation operator for our Simulated Annealing is defined as follows:

$$\hat{\mathcal{X}}^{\sigma}_{r^{\sigma}} = \mathcal{X}^{\sigma}_{r^{\sigma}} + \begin{pmatrix} r_1 \sim \mathcal{N}(0,\sigma^2) \\ \vdots \\ r_m \sim \mathcal{N}(0,\sigma^2) \end{pmatrix}. \tag{9}$$

Mutation is done by adding a vector with normal distributed random values (with variance σ^2 as step size) to a randomly chosen instance $r^{\sigma} \sim \mathcal{U}(1,m)$ from $\mathcal{X}^{\sigma}$.

Not necessarily all components have to be mutated at the same time. Algorithm 1 shows a version with only 20% of the components mutated. The feasibility of the mutated training set element is checked with the help of the set of flexibility models $\{\mathcal{M}^{\mathcal{F}}\}$. If the mutated element is not feasible, it is rejected until a feasible version is found. With this barrier approach feasibility of the solution is ensured. Finally, the objective value of mutated training set is compared with the old one and accepted (or not) after the metropolis criterion.

5 RESULTS

To evaluate the effectiveness of our approach, we simulated ensembles of non-homogeneous micro-CHP of different size. We tested the performance of respectively two classifiers that make up the flexibility model (one trained with the original ensemble training set and one trained with the respective optimized training set). Table 1 shows the result for different ensemble sizes up to groups of 50 units. Up to 40 units the Simulated Annealing approach works very well. Table 1 compares for both classifiers the accuracy, the sensitivity, the specificity and the miss rate respectively calculated by comparing the classification result (feasible and thus operable by the ensemble, or not) with the simulation model (simulating whether a schedule is really operable, or not).

The accuracy denotes the rate of correctly classified schedules (feasible as well as not feasible). After optimizing and planing the training set, the trained classifiers perform almost as well as in the single unit case (Bremer et al., 2011). In the case of 50 units

Algorithm 1: Basic scheme for the Simulated Annealing approach for ensemble training set improvement.

sample units
build models $\{\mathcal{M}^{\mathcal{F}}\}_n$ for single units
build $\boldsymbol{X}^*$ and $\mathcal{X}^{\sigma}$
initialize temperature ϑ and weights w
while $\vartheta < \vartheta_{min}$ **do**
  repeat
    $\hat{\mathcal{X}}^{\sigma} \leftarrow \mathcal{X}^{\sigma}$
    $j \leftarrow r \sim \mathcal{U}(0,m)$
    for $c = 1; c < m; c$++ **do**
      if $r \sim \mathcal{U}(0,1) \leqslant 0.2$ **then**
        $\hat{\mathcal{X}}^{\sigma}_{j,c} \leftarrow \mathcal{X}^{\sigma} + r \sim \mathcal{N}(0,\sigma^2)$
      end if
    end for
    check feasibility Eq. (8)
  until mutation feasible
  if $e^{-\frac{E(\hat{\mathcal{X}}^{\sigma})-E(\mathcal{X}^{\sigma})}{\vartheta}} > r \sim U(0,1)$ **then**
    $\mathcal{X}^{\sigma} \leftarrow \hat{\mathcal{X}}^{\sigma}$
  end if
  $\vartheta \leftarrow \text{cooling}(\vartheta)$
end while

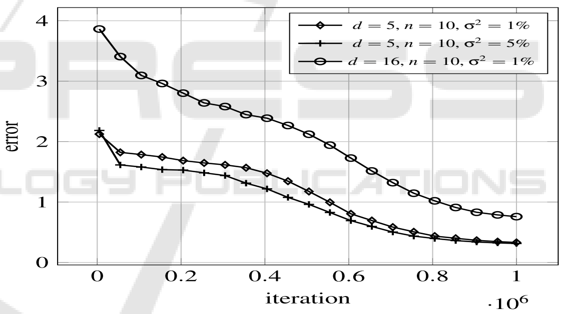
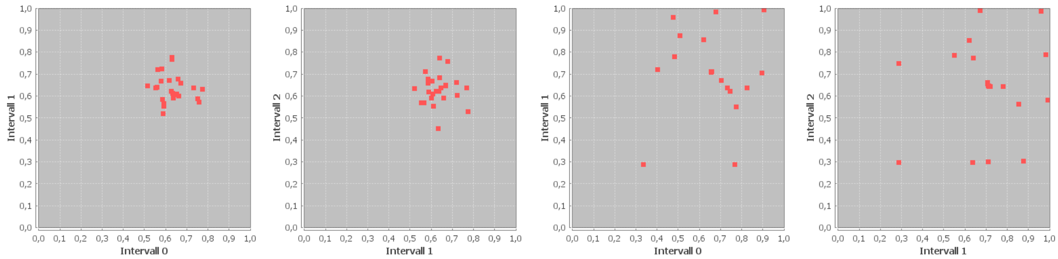


Figure 5: Convergence of the SA approach.



(a) (b)

Figure 6: Before-and-after test for an ensemble of 20 micro-co-generation plants with 5-dimensional schedules. (a) shows the aggregated training set before unfolding and the improvement after (b).

Table 1: Classifier performances of classifiers (search space models) trained with the original training set ($\mathcal{X}_{orig}$) with folded power level distributions and with the improved training sets ($\mathcal{X}_{opt}$) optimized with the SA approach.

# units	set	accuracy	sensitivity	specificity	miss
$m = 10$	$\mathcal{X}^{\sigma}_{\text{orig}}$	0.493	0.027	0.999	0.972
	$\mathcal{X}^{\sigma}_{\text{opt}}$	0.935	0.921	0.951	0.078
$m = 20$	$\mathcal{X}^{\sigma}_{\text{orig}}$	0.506	0.043	1.000	0.957
	$\mathcal{X}^{\sigma}_{\text{opt}}$	0.938	0.903	0.975	0.097
$m = 30$	$\mathcal{X}^{\sigma}_{\text{orig}}$	0.513	0.002	0.999	0.997
	$\mathcal{X}^{\sigma}_{\text{opt}}$	0.964	0.963	0.964	0.037
$m = 40$	$\mathcal{X}^{\sigma}_{\text{orig}}$	0.497	0.002	1.000	0.998
	$\mathcal{X}^{\sigma}_{\text{opt}}$	0.907	0.843	0.972	0.157
$m = 50$	$\mathcal{X}_{\text{orig}}$	0.572	0.001	1.000	0.999
	$\mathcal{X}_{\text{opt}}$	0.577	0.015	0.998	0.985

(or more) the improvement drops rapidly. The sensitivity shows the share of correctly classified feasible schedules. The specificity, on the other hand, denotes the rate of correctly classified infeasible schedules (compared with all schedules). This score of course is unbeatable high for the original training set, because with this training set the feasible region is seriously underestimated in size (only a small, central sub-region is learned) and thus almost never infeasible schedules at the boundary of the real (lots larger) regions is falsely classified. Nevertheless, the improved training set performs almost as good. The miss rate shows the rate of falsely classified infeasible schedules which drops significantly after optimizing the training set. A visual impression of the improvement that is achieved by the optimization of the training set can be seen in Figure 6. The figure shows 2-dimensional intersections through 5-dimensional schedules before and after optimization. The convergence of the algorithms has been tested on several scenarios. Figure 5 shows some of the results.

6 CONCLUSION AND FURTHER WORK

Using machine learning approaches for flexibility modeling and automatically deriving decoders from these models for efficient and domain knowledge independent implementation of (distributed) optimization methods has proven a useful tool in managing the future smart grid. So far, these models can only be applied to single energy units, because distributions of power levels in the training sets of single units fold up when aggregating them to ensemble training sets. Thus, the training set render useless for appropriately deriving a model for the joint flexibility of a group of energy units. We presented an approach to overcome the problem of folded distributions when training decoders for ensembles of energy resources in predictive scheduling. For this reason, we introduced an intermediate step prior to model training. After aggregating schedules from training sets for the flexibility of individual energy resources (resulting in a training set with an abnormally high density of training instances in the middle of the feasible region), a Simulated Annealing step attenuates this skewed density while at the same time maintaining the feasibility of the schedules.

Future work still has to show whether the process of correcting the training set density can be achieved sufficiently fast also for short term predictive scheduling. Up to now, the Simulated Annealing approach takes up to several minutes to complete. Another necessary improvement will distribute the training set correction to a fully decentralized approach. Nevertheless, the results so far demonstrate the feasibility of correcting to training set of ensembles of energy units.

With our approach also households, hotels, small businesses, schools or similar with an ensemble of co-generation, heat pump, solar power, and controllable consumers will be able to take part in agent-based decentralized predictive scheduling for providing energy services in future smart grid architectures without a need for an (expensive) individual link of each single device in the ensemble.

REFERENCES

Abarrategui, O., Marti, J., and Gonzalez, A. (2009). Constructing the active european power grid. In *Proceedings of WCPEE09*, Cairo.

Awerbuch, S. and Preston, A. M., editors (1997). *The Virtual Utility: Accounting, Technology & Competitive Aspects of the Emerging Industry*, volume 26 of *Topics in Regulatory Economics and Policy*. Kluwer Academic Publishers.

Bremer, J. and Lehnhoff, S. (2017). *Hybrid Multi-ensemble Scheduling*, pages 342–358. Springer International Publishing, Cham.

Bremer, J., Rapp, B., and Sonnenschein, M. (2011). Encoding distributed search spaces for virtual power plants. In *IEEE Symposium Series on Computational Intelligence 2011 (SSCI 2011)*, Paris, France.

Bremer, J. and Sonnenschein, M. (2013a). Constraint-handling for optimization with support vector surrogate models. In Filipe, J. and Fred, A., editors, *ICAART 2013 – Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, volume 2, pages 91–105, Barcelona, Spain. SciTePress.

Bremer, J. and Sonnenschein, M. (2013b). Model-based integration of constrained search spaces into distributed planning of active power provision. *Comput. Sci. Inf. Syst.*, 10(4):1823–1854.

Bremer, J. and Sonnenschein, M. (2013c). Sampling the search space of energy resources for self-organized, agent-based planning of active power provision. In Page, B., Fleischer, A. G., Göbel, J., and Wohlgemuth, V., editors, *27th International Conference on Environmental Informatics for Environmental Protection, EnviroInfo 2013*, pages 214–222. Shaker.

Coello Coello, C. A. (2002). 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*, 191(11-12):1245–1287.

Duckham, M., Kulik, L., Worboys, M., and Galton, A. (2008). Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224 – 3236.

European Parliament & Council (2009). Directive 2009/28/ec of 23 april 2009 on the promotion of the use of energy from renewable sources and amending and subsequently repealing directives 2001/77/ec and 2003/30/ec.

Gieseke, F. and Kramer, O. (2013). Towards non-linear constraint estimation for expensive optimization. In Esparcia-Alcázar, A., editor, *Applications of Evolutionary Computation*, volume 7835 of *LNCS*, pages 459–468. Springer Berlin Heidelberg.

Hall, P. (1927). The distribution of means for samples of size n drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika*, 19(3/4):pp. 240–245.

Hinrichs, C. (2014). *Selbstorganisierte Einsatzplanung dezentraler Akteure im Smart Grid*. PhD thesis, Carl von Ossietzky Universitt Oldenburg.

Hinrichs, C., Bremer, J., and Sonnenschein, M. (2013). Distributed Hybrid Constraint Handling in Large Scale Virtual Power Plants. In *IEEE PES Conference on Innovative Smart Grid Technologies Europe (ISGT Europe 2013)*. IEEE Power & Energy Society.

Kamphuis, R., Warmer, C., Hommelberg, M., and Kok, K. (2007). Massive coordination of dispersed generation using powermatcher based software agents. In *19th International Conference on Electricity Distribution*.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kramer, O. (2010). A review of constraint-handling techniques for evolution strategies. *Appl. Comp. Intell. Soft Comput.*, 2010:1–19.

Li, Y., Protopopescu, V. A., Arnold, N., Zhang, X., and Gorin, A. (2009). Hybrid parallel tempering and simulated annealing method. *Applied Mathematics and Computation*, 212(1):216–228.

McArthur, S., Davidson, E., Catterson, V., Dimeas, A., Hatziargyriou, N., Ponci, F., and Funabashi, T. (2007). Multi-agent systems for power engineering applications – Part I: Concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems*, 22(4):1743–1752.

Nieße, A., Lehnhoff, S., Tröschel, M., Uslar, M., Wissing, C., Appelrath, H.-J., and Sonnenschein, M. (2012). Market–based self–organized provision of active power and ancillary services. IEEE.

Padhy, N. (2004). Unit Commitment - A Bibliographical Survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205.

Pinto, R., Bessa, R. J., and Matos, M. A. (2017). Surrogate model of multi-period flexibility from a home energy management system. *CoRR*, abs/1703.08825.

Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. R. (2011). Agent-based homeostatic control for green energy in the smart grid. *ACM Trans. Intell. Syst. Technol.*, 2(4):35:1–35:28.

Schiendorfer, A., Steghöfer, J.-P., and Reif, W. (2014). Synthesised constraint models for distributed energy management. In Ganzha, M., Maciaszek, L. A., and Paprzycki, M., editors, *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 1529–1538, Warsaw, Poland.

Smith, A. and Coit, D. (1997). *Handbook of Evolutionary Computation*, chapter Penalty Functions, page Section C5.2. Oxford University Press and IOP Publishing, Department of Industrial Engineering, University of Pittsburgh, USA.

Sonnenschein, M., Lünsdorf, O., Bremer, J., and Tröschel, M. (2014). Decentralized control of units in smart grids for the support of renewable energy supply. *Environmental Impact Assessment Review*, (0):–.

Tax, D. M. J. and Duin, R. P. W. (2004). Support vector data description. *Mach. Learn.*, 54(1):45–66.