Automatic Reconstruction of Performance Indicators from Support Vector based Search Space Models in Distributed Real Power Planning Scenarios

Jörg Bremer, Michael Sonnenschein

Department for Computing Science
University of Oldenburg
Uhlhornsweg 84
26129 Oldenburg
Joerg.Bremer@uni-oldenburg.de
Michael.Sonnenschein@uni-oldenburg.de

Abstract: In order to allow for a transition of the current central market and network structure of todays electricity grid to a decentralized smart grid, an efficient management of numerous distributed energy resources will become more and more indispensable. The integration of controllable but distributed and individually configured small devices entails new challenges for the design of future control structures. Pooling and joint planning are proposed concepts for future control schemes. Each energy unit may offer an individual set of feasible schedules for some future planning horizon. If a coalition of distributed energy units is supposed to plan the provision of a wanted joint real power schedule, a many-objective problem arises that demands different evaluation criteria to weight alternative schedules against each other. Each schedule causes individual costs and may be described by individual indicators like reliability, environmental impact or robustness criteria for planning. Communication and integration into planning algorithms without having to know the individual technical setting (i.e. model, operational state, or individual constraints and their modeling) of any energy unit is the aim of this paper. We propose a combination of two approaches: surrogate modeling and a decoder approach for constraint-handling that integrates indicators that are annotated to schedules. We show the appropriateness of this approach by scrutinizing the error when reconstructing the values from the black-box model.

1 Introduction

New control structures and methods in future smart grid scenarios will have to cope with managing rather small, distributed (and operated by different owners) energy resources that are individually configured: household-size co-generation plants (CHP), photovoltaic systems (with and without coupled battery systems) or wind-energy converters. In future, a growing number of energy consumers are expected to become controllable [SBP$^+$]. Grouping energy resources to coalitions for joint production of real power (as well as for ancillary services) is a discussed topic [WLHK07, NLT$^+$12, MVO11, KE09, RVRJ11] for solving the NP-hard [GZP03] control problem of orchestrating these resources. Throughout this paper we focus on the use case of planning day-ahead real power provision. In this way, we assume a scenario where frequently and self-organized by a multi-agent system, the set of all energy resources partitions into coalitions that goto market and end up with the task of having to provide a given joint energy schedule for some future planning horizon. We concentrate on the optimization step that has to find an appropriate schedule for each participating unit.

Planning the provision of real power by coalitions of pooled distributed energy resources, usually results in distributed many objective optimization problems. The main objective is finding a feasible (operable without violating any technical constraints) schedule for each participating energy unit such that the aggregated schedule of all units in the coalition resembles a given schedule (e.g. defined by a product on some energy market). At the same time, local individual costs for different candidate schedules from the search spaces of the individual units have to be taken into account:

- Each unit wants to maximize own profit, what could be implemented egoistically by minimizing own cost or – depending on the implemented profit distribution scheme altruistically by minimizing the coalitions outcome. The latter would automatically involve knowledge about the costs in the decision spaces of other members from the coalition.
- Each unit wants to optimize weak constraints like personal comfort (temperature, no washing during night hours, etc.).
- Each unit wants to preserve own degrees of freedom in order to maintain as much options for future market choices as possible. This would require the annotation of schedules with an evaluation criterion of the operational state results from operating the schedule.

Further parameters might characterize individual schedules in terms of reliability or robustness. In [BRS10], a surrogate model has been proposed that is capable of capturing the feasible region that contains the set (or subspace) of schedules that might be realized without violating any technical constraint. The model has been extended to the inclusion of characterizing performance indicators for all schedules in terms of black-box information [Bre12], i.e. the model may be used to check whether a given schedule is feasible and has correct information about characterizing indicators attached. Thus, the means for modeling and communicating the information already exists. But, it was as yet not possible to automatically have schedules extracted from the model with correctly assigned values for the indicators. In [BS13] a decoder for constraint-handling based on the search space model has been proposed with the capability of repairing infeasible solutions.

The scheduling problem sketched above is a complex optimization problem. Based on data from simulating the respective energy resources a model is learned by a support vector approach. A mapping derived from this support vector model had so far been used to guide an algorithm towards feasible solutions [BS12]. We now extended this approach to a decoder that guides an algorithm towards good solution by integrating evaluation criteria into the support vector model and decoder. In this way, we demonstrate that the decoder approach can be extended to many-objective use cases.

In the following, we will not discuss the optimization procedure itself or the application of suitable planning algorithms as this is not the topic of this paper. We will rather focus on the necessary information on different performance indicators associated to individual, alternative schedules and on how an algorithm may harness this information when searching for good solutions.

Hence, the rest of the paper is organized as follows: We start with a description of our use case and a discussion of related work. We recap the preliminary calculations for building the model and discuss the application of the decoder for indicator integration. We conclude with some simulation results.

2 Background

We start with describing the use case from the smart grid domain where our method will be applied and a discussion of previous and related work within this field. For clarity, we restrict our discussion to the optimization part. For the overall scenario we assume the existence of a future energy market (besides a market for ancillary services) where coalitions of energy units (represented by agents) may offer individually compiled active power products [NLT$^+$12, SAH$^+$12]. Coalitions form-up of their own accord in a self-organized way. In this way, we go for a market-based self-organized provision of active power as basis for planning individual loads for future time horizons [NLT$^+$12, BS12]. Ancillary services do have related (not identical) use cases, but for now we stick with the active power use case and the optimization part within the whole process. Related work on self-organized coalition forming with an application in the smart grid field can for instance be found in [Lün12].

Effectively solving real world optimization problems often suffers from the presence of constraints that have to be obeyed when looking for feasible solutions. This holds especially true for problems from the smart grid domain where each electricity unit has its own individually configured and constrained search space of alternatively operable schedules [NLT$^+$12, BGKL11, BRS10]. Several techniques for handling constraints during optimization have been developed. Nevertheless, almost all are concerned with special cases of NLP or require a priori knowledge on the problem structure in order to be properly adapted [MS96]. Some prominent representatives of such techniques are: the introduction of a penalty into the objective function that devalues a solution that violates some constraint, the introduction of a repair mechanism for infeasible solution [LV90], or treating constraints as separate objectives. A good overview on constraints-handling techniques can for instance be found in [CC02] or, more recently, in [Kra10].

In order to give an algorithm hints on how to construct a solution, sometimes so called decoders impose a relationship between feasibility and decoder solutions. For example, [KM99] proposed a homomorphous mapping between an n-dimensional hyper cube and the feasible region in order to transform the problem into a topological equivalent one that is easier to handle. Earlier approaches used Riemannien mapping [Kim98] or Schwarz-Christoffel mapping [Squ75]. Such space mapping techniques are usually used in engineering with the objective to substitute computationally expensive models with a coarser grained surrogate model [BBC+94, BBC+95].

Another use case that exploits a space mapping approach by deriving a decoder for constrained search spaces from a support vector based surrogate model has been presented in [BS12, BS13]. With this approach the authors propose a two step process: First, they have a support vector model of the feasible region learned, i.e. a classifier that distinguishes between operable and not operable schedules. Then, they derive a mapping function that is able to map an arbitrary solution candidate to a nearby feasible solution. In this way, they get a means that guides a search algorithm where to look for feasible solutions and the constrained problem is transferred into an unconstrained one that is much easier to solve.

Tuning the operations of units in a coalition in order to achieve a wanted joint real power schedule, usually involves more objectives than just achieving an aggregated sum schedule that as close as possible resembles a given target schedule. Different schedules entail different cost on an energy unit and as the case may be on the whole coalition; depending on the market model and on the surplus distribution scheme. Anyway, each unit wants to minimize own cost. Moreover, any specific schedule always induces a specific environmental impact that might be described by specialized environmental indicators.

As argued in [Bre12] the calculation of such indicators depends on local (likely private) information and has to be done locally at unit site. In order to make them nevertheless utilizable for an optimization algorithm (whether centralized or distributed) each schedule in the surrogate model that forms the decision base for the algorithm has to be annotated with a set of respective indicators. An extension of the support vector model that is capable of incorporating a set of indicators for each schedule has already be presented and a concept for a supplementing ontology approach for correct interpretation of these indicators was given in [Bre12]. One drawback for the integration in the black-box classifier model is that so far it only answers the question for the correct assignment of a value to a schedule. As yet, an open question was: How can such indicators be integrated in the decoder for an automatic reconstruction of the values associated to a given schedule?

3 Modelling the Search Space

3.1 Use Case

A need for optimization arises when a coalition of distributed energy resources has formed-up and placed a product on some energy market. In case of acceptance, this coalition is supposed to jointly deliver an active power schedule as defined by the final product definition after market-clearing. It is not given that the original planned schedules of all energy units are still suitable and jointly resemble the schedule defined by the product.

The load distribution among the respective members of the coalition as used during negotiation might have been sub-optimal for complexity reduction by using incomplete information or a reduced objective set during coalition forming. Moreover, the partition might have become infeasible due to changed conditions for grid compatibility. Anyway, a successive optimization step is necessary after market-clearing to find an optimal or at least good partition of the product schedule.

Optimal in this sense denotes a many-objective criterion. Hence, the optimization problem that has to be solved is the following: Find exactly one schedule for each member of the coalition such that all individual (technically rooted) constraints of the energy units are met and the following objectives are achieved:

- The distance (i.e. the difference) between aggregated and product schedule is minimized.
- The sum of all individual production costs is minimized.
- The sum of the remaining degrees of freedom of all units is maximized (to preserve a maximum of opportunities for future rounds of power provision planning).
- Individual indicators devaluing unwanted schedules (e.g. CHP in the middle of the night) are minimized.
- More objectives regarding environmental impact, robustness of the plan, etc. are conceivable.

In order to fulfill this optimization task, the algorithm has to know:

- For each energy unit: which schedules are feasible (may be operated without violating any technical constraint) and which are not. This is the individual solution space of a single unit.
- For each feasible schedule: What are the individual costs that have to be considered in the evaluation function in order to achieve the above mentioned objectives?

To make the necessary information from each unit available to the algorithm, the feasible regions could be models by an surrogate model to unify access to the information as argued in [BRS10, Bre12].

3.2 SVDD-Model for Feasible Regions

As a prerequisite for our mapping, we assume that the feasible region of an optimization problem has been encoded by SVDD [TD04]. We will briefly describe this approach after [BRS10]. Given a set of data samples $x_i \in \mathcal{X}$, the inherent structure of the region where the data resides in is derived as follows: After mapping the data to a high dimensional feature space, the smallest enclosing sphere is determined that contains all images. The pre-image of this sphere then forms a contour (not necessarily connected) that encloses the data sample. This task is achieved by determining a mapping

$$\Phi : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathcal{H};\ x \mapsto \Phi(x) \tag{1}$$

such that all data from a sample from a region $\mathcal{X}$ 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 $\{\Phi(x_i)\}_N$ can be derived from minimizing $\|\Phi(x_i) - a\|^2 \leq R^2 + \xi_i$ with $\|\cdot\|$ as the Euclidean norm and slack variables $\xi_i \geq 0$ for soft constraints.

After introducing Lagrangian multipliers and further relaxing to the Wolfe dual form, the well known Mercer's theorem (cf. e.g. [SMB$^+$99]) may be used for calculating dot products in $\mathcal{H}$ by means of a kernel in data space: $\Phi(x_i) \cdot \Phi(x_j) = k(x_i, x_j)$. In order to gain a more smooth adaption, it is known to be advantageous to use a Gaussian kernel: $k_{\mathcal{G}}(x_i, x_j) = e^{-\frac{1}{2\sigma^2}\|x_i - x_j\|^2}$ [BHSHV01]. Putting it all together, the equation that has to be maximized in order to determine the desired sphere is:

$$W(\beta) = \sum_i k(x_i, x_i)\beta_i - \sum_{i,j} \beta_i \beta_j k(x_i, x_j). \tag{2}$$

With $k = k_{\mathcal{G}}$ we get two main results: the center $a = \sum_i \beta_i \Phi(x_i)$ 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:

$$R^2(x) = 1 - 2\sum_i \beta_i k_{\mathcal{G}}(x_i, x) + \sum_{i,j} \beta_i \beta_j k_{\mathcal{G}}(x_i, x_j). \tag{3}$$

Because all support vectors are 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 $\mathcal{F} = \{x \in \mathbb{R}^d | R(x) \leq R_{\mathcal{S}}\} \approx \mathcal{X}$.

The comparably small set of support vectors together with a reduced version of vector β that contains non zero weight values (denoted 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. Moreover, as the radius function Eq. 3 maps to $\mathbb{R}$ and thus allows for a conclusion about how far away a solution is from feasibility as well as for an ordering of solutions according to proximity to the feasible region.

3.3 The Decoder Approach

For integration into optimization, a decoder approach has proved useful [BS12]. Let $\mathcal{F}$ denote the feasible region within the parameter domain of some given optimization problem bounded by an associated set of constraints. It is known, that pre-processing the data by scaling it to $[0, 1]^d$ leads to better adaption [JTD02] when using SVDD. For this reason, optimization problems with scaled domains are considered for this method. $\mathcal{F}_{[0,1]}$ denotes the likewise scaled region of feasible solutions. For the decoder, one needs a mapping

$$\gamma : [0, 1]^d \to \mathcal{F}_{[0,1]} \subseteq [0, 1]^d;\ x \mapsto \gamma(x) \tag{4}$$

that is able to map the unit hypercube $[0, 1]^d$ onto the d-dimensional region of feasible solutions; in our use case: schedules $x \in [0, 1]$ with power level between 0 and 100% of max. power. This mapping is achieved as a composition of three functions:

$$\gamma = \tilde{\Phi}_\ell^{-1} \circ \Gamma_a \circ \hat{\Phi}_\ell. \tag{5}$$

The general idea is as follows: The procedure starts with an arbitrary point $x \in [0,1]^d$ from the unconstrained d-dimensional hypercube. First, x is mapped to an ℓ-dimensional manifold in kernel space that is spanned by the images of the ℓ support vectors. After drawing the mapped point to the sphere in order to pull it into the image of the feasible region, the last step searches the pre-image of the modified image to get a point from $\mathcal{F}_{[0,1]}$.

In the following, we recap these steps in more detail from [BS13] and discuss them in the context of the new use case proposed in this paper. With mapping $\hat{\Phi}_\ell$, defined as

$$\begin{aligned} \hat{\Phi}_\ell :\ \mathbb{R}^d &\to \mathcal{H}^{(\ell)}\ , \\ x &\mapsto K^{-\frac{1}{2}}(k(s_1,x),\dots,k(s_\ell,x)) \end{aligned} \tag{6}$$

with $K_{ij} = k(s_i,s_j)$ as the kernel Gram Matrix, arbitrary vectors x,y from $[0,1]^d$ are mapped to an ℓ-dimensional space $\mathcal{H}^{(\ell)}$ (spanned by the set of support vectors $\{s_1,\dots,s_\ell\}$) that contains an likewise ℓ-dimensional projection of the sphere. Vectors from $\mathcal{F}_{[0,1]}$ go onto or inside, others go outside the sphere.

In order to make schedules feasible, the images of vectors with infeasible power values have to be pulled from the outside of the sphere inside. For this purpose one can use

$$\tilde{\Psi}_x = \Gamma_a(\hat{\Psi}_x) = \hat{\Psi}_x + \mu \cdot (a - \hat{\Psi}_x) \cdot \frac{R_x - R_\mathcal{S}}{R_x} \tag{7}$$

to transform the image $\hat{\Psi}_x$ of data point x produced in step eq. (6) into $\tilde{\Psi}_x \in \hat{\Phi}_\ell(\mathcal{F}_{[0,1]})$ by drawing $\hat{\Psi}_x$ into the sphere. Alternatively, the simpler version

$$\tilde{\Psi}_x = a + \frac{(\hat{\Psi}_x - a)\cdot R_\mathcal{S}}{R_x} \tag{8}$$

may be used for drawing $\hat{\Psi}_x$ just onto the sphere but then without having to estimate parameter $\mu \in [1, R_x]$. Parameter μ allows us to control how far a point is drawn into the sphere ($\mu = 1$ is equivalent to eq. (8), $\mu = R_x$ draws each point onto the center). In this way, each image is re-adjusted proportional to the original distance from the sphere and drawn into the direction of the center. As the mapped points are not really arbitrary (all are from $[0,1]^d$), their images lie inside a somewhat larger sphere around the small sphere with images from the feasible region. Thus, one can choose μ in a way that the larger sphere is literally squeezed onto the smaller one.

As a last step, one must look for the pre-image of $\tilde{\Psi}_x$ in order to finally get the wanted mapping to $\mathcal{F}_{[0,1]}$. As it is hardly possible to find the exact pre-image [SMB+99, KT04], one usually goes for an approximate pre-image whose image lies closest to the given image using an iterative procedure after [MSS+99].

$$x^*_{n+1} = \frac{\sum_{i=1}^{\ell}(\tilde{w}_i^{\Gamma_a} e^{-\|s_i - x^*_n\|^2/2\sigma^2} s_i)}{\sum_{i=1}^{\ell}(\tilde{w}_i^{\Gamma_a} e^{-\|s_i - x^*_n\|^2/2\sigma^2})}. \tag{9}$$

As an initial guess for x_0^* a good choice is to take the original point x and iterate it towards $\mathcal{F}_{[0,1]}$. Finally, x_n^* is the sought after image under mapping γ of x that lies in $\mathcal{F}_{[0,1]}$.

4 Indicator Integration

In general, with the term indicator we denote a parameter that characterizes a schedule with respect to the progress to some goal that we want to achieve (result of optimization) and that thus can be used by the optimizer to evaluate candidate solutions for comparison. Usually, the term performance indicator is used in industry [FG90] with a broader meaning. For simplicity we will use the term indicator for the rest of the paper and denote with indicator function an instruction to assign a real value to an indicator and not just a binary membership relation as sometimes in other use cases. In the following, we start by defining an indicator that characterizes a schedule:

Let $x \in \mathbb{R}^d$ be a solution candidate (not necessarily a feasible one) and $x_{[0,1]} \in [0,1]^d$ be its scaled variant from the scaled search space of agent A with feasible region $\mathcal{F}_{[0,1]}$. Then,

$$\begin{aligned} \mathcal{I}^{(K)}(x,S) : \mathbb{R}^d \times \mathbb{S} \to \mathbb{R}, \\ x \mapsto \mathcal{I}^{(K)}(x,S) \end{aligned} \tag{10}$$

is a mapping that unambiguously assigns a value for indicator K to schedule x based on the assumption that x is operated starting from the current operational state S of the unit.

Mapping

$$\begin{aligned} \mathcal{I}^{(K)}_{[0,1]}(x,S) : \mathbb{R}^d \times \mathbb{S} \to [0,1], \\ x \mapsto \frac{\mathcal{I}^{(K)}(x,S)}{max_{x\in\mathbb{R}^d, S\in\mathbb{S}}\{\mathcal{I}^{(K)}(x,S)\}}, \end{aligned} \tag{11}$$

maps to the scaled indicator $K_{[0,1]} \in [0,1]$. This will be helpful for the integration into scaled schedules in the model. Depending on the type of indicator and the actual problem formulation at hand, it might be useful to have the indicator value scaled by the maximum of all indicators resulting from only the current state $max_{x\in\mathbb{R}^d, S=S_0}$ instead of using the theoretical maximum of the unit.

In the support vector model, indicators can be integrated as follows: data vectors containing the mean power levels for the respective time intervals are extended by one element per indicator. Thus they become mixed feature vectors. This approach has been proposed for environmental performance indicators [Bre12], but in general, arbitrary indicators can be added in this way as long as a functional relationship exists between the power part of the vector and the value of the indicator.

In this way, we can build a modified sample $\mathcal{X}^*$ (cf. equation. 1) that integrates our model with indicator values and construct

$$\mathcal{X}^* = \{(x_1,\ldots,x_d,\mathcal{I}^{(K_1)}_{[0,1]}(x_1,\ldots,\boldsymbol{x}_d),\ldots,\mathcal{I}^{(K_m)}_{[0,1]}(x_1,\ldots,x_d))\}_n, \tag{12}$$

as n vectors $x \in [0,1]^{d+m}$ with the first d elements denoting scaled real power and m trailing elements denoting scaled indicator values. This sample is fed into exactly the same SVDD learning process to build the model and mapping γ for the decoder is derived in exactly the same way as described in section 3.3.

The decoder mapping γ then likewise maps feature vectors $x \in \mathcal{X}^*$

$$\begin{aligned} [0,1]^{d+m} &\to \mathcal{F}_{[0,1]} \times [\mathcal{I}_{[0,1]}]^m \\ \gamma(x) &\mapsto (x, \mathcal{I}^{K_1}_{[0,1]}, \ldots, \mathcal{I}^{K_m}_{[0,1]}). \end{aligned} \tag{13}$$

Indicator functions $\mathcal{I}$ are not necessarily known in their explicitly given formulation. We just demand that there exists some functional relation (maybe unknown or merely defined by some hidden variables) between schedule and indicator value. It might be possible to have the value merely determined by means of simulation. In this case, anyway, a means for learning the relation from the training data is necessary. The support vector method as described above, captures during the training process the region where the data resides in. If a relation exists between the data and therefor the region (as input domain to the indicator value function) and the indicator value, the sub-region containing the values (actually a manifold in data space) is likewise captured.

5 Results

The main goal of our method is the correct (or at least a good) reconstruction of indicator values assigned to schedules when exploring the search space model (as a substitute for the simulation model) in the course of some planning or optimization procedure. For this reason, we evaluate the reconstruction error when using our decoder method instead of the original calculation function.

We simulated a co-generation plant as energy unit with a thermal buffers store and a thermal demand derived from heat loss of a simulated house and hot water drawings. To the schedules derived from this simulation model, we added three indicators:

- Production cost: linearly estimated from primary energy usage and current (static) domestic gas prices.
- Buffer charging level after operating the schedule. This measure indicates how many degrees of freedom remain for successive planning periods. An empty or fully charged buffer allows only for a reduced set of choices.
- In order to have a somewhat complex indicator we used the Rosenbrock function $f_R(\boldsymbol{x}) = \sum_{i=1}^{d-1} \left((1-x_i)^2 + 100(x_{i+1} - x_i^2)^2\right),\ x \in \mathbb{R}^d$ as an artificial indicator for testing complexity, but without any practical meaning.

For each active power schedule from the sample, we calculated all three indicator values an add them to the respective vector. This new sample has then been used to derive the respective model and decoder mapping γ.

Table 1: Comparison of reconstruction errors of different indicators for several bandwidths (given as inverse value p) used for the kernel for encoding. Sample size was 300 15-minute-schedule for 2 hours. $\pm 0^*$ is below machine epsilon and therefor considered zero although some small value > zero is assigned.

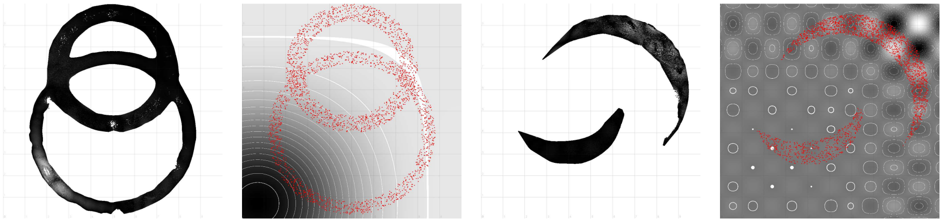
INV. BANDWITH (P)	BUFFER CHARGING LEVEL	COST	ROSENBROCK
0.1	1.825E-4 ± 5.261E-5	$\pm 0^*$	3.063E-1 ± 1.381E-2
1	1.825E-4 ± 5.223E-5	$\pm 0^*$	2.101E-1 ± 3.957E-2
5	7.543E-3 ± 1.923E-2	$\pm 0^*$	6.139E-2 ± 3.389E-2
10	9.788E-4 ± 5.287E-3	$\pm 0^*$	2.738E-2 ± 1.774E-2
20	1.608E-4 ± 6.26E-5	$\pm 0^*$	1.076E-2 ± 8.397E-3
30	1.649E-4 ± 5.972E-5	$\pm 0^*$	5.894E-3 ± 4.892E-3

Table 2: Comparison of the error for several schedule dimensions and a sample size of 300.

d	ROSENBROCK	BUFFER LEVEL
8	1.030E-4 ± 2.555E-4	1.592E-4 ± 7.773E-5
16	7.940E-5 ± 2.104E-4	1.652E-4 ± 6.948E-5
24	7.796E-5 ± 2.300E-4	1.46E-4 ± 7.282E-5
32	8.756E-5 ± 1.894E-4	1.517E-4 ± 7.549E-5
40	1.753E-4 ± 3.820E-4	1.643E-4 ± 7.542E-5
48	1.368E-4 ± 2.393E-4	1.513E-4 ± 7.188E-5
56	7.520E-5 ± 2.139E-4	1.535E-4 ± 7.011E-5
64	1.300E-4 ± 2.383E-4	1.441E-4 ± 7.260E-5
72	8.134E-5 ± 1.779E-4	1.561E-4 ± 6.855E-5
80	1.051E-4 ± 2.448E-4	1.461E-4 ± 7.433E-5
88	1.030E-4 ± 2.222E-4	1.515E-4 ± 7.330E-5
96	1.019E-4 ± 1.988E-4	1.5E-4 ± 7.721E-5

As a next step, we generated a set of random vectors $r_i \in [0,1]^{d+3} \sim U(0,1)^{d+3}$ and mapped them to feasible (in the sense of operable power levels) vectors $x = \gamma(r)$. The mapping automatically produces indicator values because these are likewise drawn towards the learned region or rather towards the learned shape of the function that describes the relation between schedule and indicator. Finally, we have vectors x that consist of feasible active power schedules $x_P = (x_1, \dots, x_d)$ and three reconstructed performance indicator values $x_K = (x_{d+1}, x_{d+2}, x_{d+3})$ that have been mapped from random values together with the active power part. Eventually, we can calculate the reconstruction error by using x_P for calculating the exact values – again with the function in the simulation model. These are compared with the reconstructed values (produced by the decoder) from x_K. Hence, the error is $\delta_j = \mathcal{I}^{K_j}_{[0,1]}(P_{max} \cdot x_{P_j}) - x_{K_j}$.

Table 1 shows the result of a comparison of the introduced indicators. All indicators have been calculated for 8-dimensional schedules (15-minute periods for 2 hours). Clearly, the smaller the bandwidth (and therefore the larger the proportion of information used for encoding the model) the better the reconstructed value. A significant decrease in reconstruction error can be especially observed for the most complex indicator: the artificial Rosenbrock indicator. The cost indicator with a value directly proportional to the power has an error below the machine epsilon of Java and is therefore considered zero. Table



(a) (b) (c) (d)

Figure 1: Dependency of the relative error size (1(a) and 1(c); brighter color denotes larger error) and derivation (1(b) and 1(d); denoted by thickness and distance of contour lines) of indicator value.

Table 3: Comparison of information increase by integrating indicators for several bandwidths (given as inverse value p) as the ratio between the number of support vectors necessary for encoding with and without indicators integrated.

INV. BANDWIDTH (P)	BUFFER CHARGING LEVEL	COST	ROSENBROCK
0.1	2.090E-1 ± 1.413E0	6.700E-2 ± 1.467E0	6.700E-1 ± 1.370E0
1	1.047E0 ± 1.861E0	1.850E-1 ± 1.824E0	2.020E-1 ± 1.790E0
5	6.433E0 ± 3.320E0	1.324E0 ± 3.215E0	3.713E0 ± 3.333E0
10	1.480E1 ± 5.080E0	3.3503E0 ± 5.016E0	7.690E0 ± 5.218E0
20	2.398E1 ± 7.015E0	5.068E0 ± 6.782E0	1.290E1 ± 6.989E0
30	2.639E1 ± 8.193E0	6.123E0 ± 8.152E0	1.412E1 ± 8.546E0

2 shows the relation of error size and schedule length. In order to test the relation between error and derivation of the indicator (with the assumption that regions with steeper changes in the indicator value exhibit larger errors), we used two further artificial indicator functions because they show a mixture of steep and flat regions: the rather smooth Branin and the many local optima Shubert function [MS05]. Figure 1 shows error plots (1(a) and 1(c)) resulting from reconstructing both artificial indicators (surface plots 1(b) and 1(d)) encoded together with two 2-dimensional artificial feasible regions (point clouds). The error plots exhibit larger errors in region with a steeper change in indicator values.

Finally, we tested the impact on the size of the model. For communication in an multi agent system, the amount of communicated information is another important issue. In our case, the number of support vectors necessary for encoding the model is an important criteria as it increases with the size and complexity of the feature space. For this reason, we tested how many additional schedules are necessary for the model when adding indicators. Table 3 shows the result. For each indicator we encoded 4-dimensional schedules without and with indicator (4+1-dimensional feature vector) and compared the mean ration between the number of support vectors. As expected, the number increases with complexity, but even for such short schedules (resulting in a respectively large share of indicators) the increase stays moderate.

6 Conclusion and Further Work

Many-objective optimization relies on the appropriate provision of indicators for the evaluation function. Within our smart grid use case, such indicators evaluate individual schedules that are offered as alternatives from different energy units to a real power planning algorithm. [Bre12] proposed the integration of performance indicators for denoting the environmental performance of different alternative schedules but with the disadvantage of only having them as a black box model. This model was only able to check the correct assignment to a schedule but did not pin how to get the correct values that are to be checked by the model and the proposed guessing method has some performance disadvantages. We extended this definition to general performance indicators and automatic reconstruction.

We have shown that an automatic reconstruction of the indicators from the model is possible when using the model as a decoder during optimization as proposed in [BS12]. In this way, all schedules do have sufficiently correct values for the attached describing indicators when an (evolutionary) search/ optimization algorithm scans through the feasible region while looking for appropriate solution candidates. Thus, we extended the decoder in a way that not only guides towards feasible but now also towards good solutions.

The next step will be to integrate the whole scenario to a distributed many-objective optimization procedure with indicators denoting cost of individual schedules as well as criteria on reliability or remaining degrees of freedom (in order not to shorten choices for successive planning periods) for a more robust planning of real power provision.

Acknowledgement

The Lower Saxony research network 'Smart Nord' acknowledges the support of the Lower Saxony Ministry of Science and Culture through the Niedersächsisches Vorab grant programme (grant ZN 2764).

References

[BBC+94] John W Bandler, Radoslaw M Biernacki, Shao Hua Chen, Piotr A Grobelny, and Ronald H Hemmers. Space mapping technique for electromagnetic optimization. *Microwave Theory and Techniques, IEEE Transactions on*, 42(12):2536–2544, 1994.

[BBC+95] John W Bandler, Radoslaw M Biernacki, Shao Hua Chen, Ronald H Hemmers, and Kaj Madsen. Electromagnetic optimization exploiting aggressive space mapping. *Microwave Theory and Techniques, IEEE Transactions on*, 43(12):2874–2882, 1995.

[BGKL11] Marita Blank, Sebastian Gerwinn, Olav Krause, and Sebastian Lehnhoff. Support Vector Machines for an efficient Representation of Voltage Band Constraints. In *Innovative Smart Grid Technologies*. IEEE PES, 2011.

[BHSHV01] Asa Ben-Hur, Hava T. Siegelmann, David Horn, and Vladimir Vapnik. Support Vector Clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[Bre12] Jörg Bremer. Ontology based Description of DER's Learned Environmental Performance Indicators. In Brian Donnellan, João Peças Lopes, João Martins, and Joaquim Filipe, editors, *Proceedings of the 1st International Conference on Smart Grids and Green IT Systems – SmartGreens 2012*, pages 107–112, Porto, Portugal, 04 2012. SciTePress.

[BRS10] Jörg. Bremer, Barbara Rapp, and Michael Sonnenschein. Support Vector based Encoding of Distributed Energy Resources' Feasible Load Spaces. In *IEEE PES Conference on Innovative Smart Grid Technologies Europe*, Chalmers Lindholmen, Gothenburg, Sweden, 2010.

[BS12] Jörg Bremer and Michael Sonnenschein. A Distributed Greedy Algorithm for Constraint-based Scheduling of Energy Resources. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Federated Conference on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proceedings*, pages 1285–1292, 2012.

[BS13] Jörg Bremer and Michael Sonnenschein. Constraint-handling for Optimization with Support Vector Surrogate Models – A Novel Decoder Approach. In J. Filipe and A.L.N. Fred, editors, *ICAART 2013 – Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, volume 2, pages 91–105, Barcelona, Spain, 2013. SciTePress.

[CC02] Carlos 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*, 191(11-12):1245–1287, January 2002.

[FG90] C.T. Fitz-Gibbon. *Performance Indicators*. BERA Dialogues Series. Multilingual Matters Limited, 1990.

[GZP03] Xiaohong Guan, Qiaozhu Zhai, and A. Papalexopoulos. Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. volume 2, page 1100 Vol. 2, 2003.

[JTD02] P. Juszczak, D.M.J. Tax, and Robert P. W. Duin. Feature scaling in support vector data description. In E.F. Deprettere, A. Belloum, J.W.J. Heijnsdijk, and F. van der Stappen, editors, *Proc. ASCI 2002, 8th Annual Conf. of the Advanced School for Computing and Imaging*, pages 95–102, 2002.

[KE09] Andreas Kamper and Anke Esser. Strategies for decentralised balancing power. In M. Randall A. Lewis, S. Mostaghim, editor, *Biologically-inspired Optimisation Methods: Parallel Algorithms, Systems and Applications*, number 210 in Studies in Computational Intelligence, pages 261–289. Springer, Berlin, Heidelberg, Juni 2009.

[Kim98] Dae Gyu Kim. Riemann mapping based constraint handling for evolutionary search. In *SAC*, pages 379–385, 1998.

[KM99] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evol. Comput.*, 7:19–44, March 1999.

[Kra10] Oliver Kramer. A review of constraint-handling techniques for evolution strategies. *Appl. Comp. Intell. Soft Comput.*, 2010:1–19, January 2010.

[KT04] J.T.Y. Kwok and I.W.H. Tsang. The pre-image problem in kernel methods. *Neural Networks, IEEE Transactions on*, 15(6):1517–1525, 2004.

[Lün12] Ontje Lünsdorf. *Selbstorganisation virtueller Geräte für das Lastmanagement von Kleinverbrauchern*. 2012. Dissertation.

[LV90] G. E. Liepins and M. D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, 2, 1990.

[MS96] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evol. Comput.*, 4:1–32, March 1996.

[MS05] M. Molga and C. Smutnicki. Test functions for optimization needs. Technical report, 2005. http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

[MSS+99] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus R. Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 536–542, Cambridge, MA, USA, 1999. MIT Press.

[MVO11] Radu-Casian Mihailescu, Matteo Vasirani, and Sascha Ossowski. Dynamic coalition adaptation for efficient agent-based virtual power plants. In *Proceedings of the 9th German conference on Multiagent system technologies*, MATES'11, pages 101–112, Berlin, Heidelberg, 2011. Springer-Verlag.

[NLT+12] Astrid Nieße, Sebastian Lehnhoff, Martin Tröschel, Mathias Uslar, Carsten Wissing, H.-Jürgen Appelrath, and Michael Sonnenschein. Market–Based Self–Organized Provision of Active Power and Ancillary Services. IEEE, 06 2012.

[RVRJ11] Sarvapali D. Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nick R. Jennings. Agent-based control for decentralised demand side management in the smart grid. In Liz Sonenberg, Peter Stone, Kagan Tumer, and Pinar Yolum, editors, *AAMAS*, pages 5–12. IFAAMAS, 2011.

[SAH+12] M. Sonnenschein, H.-J. Appelrath, L. Hofmann, M. Kurrat, S. Lehnhoff, C. Mayer, A. Mertens, M. Uslar, A. Nieße, and M. Tröschel. Dezentrale und selbstorganisierte Koordination in Smart Grids. In *VDE-Kongress 2012 Smart Grid Intelligente Energieversorgung der Zukunft*. VDE, 11 2012.

[SBP+] Rainer Stamminger, Gereon Broil, Christiane Pakula, Heiko Jungbecker, Maria Braun, Ina Rüdenauer, and Christoph Wendker. Synergy potential of smart appliances. A report prepared as part of the EIE project Smart Domestic Appliances in Sustainable Energy Systems (Smart-A) D2.3.

[SMB+99] 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*, 10(5):1000–1017, 1999.

[Squ75] W. Squire. Computer implementation of the Schwarz-Christoffel transformation. *Journal of the Franklin Institute*, 299(5):315–322, 1975.

[TD04] David M. J. Tax and Robert P. W. Duin. Support Vector Data Description. *Mach. Learn.*, 54(1):45–66, 2004.

[WLHK07] Horst F. Wedde, Sebastian Lehnhoff, Edmund Handschin, and Olav Krause. Dezentrale vernetzte Energiebewirtschaftung (DEZENT) im Netz der Zukunft. *Wirtschaftsinformatik*, 49(5):361–369, 2007.