Distributed Hybrid Constraint Handling in Large Scale Virtual Power Plants

Christian Hinrichs, Jörg Bremer, Michael Sonnenschein
Environmental Informatics, Department of Computing Science, University of Oldenburg, Germany
Email: {name}@informatik.uni-oldenburg.de

***Abstract*—In many virtual power plant (VPP) scenarios, numerous individually configured units within a VPP have to be scheduled regarding both global constraints (i.e. external market demands) and local constraints (i.e. technical, economical or ecological aspects for each unit). Approaches for global and local constraint handling have been discussed in the relevant literature independently. A hybrid approach is proposed that combines a decentralized combinatorial optimization heuristic with the encoding of individually constrained search spaces into unconstrained representations by means of support vector data description. The approach is applied to simulated VPP.**

I. Introduction

The transition of today's electricity grid to a decentralized smart grid is characterized by an increasing share of distributed energy resources (DER). In order to cope with stochastic feed-in effects (i.e. due to fluctuating meteorological conditions), an efficient management of the DER as well as further appliances has to be incorporated. However, these numerous *small active units* (generators, loads, storages) are individually configured, and operate rather dynamically in comparison to the classical large power plants. This implies an increase in complexity for the control of the system. Furthermore, due to privacy aspects as well as technical restrictions, a central control scheme may not be feasible any more. Thus, a paradigm shift to a decentralized system based on Information and Communications Technology (ICT) has been proposed repeatedly (c.f. [1] for an ongoing approach and further references).

In the contribution at hand, the problem of scheduling a pool of small active units, in order to jointly produce a predefined target power profile for a given planning horizon is considered. For example, such a task is present in the operational management of virtual power plants (VPP). Referring to [2], the *operation algorithm of VPP* includes means to adjust the supply of a VPP with regard to externally requested ancillary services. For this, an optimization process for unit commitment is employed. From a central point of view, the problem can alternatively be formulated as a combinatorial optimization problem: First the requested service (i.e. provisioning of active/reactive power) is translated into a desired target power profile for the VPP. Then a schedule has to be found for each unit within the VPP, such that the combination of all selected schedules yields the target power profile. However, a unit has additionally to obey individual operational constraints. These can be technically rooted hard constraints (e.g. minimal/maximal idle periods) as well as economically or ecologically rooted soft constraints. So the search for a combination of feasible schedules, which is optimal with respect to both the global target and these local constraints, has to be performed in a highly constrained search space. This yields a two-fold problem, comprising combinatorial aspects from a global point of view, and constraint handling aspects at a local level.

As stated in [2], larger VPPs require decentralized operation, which could be achieved by equipping the individual DER with decision making capabilities, thus forming a multi-agent system. This is in line with the call for a paradigm shift to decentralization as referenced in the first paragraph. Many problems solved by multi-agent systems are modeled as distributed constraint optimization problems (DCOP). According to [3], in a DCOP, *a number of independent agents each control the state of (a subset of) the variables in the system, with the joint aim of maximizing the global reward for satisfying constraints*. Based on the scope of constraints appearing in multi-agent systems, constraint handling approaches can be categorized into different lines of research.

A. Low-Arity (Inter-Agent) Constraint Handling

In the field of DCOPs, individual constraints usually affect only a small subset of agents. Typical DCOP approaches are therefore based on the representation of constraints in a graph, where each node describes an agent, and two nodes are connected by an edge if there is a constraint affecting these two nodes. If a constraint affects more than two nodes, some kind of relaxeration is usually performed. This graph then forms the communication network of the system. [3] gives an elaborate overview of approaches in this field.

B. High-Arity (Global) Constraint Handling

If individual constraints affect more than two or three agents, classical DCOP methods are not feasible any more. This is especially the case in (distributed) combinatorial optimization problems. In these, at least one constraint exists, which rewards combinations of values *globally*. If viewed from the DCOP perspective, this would yield a fully connected constraint graph. Since the action of a single agent in

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

Accepted for publication by the IEEE Power & Energy Society

this case affects many (all) other agents, the communication requirements in a distributed setting are enormous. Hence, approaches in this field focus on reducing the communication complexity. This can be done by means of a centralized information repository [4], by broadcasting [5], by using a hierarchical projection [6], [7] or by applying a self-organizing mechanism [8].

C. Intra-Agent Constraint Handling

Finally, constraints may additionally appear *within* single agents, affecting the decision making of these agents locally. Since these constraints are not of a distributed nature, they can be solved by the respective agents using central approaches. A widely and long since used approach for constraint handling is the introduction of a penalty into the objective function that devalues a solution that violates some constraint. 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 [9]. Another popular method treats constraints or aggregations of constraints as separate objectives, also leading to a transformation into a (unconstrained) many objective problem. A powerful, yet flexible way of constraint-handling is the use of a decoder that gives an search algorithm hints on where to look for feasible solutions [10].

In principle, the above outlined optimization problem of scheduling DER within a VPP might involve constraints from all three fields. As an example, power flow management covers field *A*, whereas the provisioning of balancing power usually involves field *B*. Constraints from field *C* will be present regardless the requested service. For the scope of this paper, and without loss of generality, the focus will be on fields *B* and *C*.

Recently, the Combinatorial Optimization Heuristic for Distributed Agents (COHDA) has been proposed, which solves the combinatorial problem of scheduling DER in a distributed, asynchronous way [8]. This is done by implementing a software agent for each unit, and letting the agents coordinate autonomously, thus forming a multi-agent system that meets the requirements given by [2]. However, the approach primarily focuses on the combinatorial part of the problem (constraints from field *B*). It therefore does not handle the operational constraints (field *C*) for each unit directly, but rather assumes unconstrained search spaces at agent level. For this, a simulation model has been employed in the referenced work, in order to generate samples of feasible schedules, which were then used as local search space for each unit, respectively. Although this eliminates the need to perform a constraint handling at agent level, such an approach clearly causes a reduction of the theoretically feasible search space. Upon investigating this property, it was found that the solution quality produced by the heuristic improves significantly with increasing sample size. Unfortunately, large sample sizes require an excessive amount of memory capacity. Also, this induces a high computational complexity for the agents in the process of schedule selection. Hence, an efficient method for searching in a constrained search space at agent level is necessary.

In the contribution at hand, an extension to the COHDA heuristic is proposed, that adds intra-agent constraint handling to the high-arity constraint handling that is already inherently present in the heuristic. For this purpose, the set of pregenerated schedules is replaced with a mathematical model of the feasible regions of the constrained search space. This yields an unconstrained representation that can directly be queried for matching feasible schedules. The approach is based on the encoding of distributed search spaces for virtual power plants, as introduced in [11]. The following section describes this method in more detail.

II. Support-Vector Surrogate Model

A. SVDD-Model for Feasible Regions

As a prerequisite, it is assumed that the feasible region of the problem has been encoded by support vector data description (SVDD) as e.g. described in [11]. The approach is recapped briefly. Given a set of schedules, the inherent structure of the region where they reside in is derived as follows: After mapping the data to a high dimensional feature space, the smallest sphere is determined that encloses all images. When mapping back the sphere to schedule space, its pre-image forms a contour (not necessarily connected) enclosing the sample.

This task is achieved by determining a mapping $\Phi : \mathcal{X} \subset \mathbb{R}^d \to \mathcal{H}, x \mapsto \Phi(x)$ 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. [12]) 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: $k_{\mathcal{G}}(x_i, x_j) = e^{-\frac{1}{2\sigma^2}\|x_i - x_j\|^2}$ [13].

With $k = k_{\mathcal{G}}$ the SVDD procedure yields 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:

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

Because all support vectors are mapped right 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 $\mathcal{F} = \{x \in \mathbb{R}^d | R(x) \leq R_{\mathcal{S}}\} \approx \mathcal{X}$, with $\mathcal{X}$ as the set of feasible schedules.

So far, such models have for example been used for efficiently communicating the feasible region of controllable energy resources in smart grid scenarios [11]. Only 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 has to be submitted. 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. Moreover, as the radius function (1) maps to $\mathbb{R}$, it allows for a conclusion about how far away a solution is from feasibility. Nevertheless, the model alone does not enable a systematic constraint handling during optimization. In the following, a more sophisticated way of integrating such SVDD surrogate models into optimization is developed.

B. The Decoder Approach

Let $\mathcal{F}$ denote the feasible region within the parameter domain of some given optimization problem bounded by an associated set of constraints. No assumptions are made on the constraints. It is known, that pre-processing the data by scaling it to $[0,1]^d$ leads to better adaption [14]. According to [11], some energy domain problems require a rescaling of the domain to $[0,1]^d$ for easier handling, too. For this reason, optimization problems with scaled domains are considered and the likewise scaled region of feasible solutions is denoted with $\mathcal{F}_{[0,1]}$. Now a mapping $\gamma : [0,1]^d \rightarrow \mathcal{F}_{[0,1]} \subseteq [0,1]^d;\ x \mapsto \gamma(x)$ is constructed to map the unit hypercube $[0,1]^d$ onto the d-dimensional region of feasible solutions.

This mapping is achieved as a composition of three functions:

$$\gamma = \tilde{\Phi_\ell}^{-1} \circ \Gamma_a \circ \hat{\Phi}_\ell. \tag{2}$$

Instead of trying to find a direct mapping to $\mathcal{F}_{[0,1]}$ the approach goes through the kernel space. The procedure starts with an arbitrary point $x \in [0,1]^d$ from the unconstrained d-dimensional hypercube and maps it 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 pre-image of the modified image is searched to get a point from $\mathcal{F}_{[0,1]}$. In the following, these steps are described briefly, for more details refer to [10].

1) Mapping to the SV induced subspace $\mathcal{H}^{(\ell)}$ with an empirical kernel map: Let $\hat{\Phi}_\ell$, defined as

$$\begin{aligned} \Phi_\ell : \mathbb{R}^d &\rightarrow \mathbb{R}^\ell, \\ x &\mapsto k(.,x)|_{\{s1,\ldots,s_\ell\}} \\ &= (k(s_1,x),\ldots,k(s_\ell,x)) \end{aligned} \tag{3}$$

be the empirical kernel map w.r.t. the set of support vectors $\{s_1,\ldots,s_\ell\}$. If Φ_ℓ is modified to

$$\begin{aligned} \hat{\Phi}_\ell :\ & \mathbb{R}^d \rightarrow \mathcal{H}^{(\ell)}\ , \\ & x \mapsto K^{-\frac{1}{2}}(k(s_1,x),\ldots,k(s_\ell,x)) \end{aligned} \tag{4}$$

with K as the kernel Gram Matrix, $K_{ij} = k(s_i,s_j)$, arbitrary points x, y may be mapped from $[0,1]^d$ to some ℓ-dimensional space $\mathcal{H}^{(\ell)}$ that contains a lower dimensional projection of the sphere. Points from $\mathcal{F}_{[0,1]}$ go onto or inside, others go outside the sphere.

2) Re-adjustment in kernel space: In the next step, the images from infeasible points from outside the sphere have to be pulled inside. This is done using

$$\tilde{\Psi}_x = \Gamma_a(\hat{\Psi}_x) = \hat{\Psi}_x + \mu \cdot (a - \hat{\Psi}_x) \cdot \frac{R_x - R_{\mathcal{S}}}{R_x} \tag{5}$$

to transform the image $\hat{\Psi}_x$ produced in step 1) into $\tilde{\Psi}_x \in \hat{\Phi}_\ell(\mathcal{F}_{[0,1]})$ by drawing $\hat{\Psi}_x$ into the sphere towards center a. Parameter μ allows us to control how far a point is drawn into the sphere ($\mu = 1$ is equivalent to drawing points right onto the sphere and therefore to the nearest feasible point, $\mu = R_x$ 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.

After this procedure $\tilde{\Psi}_x$ is the image of a point from $\mathcal{F}_{[0,1]}$ in terms of a modified weight vector $\tilde{w}^{\Gamma_a}$.

3) Finding an approximate pre-image: As a last step, the pre-image of $\tilde{\Psi}_x$ has to be found in order to finally get the wanted mapping to $\mathcal{F}_{[0,1]}$. As it is hardly possible to find the exact pre-image (c.f. [12], [15]), an approximate pre-image is searched, whose image lies closest to the given image using an iterative procedure after [16]. In the used case (Gaussian kernel), x^* is iterated to find the point closest to the pre-image and approximation $\tilde{\Phi_\ell}^{-1}$ is defined by equation

$$x^*_{n+1} = \frac{\sum_{i=1}^{\ell}(\tilde{w}_i^{\Gamma_a} e^{-\|s_i - x^*_n\|^2/2\sigma^2} s_i)}{\sum_{i=1}^{\ell}(\tilde{w}_i^{\Gamma_a} e^{-\|s_i - x^*_n\|^2/2\sigma^2})}. \tag{6}$$

As an initial guess for x^*_0 the original point x is taken and iterated towards $\mathcal{F}_{[0,1]}$. Finally, the mapping of an arbitrary point from $[0,1]^d$ into the region of feasible solutions described merely by a given set of support vectors and associated weights is achieved: x^*_n is the sought after image under mapping γ of x that lies in $\mathcal{F}_{[0,1]}$.

III. Evaluation

Originally, each agent in the COHDA heuristic was equipped with a limited set of pregenerated power profiles for the unit it represented (see [8] for details on the optimization process). Using the proposed extension, the agents now each include a SVDD-Model that represents the search space of feasible power profiles for the according unit. Using the decoder approach outlined in Section II-B, an agent is able to retrieve the nearest feasible schedule for an arbitrary target schedule from this model directly. Thus, each agent is able to search in an arbitrary large search space very efficiently. In order to determine the effect of extending the COHDA heuristic with SVDD-Models, the performance of both versions with respect to solution quality as well as run-time is compared. This is done using a small scale VPP composed of only few DER. Afterwards, the influence of larger population size (i.e. several hundred DER in a VPP) is examined. Finally, working memory requirements and computational complexity is considered.

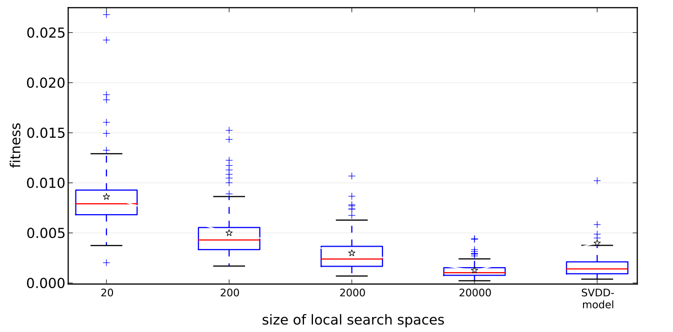

Fig. 1. Influence of the sample size of local search spaces on the fitness (normalized solution quality) in the COHDA heuristic (first four box-charts), compared to the results of the hybrid approach with SVDD-Model (rightmost box-chart).

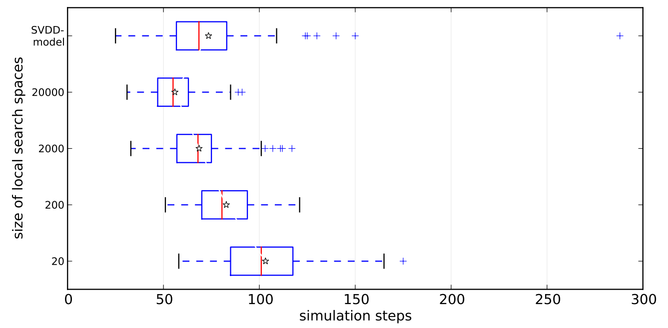

Fig. 2. Influence of the sample size of local search spaces on run-time (in simulation steps) in the COHDA heuristic (lower four box-charts), compared to the results of the hybrid approach with SVDD-Model (topmost box-chart).

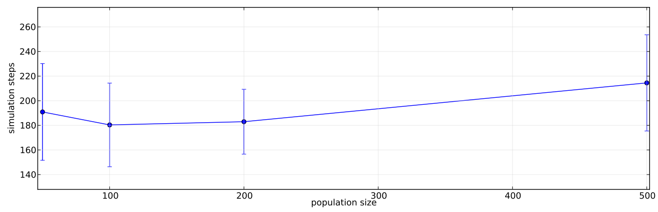

Fig. 3. Influence of the population size on run-time (in simulation steps) in the hybrid approach with SVDD-Model.

A. *Performance Comparison*

For identifying the effect of SVDD-Models on the heuristic, the performance of the original heuristic with regard to different sizes of enumerated search spaces is examined, and the findings are compared to the hybrid approach which is proposed in this contribution. The simulated VPP comprised 30 combined heat and power (CHP) devices with an $800\,l$ thermal buffer store each, using the same simulation model as in [8]. Figure 1 summarizes the main results as follows. Solution quality (vertical axis) is measured as absolute cumulated difference between the joint load profile produced by the finally selected schedules, and the target load profile. Hence, this is a minimizing fitness function for the optimization problem. In the figure, this has been normalized to the interval $[0.0,\ 1.0]$, where 0.0 is the optimal solution. Each evaluated configuration has been simulated 100 times. The results are visualized as box-charts, where the box spans from the upper to the lower quartile of the results. The median is shown as horizontal line within a box, whereas the whiskers span over $1.5\,\times$ the interquartile range. Additionally, the average is denoted with a star marker and outliers are illustrated by plus markers. The first four box-charts (from the left) show the results for simulations of the original COHDA heuristic with sample sizes $n_{sample} \in \{20, 200, 2000, 20000\}$ for the local search spaces. The rightmost box-chart represents the results for the hybrid approach with SVDD-Model. Obviously, the hybrid approach yields excellent solutions in comparison to the original heuristic.

However, due to the significantly larger search spaces, as well as some uncertainty introduced by the SVDD model, the run-time of the hybrid approach varies in a broader range than in the original heuristic with pre-sampled search spaces. Figure 2 shows this finding, using the same parameters as above. Note that, since the run-time strongly depends on the final fitness that is achieved by the approach, the simulation steps were counted only until the process reached a normalized fitness of 0.01. This way, the effect of larger search spaces being able to achieve a better fitness, and likewise spending more time on this, was eliminated.

B. *Large Scale VPPs*

As stated in the introduction, decentralized approaches are especially feasible in large scale scenarios. Hence, the behavior of the hybrid heuristic was analyzed with respect to population size. For these calculations, the simulation was stopped after reaching a normalized fitness of 0.01 for the same reason as in the previous paragraph. In [8], it was shown that in the original heuristic, run-time increases at most linearly with larger population sizes. With population sizes $n_{agents} \geq 100$, this effect could also be observed in the SVDD-equipped variant (see Figure 3). In order to determine the maximal performance of the approach regarding solution quality in large scale scenarios, the scenario with $n_{agents} = 500$ agents was considered in more detail subsequently. For this, the according simulation runs were analyzed without stopping them manually (and thus letting the heuristic run until termination, c.f. [8]). Then, a fitness of $5.6e{-}5 \pm 2.0e{-}5$ was achieved on average. The small scale scenario ($n_{agents} = 30$, as depicted in Figure 1) yielded a fitness of $3.9e-3\,\pm\,2.2e-2$. Hence, regarding the simulated setting, in large scale scenarios the maximal achievable fitness improves approximately by two orders of magnitude.

C. *Working Memory Requirements*

Given a planning horizon of q time intervals and n_{sample} pregenerated power profiles per unit, each agent would have to store $q \cdot n_{sample}$ floating point values in the classical approach. Equipping the agents with a SVDD-Model instead, the storage requirements are determined by the number of support vectors that define the search space model. This number is not fixed however, and depends on the structure of the search space (i.e. the amount of information to encode, which loosely depends on q). The experiments showed an average number of $n_{sv} = 25.37 \pm 3.04$ support vectors for the scenarios scrutinized in this paper, each comprising $q+1$ floating point values (incl. weight vector w, see Section II-A). Since solution quality in the classical approach strongly depends on n_{sample} (see Figure 1), one would usually choose $n_{sample} \ggg 20$. Following, the hybrid SVDD approach yields a reduction of space requirements from $q \cdot n_{sample}$ to $(q+1) \cdot 25.37 \pm 3.04$, which is several orders of magnitude expectedly.

D. Computational Complexity

Whenever an agent is notified by an event in its immediate vincinity, it performs a search for the best matching power profile regarding the current situation (c.f. [8]). For this, in the classical approach, an agent performs a sequential search in the set of pregenerated power profiles. Since the computational complexity of a sequential search is linear in the number of elements, this yields a computational complexity of $\mathcal{O}_{classic}\left(n_{sample} \cdot q\right)$ for a single solution search, where q denotes the length of each element in the set of pregenerated power profiles (i.e. the planning horizon).

In the case of using SVDD-Models one has to look at the complexity of the mapping. Decisive for the complexity of the mapping is the matrix vector multiplication growing quadratically with the number of support vectors n_{sv}. Another crucial question is the number of iterations i necessary for finding the pre-image with sufficient closeness. Empirically, during the experiments, a mean number of $i = 6.99 \pm 6.72$ iterations was observed, for instance, in order to to achieve a convergence $\|x^*_{m+1} - x^*_m\|_{\forall m>i} \leq \epsilon$ below an $\epsilon = 1e{-}7$ (see Section II-B), so the iterations can be neglected in complexity considerations. Furthermore, as n_{sv} loosely depends on q (see Section III-C), the computational complexity for the hybrid approach can be approximated by $\mathcal{O}_{hybrid}\left(n_{sv}^2 + i\right) \approx \mathcal{O}_{hybrid}\left(q\right)$.

IV. Conclusion

In the operational management of virtual power plants (VPP), the contained distributed energy resources (DER) have to be scheduled regarding a global goal (i.e. external market demands). Such a global constraint is accompanied with arbitrary individual constraints, that confine the feasible action space of individual units with regard to technical, economical or ecological aspects. The COHDA heuristic [8] solves combinatorial problems by handling global constraints in a decentralized way. On the other hand, the SVDD approach [11] is able to deliver an unconstrained representation from a highly constrained local search space. In the constribution at hand, the approaches are combined, which yields a hybrid heuristic that is able to handle both global and local constraints in a distributed combinatorial optimization problem. The approach is thus able to schedule the DER of a VPP with regard to an external global target schedule in a decentralized manner, while simultaneously respecting individual private constraints at unit level.

The evaluation shows that the hybrid approach yields excellent solutions regarding the fitness of the optimization. The run-time of the approach lies within the same range as the original COHDA heuristic. Furthermore, experiments with varying population sizes show that the run-time for reaching a specific solution quality scales linearly with the number of participating DER. The maximal achievable solution quality improves in large scenarios. Furthermore, space requirements for working memory as well as the computational complexity are reduced by several orders of magnitude.

References

[1] A. Nieße, S. Lehnhoff, M. Troschel, M. Uslar, C. Wissing, H.-J. Appelrath, and M. Sonnenschein, "Market-based self-organized provision of active power and ancillary services: An agent-based approach for Smart Distribution Grids," in *2012 Complexity in Engineering (COMPENG). Proceedings*. IEEE, Jun. 2012, pp. 1–5.

[2] K. Bakari and W. Kling, "Virtual power plants: An answer to increasing distributed generation," in *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*. IEEE, 2010, pp. 1–6.

[3] A. C. Chapman, A. Rogers, N. R. Jennings, and D. S. Leslie, "A unifying framework for iterative approximate best-response algorithms for distributed constraint optimization problems," *The Knowledge Engineering Review*, vol. 26, no. 04, pp. 411–444, 2011.

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

[5] Y. Penya, "Optimal Allocation and Scheduling of Demand in Deregulated Energy Markets," Ph.D. dissertation, Vienna University of Technology, 2006.

[6] E. Pournaras, M. Warnier, and F. M. Brazier, "Local Agent-based Self-stabilisation in Global Resource Utilisation," *International Journal of Autonomic Computing*, vol. 1, no. 4, p. 350, 2010.

[7] E. Pournaras, "Multi-level Reconfigurable Self-organization in Overlay Services," Ph.D. dissertation, Technische Universiteit Delft, 2013.

[8] C. Hinrichs, M. Sonnenschein, and S. Lehnhoff, "Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces," in *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 1 - Agents, Barcelona, Spain, 15-18 February, 2013*, J. Filipe and A. L. N. Fred, Eds. SciTePress, 2013.

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

[10] J. Bremer and M. Sonnenschein, "A distributed greedy algorithm for constraint-based scheduling of energy resources," in *Federated Conference on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proceedings*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2012, pp. 1285–1292.

[11] J. Bremer, B. Rapp, and M. Sonnenschein, "Encoding distributed search spaces for virtual power plants," in *IEEE Symposium Series in Computational Intelligence 2011 (SSCI 2011)*, Paris, France, 4 2011.

[12] 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.

[13] 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.

[14] 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.

[15] J. Kwok and I. Tsang, "The pre-image problem in kernel methods," *Neural Networks, IEEE Transactions on*, vol. 15, no. 6, pp. 1517–1525, 2004.

[16] S. Mika, B. Schölkopf, A. Smola, K. R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Proceedings of the 1998 conference on Advances in neural information processing systems II*. Cambridge, MA, USA: MIT Press, 1999, pp. 536–542.