Managing Energy in a Virtual Power Plant Using Learning Classifier Systems

Oliver Kramer, Benjamin Satzger, Jörg Lässig
International Computer Science Institute, Berkeley CA 94704, USA
{okramer, satzger, jla}@icsi.berkeley.edu
http://www.icsi.berkeley.edu/

***Abstract*—Many renewable energy resources like wind-turbines or solar collectors are volatile and unstable. Distributed energy producing entities are clustered to virtual power plants that – from a black-box perspective – should act like conventional stable power plants. It is the task of intelligent control strategies to compensate fluctuations and to exploit renewable recourses to a maximal extend. In this paper we present a simple simulation model of a virtual power plant that is based on real-world power consumption and wind production data. A power storage system allows to save overcapacities and to supply energy in case of demand bursts. A reserve power plant allows additional energy production to compensate shortcomings. The simulated power plant can easily be extended by parameterizable modules. We use a learning classifier system to evolve rules that control the energy storage system and the reserve plant to compensate energy fluctuations. A study of evolutionary parameters as well as a discussion of the evolved rules complement the experimental analysis. A simple demand side management example demonstrates the influence of pricing on the virtual power plant.**

I. Introduction

Renewable electric energy systems like wind mills, biomass or solar energy have an important part to play in modern energy grids. Renewable electric energy production moves today's energy grids from a centralized single supply system towards a decentralized bidirectional grid of suppliers and consumers. But renewable resources are often volatile and temporally unsteady. Their determining factors cannot be controlled, and are difficult to predict. To compensate instabilities and fluctuations in energy production, modules are clustered to virtual power plants. The movement towards renewable plants with higher degrees of efficiency, e.g., based on fuel cells, and towards large energy storage systems, increase the chances of successful virtual power plants that may be able to replace conventional producers. Energy management systems of the future will have the task to predict power demand and production of fluctuating producers like wind turbines or solar collectors, and to develop control strategies to balance power consumption and production under uncertainty.

In this paper we describe a virtual power plant simulator that models power consumption based on real-world data on the demand side, and various types of power producers based on parameterized entities. The simulation model can be used to learn energy system management strategies that compensate fluctuations, and to analyze the interplay between various types of controllers. Methods from computational intelligence (CI) turn out to be well appropriate to solve these problems. In Section II we summarize related work in the field of virtual power plant models and optimization. Section III concentrates on a description of the virtual parameterizable energy consumers and producers. We will learn controller strategies for energy storage systems and compensating power producers, as well as pricing for simple demand side management with a learning classifier system (LCS). The LCS will be described in Section IV, while the experimental results will be presented in Section V. The last Section VI summarizes the results and provides an outlook to future research directions.

II. Related Work

Distributed energy supply by renewable resources becomes more and more important. Willis and Scott [10] describe planning and evaluation of distributed power systems in detail. Davis [6] compares existing central station generation and distributed resources with regard to various aspects like liability, power quality, infrastructure requirements or electrical environmental effects. Based on this discussion he shows the significant advantages of distributed power systems such as micro-grids. The idea to cluster distributed generators to virtual power plants is discussed by Stothert *et al.* [16] or by Dondi *et al.* [7].

Some examples show that evolutionary algorithms (EAs) and other CI techniques can be successfully applied to problems in the field of renewable energy and smart grids. Mosetti *et al.* [15] optimized the wind turbine distribution of a wind farm in order to extract maximum energy for minimal installation cost. The optimization process is based on a wind farm simulation model subdividing an area into 100 square cells of possible turbine locations. Caldon *et al.* [5] have introduced a virtual power plant simulation model taking into account electric and heat production. They use a stochastic optimization algorithm and concentrate on objectives like minimization of short term variable production. Wang and Singh [2] address the problem to integrate different power sources like wind turbines, solar panels and storage batteries with regard to the objectives cost, reliability and pollutant emissions. To solve this multi-objective optimization problem, they introduce a multi-objective particle swarm optimization algorithm. Recently, Kramer *et al.* [13] introduced a memetic approach of evolutionary optimization and a local variant of kernel regression to forecast power consumption in smart grids.

III. THE VIRTUAL POWER PLANT MODEL

In the following, we describe the virtual power plant model. It is based on various entities modeling typical components, e.g., renewable power resources, an energy storage system, and an additional energy production entity that is able to compensate consumption bursts or a lack of renewable energy. Figure 1 illustrates the participating modules of the virtual plant. Consumers have to be supplied with energy. For this sake various energy resources are available. Wind mills play the role of the renewable volatile energy sources in our virtual plant. A storage system[1] stores overcapacities and is able to deliver stored energy with regard to an energy loss factor. A reserve plant can produce additional energy, and has the capacity to cover the whole energy demand. The total power of the system taking into account all consumers and producers at time t is denoted by $\eta(t) = \sum_i e_i(t)$, and is the sum of all producers with $e_p(t) \geq 0$ and consumers with $e_c(t) \leq 0$.

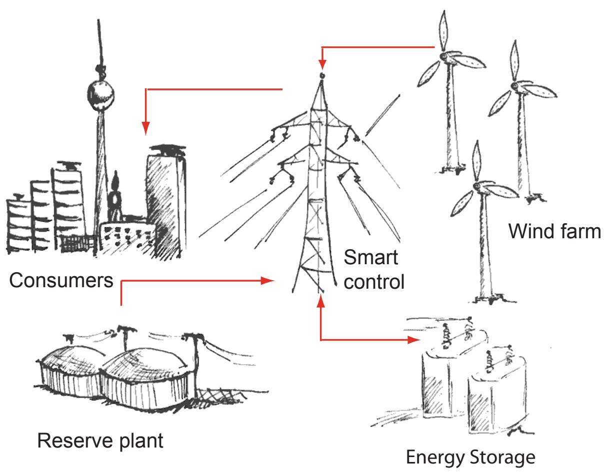

Fig. 1. Illustration of the virtual power plant model consisting of consumers that get their energy from a wind farm, a reserve plant, and an energy storage system.

A. Consumer Model

Consumers are modeled as follows. We assume that each entity $c \in \mathcal{C}$ in the set of energy consumers $\mathcal{C}$ consumes a certain amount of energy $e_c(t)$ at time t. To reflect a realistic consumption scenario, the basis of the energy consumption is real energy data $e_{\text{EG}}(t)$ measured and published by the Irish energy company EIRGRID [1]. Each day consists of 96 values measured in mega-watt (MW), i.e., one value every 15 minutes. Figure 2 shows the power consumption on two Sundays in February 2010[2]. To simulate fluctuations that may occur depending on consumer behavior and number of modules on the demand side, and to avoid an overadaptation, we model noise that may be added to the original data as follows:

$$e_c(t) = -e_{\text{EG}}(t) \cdot (1 + \gamma_1 \cdot \mathcal{N}(0,1)). \quad (1)$$

The noise strength can be controlled by parameter γ_1.

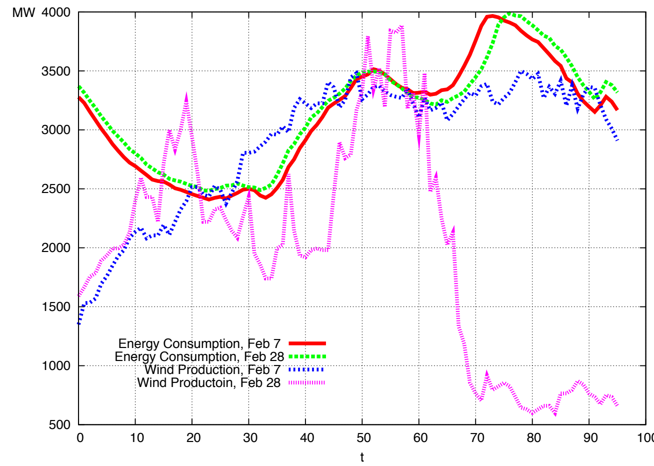

Fig. 2. Examples for energy consumption and wind data from the EIRGRID data source for the two Sundays Feb 7, and Feb 28 that will be subject to the experimental analysis in Section V. To allow interesting interactions between storage systems and reserve plant, the wind production is scaled into the range of the power consumption, see Section V-A.

B. Wind Energy

In our current virtual power plant model, wind is the renewable "fluctuating" resource we plan to concentrate on. The data for the produced wind energy is also taken from the EIRGRID data resources, i.e., from an EIRGRID wind farm. Figure 2 shows the wind production on Feb 7, and Feb 28, scaled by factors $\varphi = 5.0$, and $\varphi = 15.0$ to reach levels that allow interesting interactions between renewable energy production and consumption. We assume that module $w \in \mathcal{P}$ in the set of energy producers $\mathcal{P}$ generates a certain amount of energy $e_w(t)$ at time t. Again, we allow to add noise with strength γ_2 to the original data:

$$e_w(t) = e_{\text{WIND}}(t) \cdot (1 + \gamma_2 \cdot \mathcal{N}(0,1)). \quad (2)$$

Wind energy is volatile and its prediction is no easy undertaking. Wind power prediction can help to improve turbine control and the integration of wind power into the grid [8]. Various approaches have been introduced to predict the production of wind energy based on weather forecasts [14], [12]. According to Focken *et al.* [19] the prediction of energy produced by wind has sufficiently been solved. A recent discussion about the integration of wind energy has been published by Grant *et al.* [9]. Forecasting is a rather important aspect. To model uncertainty in wind prediction, we recommend two variants. First, the future time steps can be subject to an increased magnitude of noise $\hat{\gamma}_2$. Second, besides the actual energy that has been produced, EIRGRID has published an estimation of produced energy that can alternatively be used for prediction,

[1] or battery

[2] We have chosen February 7, and February 28, because of characteristic wind production on these days.

while the actual production can be used to determine the total energy $\eta(t)$ in the virtual power plant.

C. Energy Storage System

An energy storage system $s \in \mathcal{S}$ that is able to store overcapacities is an important part of smart virtual power plants. With an energy storage system overcapacities, e.g., in case of active wind, do not have to be wasted, but energy can be stored and retrieved to reduce additional power production, if the renewable resources are not able to cover the demand. We assume a storage system s with capacity $c_s(t)$ stored at time t and a maximum capacity of $(c_s)_{\max}$. We assume the actions *load*, and *deliver* with regard to parameter Δ_s. In case of action *load*, and $\eta(t) > 0$, the storage system is loaded by volume $\min(\Delta_s, \eta(t))$ with the constraint $c_s(t) \leq (c_s)_{\max}$, while $\eta(t)$ is decreased by the same amount. In case of $\eta(t) \leq 0$, no energy is stored. In case of $c_s(t) > 0$, and action *deliver*, the storage system delivers $c_s(t) = \min(\Delta_s, c_s(t))$. Furthermore, we assume that energy is lost over time with factor $0 < \gamma_3 < 1$. To model the energy loss, we assume that in each time step the stored energy is reduced by $c_s(t) = c_s(t-1) \cdot \gamma_3$. Table I shows a survey of all commands of the virtual power plant, also for the modules described in the following. Furthermore, the table shows typical parameterization that will be used in Section V.

TABLE I
SURVEY OF COMMANDS AND TYPICAL PARAMETERS THAT WILL BE USED IN SECTION V

Entity	commands			parameters
Storage	load	deliver	-	$(c_s)_{\max} = 2,000$
	+ 200	-200	-	$\gamma_3 = 0.9$
Reserve	increase	decrease	hold	$(e_r)_{\max}$
	+ 300	-300	0	$= 10,000$
Price	increase	decrease	hold	
	+0.1	-0.1	0	$0.8 \leq p \leq 1.2$

D. Reserve Plant

To cover energy demand in case of energy shortcomings and empty energy storages, we assume that an additional energy source can be activated, e.g., a CHP generation plant or a biogas plant. Such a device belongs to the set $\mathcal{P}$ of producers. We assume that reserve plant r can deliver energy $e_r(t) > 0$ at time t. In each step the plant can *increase* and *decrease* its energy production by Δ_r. Also here, we assume a maximum production capacity of $(e_r)_{\max}$. Usually, it is one goal to minimize the reserve energy production in order to reduce additional costs and CO_2 emissions.

E. Aggregation of Consumers and Producers

At each time step t the status of all components of the virtual power plant is updated. Then, the total energy $\eta(t)$ of the plant is computed by aggregating the energy of each entity, i.e., summing up the energy of all consumers $\mathcal{C}$, producers $\mathcal{P}$, and storage systems $\mathcal{S}$:

$$\eta(t) = \sum_{i \in \mathcal{C}, \mathcal{P}, \mathcal{S}} e_i(t). \tag{3}$$

In the scenario of compensating fluctuating energy, it is of interest to supply all consumers, while at the same time minimizing $\eta(t)$, i.e., to maintain $\eta(t) > \epsilon$ for a safety threshold ϵ that guarantees the supply of all consumers. In the following, we will evolve rules to achieve this goal using an LCS.

IV. THE LEARNING CLASSIFIER SYSTEM

The first LCS has been introduced by John Holland in the 1970s [11]. Learning classifiers are strongly connected to EAs. They are rule based systems, automatically building a rule set for classification, regression, reinforcement learning, and general prediction problems. Their scope is to build and manipulate a compact set of rules or classifiers with generalization abilities. LCS can be divided into Michigan-style and Pittsburgh-style classifiers. Michigan-style LCS encode one complete problem solution in a large population with many local rules. An individual represents a single rule. This rule is evolved by genetic operators and evaluated competitively and individually. Usually, a Michigan-style LCS learns iteratively from single problem instances. In a Pittsburgh-style LCS a single individual encodes an entire problem solution with a whole set of rules. It evolves the whole rule sets by means of genetic operators. In contrast to the Michigan-style LCS, a Pittsburgh-style LCS learns offline from sets of problem instances. We will use a Pittsburgh-style LCS in the following.

1) LCS architecture: Our LCS consists of a set $\mathfrak{R} = \{r_1, \ldots, r_N\}$ of rules r_n, $1 \leq n \leq N$. Each rule r_n consists of a conditional part χ_n describing the state of the system, and an action part α_n determining the commands for the power plant's future activities. Rule $r_n = (\chi_n, \alpha_n)$ is interpreted as:

$$r_n : \quad \text{IF } \chi_n \text{ THEN } \alpha_n. \tag{4}$$

Let s_t be the actual state of the power plant. It consists of $\mathcal{T}$ parts, corresponding to the discretization of the current total power of the system $s_1 = \eta(t)$ and the predicted power for the next $i = 1, \ldots \mathcal{T}-1$ time steps $s_i = \eta(t+i)$. The rule $r_j \in \mathfrak{R}$ with the most similar conditional part χ_j is chosen, i.e., for a minimal $d(\chi_j, s_t)$. For measure $d(\cdot)$ we use the Manhattan distance. If A is the number of controllable entities, we assume that each command α_n consists of $\mathcal{T} \cdot A$ actions, i.e., for each future time step $t+i$ each controllable entity j performs action a_{ij}.

2) Population Scheme and Mutation: At the beginning, all rules are randomly initialized. We apply a $(\mu + \lambda)$-population scheme, oriented to evolution strategies [3]. In each generation λ solutions are produced by randomly selecting one of the μ parent solutions, and applying the mutation operator. In the experiments in Section V we will use $\mu = 10$ parents, and $\lambda = 50$ offspring solutions. Mutation works as follows. In the course of the stochastic optimization process, a rule is selected randomly with probability σ_1, and each component is mutated with probability σ_2. On the component level, mutation means that each component is randomly chosen from the set of possible values, e.g. actions $\mathcal{A} = \{-1, 0, +1\}$. In each

TABLE II

STUDY OF MUTATION STRENGTHS. THE FIGURES SHOW THE FITNESS VALUES (SEE EQUATION 5) ACHIEVED AFTER $t_{\max} = 50$ GENERATIONS OF THE LCS WITH VARIOUS COMBINATIONS OF MUTATION STRENGTHS, I.E., FOUR COMBINATIONS FOR $\sigma_i = 0.1, 0.01$, AND FOUR WITH $\sigma_i = 0.1, 0.2$.

	σ_1 =0.01					σ_1 =0.1				
	best	median	worst	mean	dev	best	median	worst	mean	dev
σ_2 =0.01	23,066	28,902	33,535	28,858	3,019.9	19,523	24,551	27,961	24,093	2,608.9
σ_2 =0.1	18,235	24,741	31,585	24,730	3,883.1	14,599	18,612	21,922	18,827	2,085.2
	σ_1 =0.1					σ_1 =0.2				
σ_2 =0.1	14,599	18,612	21,922	18,827	2,085.2	15,098	17,672	20,338	17,770	1,555.5
σ_2 =0.2	15,956	18,469	21,312	18,297	1,400.5	15,396	18,006	19,552	17,724	1,401.6

iteration of the LCS a population of λ rule bases, each consisting of N rules are generated with regard to the described stochastic modifications. Each rule set is tested independently and causes a whole run of the virtual power plant. Each rule set is assigned to a user-defined quality measure f. In the experiments presented in the next section, the goal will be to minimize η. In the following generation, the μ-best rule bases are selected and are the basis for generation of the next λ solutions. The optimization process is repeated until a user-defined termination condition is reached.

V. EXPERIMENTAL ANALYSIS

In this section we present experimental results of our LCS. First, we concentrate on a parameter study, then we illustrate the activities of an evolved rule set. We conduct experiments that analyze the ability to generalize, i.e., to show satisfying results in unknown situations. Last, we extend the system by a simple demand side management mechanism.

A. Parameter Study

At the beginning, the question arises how to choose the parameters for the LCS. To answer this question, we have conducted a series of experiments that are presented in the following. The experiments are focused on training the rules for one particular day. Basis of our parameter analysis is the power consumption and wind energy production on February 7, 2010. For the experiments, we scale the wind energy production into the range of the power consumption. The wind energy of Feb 7 is multiplied with factor $\varphi = 5.0$, while the produced wind energy of Feb 28 is scaled by $\varphi = 15.0$ to make the situation more dynamic. Without scaling the energy production, no overcapacities that are very interesting for interactions with the storage system will occur. The noise parameter for consumers is set to $\gamma_1 = 0.01$, the noise for producers is set to $\gamma_2 = 0.01$. We choose the energy storage loss factor $\gamma_3 = 0.9$

The optimization goal is to minimize the absolute sum of energy in each time step, i.e., the sum of energy produced by the producers minus the energy demand of all consumers in all times steps:

$$f = \sum_{t=1}^{T} |\eta_t| \rightarrow \min . \tag{5}$$

In practice, a safety margin ϵ of energy overcapacity will be necessary to guarantee that the energy demand never exceeds the sum of energy in the system. But for our toy example we assume that the condition of Equation 5 is sufficient.

All experiments of the parameter studies use a (10+50)-EA, and terminate after $t_{\max} = 50$ generations. At first, we concentrate on the influence of mutation strengths σ_1 and σ_2. We use 15 states to describe each power situation, and 20 rules. Table II shows the corresponding experiments. Each trial has been repeated 15 times. The figures show the best, median, worst, and mean result as well as the corresponding standard deviation of the achieved fitness until the termination condition has been reached. The results show a clear tendency towards higher mutation strengths. The overall best fitness has been achieved for $\sigma_1 = 0.1$, and $\sigma_2 = 0.1$, while the best mean value has been achieved for $\sigma_1 = 0.2$, and $\sigma_2 = 0.2$. The best median has been achieved with settings $\sigma_1 = 0.2$, and $\sigma_2 = 0.1$. We have to point out that a non-parametric Wilcoxon test has not shown significant superiority for the latter variants. But any setting with $\sigma_i = 0.2$ is significantly better than any setting with $\sigma_i = 0.01$. For the following experiments, we use the setting $\sigma_1 = 0.1$, and $\sigma_2 = 0.1$ that has also shown stable results in further experiments.

An important parameter of LCS is the size of rule set $\mathfrak{R}$. In the next experiment, we vary the rule set size and test the settings $N = 10, 20, 30, 40$. Table III shows the outcome of this series of experiments. The results show an increase of variance with a decrease of $\mathfrak{R}$. On the one hand, for $N = 10$ the worst median and mean can be reported, but on the other hand, the same setting achieved the best overall result. The high standard deviation of $dev = 3,336.8$ confirms this result, in comparison to $dev = 1,777.7$ in case of $N = 40$. As no statistical significance can be observed, when increasing the rule set size to values higher than $N = 20$, we apply this rule set size in the following experiments, and it confirms the choice for the previous mutation strengths analysis.

TABLE III

STUDY OF THE NUMBER OF RULES

	best	median	worst	mean	dev
10	14,059	19,426	29,159	19,911	3,336.8
20	14,599	18,612	21,922	18,827	2,085.2
30	14,785	18,361	22,014	18,419	1,838.4
40	15,609	18,887	21,722	18,808	1,777.7

B. Illustration of Activities

To illustrate the behavior of the virtual power plant evolved by the LCS, we discuss the behavior of typical trained rule sets. Figure 3, upper part, shows the activities of the power plant's modules after training on the data sets of Feb 7 (left upper part), and Feb 28 (right upper part). First, we can

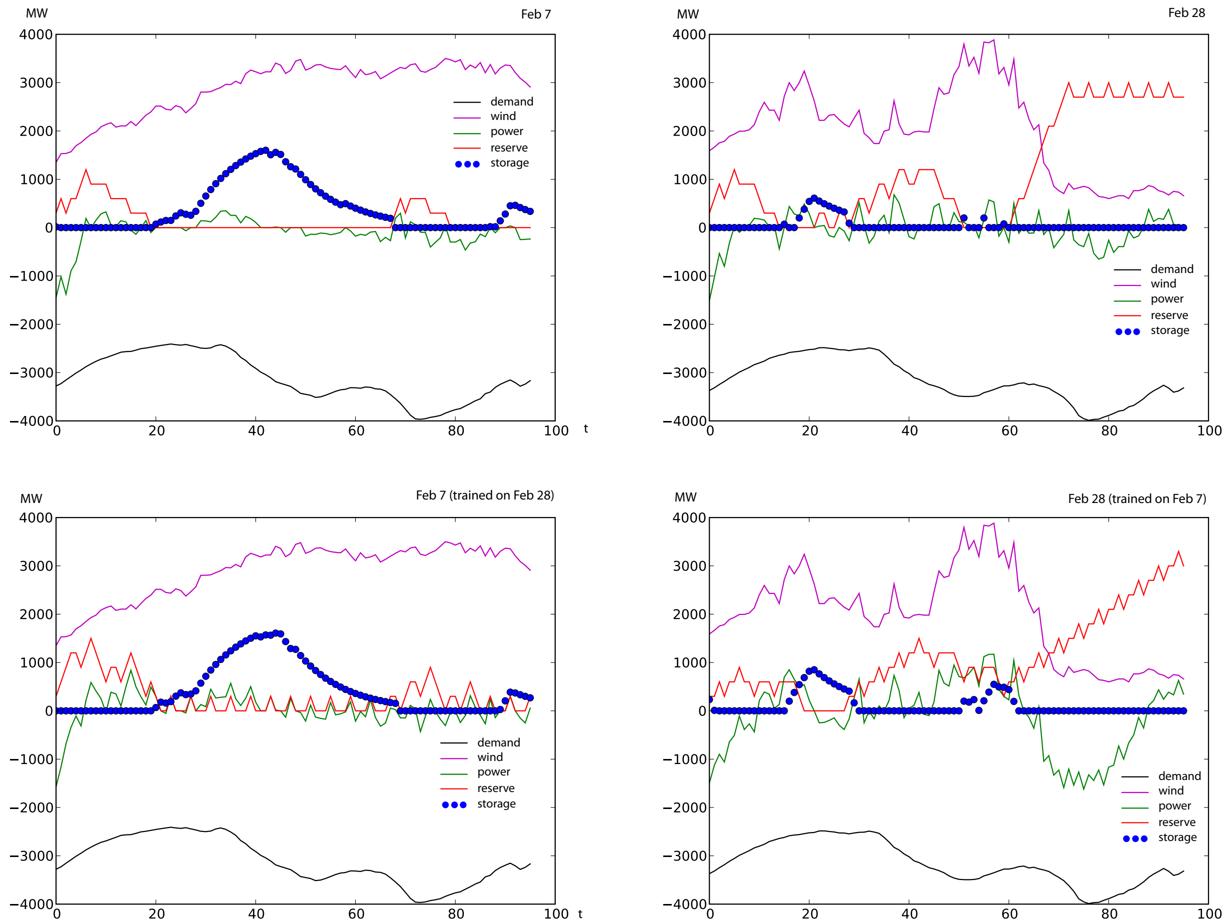

Fig. 3. Activities of the simulated virtual power plant after $t_{\max} = 200$ generations of learning on Feb 7 (left upper part), and on Feb 28 (right upper part). The lower part shows the behavior of the rule sets on both days, but each evolved on the other day.

observe that the overall energy in the system (green line) is fluctuating around the zero-point, i.e., the objective has been reached to avoid overcapacities while maintaining supply of the energy consumers. If we have a closer look at how this objective has been reached, we can observe that the reserve plant, which is starting with zero energy at the first time step, is increasing its production at the beginning to compensate the energy deficit at the start[3]. On Feb 7, the wind is blowing more and more. Hence, the reserve plant can close down. Later, when the demand is decreasing, overcapacities are produced that are saved by the storage system. The energy storage is loaded to approximately $1,500$ MW and afterwards, when the demand is increasing, it is delivering energy to the system. The reserve plant does not have to be run up in this situation. As soon as the storage is empty, the reserve plant has to help to cover the demand again. At the end, overcapacities can be stored again by the energy storage system.

On Feb 28, the power consumption is similar, but the wind is more dynamic. As a consequence, it is more difficult for the virtual plant to compensate the overall energy fluctuations in the system. Again, the reserve plant has to compensate the energy deficit at the beginning. When the wind energy reaches a peak, the reserve plant has learned to power down. Only little overcapacities are produced, but are stored by the battery at once. When the energy demand is increasing in the second quarter, the reserve plant has learned to increase its production, as the wind level does not allow to cover the demand side. Then, the wind is increasing, but also the power demand. In the last third the wind is decreasing, and the reserve plant increases its production.

[3]Of course, in real-world systems a deficit at the beginning of each day can be avoided in a close-loop day-to-day model.

To summarize, the LCS was able to evolve rules that manage the storage systems and the reserve plant. Although the latter is equipped with a comparatively high production capacity, it can be observed that the solely objective to minimize overcapacities while covering the demand is sufficient to minimize the reserve energy production at the same time. Figure 4 shows the corresponding fitness development of Feb 7 (red line) and Feb 28 (blue dots) that have led to the discussed rule bases. The fitness developments show that most of the evolutionary process, i.e., the highest fitness achievements, takes place within the first 120 generations. This observation confirms the choice of the termination condition, i.e., terminating after $t_{\max} = 200$ generations.

C. Testing Rule Sets in Unknown Situations

In the previous parts of the experimental analysis we concentrated on the behavior of the system with regard to the training set. Now, we will analyze the ability of the evolved rule sets to control the virtual plant in unknown situations. For this sake we try two settings. We let the rule set, evolved on Feb 28, control the situation in Feb 7, and in turn, test the rule set of Feb 7 to control day Feb 28. Figure 3, lower part, shows the corresponding experimental results. The rule set evolved on Feb 28 shows excellent results in controlling the virtual power plant on Feb 7. Reserve plant and storage system are controlled in the same way like in Figure 3, upper part. The rule set evolved on Feb 7 shows worse results on Feb 28. The power fluctuations are higher within the whole control process. In particular in the last third of Feb 28 the rule base is not able to compensate the power shortcomings caused by an increasing demand and a decreasing wind generation. The reason for this is obvious: the situation that the wind decreases while the demand is high does not occur on Feb 7. Hence, no rule has been evolved that is able to handle this situation.

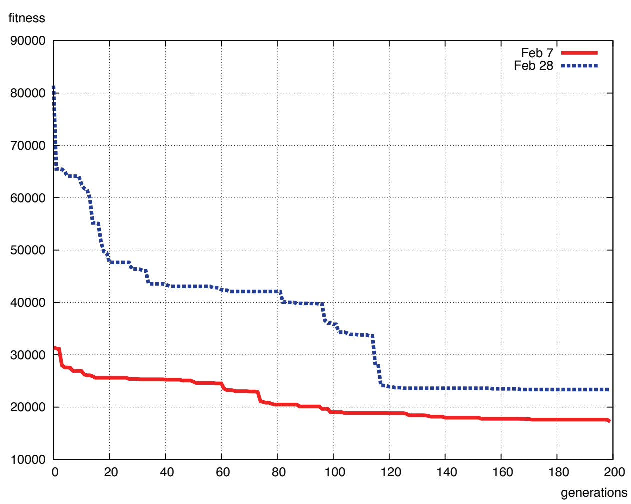

Fig. 4. Fitness development of the (10,50)-LCS for 200 generations during the optimization of the rule sets for the two days shown in Figure 3.

D. Demand Side Management

Besides the prediction of power consumption, demand side management has an important part to play in virtual power plants, i.e., the influence of consumption to avoid bursts and make consumption in case of energy overcapacities more attractive [4], [17]. We have implemented a simple demand side management model considering two aspects. We assume that the power consumption depends on pricing, and on urgency. To model this, we extend Equation 1 by price- and urgency-dependent variables, i.e., we assume price $p(t)$ with $0.8 \leq p(t) \leq 1.2$ and express urgency $u(t)$:

$$\hat{e}_c(t) = \prod_{x=e_c(t),p(t),u(t)} x. \tag{6}$$

The urgency $u(t)$ is modeled by the fraction of the sum of energy that would usually have been consumed determined by Equation 1 and the sum of consumed energy in the past:

$$u(t) = \left(\frac{\sum_{q=1}^{t-1} e_c(q)}{\sum_{q=1}^{t-1} \hat{e}_c(q)} \right)^{\gamma_4}. \tag{7}$$

Parameter $\gamma_4 \geq 0$ allows to adjust the magnitude of influence of the urgency in Equation 6. Our first tests have revealed interesting effects. On Feb 7 this demand side modeling leads to a decrease of consumed energy (with $\gamma_4 = 2.0$) by increasing the price to a maximum of $p(t) = 1.2$. The total energy consumption deficit over the whole day was $21,330.5$ MW, but the achieved fitness was $f = 13,319.5$. For higher settings, e.g. $\gamma_4 = 5.0$, no significant deviation of the demand curve could be observed. On Feb 28 the demand curve was changed (using $\gamma_4 = 2.0$), leading to a demand burst by decreasing the price to a minimum $p(t) = 0.8$ in the middle of the day, i.e., when the wind production has reached a maximum. The total consumption even increased by 561.5 MW. The other parts of the demand curve have been flattened. A detailed analysis of the interactions between pricing and the modeled entities will be subject to future work.

VI. Summary and Outlook

Virtual power plants have an important part to play in the development towards smart energy grids. We have presented a simple toy simulation model that allows to model important entities like energy storage systems, reserve power plants, and demand side management. Furthermore, we have shown, how an LCS is able to evolve rules with the goal of compensating fluctuations while maintaining energy supply to cover the total energy demand. The introduced power plant model may look like a toy problem, and – in the form used for our first experimental analysis – be also analytically solvable. But the experiments of Section V have only been a first attempt to show the potentials of the model, and to demonstrate how an LCS is able to solve it. By introducing multiple modules with different parameterization, it is scalable to a complex model that is not analytically solvable anymore. For these situations the rules evolved for simpler models can be used as the basis for learning and optimization.

In the future we are going to concentrate on various directions. First, we plan to improve the simulator by adapting its components to observed real-world data. The usage of the EirGrid data sources have only been a first attempt into this direction. We will integrate various new renewable energy modules like solar energy, water power or biomass that can arbitrarily be aggregated to virtual plants. Furthermore, we will concentrate on demand side management and pricing strategies, and analyze their interactions with other entities of the virtual power plant in detail.

It turns out that virtual power plants are an excellent testbed for CI methods. We are going to implement and compare further CI methods, in particular concentrating on the LCS variant XCS by Wilson [18] and various reinforcement learning variants like Q-learning or TD(λ) that allow online-adaptation to feedback. Furthermore, we will compare our results to analytical methods, and elaborate the boundary where analytical models may fail and heuristics have to be used instead.

ACKNOWLEDGEMENT

The authors would like to thank the German Academic Exchange Service (DAAD) for funding their research. Furthermore, we thank EIRGRID for the permission to use their consumption and wind production data.

REFERENCES

[1] Download centre EIRGRID. http://www.eirgrid.com/.

[2] L. Wang and C. Singh. PSO-Based Multi-Criteria Optimum Design of A Grid-Connected Hybrid Power System With Multiple Renewable Sources of Energy. In *Swarm Intelligence Symposium SIS*, pages 250–257, 2007

[3] H.-G. Beyer and H.-P. Schwefel. Evolution strategies - A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[4] F. Born. *Aiding Renewable Energy Integration through Complementary Demand-Supply Matching*. PhD thesis, Glasgow, University of Strathelyde, 2001.

[5] R. Caldon, A. R. Patria, and R. Turri. Optimal control of a distribution system with a virtual power plant. In *Bulk Power System Dynamics and Control - VI*, pages 278–284, 2004.

[6] M. W. Davis. Distributed resource electric power systems offer significant advantages over central station generation and T&D power systems. In *IEEE Power Engineering Society Summer Meeting*, pages 153–160, 2002.

[7] P. Dondi, D. Bayoumi, C. Haederli, D. Julian, and M. Suter. Network integration of distributed power generation. *Journal of Power Sources*, 106(1-2):1–9, 2002.

[8] G. Giebel, R. Brownsword, and G. Kariniotakis. State of the art on short-term wind power prediction. Technical report, ANEMOS, 2003.

[9] W. Grant, D. Edelson, J. Dumas, J. Zack, M. Ahlstrom, J. Kehler, P. Storck, J. Lerner, K. Parks, and C. Finley. Operational challenges in wind-power production and prediction. *IEEE Power and Energy*, 7(6):131–145, 2009.

[10] W. G. S. H. L. Willis. *Distributed Power Generation Planning and Evaluation*. Marcel Deker Edition, Amsterdam (NL), 2000.

[11] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[12] R. Jursa. Variable selection for wind power prediction using particle swarm optimization. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 2059–2065, New York, NY, USA, 2007. ACM.

[13] O. Kramer, B. Satzger, and J. Lässig. Power prediction in smart grids with evolutionary local kernel regression. In *Hybrid Artificial Intelligence Systems, 4th International Conference, HAIS 2010*, in print, 2010.

[14] A. Kusiak, H. Zheng, and Z. Song. Wind farm power prediction: a data-mining approach. *Wind Energy*, 12:275–293, 2009.

[15] G. Mosettia, C. Poloni, and B. Diviaccoa. A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews*, 51(1):105 – 116, 1994.

[16] A. Stothert, O. Fritz, and M. Sutter. Optimal operation of a virtual utility. In *International Symposium on Distributed Generation: Power System and Market Aspects*, pages 11–13, 2001.

[17] U. Vogel and M. Sonnenschein. Optimization of adaptive consumers to a time-varying electricity supply. In *ITEE*, pages 119–131, 2007.

[18] S. W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.

[19] S. W. Wilson. Short-term prediction of the aggregated power output of wind farms - a statistical analysis of the reduction of the prediction error by spatial smoothing effects. *Journal of Wind Engineering and Industrial Aerodynamics*, 90:231–246, 2002.