ORIGINAL PAPER

Computational intelligence based optimization of hierarchical virtual power plants

Stefan Rädle[1] · **Johannes Mast[1]** · **Joachim Gerlach[1]** · **Oliver Bringmann[2]**

© The Author(s) 2020

Abstract
In the context of renewable energy sources, virtual power plants (VPP) are regarded as a key technology for an intelligent control of the complex, decentralized, distributed and heterogeneous power generation process. However, an economic and ecological control of a VPP turns out to be a highly critical task: due to the strongly varying characteristics of VPPs, in terms of complexity, technology mix, environmental conditions and target objectives to be optimized during operation, the control of an individual VPP needs to be able to effectively take into account all of those individual constraints. Therefore, we propose in this paper an abstract control methodology for a VPP in combination with computational intelligence (CI) metaheuristics, which is designed to be flexibly applicable for different VPP sizes, target objectives and power plant types. The methodology furthermore provides the possibility to build hierarchical VPPs as they are often demanded by the system operators. To demonstrate the effectiveness of the control methodology, three exemplary optimization targets are considered and applied to different compositions of flat/hierarchical VPPs: the minimization of operating reserve requirements, the minimization of CO_2 emissions and the maximization of the power plant flexibility. Furthermore, the methodology is combined and evaluated with three exemplary CI metaheuristics: simulated annealing (SA), particle swarm optimization (PSO) and ant colony optimization (ACO). To legitimize the use of such advanced CI metaheuristics for the optimization problem, gradient descent optimization (GDO) as a traditional optimization technique is regarded as well. On the basis of concrete example scenarios as well as extensive, aggregated test runs, the results show that the control methodology is capable of efficiently optimizing various compositions of VPPs towards the given objectives.

Keywords Metaheuristic optimization · Renewable energy integration · Distributed power generation · Control optimization

Extended author information available on the last page of the article

Abbreviations

ACO	Ant colony optimization
CHP	Combined heat and power unit
CI	Computational intelligence
GDO	Gradient descent optimization
IPCC	Intergovernmental panel on climate change
PSO	Particle swarm optimization
SA	Simulated annealing
VPP	Virtual power plant

Methodology symbols

$P_{i,min}(t)$	Theoretical minimum power limit of sub-plant i at time point t
$P_{i,max}(t)$	Theoretical maximum power limit of sub-plant i at time point t
$\tilde{P}_{i,min}(t)$	Actual minimum power limit of sub-plant i at time point t
$\tilde{P}_{i,max}(t)$	Actual maximum power limit of sub-plant i at time point t
$P_{i,s}(t)$	Step size of sub-plant i at time point t
$C_i(t)$	Specific CO_2 emissions of sub-plant i at time point t
$k_i(t)$	Control coefficient of sub plant i at time point t
$P_i(t)$	Power output of sub-plant i at time point t
$\mathbf{P_s(t)}$	Step size vector for a VPP at time point t
$\mathbf{k(t)}$	Control coefficient vector for a VPP at time point t
$P(t)$	Power output of a VPP at time point t
$P_{VPP,min}(t)$	Theoretical minimum power limit of a VPP at time point t
$P_{VPP,max}(t)$	Theoretical maximum power limit of a VPP at time point t
$\tilde{P}_{VPP,min}(t)$	Actual minimum power limit of a VPP at time point t
$\tilde{P}_{VPP,max}(t)$	Actual maximum power limit of a VPP at time point t
$P_{VPP,s}(t)$	Step size of a VPP at time point t
$C_{VPP}(t)$	Specific CO_2 emissions of a VPP at time point t
$L(t)$	Consumer load demand at time point t
$O(t)$	Error function for the operating reserve objective
$M(t)$	Total CO_2 emissions of a VPP at time point t
$NM(t)$	Error function for the CO_2 emissions objective
$F_i(t)$	Flexibility potential of sub-plant i at time point t
$F(t)$	Error function for the flexibility objective of a VPP
W_O	Weighting factor for the operating reserve objective
W_M	Weighting factor for the CO_2 emissions objective
W_F	Weighting factor for the flexibility objective
$E(t)$	Error function to be optimized by the control methodology

SA symbols

ϑ_{init}	Initial temperature
ϑ_j	Temperature value after j iterations
ϑ_{min}	Minimum temperature value
α	Exponential cooling factor for the temperature value

PSO symbols

$x_i(j)$	Position of particle i at iteration j
$\hat{x}_i(j)$	Best found state for particle i until iteration j
$g(j)$	Globally best found state until iteration j
$v_i(j)$	Velocity of particle i at iteration j
w	Weighting factor for the current particle velocity
c_1	Weighting factor for the particle best found state velocity
c_2	Weighting factor for the globally best found state velocity
r_1	Random factor for the particle best found state velocity
r_2	Random factor for the globally best found state velocity

ACO symbols

e_{kl}	Directed edge leading from node k to node l in the graph
τ_{kl}	Pheromone value for the edge from node k to node l
p_{kl}	Probability for choosing the edge from node k to node l
α	Probability calculation constant
S_k	List of all edges starting from node k
ω	Pheromone constant
ρ	Evaporation constant

GDO symbols

η	Step factor for the gradient value
$\delta\eta$	Step change applied after each iteration

1 Introduction

1.1 Motivation

According to a report by the Intergovernmental Panel on Climate Change (IPCC), global warming must be permanently limited to 1.5 °C in order to minimize the risk of irreversible consequences for the environment [1]. A key driver is given by the reduction of CO_2 emissions in the energy sector, which with 42% (in 2016) accounts for the largest share of the global CO_2 production [2].

The reduction of these CO_2 emissions requires a massive expansion of renewable energy sources such as wind turbines or solar power plants to replace the existing widespread generation of electrical energy from fossil fuels. However, the integration of such renewable energy sources into the existing electricity infrastructure also causes numerous technical problems. For example, their unstable and difficult to plan power generation causes an increasing instability of the power grid due to their weather dependency, which in turn has to be compensated by conventional power plants in a costly and environmentally damaging way.

In this context, the formation of so-called "virtual power plants" (VPP)—the logical combination of several power plants of different types—plays an increasingly important role. The various power plants inside the VPP should balance each other and thus form a better controllable network. However, the optimization of such

power plant networks turns out to be a difficult task due to complex constraints such as power plant specific control requirements, different VPP sizes (e.g. within an urban area to cover local consumption or distributed over large areas as a regulation mechanism for transmission system operators—possibly also hierarchically structured) as well as different compositions (conventional power plants and/or renewable energy sources etc.) and various target objectives based on the use case. Our motivation was therefore to develop an abstract and flexible control methodology for such VPPs, which can be easily adapted to the various requirements mentioned. In order to implement this control methodology efficiently we used techniques from the field of Computational Intelligence (CI). According to the IEEE Computational Intelligence Society the field of CI can be defined as the "theory, design, application and development of biologically and linguistically motivated computational paradigms" which are organized in three main pillars: neural networks, fuzzy systems and evolutionary computation [3]. In this paper we focus on the usage of evolutionary computation algorithms.

1.2 Related work

Due to the growing importance of VPPs in energy technology, there has been a steady increase in the number of publications in this area in recent years [4, 5]. Therefore, the following sections give a representative overview of the existing approaches.

In the area of VPP optimization, most approaches focus on optimizing profit, for example by generating revenue on the electricity market or by selling to local consumers. The applied optimization techniques include the active control of the consumer load demand [6] as well as the control of the VPP itself [7, 8], optionally with the use of energy storage devices [9]. Other, less popular optimization goals are, for example, the adaptation of the local electricity production to the local consumption in order to minimize the operating reserve requirements by actively regulating the VPP and/or the consumers [10].

A representative approach for the optimized control of a VPP is given in [11]. The optimization is done by controlling the voltage buses in a VPP with a diesel generator, a solar power plant and an energy storage device with a matlab simulation. To reach the goal of covering the local consumption, the electricity generated by the solar power plant is used primarily. If the current power output of the solar power plant is not sufficient, the diesel generator is switched on in a second step. This power plant prioritization is intended to produce the required electricity in a preferably climate-neutral way. Should the total electricity provided by the power plants be greater or smaller than the consumer load demand, the difference will be compensated in a third step by the energy storage device, if possible.

A second representative approach uses the optimization of the consumer side in order to achieve the highest possible coverage by renewable energy sources [12]. Here, the authors assume a decentralized server system of a cloud provider. Since there are large spatial distances between the server farms, different weather conditions also occur between the farms at a given time. The presented approach therefore

tries to distribute the required computing load to the individual server farms in such a way that farms with a currently high availability of renewable energy power carry an increased computing load.

In [13] the authors present a central control approach for a VPP consisting of several wind turbines, solar power plants, bio power plants and an energy storage device. The approach regulates the power output of the individual power plants as well as the energy exchange with the energy storage device in order to optimize the entire power output of the VPP to a selected target objective. The authors present several target objectives that a system operator can select from including, but not limited to, profit maximization and operating reserve maximization.

The usage of CI techniques is also increasingly widespread in the field of VPPs. However, these are mainly used for dimensioning and prediction of VPPs. The dimensioning approaches are also mainly aimed at covering local consumption as well as possible [14–16] or maximizing the profit on the electricity market [17]. In the field of forecasting, artificial neural networks are increasingly used to determine the expected power output [18, 19] or the flexibility potential [20] of a VPP, for example.

A representative example for a CI based optimization approach is given in [21]. Here the authors use an evolutionary optimization algorithm called "Grenade Explosion" to optimize the output of power buses against six target objectives. These objectives include, but are not limited to, fuel costs, the voltage deviation and greenhouse gas emissions. Furthermore, several approaches are shown to consider the various objectives in a bundled way.

A second representative CI based optimization example is given in [22]. Here, the authors use a three-step optimization process using PSO, among other components. This process uses electric vehicles as energy storages and controllable consumers to optimize the compensation cost, the abandoned energy cost and the operation revenue of the VPP. It is also possible to combine the three objectives in a weighted manner.

1.3 Contribution of this paper

As previously mentioned we propose in this paper an abstract control methodology for a VPP that can easily be applied to any given (also hierarchical) VPP structure and target objectives. To demonstrate the effectiveness of the control methodology, three exemplary target objectives are considered: the minimization of operating reserve requirements, the minimization of CO_2 emissions and the maximization of the power plant flexibility which will be explained in separate sections. To implement this control methodology efficiently, we used three metaheuristics form the CI field of evolutionary computation algorithms: simulated annealing (SA), particle swarm optimization (PSO) and ant colony optimization (ACO). To legitimize the use of such advanced CI metaheuristics, we also applied gradient descent optimization (GDO) as an alternative non-CI approach.

Finally, we want to mention what in our opinion are the new contributions of this methodology compared to the state of the art:

– Abstract view: Most of the existing optimization approaches for VPPs are based on a fixed power plant composition. The authors assume a given number of power plants with predefined power plant types, such as a combination of a solar power plant farm with a diesel generator as well as predefined, fixed target objectives. This makes it difficult to transfer the approaches to VPPs that have different sizes, use different types of power plants or pursue different target objectives. In our approach, we present an abstract view that allows the neutral combination and optimization of different power plants types as well as a neutral formulation of the target objectives to be optimized. Thus, the approach can be easily transferred to VPPs of different sizes and power plant compositions.
– Robustness against large VPPs: Many of the approaches shown, such as in [13], have the problem that the complexity of the optimization process increases significantly with larger VPPs. Through the abstract view in our approach and the usage of CI metaheuristics in the optimization process, we are able to efficiently control even larger VPPs.
– Active control of renewable energy sources: In the active control of VPPs, most approaches take the power output of renewable energy sources as "given" and only optimize the control of conventional power plants. However, this results in a loss of optimization potential, especially with regard to target objectives such as the operating reserve requirements. In our abstract approach, we don't distinguish between the different power plant types inside the VPP and thus automatically include the active control of renewable energy sources in the optimization process. Thereby we're able to further increase the optimization potential of the VPP.

1.4 Organization of the paper

The paper is structured as follows: Sect. 2 gives an overview of the developed abstract control methodology. Section 3 describes the exemplary target objectives pursued here as well as their weighted combination. Section 4 explains how the control methodology is built up to the different CI metaheuristics. In Sect. 5, an exploration of the control methodology is done in two steps: first the functionality of the methodology is evaluated in terms of a concrete example scenario. Next, a comparison of the different metaheuristics is carried out. Finally, Sect. 6 summarizes the results and gives a brief outlook on future planned extensions.

This paper is an extension of our previously published control methodology for flat VPPs using SA [23].

2 Methodology

2.1 Optimization approach

This section explains the basic functionality of our control methodology proposed in this contribution. The basic goal is to create an optimized control configuration

for a VPP. In this context we define a VPP as the logical combination of several decentralized power plants of any type and in any number, which are individually controlled by a central unit (using the presented control methodology). In addition, it is possible to combine several power plants into a smaller VPP and thus generate a hierarchical structure. The main purpose of the control configuration generated by the methodology is to adapt the electric power output of the VPP so that any given target objective will be minimized. The generation of the control configuration is based on forecasts (e.g. weather forecasts for wind turbines and solar power plants) as well as simulations (e.g. load demand simulations for consumers and power output simulations for power plants). A typical time horizon for this optimization would be 1 day: on a given day, the methodology collects the forecast data for the next day, simulates the expected power output of the power plants in the VPP as well as any required consumer load demands on the basis of this data, and finally determines the control configuration for the next day based on the chosen target objectives to help the power plant operators with their planning work. If the actual environmental conditions of the next day differ from the forecasts made (e.g. different weather conditions), the power plant operators can decide at any time whether a new control configuration should be created on the basis of the updated data (e.g. if there is a high demand on the accuracy of the power output) or the old configuration should be retained.

This contribution is focused on the development of the control methodology. The forecast data required for this optimization were thus obtained from external sources, which are identified at the appropriate places. All simulations required for the optimization (both of power plants and consumers) were generated with a simulation environment developed in previous works [24]. The power output of the VPP is optimized in time steps of 15 min (according to the frequency-response reserve), resulting in a total of 96 time points for 1 day, but other time intervals are also possible. The control methodology has been developed in such a way that it can handle any type of power plant. For the purpose of clarity, however, only three types of power plants are considered in this contribution: wind turbines, solar power plants and combined heat and power units (CHP). Power plants that are within the control logic of a VPP are referred to as sub-plants in the following.

Figure 1 shows an abstract representation of the optimization procedure. To alter/optimize the total power output of the entire VPP (here: VPP one), each time point from left to right (here: t1–t6) is processed successively in a separate call of the control methodology. The control methodology optimizes the power output of VPP one at this time point by setting a desired power output for all its sub-plants (here: wind turbine one, VPP two and solar power plant one). If one or more of the sub-plants are another VPP (in a hierarchical structure; here: VPP two), the control methodology is called recursively for these VPPs until the entire hierarchical tree has been processed. The desired power output for the sub-plants is determined according to the chosen target objectives. Once the control methodology has been called for all time points, an optimized power output for VPP one has been found and its distribution to the individual sub-plants has been carried out at the same time. The determined power outputs of the individual sub-plants inside the hierarchical tree
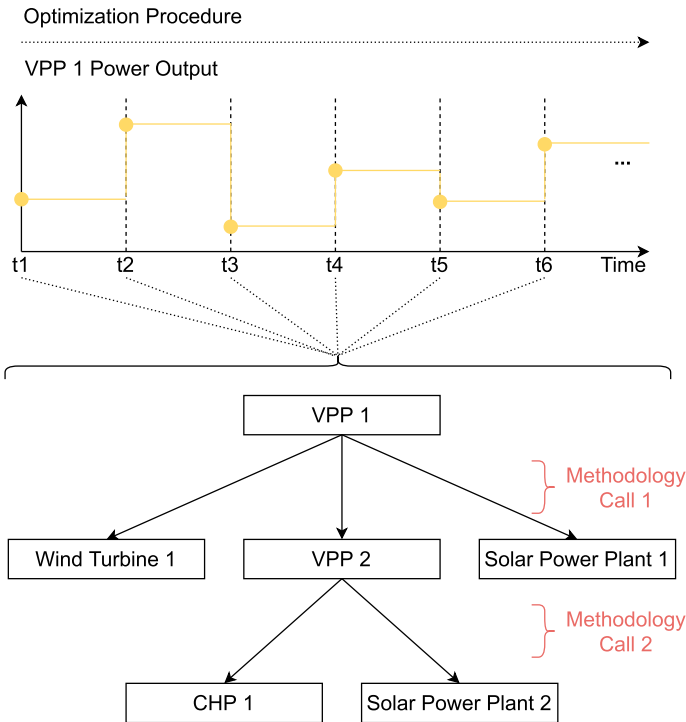
Optimization Procedure

VPP 1 Power Output

Fig. 1 Optimization process of the control methodology

can then be used to generate control signals and thus implement the solution for the considered time period.

2.2 Abstraction of power plant types

In order to enable the optimization of any VPP, each sub-plant type provides a series of functions dependent on the time t, defining an abstract interface for the usage of the methodology. These functions are explained below.

The functions $P_{i,min}(t)$ and $P_{i,max}(t)$ represent the *theoretical minimum and maximum power production limits* of a sub-plant i at a time point t (in Watt). The minimum power limit of the sub-plants is usually zero—since they can all be switched off—but is still provided here as an independent function in order to maintain compatibility for any subsequent adjustments. In the case of a CHP unit, the maximum power limit at any time point corresponds to the rated power of the engine. In case of solar power plants and wind turbines, the maximum power limit is determined by the previously executed simulation of those power plants [23].

The functions $\widetilde{P}_{i,min}(t)$ and $\widetilde{P}_{i,max}(t)$ represent the *actual minimum and maximum power production limits* accordingly (in Watt). The differences to the theoretical limits are restrictions due to the control requirements of a sub-plant that arose from

control decisions at earlier time points. Such restrictions can be, for example, the minimum running time and minimum cooling time of a CHP unit [23].

The function $P_{i,s}(t)$ determines the *step size* of a sub-plant i at a time point t (in Watt) and corresponds to the minimum power quantum by which the power output of a sub-plant can be increased or decreased within its limits. In this way, the different control methods of the various sub-plant types can be mapped [23].

Finally, the function $C_i(t)$ determines the *specific* CO_2 *emissions* of a sub-plant i at time point t. The specific CO_2 emissions describe the amount of CO_2 which is produced by the sub-plant's power generation process over time (in g/kWh). For a CHP unit, this value depends on the fuel which is used to power the engine. Since wind turbines and solar power plants don't produce any CO_2 during power generation but only by their construction, this value would normally be fixed at zero for both sub-plant types in a real-world application scenario. However, the presented scenarios in this paper use values greater than zero (and thus assume that those sub-plant types produce CO_2 during power generation) in order to better demonstrate the effect of the various target objectives. To keep these fictional CO_2 emissions as realistic as possible, the amount of CO_2 produced during the construction was divided by the average amount of energy produced during their typical lifespan.

To illustrate these functions, Fig. 2 shows an exemplary power curve for a CHP unit with a rated power output of 100 kW and the associated function values. Since the theoretical power limits and the step size are constant over the entire time series, they are given only once (in the upper left corner). The actual power limits are given at any time point by a tuple of minimum and maximum limit. At the beginning of the time series it is assumed that the CHP unit is not yet affected by any control restrictions. Consequently, the theoretical and actual limits coincide. At 00:30, the control methodology decides to switch the CHP unit from off to on by adding the step size once to the power output value. From here on, the minimum running time of—for example—1 h takes effect. The CHP unit therefore must not be switched off again during the following three time points. The CHP unit passes this information to the control methodology by setting the actual minimum power limit to 100 kW and thus reducing the power mobility (maximum minus minimum limit) to zero. The control methodology therefore cannot change the power output of the CHP unit during these time points. From 01:30
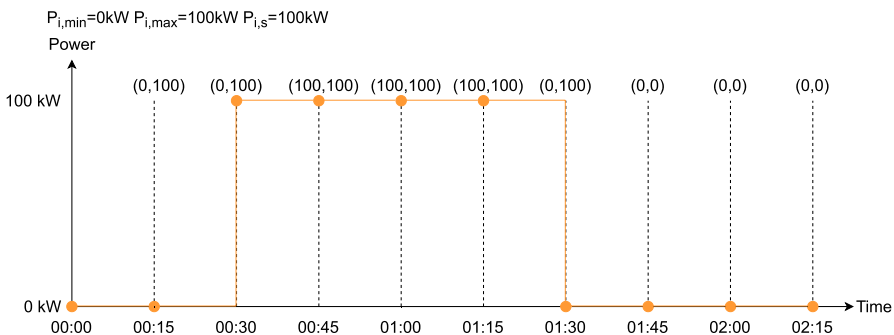

Fig. 2 Exemplary curve of the functional values [23]

on, the minimum running time is satisfied and the CHP unit resets the minimum power limit to zero: The power mobility of the sub-plant increases and the control methodology can switch off the CHP unit again (if necessary). After the switch-off process at 01:30, the minimum cooling time of—for example—1 h applies accordingly. Here, the CHP unit sets the actual maximum power limit to zero in order to prevent a switch-on process during the next time points. The CHP unit could do the same if it has to be switched on or off due to an additional thermal load demand that needs to be satisfied. A wind turbine or a solar power plant can use the actual power limit accordingly to transmit its simulated maximum possible power output to the control methodology at any time point [23].

The abstract usage of these functions allows a simple extension of the methodology by new power plant types, since a new type only has to provide these functions in order to be integrated into the process. This significantly increases the flexibility of the control methodology.

2.3 Formal capturing of the power generation process

Using the previously defined abstract interface, it is possible to define the power output $P_i(t)$ of a sub-plant i in general [23]:

$$P_i(t) = \widetilde{P}_{i,min}(t) + k_i(t) \cdot P_{i,s}(t) \tag{1}$$

$$k_i(t) \in \mathbb{N}_0 \: : \: k_i(t) \leq \left\lfloor \frac{\widetilde{P}_{i,max}(t) - \widetilde{P}_{i,min}(t)}{P_{i,s}(t)} \right\rfloor \tag{2}$$

The power output of a sub-plant i is the sum of its actual minimum power limit and a natural multiple $k_i(t)$ of its step size (Eq. 1), which is hereinafter referred to as the control coefficient. In order to not exceed the actual maximum power limit, an upper border is defined for the control coefficient $k_i(t)$ (Eq. 2). Assuming that the VPP consists of a total of N sub-plants, the total power output $P(t)$ of the VPP is given by [23]:

$$P(t) = \sum_{i=1}^{N} P_i(t) \tag{3}$$

The resulting sum contains N values for the step sizes and the control coefficients. These can alternatively be formulated as vectors to transform the problem into a solvable form for the CI metaheuristics [23]:

$$P(t) = \mathbf{k(t)} \cdot \mathbf{P_s(t)} + \sum_{i=1}^{N} \widetilde{P}_{i,min}(t) \tag{4}$$

$$\mathbf{k(t)} = \left(k_1(t), k_2(t), \ldots, k_N(t) \right) \tag{5}$$

$$\mathbf{P_s}(t) = \big(P_{1,s}(t), P_{2,s}(t), \ldots, P_{N,s}(t) \big) \tag{6}$$

In the shown form, the CI metaheuristics can control the sub-plants and therefore optimize the power output by simply modifying the coefficient vector $\mathbf{k}(t)$

2.4 Realization of the hierarchy concept

As already mentioned, it should be possible to consider hierarchical power plant structures in which a VPP can contain a combination of several smaller VPPs and/or simple power plants. The need of hierarchical VPPs results from the fact that often smaller VPPs already exist with their own, static control mechanisms. If such a VPP has to be combined with other power plants to form a larger VPP, the higher-level control system can only set target values for the smaller network as a whole and not directly influence its sub-plants. An example of this would be an existing network of solar power plants and CHP units in a village which should be combined with new built wind turbines nearby the village: a higher-level control system can possibly only perform an optimization for the wind turbines and the smaller village network as a whole, but not for individual solar power plants/CHP units inside the village. In a hierarchical VPP model each sub-VPP can define whether the control methodology can also access its sub-plants or not. If not, a power output value is only determined for the sub-VPP as a whole and the recursive call of the control methodology (as shown in Fig. 1) is omitted.

To implement the hierarchical model, a sub-VPP must also provide the abstract interface for simple sub-plants presented in Sect. 2.2. Since the control methodology only uses this interface for the optimization, it doesn't have to distinguish between a simple power plant and a smaller VPP if both provide the previously interface. The methodology therefore uses the idea of the composite pattern, which is a commonly used design pattern for hierarchical structures in computer science [25].

The theoretical and actual power output limits of a VPP refer to the sum of all its sub-plants:

$$P_{VPP,min}(t) = \sum_{i=1}^{N} P_{i,min}(t) \tag{7}$$

$$P_{VPP,max}(t) = \sum_{i=1}^{N} P_{i,max}(t) \tag{8}$$

$$\widetilde{P}_{VPP,min}(t) = \sum_{i=1}^{N} \widetilde{P}_{i,min}(t) \tag{9}$$

$$\widetilde{P}_{VPP,max}(t) = \sum_{i=1}^{N} \widetilde{P}_{i,max}(t) \tag{10}$$

However, the specific CO_2 emissions of a VPP must be determined differently: since the actual emissions vary depending on the control state of the sub-plants, they cannot be simply added up, but must be estimated during the optimization process.

Figure 3 shows an example of this problem for a hierarchical model. The top-level VPP (VPP one) consists of two smaller VPPs (VPP two and three) which both in turn consist of one wind turbine and one CHP unit. The specific CO_2 emissions $C_i(t)$ are given both for the CHP units and the wind turbines and are the same for VPP two and three. In the first optimization step, a load demand $L(t)$ of 1000 W to be met by the VPP is split from VPP one to VPP two (400 W) and VPP three (600 W). In the second step VPP two splits the required power to wind turbine one (300 W) and CHP unit one (100 W). In the third step, the power is split from VPP three to wind turbine two (100 W) and CHP unit two (500 W). In order to minimize the overall CO_2 emissions, the specific emissions of VPP two and three must be known at the first step to divide the load demand appropriately between them. However, those values can only be determined precisely after step two and three: since the load demand was stronger distributed to the CHP unit in VPP three than in VPP two, the specific CO_2 emissions of VPP three are higher than those of VPP two, although they both have the same structure. In order to be able to carry out the optimization in step one, the specific CO_2 emissions of VPP two and three must be estimated in advance.

The larger the difference between the actual minimum and maximum power limit of a sub-plant i is (i.e. the more electricity it can generate), the more likely the power output of this sub-plant will be increased by the control methodology to satisfy the target objectives. Therefore, the specific CO_2 emissions of a VPP $C_{VPP}(t)$ can be approximated by a weighted sum of the specific emissions of its sub-plants, using the actual power limit difference of the sub-plants as a weighting factor:
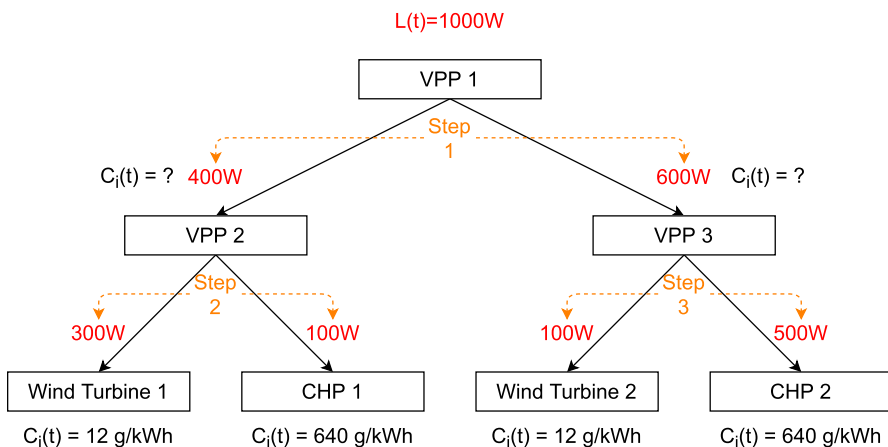

Fig. 3 Example of the hierarchical control methodology

$$C_{VPP}(t) = \frac{\sum_{i=1}^{N} \left[\left(\widetilde{P}_{i,max}(t) - \widetilde{P}_{i,min}(t) \right) \cdot C_i(t) \right]}{\sum_{i=1}^{N} \left[\widetilde{P}_{i,max}(t) - \widetilde{P}_{i,min}(t) \right]} \tag{11}$$

The situation is similar with the step size of a VPP $P_{VPP,s}(t)$: since it represents the control methods of the various power plant types, these values also cannot simply be summed up. In this paper, the smallest step size among all the sub-plants is selected for the entire VPP as it turned out to produce the best results:

$$P_{VPP,s}(t) = \min_{1 \le i \le N} P_{i,s}(t) \tag{12}$$

3 Target objectives

Using the formulations in Sect. 2.3, the methodology aims to minimize a given target objective $E(t)$ by adjusting the coefficient vector $\mathbf{k(t)}$ and thus manipulating the power output $P(t)$ of the VPP:

$$\underset{\mathbf{k(t)}}{\text{minimize}}\, E(t) \tag{13}$$

To illustrate the methodology and how the abstract interface can be used to formulate objectives, we have opted for a weighted combination of three individual target objectives in terms of error functions: the minimization of operating reserve requirements, the minimization of CO_2 emissions and the maximization of the power plant flexibility, which will be explained further in the following sections. To combine the individual objectives in a weighted manner, each objective will be scaled to the interval [0, 1]. A scaled value of zero represents the best matching result; a value of one represents the worst matching result.

3.1 Minimization of operating reserve requirements

In order to compensate the increasing grid instability caused by renewable energy sources, transmission system operators need a large amount of operating reserves to buffer any fluctuations, which causes high costs. To minimize the need for operating reserves and thus the costs, a VPP should try to reproduce a predicted load demand as accurately as possible so that ideally no operating reserve is needed anymore. Therefore, the following error function $O(t)$ determines how much the power output of a VPP $P(t)$ deviates from a given load demand $L(t)$ [23]:

$$O(t) = \left(\frac{P(t) - L(t)}{max(P(t), L(t))} \right)^2 \tag{14}$$

3.2 Minimization of CO$_2$ emissions

An additional target objective is the minimization of CO$_2$ emissions. When reproducing a load demand, sub-plants with low specific CO$_2$ emissions should be regulated up, whereas sub-plants with high specific CO$_2$ emissions should be regulated down. One indicator for the objective error value therefore is the total amount of CO$_2$ produced by the all sub-plants. To approximate the emissions of a sub-plant i, the power produced is multiplied by the specific CO$_2$ emissions $C_i(t)$ of the sub-plant i. The emissions of the entire VPP $M(t)$ correspond to the sum of all its sub-plants (see Eq. 15) [23]. Finally, the emission amount $M(t)$ is scaled by dividing it by the total possible amount of CO$_2$ emissions for that time point (Eq. 16):

$$M(t) = \sum_{i=1}^{N} P_i(t) \cdot C_i(t) \tag{15}$$

$$NM(t) = \frac{M(t)}{\sum_{i=1}^{N} \widetilde{P}_{i,max}(t) \cdot C_i(t)} \tag{16}$$

3.3 Maximization of power plant flexibility

The maximization of the power plant flexibility has the purpose to define the control of a sub-plant i in such a way that it can act as freely as possible for subsequent time points. Examples of this are the minimum running time and the minimum cooling time of a CHP unit: if a CHP unit is switched on or off at a time point, it must not be switched again over a certain time period. This limits its flexibility, which leads to a loss of possibly better states for the following time points. This in turn may result in an overall worse outcome for the optimization of the entire time period. It therefore makes sense to keep power plants as flexible as possible.

In order to determine to what extent a control decision at a time point t affects the flexibility of the subsequent time points on average (with a total of T time points), the ratio between the theoretical and actual power limit of a sub-plant i is determined for all subsequent time points. Since the two limits differ only in the influence of control decisions, the greater the difference between the limits is, the more limited is the sub-plant. Based on the assumption that a control decision for a sub-plant i was made at time point t, the impact on the sub-plant flexibility $F_i(t)$ is calculated as shown in Eq. (17) [23]:

$$F_i(t) = \frac{1}{T-t} \cdot \sum_{t'=t+1}^{T} \frac{\left(P_{i,max}(t') - P_{i,min}(t')\right) - \left(\widetilde{P}_{i,max}(t') - \widetilde{P}_{i,min}(t')\right)}{P_{i,max}(t') - P_{i,min}(t')} \tag{17}$$

The fractional term within the sum is formed so that a value zero results if the theoretical and actual power limits of sub-plant i coincide and thus there are no

restrictions due to the control decision made. In the other extreme case, when sub-plant i can no longer be controlled at all in subsequent time points due to the control decision (the difference of the actual maximum and minimum power limit results in zero), a value of one is returned. This fractional term is calculated and averaged for all time points following the currently considered time point t (all time points that are potentially affected by the control decision).

The flexibility effects for the entire VPP $F(t)$ can be determined as the average value of all sub-plants [23]:

$$F(t) = \frac{1}{N} \cdot \sum_{i=1}^{N} F_i(t) \tag{18}$$

3.4 Combination of the target objectives

The final error function $E(t)$ to be optimized by the methodology is made up of the weighted sum of the three target objectives presented above. For this purpose, three weighting coefficients (W_O, W_M, W_F) in the interval [0, 1] are defined which a system operator could use to define what objectives have to be considered and to what extent [23]:

$$E(t) = W_O \cdot O(t) + W_M \cdot NM(t) + W_F \cdot F(t) \tag{19}$$

For the scenarios considered in this paper, a weighting of 80% for the operating reserve requirements W_O, 10% for the CO_2 emissions W_E and 10% for the power plant flexibility W_F was assumed. The minimization of the operating reserve requirements has the largest weighting because serving the load demand is considered as the main objective of a VPP here. This means that the VPP should always try to cover the load demand as good as possible. The other two objectives serve as secondary targets to fulfill the load demand as flexible and CO_2-neutral as possible.

4 Considered metaheuristics

According to Sect. 2.3, a VPP can be controlled by manipulating the coefficient vector **k(t)**. Therefore, the following sections explain how to adjust the coefficient vector to optimize the VPP power output using the three CI metaheuristics: SA, PSO and ACO. GDO is also considered as a traditional, non-CI algorithm to demonstrate the necessity of the other three CI metaheuristics. To illustrate the application to the optimization problem, the process for each metaheuristic is additionally displayed in pseudo code. Furthermore, a Grid Search parameter tuning process was performed for all four approaches in order to find the best configurations for the optimization problem. The chosen parameters are presented at the end of the corresponding sections.

4.1 Simulated annealing

SA is a metaheuristic that mimics the cooling process of hot metals to iteratively search a state space. The cooling process is expressed by the fact that at the beginning also worse states (with regard to the target objective) found are adopted as the best state by chance. However, this probability decreases with the number of completed iterations. By adopting worse states at the beginning, local minima of an (to be optimized) error function can be overcome in order to increase the chance of finding the global minimum. The overall best state found represents the final solution of the optimization problem. For this purpose, SA randomly changes the current best state slightly in each iteration step. If the new state is better than the old one, it will be adopted in any case. If not, it will still be adopted with the probability described above. When determining the probability, a temperature value is included which is reduced after each iteration step, simulating the cooling process [26].

A possible coefficient vector of the VPP represents a state for the SA metaheuristic. SA selects a random vector element at each iteration step and enlarges/reduces it. This increases/reduces the power output of the corresponding sub-plant in the VPP, which results in an alternative state to be evaluated with respect to the target objective $E(t)$. An exponential variant was chosen for the cooling process as shown in Eq. (20). The optimization process terminates as soon as the current temperature at iteration j ϑ_j is smaller than the minimum temperature ϑ_{min} [23].

$$\vartheta_j = \vartheta_{init} \cdot \alpha^j \tag{20}$$

Algorithm 1 SA Procedure

---

set iteration $j = 0$
set $\vartheta_j = \vartheta_{init}$
randomly set $\mathbf{k}(\mathbf{t})$
evaluate $\{best\_score\}$ according to $\mathbf{k}(\mathbf{t})$ and $E(t)$
while $\vartheta_j \geq \vartheta_{min}$ **do**
    randomly choose an element $k_i(t)$ from $\mathbf{k}(\mathbf{t})$
    randomly change $k_i(t)$ according to equation 2
    evaluate $\{new\_score\}$ according to $\mathbf{k}(\mathbf{t})$ and $E(t)$
    if $\{new\_score\} < \{best\_score\}$ **then**
        set $\{best\_score\} = \{new\_score\}$
        keep changes to $\mathbf{k}(\mathbf{t})$
    else
        if $e^{(\{best\_score\}-\{new\_score\})/\vartheta_j} < 1.0$ **then**
            set $\{best\_score\} = \{new\_score\}$
            keep changes to $\mathbf{k}(\mathbf{t})$
        else
            undo changes to $\mathbf{k}(\mathbf{t})$
        end if
    end if
    set $j = j + 1$
    update ϑ_j according to equation 20
end while
return $\mathbf{k}(\mathbf{t})$ as result

The pseudo code 1 illustrates the exact procedure. The tuned parameters for SA are: 481.0 for the initial temperature ϑ_{init} and 2.76e−8 for the minimum temperature ϑ_{min}. The exponential cooling factor α is automatically determined to fit the desired number of iterations.

4.2 Particle swarm optimization

PSO is a metaheuristic which—similar to SA—iteratively searches a state space for the best possible solution. However, PSO uses aspects of swarm intelligence by observing several states simultaneously; so-called particles. Each particle has a position $x_i(j)$—the currently selected state in the state space—and a velocity $v_i(j)$ that points in the direction of an expected better state. At the beginning of an iteration, each particle is shifted by its velocity to a different state in the state space, which is then evaluated against the target objective $E(t)$. After doing this, each particle also updates its best state ever found if necessary. At the same time, the globally (across all particles) best found state is updated if necessary. Finally, the velocity of each particle is updated by the following Eq. (21). Here, both the position of the locally $\hat{x}_i(j)$ and globally $g(j)$ best found state as well as the current particle state $x_i(j)$, various weighting factors w, c_1, c_2 and random values r_1, r_2 are included. At the end the global best state found is the proposed solution for the optimization problem [27].

$$v_i(j + 1) = w \cdot v_i(j) + c_1 \cdot r_1 \cdot \left[\hat{x}_i(j) - x_i(j)\right] + c_2 \cdot r_2 \cdot \left[g(j) - x_i(j)\right] \tag{21}$$

A possible state $x_i(j)$ of a particle is represented by the coefficient vector. The particle velocity $v_i(j)$ is represented by a difference vector of the same length. When adding a difference vector to a coefficient vector (displacement of a particle), care must be taken that no element of the new coefficient vector exceeds its defined maximum value (as expressed in Eq. 2).

Algorithm 2 PSO Procedure

---

randomly set $x_i(j)$ for all particles i
randomly set $v_i(j)$ for all particles i
set iteration $j = 0$
while $j < \{num\_iterations\}$ **do**
    for all particles i **do**
        update particle position $x_i(j+1) = x_i(j) + v_i(j)$
        clip particle position $x_i(j+1)$ according to equation 2
        evaluate $\{new\_score\}$ according to $x_i(j+1)$ and $E(t)$
        if $\{new\_score\} < \{particle\_best\_score\}$ **then**
            set $\{particle\_best\_score\} = \{new\_score\}$
            update particle best position $\hat{x}_i(j+1) = x_i(j+1)$
            if $\{new\_score\} < \{global\_best\_score\}$ **then**
                set $\{global\_best\_score\} = \{new\_score\}$
                update global best position $g(j+1) = x_i(j+1)$
            end if
        end if
    end for
    for all particles i **do**
        update particle velocity $v_i(j+1)$ according to equation 21
    end for
    set $j = j+1$
end while
return $g(j)$ as result

The pseudo code 2 illustrates the exact procedure. The tuned parameters for PSO are: 14 particles, 1.1 for the weight of the old velocity (w), 0.62 for the weight of the local best state (c_1) and 0.94 for the weight of the global best state (c_2).

4.3 Ant colony optimization

ACO is a metaheuristic that is based on the natural behavior of ant colonies. If—for example—ants are looking for food, each ant leaves a pheromone trail on its way, which evaporates over time. If an ant has found a particularly short (good) path, the concentration of the pheromone track increases faster, which leads other ants to take this path as well. Similarly, ACO uses a directed graph with several nodes and edges connecting those nodes. This graph is then traversed by several ants from node to node over the edges. The resulting node sequence generated by an ant represents a possible solution to the optimization problem. The structure of the graph is created specifically for the optimization problem and must be such that all possible solutions of the problem are represented by a path within the graph. The goal is to find a preferably good path through the graph (a good solution for the optimization problem). To increase the chance that an ant finds a good path, each edge e_{kl} from node k to node l is provided with a pheromone value τ_{kl} which influences the probability p_{kl} that an ant will choose that edge. The probability for an edge is calculated according to Eq. (22) using a probability constant α and the list of all edges starting from node k: S_k. After all ants have traversed the graph (one iteration), each ant a evaluates its found path with regard to the target objective $E(t)$. Based on the objective score, each ant increases the pheromone values of all edges on its path using a pheromone constant ω. The better the score of a path, the more the pheromone values of the corresponding edges are increased. At the same time, all pheromone values inside the graph are evaporated (decreased) using an evaporation constant ρ. These two steps to adjust the pheromone values are represented by Eq. (23). By iteratively increasing and evaporating the pheromone values, edges that are often part of a good solution get an increased probability over time. At the same time the probability of edges which are rarely part of a good solution is continuously reduced. This adaptive behavior should result in preferably good paths being found as the number of iterations increases. After all iterations are finished, the best path found overall is the solution to the optimization problem [28].

$$p_{kl} = \frac{\left(\tau_{kl}\right)^{\alpha}}{\sum_{\tau_{km} \in S_k} \left(\tau_{km}\right)^{\alpha}} \tag{22}$$

$$\tau_{kl} = (1 - \rho) \cdot \tau_{kl} + \sum_{a}^{\{\text{ants with } e_{kl} \text{ in path}\}} [(1 - E(t)) \cdot \omega] \tag{23}$$

In order to adapt the presented methodology to the ACO metaheuristic, a graph with several layers is formed, whereby each layer represents a sub-plant of the

VPP (see Fig. 4). The number of nodes in a layer corresponds to the maximum value of the control coefficient for the sub-plant. Each node of a layer has directed edges to all nodes of the next layer, allowing an ant to form any path between the layers. The nodes of a selected path represent the selected control coefficient values of the sub-plants and thus their power output. To generate such a path, an ant starts at a random node of the first layer and works its way up to the last layer. The pseudo code 3 illustrates the exact procedure.

Algorithm 3 ACO Procedure

---

set iteration $j = 0$
randomly set $\{best\_state\}$
evaluate $\{best\_score\}$ according to $\{best\_state\}$ and $E(t)$
while $j < \{num\_iterations\}$ **do**
    for all ants a **do**
        place ant a on a random node k of the first layer
        empty $\{path\}$ of ant a
        while ant a isn't on last layer **do**
            calculate probabilities of all outgoing edges from node k according to equation 22
            choose a random edge based on the calculated probabilities
            store the chosen edge into $\{path\}$ of ant a
            move ant a over the chosen edge to the node k on the next layer
        end while
        create a coefficient vector $\mathbf{k(t)}$ based on $\{path\}$ of ant a
        evaluate $\{new\_score\}$ according to $\mathbf{k(t)}$ and $E(t)$
        if $\{new\_score\} < \{best\_score\}$ **then**
            set $\{best\_score\} = \{new\_score\}$
            set $\{best\_state\} = \mathbf{k(t)}$
        end if
    end for
    update pheromone values of all edges according to equation 23
    set $j = j + 1$
end while
return $\{best\_state\}$ as result

The tuned parameters for ACO are: 13 ants, 0.32 for the evaporation constant ρ, 0.25 for the probability constant α and 9.58 for the pheromone constant ω.
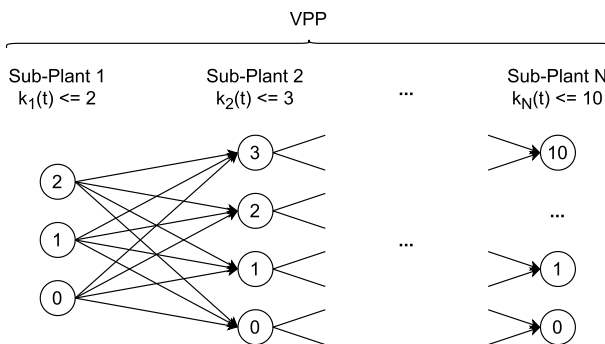

Fig. 4 Structure of the ACO VPP graph

4.4 Gradient descent optimization

GDO is a common non-CI optimization algorithm and popular especially in the training process of neural networks. The idea of GDO is to find a minimum of a given function by altering the input variables in the direction of the negative gradient. This causes the function settle around the closest minimum of the function from a given starting position [29].

One drawback of GDO is that it can't overcome any local minima of the function to be optimized and thus stays at the nearest minimum. This especially may lead to poor results when optimizing functions which have many local minima.

For the given optimization problem, a vanilla gradient descent variant was chosen which manipulates the input variables of a given function by the product of the negative gradient and a step factor η [29]. The algorithm manipulates the elements of the coefficient vector as shown in Eq. (24) using the gradient of the error function $E(t)$.

$$k_i'(t) = k_i(t) - \eta \cdot \frac{\partial E(t)}{\partial k_i(t)} \tag{24}$$

Additionally, the step factor η is reduced by a step change $\delta\eta$ after each iteration so that the size of the steps decreases over time. This should prevent the coefficients from oscillating around the minimum.

The tuned parameters for GDO are: 0.154 for the step factor η and 0.057 for the step change $\delta\eta$.

5 Validation

The validation of the optimization described in the previous sections will take place in two steps: in the first step, the general functionality of the control methodology—both with a flat and a hierarchical VPP—is demonstrated in terms of an example scenario using the PSO metaheuristic. In the second step, a comparison of all the selected metaheuristics takes place in order to examine their suitability for the optimization problem.

5.1 Validation of functionality

To validate the functionality, Fig. 5 shows an exemplary curve for 1 day of a load demand to be met, the optimized power curve of a flat VPP as well as the power curves of its three sub-plants: a wind turbine, a solar power plant and a CHP unit. A rated output of 100 kW and a minimum running time and cooling time of 90 min each were assumed for the CHP unit. The maximum possible power output of the solar power plant and the wind turbine is based on simulation data for the concrete weather conditions on 08-25-2016 in the region of Stuttgart, Germany [30] and shown in Fig. 6. The load demand is based on a simulation for

residential areas [24]. For the optimization of the CO_2 emissions, the specific CO_2 emissions of the sub-plants listed in Table 1 were assumed.

Comparing the load demand with the power curve of the VPP in Fig. 5, it can be seen that the power output reproduces the load demand very well at times when there is sufficient power available at all. The range from approximately 13:00 is of interest: as soon as the weather conditions make it possible to increasingly receive power from the wind turbine, the control methodology decides to gradually reduce the output of the solar power plant for this purpose, since its specific CO_2 emissions are higher than those of the wind turbine. An alternative could be to switch off the CHP unit to further reduce the CO_2 emissions. However, this would contradict the target objective of the power plant flexibility, since the occurring minimum cooling time would limit the CHP unit for several time points. Looking at the curve from approximately 15:15 on, an area can be seen in which the load demand can be completely covered by the wind turbine. Here even the CHP unit was switched off (despite the fact that this results in a limited flexibility), since the exclusive covering of the load demand by the wind turbine can massively reduce the CO_2 emissions. As soon as the available power of the wind turbine drops, the control methodology switches the CHP unit on again, although this leads to a further restriction of its flexibility (minimum running time) and an increase of CO_2 emissions. However, this is necessary in order to be able to sufficiently cover the load demand, which is the dominant part with a weighting of 80%. These examples clearly demonstrate the trade offs of the target objectives and thus the importance of their chosen weightings.

In order to evaluate the results in relation to the structure of a VPP, a comparison between a flat and a hierarchical VPP is given in Fig. 7. Both VPPs consist of the same 30 sub-plants: ten wind turbines, 15 solar power plants and five CHP
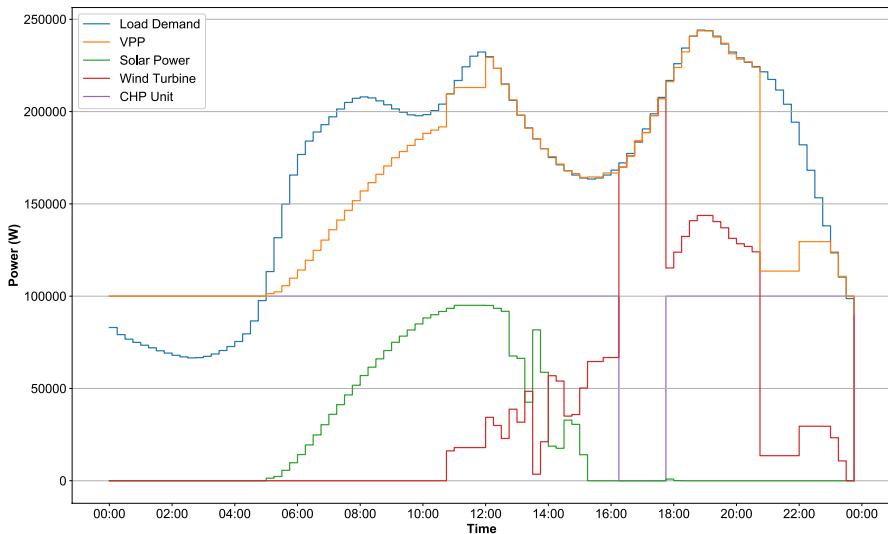
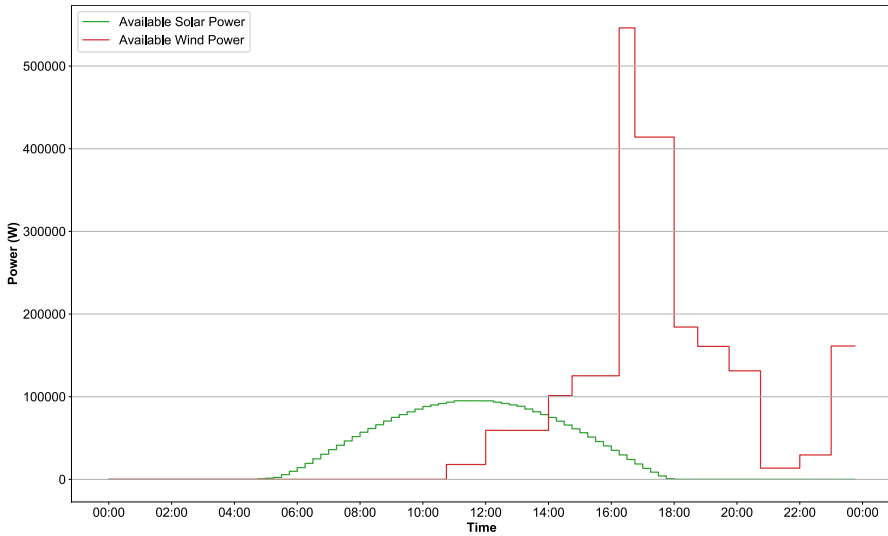

Fig. 5 Exemplary power curves for 1 day

Fig. 6 The maximum available renewable energies

Table 1 Assumed specific CO_2 emissions of the sub-plants [31]

Power plant type	Specific CO_2 emissions (g/ kWh)
CHP unit	607.6
Wind turbine	15.7
Solar power plant	55.8

units. The sub-plants have the same characteristic features as in the previously described scenario. The load demand has been upscaled so that the ratio of average consumption and average production is the same in both scenarios. In the flat VPP variant, all 30 sub-plants were evenly combined into one VPP. The hierarchical VPP consists of seven smaller sub-VPPs: one sub-VPP containing the ten wind turbines, one sub-VPP containing ten solar power plants and five sub-VPPs each containing one solar power plant and one CHP unit. This design represents a rural structure with several private houses and/or company buildings with CHP units and solar power plants on the roof as well as some extra-local, bigger solar power plants and wind turbines.

As it can be seen in Fig. 7, the optimized power output of the hierarchical VPP reaches almost the same values as the flat variant, which demonstrates that the control methodology is able to optimize an estimation-based hierarchical VPP nearly as efficiently as a flat VPP. The average error value $E(t)$ for a time point is 0.057 for the flat VPP (0.022 for $O(t)$, 0.251 for $NM(t)$ and 0.135 for $F(t)$) and 0.077 for the hierarchical VPP (0.044 for $O(t)$, 0.231 for $NM(t)$ and 0.204 for $F(t)$).

5.2 Comparison of the metaheuristics

To compare the selected metaheuristics, a scalable test environment with a collection of flat and hierarchical VPPs containing different numbers of sub-plants was set up. The more sub-plants a VPP contains, the larger is the possible state space and the complexity for the control methodology. Furthermore, each of the four selected approaches was applied to all VPPs with different iteration numbers.

Figure 8 shows the results obtained from that test environment. The figure consists of four sub-charts for each approach (SA, PSO, ACO and GDO). For each approach, multiple combinations of a VPP and an iteration number are considered. A VPP consists of several wind turbines ("#WT"), solar power plants ("#SPP") and CHP units ("#CHP"). Each of the three power plant types has either 10, 30 or 50 units inside the VPP, resulting in 27 different VPP configurations per approach. Furthermore, each VPP configuration was successively optimized with 100, 500 and 2000 iterations ("#Iteration"), resulting in a total of 81 test cases per approach. For each test case, a time span of 1 day was optimized with 10 consecutive runs and the average error value $E(t)$ over all time points and runs was stored. The stored error value for each test case is displayed in the figure with a number (rounded to two decimal places) and a color. The day to be optimized was randomly selected with a uniform distribution from the year 2016 and modelled analogously to the previous examples with weather data, power plant simulations and consumer simulations.

The comparison of the four approaches shows that PSO and ACO could reliably produce good results independent of the iteration number and VPP size with only small fluctuations of the error value due to the heuristic aspect of the algorithms. SA also achieves good error values, but requires more iterations as the size of the VPP increases. This is because the logic of SA only allows small changes of the state
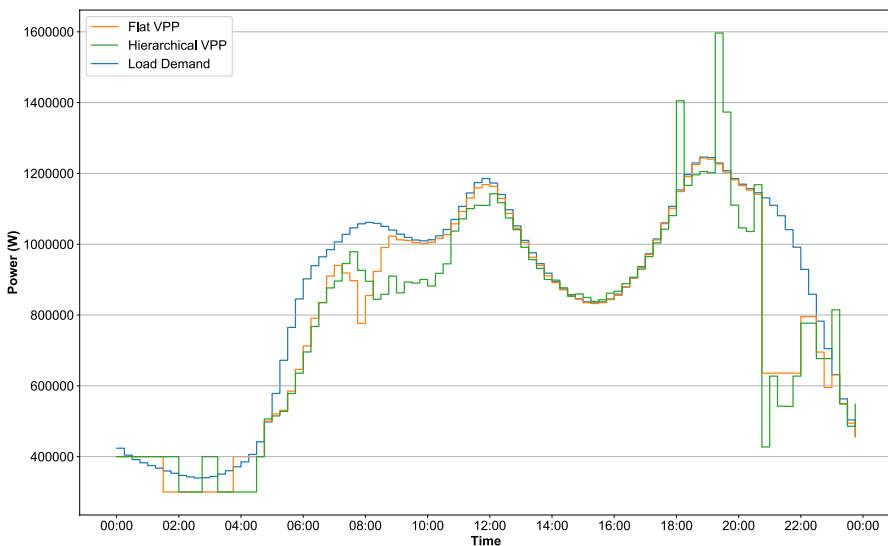

Fig. 7 Comparison of a flat and a hierarchical VPP

within an iteration. When the size of the state space increases, SA therefore needs more such small changes to achieve a good result. The PSO and ACO logics ensure that the size of the state changes automatically adapts to the growing state space and thus perform better. GDO doesn't manage to achieve comparable good results to the other three metaheuristics. Even with small VPPs and high iteration numbers, worse error values result. The randomly looking distribution of good and bad error values for GDO shows that the error function has several local minima which GDO cannot overcome.

Since PSO and ACO produce similarly good scores when comparing the optimization results, a further evaluation was done with these two approaches regarding the runtime to identify the faster approach.

Figure 9 shows the measured runtime values (in seconds) divided according to the number of sub-plants in the VPP ("#Power plant") and the number of iterations ("#Iteration") used. The measured time for each combination of iteration number and count of sub-plants inside the VPP results as the average from ten consecutive runs. As it can be seen in the diagram, the runtime of ACO increases significantly more with increasing numbers of sub-plants and iterations. At the smallest combination (with three sub-plants in the VPP and 80 iterations) the PSO approach is twice as fast as the ACO approach (approximately 13–26 s). At the largest combination (with 150 sub-plants in the VPP and 2000 iterations) the PSO approach is already four times faster than the ACO approach (approximately 16–64 s). Thus, PSO represents the more efficient optimization approach, especially for larger VPP configurations and is—according to the opinion of the authors—also the overall best approach for this scenario.
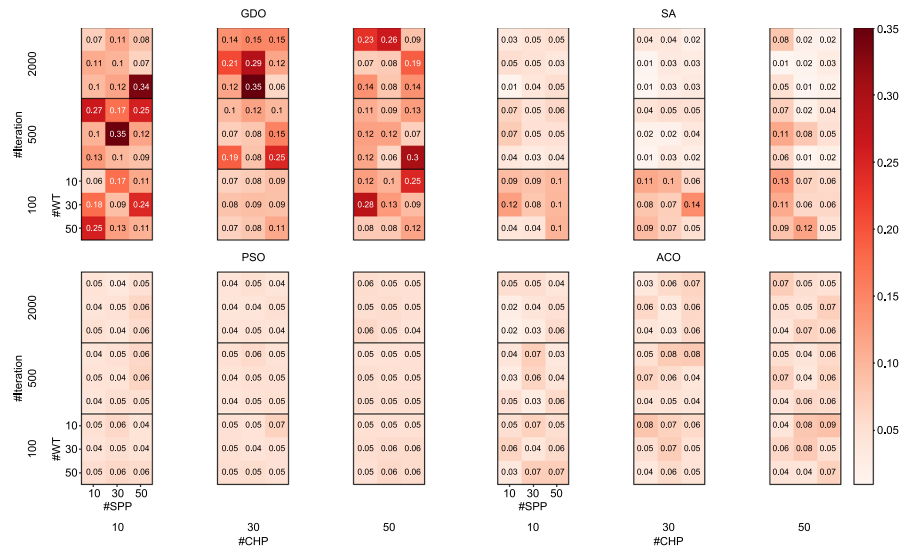

Fig. 8 Comparison of the metaheuristics

6 Conclusion

This paper describes an abstract control methodology for a VPP that can be easily applied to VPPs of different size and composition which can potentially pursue various target objectives. To apply the methodology to different power plant types, an abstract interface has been defined which each power plant type can implement according to its specific control constraints. Since the methodology only uses this interface to generate the control configuration, various power plant types in different (and hierarchical) compositions and sizes can be optimized neutrally. The three exemplary target objectives pursued by the methodology in this paper included the minimization of operating reserve requirements, the minimization of CO_2 emissions and the maximization of the power plant flexibility, which were combined in a weighted manner to perform a systematical trade-off analysis. To implement the methodology, three exemplary Computational Intelligence metaheuristics where used: Simulated Annealing, Particle Swarm Optimization and Ant Colony Optimization. To prove the necessity of such complex metaheuristics for the optimization problem, a more traditional approach was also regarded and compared to the others, Gradient Descent Optimization.

An exemplary scenario of a concrete VPP composition for 1 day showed that the methodology is able to perform a highly accurate trade-off optimization of the VPP based on the selected weighting of the target objectives. A similar example for 1 day with a comparison between a flat and a hierarchical VPP showed that the methodology achieved only slightly worse results when optimizing the hierarchical variant.

Finally, a test environment containing 27 different flat and hierarchical VPPs showed that the methodology is able to optimized various VPP sizes and compositions without knowing its exact structure. Since the test environment is further
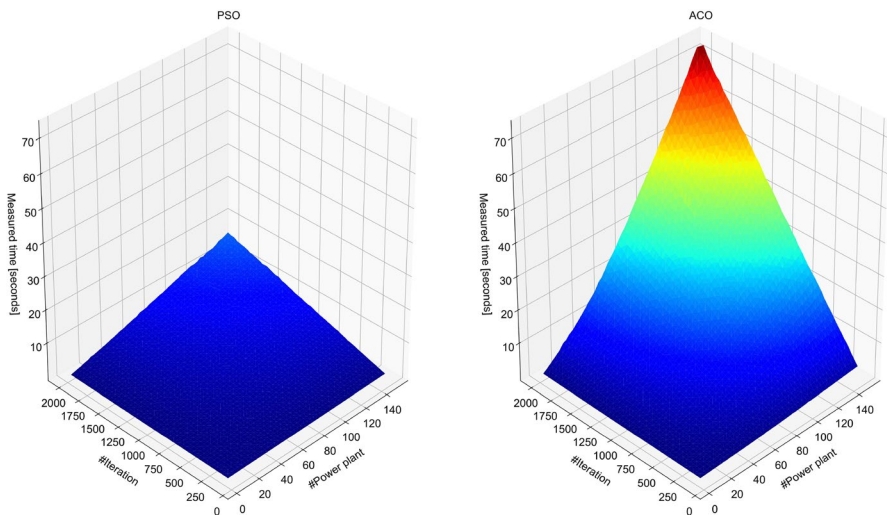

Fig. 9 Runtime comparison of PSO and ACO

divided according to the optimization algorithm and iteration number used, it could be shown that Particle Swarm Optimization and Ant Colony Optimization generated good results for all test cases independent of the iteration number. A direct comparison of these two metaheuristics with respect to their running time showed that Particle Swarm Optimization has a higher performance and is therefore considered as the best choice.

To further increase the optimization potential of the methodology, a second abstract interface is to be integrated in the future, which allows the neutral consideration of different energy storage technologies (e.g. battery storages, hydrogen storages etc.). The goal of this interface and the methodology will be to determine a temporal redistribution of the energy quantities between the energy storages and the VPP in order to fulfill the target objective even more accurately.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Masson-Delmotte V., et al.: Summary for policymakers. In: Global Warming of 1.5 °C. An IPCC Special Report on the impacts of global warming of 1.5 °C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty, pp 1–32. World Meteorological Organization, Geneva, Switzerland (2018)
2. International Energy Agency, Paris, France. CO_2 emissions from Fuel Combustion-Highlights (2018)
3. IEEE Computational Intelligence Society. What is Computational Intelligence? https://cis.ieee.org/about/what-is-ci. Accessed 18 Feb 2020
4. Guerrero, J.M., Chandorkar, M., Lee, T., Loh, P.C.: Advanced control architectures for intelligent microgrids—part i: decentralized and hierarchical control. IEEE Trans. Ind. Electron. **60**(4), 1254–1262 (2013)
5. Tuballa, M.L., Abundo, M.L.: A review of the development of smart grid technologies. Renew. Sustain. Energy Rev. **59**, 710–725 (2016)
6. Al-Awami, A.T., Amleh, N.A., Muqbel, A.M.: Optimal demand response bidding and pricing mechanism with fuzzy optimization: application for a virtual power plant. IEEE Trans. Ind. Appl. **53**(5), 5051–5061 (2017)
7. Bakari, K.E., Kling, W.L.: Virtual power plants: an answer to increasing distributed generation. In: 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), pp 1–6 (2010)

8. Baringo, A., Baringo, L., Arroyo, J.M.: Day-ahead self-scheduling of a virtual power plant in energy and reserve electricity markets under uncertainty. IEEE Trans. Power Syst. **34**(3), 1881–1894 (2019)

9. Rahimiyan, M., Baringo, L.: Strategic bidding for a virtual power plant in the day-ahead and real-time markets: a price-taker robust optimization approach. IEEE Trans. Power Syst. **31**(4), 2676–2687 (2016)

10. Pasetti, M., Rinaldi, S., Manerba, D.: A virtual power plant architecture for the demand-side management of smart prosumers. Appl. Sci. **8**(3), 432 (2018)

11. Belabbas, B., Allaoui, T., Tadjine, M., Denai, M.: Power management and control strategies for off-grid hybrid power systems with renewable energies and storage. Energy Syst. **10**(2), 355–384 (2019)

12. Camus, B., Dufossé, F., Orgerie, A.: A stochastic approach for optimizing green energy consumption in distributed clouds. In: Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems—volume 1: SMARTGREENS, pp. 47–59. INSTICC, SciTePress (2017)

13. Richter, A., Hauer, I., Wolter, M.: Algorithms for technical integration of virtual power plants into German system operation electricity market. Adv. Sci. Technol. Eng. Syst. J. **3**(1), 135–147 (2018)

14. Ekren, O., Ekren, B.Y.: Size optimization of a pv/wind hybrid energy conversion system with battery storage using simulated annealing. Appl. Energy **87**(2), 592–598 (2010)

15. Klonari, V., Toubeau, J., Lobry, J., Vallée, F.: Photovoltaic integration in smart city power distribution: A probabilistic photovoltaic hosting capacity assessment based on smart metering data. In: 2016 5th IEEE International Conference on Smart Cities and Green ICT Systems (SMART-GREENS), pp. 1–13. Rome, Italy (2016)

16. Mellit, A., Kalogirou, S.A., Drif, M.: Application of neural networks and genetic algorithms for sizing of photovoltaic systems. Renew. Energy **35**(12), 2881–2893 (2010)

17. Kornelakis, A.: Multiobjective particle swarm optimization for the optimal design of photovoltaic grid-connected systems. Sol. Energy **84**(12), 2022–2033 (2010)

18. Labati, R.D., Genovese, A., Piuri, V., Scotti, F., Sforza, G.: A decision support system for wind power production. IEEE Trans. Syst. Man Cybern. Syst. **50**(1), pp. 1–15 (2018)

19. Ogliari, E., Niccolai, A., Leva, S., Zich, R.: Computational intelligence techniques applied to the day ahead pv output power forecast: phann, sno and mixed. Energies **11**, 1487 (2018)

20. MacDougall, P., Kosek, A.M., Bindner, H., Deconinck, G.: Applying machine learning techniques for forecasting flexibility of virtual power plants. In: 2016 IEEE Electrical Power and Energy Conference (EPEC), pp. 1–6 (2016)

21. Bouchekara, H.R.E.H., Chaib, A.E., Abido, M.A.: Multiobjective optimal power flow using a fuzzy based grenade explosion method. Energy Syst. **7**(4), 699–721 (2016)

22. Liwei, J., Li, H., Zhao, J., Chen, K., Tan, Q., Tan, Z.: Multi-objective stochastic scheduling optimization model for connecting a virtual power plant to wind-photovoltaic-electric vehicles considering uncertainties and demand response. Energy Convers. Manag. **128**, 160–177 (2016)

23. Rädle, S., Mast, J., Gerlach, J.: Anwendung eines heuristischen approximationsverfahrens zur multikriterien optimierung von energieclustern. In: 60. Workshop der Multi Projekt Chip-Gruppe (MPC), pp. 19–27. Konstanz, Germany. IEEE German Section Solid-State Circuit Society (2018)

24. Mast, J., Rädle, S., Gerlach, J., Bringmann, O.: Development of a modelling and simulation methodology for hierarchical energy system scenarios. Int. J. Smart Grid Clean Energy **8**(4), 383–391 (2019)

25. Gamma, E., Vlissides, J., Helm, R., Johnson, R.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston (1994)

26. Poole, D.L., Mackworth, A.K.: Artifical Intelligence: Foundations of Computational Agents, 2nd edn. Cambridge University Press, Cambridge (2017)

27. Marini, F., Walczak, B.: Particle swarm optimization (pso): a tutorial. Chemom. Intell. Lab. Syst. **149**, 153–165 (2015)

28. Katiyar, S., Nasiruddin, I., Ansari, A.Q.: Ant colony optimization: a tutorial review. In: National Conference on Advances in Power and Control, pp. 99–110. Faridabad, India. Manav Rachna International University (2015)

29. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv e-prints arXiv :1609.04747 (2016)

30. Deutscher Wetterdienst (DWD). Deutscher wetterdienst: climate data center (cdc). https://www.dwd.de/EN/climate_environment/cdc/cdc_node.html. Accessed 09 Aug 2019

31. Wang, Y., Sun, T.: Life cycle assessment of CO_2 emissions from wind power plants: Methodology and case studies. Renew. Energy **43**, 30–36 (2012)

Affiliations

Stefan Rädle[1] · **Johannes Mast[1]** · **Joachim Gerlach[1]** · **Oliver Bringmann[2]**

✉ Stefan Rädle
  raedle@hs-albsig.de

1   Department of Computer Engineering, Albstadt-Sigmaringen University, 72458 Albstadt,
    Germany

2   Department of Embedded Systems, University of Tübingen, 72076 Tübingen, Germany