A New Hybrid-Adaptive Differential Evolution for a Smart Grid Application Under Uncertainty

Fernando Lezama[1], *Member, IEEE,* João Soares[1], *Member, IEEE,* Ricardo Faia[1], Tiago Pinto[2], *Member, IEEE,* and Zita Vale [1], *Senior Member, IEEE*

[1]GECAD-Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development
School of Engineering, Polytechnic of Porto, Porto, Portugal 4200-072
Email: {flzcl, joaps, rfmfa, zav}@isep.ipp.pt
[2]BISITE Research Center
University of Salamanca, Calle Espejo, 2, 37007 Salamanca, Spain
Email: {tpinto}@usal.pt

Abstract—**Power systems are showing a dynamic evolution in the last few years, caused in part by the adoption of smart grid technologies. The integration of new elements that represent a source of uncertainty, such as renewables generation, electric vehicles, variable loads and electricity markets, poses a higher degree of complexity causing that traditional mathematical formulations struggle in finding efficient solutions to problems in the smart grid context. In some situations, where traditional approaches fail, computational intelligence has demonstrated being a very powerful tool for solving optimization problems. In this paper, we analyze the application of Differential Evolution (DE) to address an energy resource management problem under uncertain environments. We perform a systematic parameter tuning to determine the best set of parameters of four state-of-the-art DE strategies. Having knowledge of the sensitivity of DE to the parameter selection, self-adaptive parameter control DE algorithms are also implemented, showing that competitive results can be achieved without the application of parameter tuning methodologies. Finally, a new hybrid-adaptive DE algorithm, HyDE, which uses a new ”$DE/target-to-perturbed\_best/1$” strategy and an adaptive control parameter mechanism, is proposed to solve the problem. Results show that DE strategies with fixed parameters, despite very sensitive to the setting, can find better solutions than some adaptive DE versions. Overall, our HyDE algorithm excelled all the other tested algorithms, proving its effectiveness solving a smart grid application under uncertainty.**

I. INTRODUCTION

Due to the importance of the energy resource management (ERM) in the smart grid field, several mathematical formulations have been successfully proposed in the literature [1]. However, the very dynamic evolution of electrical grids, mainly due to the development of smart grid technologies, has caused that the traditional formulations, which were designed for different scenarios, sometimes cannot deal with the problem efficiently. Challenges in the management of energy resources have been recently identified [2], [3], [4]. Common issues have been consistently highlighted concerning, namely the growing complexity of a large number of distributed energy resources, the increase of electric vehicles (EVs) penetration and the increase of sources of uncertainty. Since the aggregator performs the scheduling of resources for the day-ahead (i.e., the next 24 hours), it relays in the forecast of weather conditions (to predict renewable generation), load demand, EV trips, and market prices. However, the assumption of perfect or highly accurate forecast might bring unforeseen consequences into the operation of the grid when the realizations do not follow the expected predictions [5]. Due to this situation, it is desired that the aggregator determines solutions that are robust to the uncertainty inherent in some parameters and the environment. Four aspects of uncertainty that affect the performance of a solution are considered in this paper, namely: a) weather conditions, b) load forecast, c) planned EVs trips, and d) market prices. Regarding the aforementioned aspects, some efforts have been made in the literature to tackle different sources of uncertainty but essentially most proposed models are based on mathematical approaches, namely stochastic models (e.g., two-stage stochastic models [6], [7], [8]) or robust optimization models [9], [10]. Those approaches are usually solved as deterministic Mixed-integer Linear Programming (MILP) models after linearization and conversion of the original formulation. The limitations of those traditional deterministic approaches are that they cannot efficiently incorporate nonlinear functions (e.g., generator quadratic function, AC power flow, and other complex conditions or resources constraints), a large number of resources and considerable accuracy of uncertainty representation (e.g., the number of scenarios in stochastic models). Also, due to different legal aspects of energy markets all over the world, the proposed ERM models do not consider the same business model which adds up more complexity to this research field [3].

In this paper, we propose to solve the ERM problem using evolutionary computation (EC) techniques, namely various versions of the differential evolution (DE) algorithm to tackle the problem with several sources of uncertainty, as highlighted early. DE and other metaheuristics have been applied with satisfactory results to the ERM problems [11], [12], yet few EC works, we are aware of, have considered parameter uncertainty and scenario-based information into the ERM. Since traditional DE is highly dependent on the control parameters' selection, a proper tuning is required to obtain good results but such tuning is typically time consuming and tedious process. Adaptive Control parameters DE versions

solve that problem by selecting the parameters automatically during iterations. Yet, the adaptive versions do not cope well with all the problems with different characteristics, and their performance might be improved in different directions. Hence, we propose a new hybrid-adaptive DE, HyDE, which uses a "$DE/target - to - perturbed\_best/1$" strategy combined with a self-adaptive parameter selection of the jDE algorithm [13]. Our HyDE algorithm is suited to overcome the limitations of traditional techniques, e.g., a higher number of scenarios representing the problems uncertainty, and shows promise performance, excelling fixed and adaptive parameter DE strategies when solving the smart grid problem proposed in this work.

II. PROBLEM FORMULATION

We divide this section into three parts for better understanding: A) Objective function, B) Uncertainty modelling, and C) encoding of solutions and fitness function.

A. Objective Function

The problem can be modelled as a combinatorial Mixed-Integer Linear Programming (MILP) problem due to the presence of continuous, discrete and binary variables. The objective of the aggregator is to minimize operational costs (OC) while maximizing incomes (In). OC are associated with the management of resources by the aggregator and represented by:

$$
\begin{aligned}
OC = & \sum_{t=1}^{T} \left(\sum_{i=1}^{N_{DG}} p_{DG(i,t)} \cdot C_{DG(i,t)} + \sum_{i=1}^{N_k} p_{ext(k,t)} \cdot C_{ext(k,t)} \right) + \\
& \sum_{s=1}^{N_s} \sum_{t=1}^{T} \begin{pmatrix} \sum_{j=1}^{N_{PV}} p_{PV(j,t,s)} \cdot C_{PV(j,t)} + \\ \sum_{e=1}^{N_e} p_{ESS-(e,t,s)} \cdot C_{ESS-(e,t)} + \\ \sum_{v=1}^{N_v} p_{EV-(v,t,s)} \cdot C_{EV-(v,t)} + \\ \sum_{l=1}^{N_L} p_{curt(l,t,s)} \cdot C_{curt(l,t)} + \\ \sum_{l=1}^{N_L} p_{imb-(l,t,s)} \cdot C_{imb-(l,t)} + \\ \sum_{i=1}^{N_{DG}} p_{imb+(i,t,s)} \cdot C_{imb+(i,t)} \end{pmatrix} \cdot \pi(s)
\end{aligned}
\tag{1}
$$

Eq. 1 considers the cost associated with Distributed Generation (DG), external suppliers, discharge of ESS and EVs, DR by direct load control programs (curtailable loads), penalization of non-supplied demand (negative imbalance) and penalization for excess of DG units generation (positive imbalance). On the other hand, the aggregator can receive its incomes (In) from market transactions as follows:

$$
MT = \sum_{s=1}^{N_s} \sum_{t=1}^{T} \left(\sum_{m=1}^{N_m} \left(p_{buy(m,t)} - p_{sell(m,t)} \right) \cdot MP_{(m,t,s)} \right) \cdot \pi(s).
\tag{2}
$$

where bids and offers are allowed into two markets with distinctive characteristics, namely wholesale and local markets.

Notice that both equations, Eqs. 1 and 2, can be linearly combined given as a result the objective minimization function:

$$
\text{Minimize } f(\vec{x}) = OC_{Total}^{Day+1} - In_{Total}^{Day+1}
\tag{3}
$$

where $f(\vec{x})$ is the fitness function that DE aims to optimize. The minimum value of $f(\vec{x})$ is the total cost (or profits if negative) for the energy aggregator.

The Eq. (3) is also subject to resource limit capacities and balanced energy constraints, which enhance the complexity of the problem. The reader can be referred to the appendix section for the nomenclature used in this work, and to [7] to consult a complete mathematical model of this problem (i.e., including all the constraints).

B. Uncertainty modelling

The aggregator relays in forecast of weather conditions (to predict renewable generation), load demand, EV trips, and market prices to perform the scheduling of resources for the day-ahead (i.e., the next 24 hours). However, the assumption of perfect or highly accurate forecast might bring catastrophic consequences into the operation of the grid when the realizations do not follow the expected predictions.

To overcome this issue, we assume that a correct set of scenarios that simulate real-world conditions can be generated considering forecast and associated errors based on historical data or previous experiences. The uncertainty considered in this paper comes from: i) PV renewable sources, ii) load profiles, iii) EVs scheduling, and iv) market prices for wholesale and local markets.

We apply a technique for scenario generation (and scenario reduction) used in [7]. In a first step, a large number of scenarios is generated by Monte Carlo Simulation (MCS). The MCS uses the probability distribution function of the forecasted errors (which can be obtained from historical data) to create a number of scenarios according to:

$$
X_s(t) = x^{forecast}(t) + \mathcal{N}(0, \sigma_{error})
\tag{4}
$$

where $x^{(error,s)}$ is a normal distribution function with zero-mean and standard deviation σ_{error}, and $x^{forecast}(t)$ is the forecasted value of variable x at time t. To simplify, all forecast errors for the uncertain inputs are represented by a normal distribution function. In a second step, a standard scenario reduction technique is applied that excludes scenarios with low probabilities and combines those that are close to each other in terms of statics metrics (for a complete description of these techniques see [7]).

C. Encoding of the solutions and fitness function under uncertainty

The solution representation (i.e., an individual in DE) is a fundamental part in the development of EC algorithms. In this paper, we adopt the same encoding used in the WCCI2018

Competition Evolutionary Computation in Uncertain Environments: A Smart Grid Application [14].

Each solution is therefore encoded as a vector with '6' groups of variables that are repeated sequentially across the 24 periods (hours) of optimization. In the vector representation, all variables, apart from group (2), are continuous variables with bounds matching the power or capacity limits of the associated variables. Group (2), generator binaries, corresponds to binary variables that are used to indicate if a generator is connected ('1' value) or disconnected ('0' value). A special attention is pointed to variables representing PV generation. PV generation cannot be controlled, so even when it is part of the vector solution, the variable corresponding to the PV generation (last variable of group (1)) will take a specific, and thus unalterable, value depending on the considered scenario.

The fitness function in the optimization process is evaluated through all the available scenarios generated by the method presented in Sect. II-B. This means that for a given input solution, the fitness function returns an array of fitness values over all the set of available scenarios. Figure 1 illustrates the design of the fitness function.
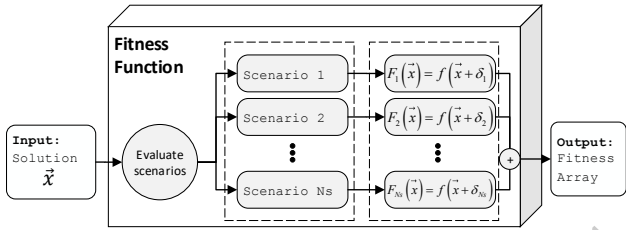

Fig. 1: Fitness function design. A single solution returns a set of values based on the considered scenarios.

A solution $\vec{x}$ is evaluated into different pre-computed scenarios as follows:

$$F_s(\vec{x}) = f(\vec{x} + \delta_s) \qquad (5)$$

where δ_s is the disturbance of variables and parameters in scenario s, and $F_s(\vec{x})$ is the fitness value associated to the s Monte Carlo sampling. Since the fitness of a given individual is a set of values and not a single fitness value (as occurs typically in problems without uncertainty consideration), a performance criterion (e.g., mean value, best value, less standard deviation) should be selected to guide the search and select the best individuals in the population.

In this paper, we have chosen a robust approach, meaning that the fitness of a given solution correspond to the worst-case scenario. Other fitness metrics should be analyzed in further studies. The reader can be referred to [14] for specific details of the encoding and fitness function design.

III. DIFFERENTIAL EVOLUTION ALGORITHM

The basic DE algorithm uses a population (Pop) of individuals $\vec{x}_{j,i,G} = [x_{1,i,G}, ..., x_{D,i,G}]$, where G is the generation number, and $i = [1, ..., NP]$ is the number of individuals in the population, to optimize a function with D variables (i.e., the dimension of the problem). DE iterates by creating new offspring using mutation and recombination operators.

At each generation G, all individuals $\vec{x}_{i,G} \in$ Pop are evaluated in a fitness function. The individual being evaluated is called the target vector ($\vec{x}_{i,G}$). For each target vector $\vec{x}_{i,G}$, a mutant individual $\vec{m}_{i,G}$ is generated using the mutation operator:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) \qquad (6)$$

where $\vec{x}_{r1,G}, \vec{x}_{r2,G}, \vec{x}_{r3,G} \in$ Pop are three random individuals from the Pop, mutually different and also different from the current target vector $\vec{x}_{i,G}$. This is the standard DE mutation strategy also known as DE/rand/1.

After create the mutant vector, the recombination operator combines the mutant individual $\vec{m}_{i,G}$ with the target vector $\vec{x}_{i,G}$ creating the trial vector $\vec{t}_{i,G}$. Particularly, for each component j, where $j = \{1, 2, \ldots, D\}$, we choose the jth element of the $\vec{m}_{j,i,G}$ with probability RC, otherwise from the $\vec{x}_{j,i,G}$. Moreover, a random integer value Rnd is chosen from the interval $[1, D]$ to guarantee that at least one element is taken from $\vec{m}_{i,G}$. The $\vec{t}_{i,G}$ is created as follows:

$$\vec{t}_{j,i,G} = \begin{cases} \vec{m}_{j,i,G} & \text{if } (rand_{i,j}[0,1] < \text{Cr}) \vee (j = \text{Rnd}) \\ \vec{x}_{j,i,G} & \text{otherwise} \end{cases} \qquad (7)$$

After we create $\vec{t}_{i,G}$, it is necessary to verify the boundary constraints of each element of $\vec{t}_{i,G}$ to avoid creating infeasible solutions. If any element of the trial vector violates the constraints, it should be repaired either with a random number in the allowed range, or any other technique such as back-bounce or boundary reinitialization [11].

Finally, the selection operator in the basic DE is a simple rule of elitist done by comparing the fitness between the trial vector $\vec{t}_{i,G}$, and the target vector $\vec{x}_{i,G}$ in the objective function:

$$\text{Pop}_{i,G+1} = \begin{cases} \vec{t}_{i,G} & \text{if } f(\vec{t}_{i,G}) \leq f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} \qquad (8)$$

where $\text{Pop}_{i,G+1}$ is the population of the next generation, that changes by accepting or rejecting new individuals, and $f(.)$ is the fitness function used to measure the performance of an individual (i.e., Eq. (3)).

A. DE mutation strategies with fixed parameters

For each target vector $\vec{x}_{i,G}$, a mutant individual $\vec{m}_{i,G}$ is generated using typically the mutation operator from Eq. 6.In this paper we analyze also some other state-of-the-art DE strategies, namely:

- DE/target-to-best/1: In this strategy, the DE operator is modified to generate base vectors that lie on the line defined by the target vector $\vec{x}_{i,G}$ and the best-so-far found vector $\vec{x}_{best}$:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F(\vec{x}_{best} - \vec{x}_{i,G}) + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \quad (9)$$

This strategy has some similarities with the velocity update formula of the PSO algorithm, in which a stochastic attraction towards the best particle and neighborhood best position are modeled by scaled difference vectors as well [15].

- DE/rand/1 with dither: In this strategy, F is randomly varied (this variation is known as dither) in certain range:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + rand(F,1) * (\vec{x}_{r2,G} - x_{r3,G}) \quad (10)$$

where $rand(F,1)$ is a random number in the range [F, 1] for each member of the population. The dither variation has proved to improve the performance of DE in different problems [15].

- DE/rand/1/either-or: In this strategy, the mutant vector is generated either by a three-vector pure mutation scheme (as in standard DE) with probability PF or as a randomly recombination scheme with probability $1 - PF$:

$$\vec{m}_{i,G} = \begin{cases} \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) & \text{if } (rand < 0.5) \\ \vec{x}_{r1,G} + k(\vec{x}_{r2,G} + \vec{x}_{r3,G} - 2\vec{x}_{r1,G}) & \text{o.w.} \end{cases} \quad (11)$$

where a recommended a value of $k = 0.5(F + 1)$ and $PF = 0.4$ is a good choice for these two parameters [16]. This strategy has shown competitive results against classical DE strategies[15].

B. Adaptive DE algorithms

DE strategies' performance is highly dependent on a proper selection of parameters, as has been shown in many studies [11], [15]. To overcome this situation and obtain acceptable performance, a preliminary phase of tuning is required to determine the best set of parameters for each problem. This tuning phase, either by a trial-and-error approach or a systematic framework, usually requires tedious and time consuming optimization trials [17].

The aforementioned situation has motivated the development of different adaptive parameter control mechanisms to dynamically determine the best set of parameters without involving previous knowledge on the characteristics of the problem. In addition, self-adaptive parameter DE versions are capable of improving the convergence of the algorithms in different problems. Among the vast variety of self-adaptive DE versions, JADE [18] and jDE [19] algorithms have proven to be effective in various benchmark functions with different characteristics. These two adaptive versions are selected in this paper for comparisons purposes. The adaptive mechanisms, very simple and effective, are briefly described next.

1) JADE: The JADE algorithm is an adaptive parameter version of DE based on a mutation strategy called "DE/current-to-*p*best" [18]. The JADE algorithm also incorporate an external archive of solutions to extract historical knowledge about the success and failure of solutions in the evolution process. The "DE/current-to-*p*best" used in JADE is a less greedy generalization of the DE/target-to-best/1 strategy (i.e., Eq.), and is defined as:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F(\vec{x}_{best}^{p} - \vec{x}_{i,G}) + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}^{h}) \quad (12)$$

where $\vec{x}_{best}^{p}$ is randomly chosen as one of the top 100p% individuals of the current population with $p \in (0, 1]$, and $\vec{x}_{r2,G}^{h}$ is chosen randomly from the union of and external archive of solutions A and the current population P (i.e., from $P \cup A$).

In addition, JADE update the control parameters F and Cr at each generation, and for each individual, using simple update rules. The Cr parameter is updated as follows:

$$Cr_i = randn_i(\mu_{Cr}, 0.1) \quad (13)$$

where $randn_i(\mu_{Cr}, 0.1)$ is a random number taken from a normal distribution with mean μ_{Cr} and standar deviation 0.1. μ_{Cr} is also updated at the end of each generation as:

$$\mu_{Cr} = (1 - c) \cdot \mu_{Cr} + c \cdot mean_A(S_{Cr}) \quad (14)$$

where c is a positive number between 0 and 1, and $mean_A(S_{Cr})$ is the arithmetic mean of the set of all successful Cr_i parameters at generation g.

Similarly, the F parameter is updated using a cauchy distribution as:

$$F_i = randc_i(\mu_F, 0.1) \quad (15)$$

The mean value μ_F is also updated at the end of each generation according to:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot mean_L(S_F) \quad (16)$$

where c is a positive number between 0 and 1, and $mean_L(S_F)$ is the Lehmer mean of the set of all successful F_i parameters at generation G. Further details and the application of the adaptive JADE algorithm can be found in [18].

2) jDE: Another simple, yet successful self-adaptive DE variant is the jDE algorithm proposed in [13].

jDE algorithm in its original form incorporates a simple self-adapting control parameter F and Cr mechanism applied in the standard "DE/rand/1" strategy (Eq. 6). New control parameters for each individual in the population are calculated as:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 * F_u, & \text{if } (rand_2 < \tau_1) \\ F_{i,G} & \text{o.w.} \end{cases} \quad (17)$$

$$Cr_{i,G+1} = \begin{cases} rand_3, & \text{if } (rand_4 < \tau_2) \\ Cr_{i,G} & \text{o.w.} \end{cases} \quad (18)$$

where $rand_j$, $j \in 1, 2, 3, 4$ are uniform random values in the range $[0, 1]$, F_l and F_u are the lower and upper limits of F, and τ_1 and τ_2 represent probability factors. In the results section of this paper, the original jDE set of parameters is used, namely $\tau_1 = \tau_2 = 0.1$, and $F_l = 0.1$ and $F_u = 0.9$.

jDE was later improved in [19], where a population multi-population method with aging mechanism was designed to handle dynamic landscapes. With this modification, the jDE

algorithm achieved the first rank in the competition on "Evolutionary Computation in Dynamic and Uncertain Environments" in CEC2009 [20], which indeed attract our attention making it a good choice for the application presented in this paper.

IV. Hybrid-Adaptive Differential Evolution with "$DE/target-to-perturbed\_best/1$"

Self-adaptive DE versions eliminate the necessity of a parameter tuning and typically show acceptable performance in different type of problems. DE versions with fixed parameters, after a proper parameter tuning, have also proven excelled performance in a wide variety of problems. Inspired by this context, in this paper we have developed a hybrid-adaptive DE algorithm (HyDE). Our HyDE combines the DE/target-to-best/1 strategy with a perturbation of the best individual (inspired by EPSO [21]), called "$DE/target-to-perturbed\_best/1$" strategy, and the self-adaptive mechanism of jDE algorithm. The main operator in HyDE is defined as follows:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F_i^1((\vec{x}_{best} \cdot \mathcal{N}(F_i^2, 1)) - \vec{x}_{i,G}) + F_i^3(\vec{x}_{r1,G} - \vec{x}_{r2,G})$$
$$(19)$$

where F_i^1, F_i^2, and F_i^3, are scale factors in the range [0,1] independent for each individual i, updated each iteration following the same rule of jDE algorithm (see Sect. III-B2, Eq. 17). Notice that HyDE uses a perturbation of the $\vec{x}_{best}$ best individual by a random value taken from a Normal distribution with mean F_i^2, which is also an adaptive parameter updated using the same rule of jDE. Such perturbation is inspired in EPSO [21], and has proven to improve the convergence capabilities of the algorithms when solving optimization problems in the energy domain. Our HyDE algorithm also update Cr parameter using the same rule as in jDE algorithm (Eq. 18).

V. Results and discussion

In addition to the proposed HyDE (Sect. IV), one of the contribution of this paper is to compare the performance of DE strategies with fixed parameters with adaptive DE algorithms. To provide a comparison between DE strategies with fixed parameters (requiring a proper set of parameters) and adaptive DE versions, the results section is divided into two parts. In a first part, we conduct a parameter tuning methodology (as in [11]) to provide the most suitable values for F, Cr and NP parameters for the standard DE/rand/1 strategy and the three DE strategies presented in Sect. III-A. In the second part, we compare the DE strategies (with the most-suitable fixed setting found in part 1) with two of the most popular adaptive DE versions (i.e., JADE and jDE), and also our new adaptive DE with Target-to-Best-Perturbation, HyDE. The experiments were performed using MATLAB 2014b 64 bits in a computer with Intel Xeon(R) E5-2620v2@2.1 GHz processor with 16 GB of RAM running windows 10.

The case study used in this paper is based on a 25-bus microgrid that represents a residential area with 6 DGs (5 dispatchable units and 1 PV generator), 1 external supplier, 2 ESSs, 34 EVs, and 90 loads with demand response capability. Moreover, it is considered that two markets (wholesale and local) are available for buy/sale energy for the day-ahead 24 hours. Table I outlines the resources available in the microgrid.

TABLE I: Available Energy Resources

Energy resources		Prices (m.u./kWh)	Capacity (kW)	Units
Dispatchable DGs		0.07-0.11	10-100	5
External suppliers		0.07-0.16	0-150	1
ESS	Charge	-	0-16.6	2
	Discharge	0.03	0-16-6	
EV	Charge	-	0-111	34
	Discharge	0.06	0-111	
DR curtailable loads		0.04	4.06-8.95	90
Forecast (kW)				
Photovoltaic		-	0-106.81	1 (17 agg)
Load		-	35.82-83.39	90
Limits (kW)				
Market 1 (WS)		0.021-0.039	0-85	1
Market 2 (LM)		0.021-0.039	0-40	1

A. Tuning of the parameters

We carried out two experiments for the tuning of parameters. In the first experiment, we fixed $NP = 30$ and $GEN = 100$ for all DE strategies, and performed a swept of parameters F and Cr in the range [0,1] in steps of 0.1 (as in [11]). The swept of parameters allows us to try all combinations of parameters F and Cr. With the help of heatmaps showing the average results of each combination of F and Cr in the [0,1] range, it is possible to find the combination of these parameters that lead to best performance.

Figure 2 shows heatmaps to represent the average fitness value when varying F and Cr parameters. In these figures, a darker color represents a better fitness (i.e., a low value of Eq. (3)), whereas a lighter color represents a poor performance. It can be noticed that all DE strategies have a different set (or area) of parameters were they perform better. This is a very important result to point out, since highlight the importance of a parameter tuning methodology when fixed parameter DE strategies are used. For instance, DE/rand/1 present a good performance when both F and Cr parameters are chosen in the range of [0,0.4] (see Fig. 2a). On the other hand, DE/target-to-best/1 shows a better performance when F is in the range of [0.5,0.7] and Cr in the range [0.2,0.6] (see Fig. 2b).

Overall, the parameter tuning clearly shows that the most-suitable combination of these parameters is different depending on the DE strategy, and a bad selection of parameters might lead to a poor performance for this case study. To summarize, Table II present the most-suitable setting of parameters F and Cr. The table also includes the average fitness value, standard deviation (std), minimum and maximum values, and execution time after the ten runs. It can be seen that DE/rand/1, the classic and simple version of DE, presents the best average fitness (with a slightly higher standard deviation) of all the tested DE strategies.
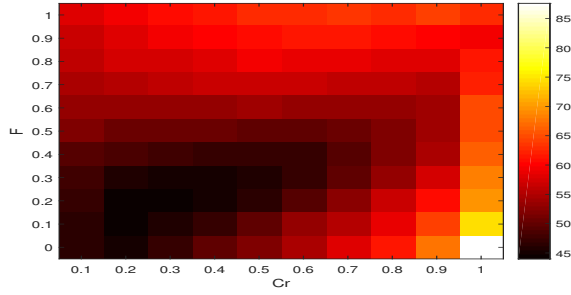
In the second experiment of the parameter tuning, we fixed the values of F and Cr to the ones from Table II, and varied

NP in the range [10,70] in steps of 10. Ten optimization runs were done for each strategy. Since increasing the size of the population leads to more function evaluation (FE) in each iteration (i.e., NP evaluations are required in each generation),
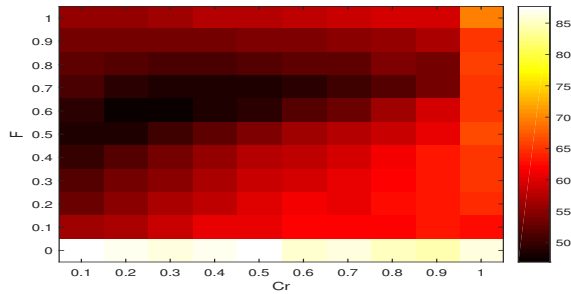
TABLE II: Best DE tuning of F and Cr parameters.

Strategy	F	Cr	Fitness $\pm$ Std	(Min-Max)
DE/rand/1	0.2	0.3	43.98 ± 0.92	(42.70-45.51)
DE/target-to-best/1	0.6	0.2	46.89 ± 0.47	(46.34-47.69)
DE/rand/1 with dither	0	0.4	47.64 ± 0.44	(46.84-48.22)
DE/rand/1/either or	0.3	0.2	44.16 ± 0.45	(43.25-44.74)

*All algorithms used a fixed NP=30 and Gen=100.

we fixed also the number of functions evaluations to 5000, and calculate the number of generations according to $GEN = \lceil 5000/NP \rceil$.

Figure 4 shows the mean fitness value of the DE strategies when NP is varied. It is important to notice that all DE strategies are sensitive also to a proper selection of NP. From this analysis, we can observe that a small NP lead to most of the DE strategies to poor performance. A population of NP= 30 results in better performance for all the DE strategies. Contrary to what intuition might suggest, increasing the value of the population (i.e., NP > 30) is not beneficial for any of the tested strategies.

Additionally, we tested the behavior of DE strategies when the number of generation were increased systematically from 100 to 500 in steps of 100 generations. We noticed that the quality of the solution improves when the number of generations grows for all the DE strategies (i.e., the best fitness values for all the strategies were found with Gen=500). However, more generations imply more function evaluations and time, so the user should choose this parameter carefully to avoid an excessive computational time. For this application, a number of generation Gen=1000 (resulting in 30,000 function evaluations when NP=30) was selected, which results in execution times of 5 minutes each experiment.

To close the parameter tuning analysis, we use the best setting of F and Cr for each strategy (reported in Table II), a value of $NP = 30$, and $Gen = 1000$ in Sect. V-B to compare with the adaptive DE versions and our new HyDE.
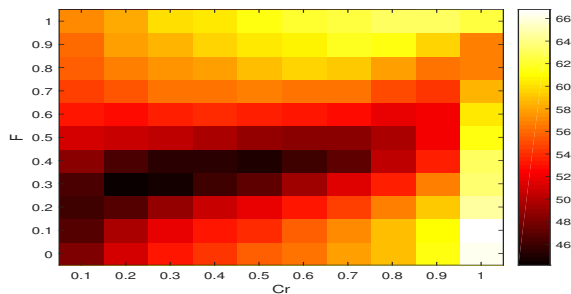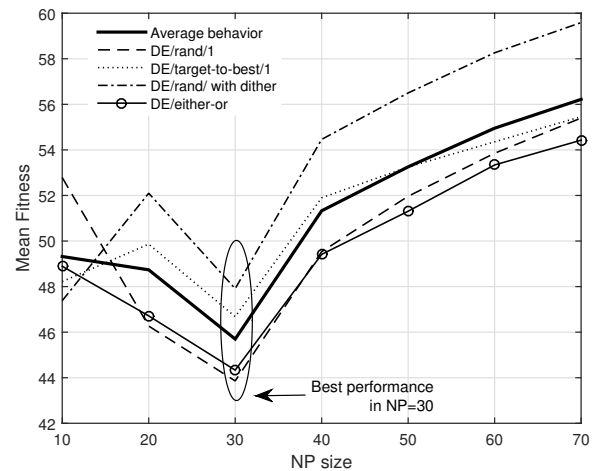

Fig. 2: Heatmap of analyzed DE strategies. (a) DE/rand/1. (b) DE/target-to-best/1. (c) DE/rand/1 with dither. (d) DE/rand/1/either-or.

Fig. 3: Fitness and time relation according NP.

TABLE III: Results and comparison of DE variants and the new HyDE.

DE strategy	Expected Fitness	Std	Min	Max	Time	% Improvement
DE/rand/1	29.628	0.798	27.840	30.168	276.490	14.248
DE/target-to-best/1	31.520	0.695	29.931	31.983	282.750	8.773
DE/rand/1 with dither	31.656	0.799	29.811	32.195	285.392	8.377
DE/either-or-algorithm	31.977	0.648	30.489	32.400	280.184	7.450
JADE	33.781	0.625	32.401	34.221	285.629	2.227
jDE	34.551	0.687	33.068	35.048	274.813	0
HyDE	**17.326**	**0.810**	**16.052**	**18.267**	**289.843**	**49.855**

B. Performance of HyDE and comparison with diverse DE strategies.

In the second part of the results, we compare the DE strategies (with the best set of fixed parameters found), two self-adaptive DE algorithms (JADE and jDE), and our new HyDE algorithm. The reported results correspond to the average of 30 runs.

Figure 4 shows the average worse-case scenario fitness convergence of the tested algorithms. From the figure, it can be noticed that DE/rand/1 presents the best performance from the fixed parameters DE strategies. It can be also seen that the self-adaptive version of DE perform slightly worse than the DE strategies with fixed parameters. However, we must keep in mind that the self-adaptive DE versions can achieve an acceptable performance without any tuning in the first place, which turns out in saving considerable amount of time compared with DE strategies with fixed parameters. Our new HyDE, however, does not require any parameter tuning and presents outstanding performance for this application, achieving a worst-case fitness value of around 20 m.u., and defeating both, adaptive and fixed, DE versions tested in this paper.

Since each final solution provided by the algorithms is subject to the uncertainty of some parameters, an analysis of the robustness of the solutions is worth it. We have

Fig. 4: Convergence of the average worst-case value over 30 runs.

computed the value of the solutions over all the available scenarios (10 scenarios have been considered) to appreciate the sensitive of the solutions to these variations. Table III presents the expected mean fitness value, standard deviation, minimum and maximum values and execution time of the final solutions over all the considered scenarios. Table III also provides the percentage of improvement with regard to the worse performance algorithm, i.e., the jDE algorithm. It can be noticed that all the algorithms present similar execution times (slightly below 5 minutes each). The two self-adaptive versions achieve an expected fitness around 33 m.u., which is slightly worse to the one obtained with the fixed parameter DE strategies (after an extensive parameter tuning), which was about 30 m.u. on average. HyDE showed excelled convergence capabilities, achieving a fitness value of 17.32 m.u., which represents the best performance of the tested algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper, we compared fixed parameters and self-adaptive DE algorithms with a new proposed DE algorithm HyDE, applied to an ERM problem in SGs under uncertainty. DE strategies with fixed parameters are a simple, yet effective algorithms that depend on the proper selection of a few control parameters, namely F, Cr, and NP, to provide acceptable solutions. Such proper selection of parameters can be achieved by a systematic, yet time consuming, preliminary optimization tuning. Self-adaptive DE versions eliminate such tuning, being able to achieve a similar performance for the analyzed SG problem. Motivated by this situation, we have combined a fixed parameter strategy, the DE/target-to-best/1, with the self-adaptive mechanism of parameters from jDE. Our new adaptive algorithm, HyDE, also incorporates a perturbation of the best individual inspired by EPSO and IMPSO. Results showed that the fixed parameter DE versions, if a proper tuning of parameters is conducted, show better performance of other self-adaptive algorithms. However, the tested self-adaptive versions, JADE and jDE, obtain competitive results without any tuning of parameters, saving valuable time in this process. Overall, our proposed HyDE algorithm presents excelled performance in this application, obtaining the best results of the tested algorithms. As future work, the rational used to develop HyDE can be adapted to other metaheuristics. In addition, an extension of this paper comparing HyDE with a broader set of EA and showing its performance in a set of benchmark functions as well as problems with different characteristics in the context of SGs is of special interest.
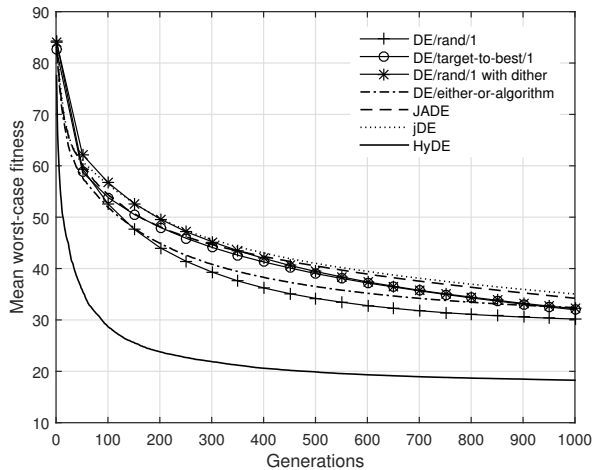
Appendix

Indices:

t	period
i	DG units
j	PV units
k	external suppliers
l	loads
m	markets
e	ESS
v	EVs
s	scenarios

Parameters:

T	number of periods
N_{DG}	number of DG
N_{PV}	number of PV
N_k	number of external suppliers
N_l	number of loads
N_m	number of markets
N_e	number of ESS
N_v	number of EVs
N_s	number of scenarios
C_{DG}	generation cost of DG (m.u./kWh)
C_{PV}	cost of PV generation (m.u./kWh)
C_{ext}	energy price of external supplier (m.u./kWh)
C_{ESS-}	discharging cost of ESS (m.u./kWh)
C_{EV-}	discharging cost of EV (m.u./kWh)
C_{curt}	Load curtailment cost (m.u./kWh)
C_{imb}	imbalance cost (m.u./kWh)
MP	electricity market price (m.u./kWh)
P_{load}	Forecasted load (kW)
P_{PV}	photovoltaic generation (kW)
$\pi(s)$	Probability of scenario s

Variables:

OC_{Total}^{Day+1}	total day-ahead operation cost (m.u.)
In_{Total}^{Day+1}	total day-ahead income (m.u.)
P_{DG}	active power generation of DG (kW)
P_{ext}	active power of external supplier (kW)
P_{ESS-}	discharging power of ESS (kW)
P_{EV-}	discharging power of EV (kW)
P_{curt}	Power reduction of Load (kW)
P_{imb-}	Non-supplied power to load (kW)
P_{imb+}	Exceeded power of DG unit (kW)
P_{buy}	Power buy to the market (kW)
P_{sell}	Power sell to the market (kW)
x_{DG}	Binary variable for DG unit status

References

[1] W. L. Theo, J. S. Lim, W. S. Ho, H. Hashim, and C. T. Lee, "Review of distributed generation (DG) system planning and optimisation techniques: Comparison of numerical and mathematical modelling methods," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 531 – 573, 2017.

[2] J. Hu, H. Morais, T. Sousa, and M. Lind, "Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1207 – 1226, 2016.

[3] T. Soares, M. Silva, T. Sousa, H. Morais, and Z. Vale, "Energy and reserve under distributed energy resources managementday-ahead, hour-ahead and real-time," *Energies*, vol. 10, no. 11, p. 1778, 2017.

[4] P. Kaufmann, O. Kramer, F. Neumann, M. Wagner *et al.*, "Optimization methods in renewable energy systems design," *Renewable Energy*, vol. 87, pp. 835–836, 2016.

[5] M. Severini, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Energy management with support of PV partial shading modelling in micro grid environments," *Energies*, vol. 10, no. 4, 2017.

[6] H. Liang and W. Zhuang, "Stochastic modeling and optimization in a microgrid: A survey," *Energies*, vol. 7, no. 4, pp. 2027–2050, 2014.

[7] J. Soares, B. Canizes, M. A. F. Ghazvini, Z. Vale, and G. K. Venayag-amoorthy, "Two-stage stochastic model using benders decomposition for large-scale energy resource management in smart grids," *IEEE Transactions on Industry Applications*, vol. 53, no. 6, pp. 5905–5914, Nov 2017.

[8] W. Su, J. Wang, and J. Roh, "Stochastic energy scheduling in microgrids with intermittent renewable energy resources," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1876–1883, July 2014.

[9] M. Rahimiyan and L. Baringo, "Strategic bidding for a virtual power plant in the day-ahead and real-time markets: A price-taker robust optimization approach," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2676–2687, 2016.

[10] L. Ju, Z. Tan, J. Yuan, Q. Tan, H. Li, and F. Dong, "A bi-level stochastic scheduling optimization model for a virtual power plant connected to a wind–photovoltaic–energy storage system considering the uncertainty and demand response," *Applied Energy*, vol. 171, pp. 184–199, 2016.

[11] F. Lezama, L. E. Sucar, E. M. de Cote, J. Soares, and Z. Vale, "Differential evolution strategies for large-scale energy resource management in smart grids," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 2017, pp. 1279–1286.

[12] F. Lezama, E. M. de Cote, L. E. Sucar, J. Soares, and Z. Vale, "Evolutionary framework for multi-dimensional signaling method applied to energy dispatch problems in smart grids," in *International Conference on Intelligent System Application to Power Systems*, 2017, pp. 1–6.

[13] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[14] F. Lezama, J. Soares, Z. Vale, and J. Rueda. (2017, Dec.) Guidelines for the CEC 2018 competition on evolutionary computation in uncertain environments: A smart grid application. http://www.gecad.isep.ipp.pt/WCCI2018-SG-COMPETITION/.

[15] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2011.

[16] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[17] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[18] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

[19] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec, and V. Zumer, "Dynamic optimization using self-adaptive differential evolution," in *IEEE Congress on Evolutionary Computation*, May 2009, pp. 415–422.

[20] C. Li, S. Yang, T. Nguyen, E. Yu, X. Yao, Y. Jin, H. Beyer, and P. Suganthan, "Benchmark generator for CEC 2009 competition on dynamic optimization," Tech. Rep., 2008.

[21] V. Miranda and N. Fonseca, "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems," in *IEEE/PES Transmission and Distribution Conference and Exhibition*, vol. 2, Oct 2002, pp. 745–750.