Differential Evolution Strategies for Large-Scale Energy Resource Management in Smart Grids

Fernando Lezama[1], Luis Enrique Sucar[1],
Enrique Munoz de Cote[1,2]
[1]Computer science department

Instituto Nacional de Astrofísica, Óptica y Electrónica
(INAOE), México
[2]PROWLER.io Ltd, Cambridge, United Kingdom
{f.lezama,e.sucar,jemc}@inaoep.mx

João Soares, Zita Vale
GECAD-Research Group on Intelligent Engineering and
Computing for Advanced Innovation and Development
Polytechnic of Porto, Portugal
{joaps,zav}@isep.ipp.pt

ABSTRACT

Smart Grid (SG) technologies are leading the modifications of power grids worldwide. The Energy Resource Management (ERM) in SGs is a highly complex problem that needs to be efficiently addressed to maximize incomes while minimizing operational costs. Due to the nature of the problem, which includes mixed-integer variables and non-linear constraints, Evolutionary Algorithms (EA) are considered a good tool to find optimal and near-optimal solutions to large-scale problems. In this paper, we analyze the application of Differential Evolution (DE) to solve the large-scale ERM problem in SGs through extensive experimentation on a case study using a 33-Bus power network with high penetration of Distributed Energy Resources (DER) and Electric Vehicles (EVs), as well as advanced features such as energy stock exchanges and Demand Response (DR) programs. We analyze the impact of DE parameter setting on four state-of-the-art DE strategies. Moreover, DE strategies are compared with other well-known EAs and a deterministic approach based on MINLP. Results suggest that, even when DE strategies are very sensitive to the setting of their parameters, they can find better solutions than other EAs, and near-optimal solutions in acceptable times compared with an MINLP approach.

CCS CONCEPTS

•**Computing methodologies** → *Search methodologies;* •**Applied computing** → **Engineering;**

KEYWORDS

Differential Evolution, Energy Resource Management, Evolutionary Algorithms, Optimization, Smart Grid

1 INTRODUCTION

Over the last years, the electric grid has evolved into an advanced power network, widely known as Smart Grid (SG), with the aim of running a high number of Distributed Energy Resources (DER) under market conditions [19]. However, some issues should be considered before high penetration of DER into the SG. For instance, DER are not able to participate in the current markets because their small size and unpredictable nature (e.g., the wind and solar generation) limit their contribution to the grid operation causing economics penalties due to unexpected unbalances. One way to overcome such issues is the aggregation of DER through Virtual Power Plant (VPP) enabling same visibility, controllability, and market functionality as conventional generation [5].

The problem addressed in this paper concerns the day-ahead Energy Resource Management (ERM) to provide efficient operational support of VPPs into the SG. The VPP should efficiently control energy resources with the objective of maximizing profits by reducing the need to buy energy from the day-ahead market or external suppliers at high prices. To achieve this task in realistic scenarios, the ERM must consider a huge variety of resources including Electric Vehicles (EVs), Energy Storage Systems (ESS) and different types of Distribute Generation (DG) [9]. Moreover, the incorporation of Demand Response (DR) programs, Vehicle-to-grid (V2G) functionalities, market bids and external suppliers participation, along with AC network power balance constraints turns the ERM problem into a Mixed Integer Non-Linear Problem (MINLP) [13, 17]. It is well-known that an MINLP is tough to solve with classical approaches, but are suitable to be addressed with Evolutionary Algorithms (EA) [8, 18].

In this paper, we propose the application of Differential Evolution (DE) optimization to the large-scale ERM problem in the SG. DE is a very simple, yet efficient, stochastic global optimizer. In its standard form, DE is initialized with a random set of candidate solutions and in every iteration follows similar computational steps as employed by most of the standard EAs (i.e., mutations, recombination, and selection). However, DE differs from well-known EAs in the fact that it mutates base vectors (secondary parents) with scaled differences of distinct members from the current population. These differences tend to adapt to the fundamental levels of objective landscape improving the search moves of the algorithm. A complete theoretical analysis and successful applications of DE in diverse domains of science and technology can be found in [4].

We perform a parameter analysis of different state-of-the-art DE strategies, namely DE/rand/1, DE/target-to-best/1, DE/rand/1 with dither and DE/rand/1/either-or algorithm. After determining the best parameter setting for each DE strategy, we provide a comparison not only between them, but also with some other EAs such as Particle Swarm Optimization (PSO) [7], Quantum-behaved PSO (QPSO)[16], and Dynamic Search Algorithm (DSA) [2]. Moreover, we present results of a deterministic approach based on MINLP as a benchmark. Results indicate that a proper setting of parameters leads DE to find better solutions than other EAs and near-optimal solutions in acceptable time.

2 PROBLEM FORMULATION

In the SG context, a VPP aims to reduce operational costs (Eq. (1)) while maximizing the incomes (Eq. (2)) over a time horizon T. For instance, the ERM model under study includes, but is not limited to, decision variables for generation power of energy resources, the commitment of DG units, ESSs and EVs schedules, DR loads, all of these for each unit and each considered period. Besides, voltage and angles in each bus during the scheduling period should be considered as well.

The ERM model used in this work consider the following assumptions: a) The VPP can predict the demand profiles with small errors by implemented high accuracy models of load forecasts developed in [10, 15]. b) The VPP is equipped with advanced Information and Communication Technologies (ICT) infrastructure for actively monitoring and controlling guaranteeing optimal operation of the grid. c) Discharging prices of EESs and EVs cover degradation costs and remunerates discharging services. d) The VPP can sell/buy energy to the main grid and external entities.

2.1 Objective Function

The day-ahead ERM model is based on a recent work [13], namely multi-period optimization with 24 periods of 1 hour each. The objective function is formed by two equations describing the operational costs and incomes that a VPP aims to optimize. On one hand, Eq. (1) models the operational costs of electricity acquisition, and includes the generation costs of DG (first term), the generation curtailment (GCP) and non-supplied demand (NSD) penalizations (second and third terms), the cost of DR programs (fourth term), the cost of external suppliers energy (fifth term), and the cost associated with the discharging of EVs and ESS (sixth and seventh terms respectively).

$$\text{Minimize } \boldsymbol{OC_{Total}^{Day+1}} =$$

$$\sum_{t=1}^{T}\left(\begin{array}{l}\sum_{I=1}^{N_I} P_{DG(I,t)} \cdot C_{DG(I,t)} + \sum_{I=1}^{N_I} P_{GCP(I,t)} \cdot C_{GCP(I,t)} + \sum_{L=1}^{N_L} P_{NSD(L,t)} \cdot C_{NSD(L,t)} \\ + \sum_{L=1}^{N_L} P_{LDR(L,t)} \cdot C_{LDR(L,t)} + \sum_{J=1}^{N_J} P_{Sup(J,t)} \cdot C_{Sup(J,t)} \\ + \sum_{K=1}^{N_K} P_{Sdis(K,t)} \cdot C_{Sdis(K,t)} + \sum_{M=1}^{N_M} P_{Vdis(M,t)} \cdot C_{Vdis(M,t)}\end{array}\right) \quad (1)$$

On the other hand, Eq. (2) models the incomes obtained by selling energy to loads (first term) and the pool market (second term), and also the incomes for charging EVs and ESS (fourth and fifth terms respectively).

$$\text{Maximize } \boldsymbol{In_{Total}^{Day+1}} =$$

$$\sum_{t=1}^{T}\left(\begin{array}{l}\sum_{L=1}^{N_L} P_{Load(L,t)} \cdot U_{Load(L,t)} + \sum_{N=1}^{N_N} P_{Sell(N,t)} \cdot U_{Sell(N,t)} \\ + \sum_{K=1}^{N_K} P_{Scha(K,t)} \cdot U_{Scha(K,t)} + \sum_{M=1}^{N_M} P_{Vcha(M,t)} \cdot U_{Vcha(M,t)}\end{array}\right) \quad (2)$$

Both equations (i.e., Eq. (1) and Eq. (2)) can be combined into a single equation representing the profits that a VPP can obtain, such as:

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

where $f(\vec{x})$ is the fitness function that EAs aims to optimize to increase profits of the VPP. Moreover, Eq. (3) is also subject to nonlinear network constraints and resource limit capacities which enhance the complexity of the problem. The reader can be referred to [13] to consult the complete mathematical model (i.e., including all the network constraints); to the appendix section for the nomenclature used in this work; and to Sect. 3.1 for the details on the structure of a solution $\vec{x}$ adopted here (e.g., type of variables, dimensionality, bounds).

3 DIFFERENTIAL EVOLUTION ALGORITHM

Differential Evolution (DE) is a search strategy that can be used to maximize or minimize any given multi-dimensional function $f(x_1, x_2, ..., x_D)$, where D is the number of variables (i.e., the dimension of the problem). The classic 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 $i = [1, ..., \text{NP}]$ is the number of individuals in the population. DE iterates by creating new offspring using mutation and recombination operators. Then, it selects the ones with better fitness (e.g., evaluating them in an objective function) replacing the worse individuals in Pop. The process is repeated for a fixed number of generations (GEN) until a satisfactory solution is obtained or a computational condition is reached [14].

DE has three crucial control parameters: the mutation constant ($F \in [0, 1]$), which controls the mutation strength; the recombination constant ($Cr \in [0, 1]$), which increases the diversity in the mutation process; and the population size (NP), which is an integer that depends on the dimension of the problem or the DE strategy selected. Throughout the execution process, the user defines F, Cr, and NP. These parameters are maintained fixed throughout the execution of the algorithms [1]. In the next subsection, we briefly describe the specifics (including the mutation DE strategies used in this paper) of the DE algorithm applied to the ERM problem.

3.1 Encoding of the Individuals

The encoding of individuals that represent solutions to the problem is a crucial part of any evolutionary algorithm. In DE (and the other EAs used in this paper), a solution vector $\vec{x} = (x_1, x_2, ..., x_D)$ should contain the sufficient information to evaluate the objective function of Eq. (3), i.e., continuous variables corresponding to active and reactive power of DG and charge and discharge values for V2G and ESS; and also DG units binary variables indicating a connection ('1' value) or a disconnection ('0' value) of the corresponding unit. Since the analyzed DE strategies and EAs used in this paper are not designed to work with binary variables, such variables are treated as continuous values in the range of [0,1]. When evaluating the objective function, a value in the range [0,0.5] will correspond to a binary '0', otherwise it will correspond to 1.

For instance, the dimension of a solution $\vec{x}$ considering a future scenario (Sect. 4) of a distribution network with 66 DG units and 1800 V2G contracts, for the day-ahead optimization problem, in a

[1] We are aware of adaptive-DE variants in the literature. However, in an initial step, we decided to use off-the-shelf DE strategies to see it such strategies are good enough to solve this particular problem

scheduled for 24 periods intervals, would correspond to 66 DG×24 periods× 3 (active and reactive power and DG units binary variables) + 1,800 V2G × 24 periods × 2 (discharge and charge active power)= 95,904 variables. When including network constraints and more resources such as demand response, this value can easily reach 500,000 variables without even increasing the number of V2G resources.

3.2 DE Mutation Strategies

At each generation G, all individuals $\vec{x}_{i,G} \in$ Pop are evaluated. 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 a particular mutation operator that depends on the selected DE strategy. There have been many modifications of the DE mutation operator in the literature. The reader can refer to [4] for further explanation of the DE strategies used in this paper, and others as well. In the next subsection, we briefly present four well-known state-of-the-art DE strategies used in this paper to solve the ERM problem.

3.2.1 *DE/rand/1.*

This is the standard DE mutation strategy. In this strategy, the mutant individual $\vec{m}_{i,G}$ is created by the linear combination of random solutions as follows:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) \tag{4}$$

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 $\vec{x}_{i,G}$.

3.2.2 *DE/target-to-best/1.*

In this strategy, the term “target-to-best” means that base vectors are chosen to lie on the line defined by the target vector $\vec{x}_{i,G}$ and the best-so-far vector $\vec{x}_{best}$ as follows:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F(\vec{x}_{best} - \vec{x}_{i,G}) + F(\vec{x}_{r1,G} - \vec{x}_{r2,G}) \tag{5}$$

3.2.3 *DE/rand/1 with dither.*

In this strategy, a random variation of F parameter, known as dither in the literature, is incorporated to the standard mutation operator as follows:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + rand(F,1) * (\vec{x}_{r2,G} - x_{r3,G}) \tag{6}$$

where $rand(F,1)$ indicates that F value is varied randomly in the range $[F,1]$ for each member of the population. The so-called dither variation has proved to improve the performance of DE in different problems [3].

3.2.4 *DE/rand/1/either-or algorithm.*

In this strategy, the mutant vector is generated either by a three-vector pure mutation scheme (such as standard DE) with probability P_F or as a randomly recombinant scheme with probability $1 - P_F$:

$$\vec{m}_{i,G} = \begin{cases} \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}) & \text{if } (rand < P_F) \\ \vec{x}_{r1,G} + k(\vec{x}_{r2,G} + \vec{x}_{r3,G} - 2\vec{x}_{r1,G}) & \text{o.w.} \end{cases} \tag{7}$$

where Price *et* al. recommended a value of $k = 0.5(F+1)$ as a good choice for the parameter k for a given F and $P_F = 0.4$ [11]. This strategy has shown competitive results against classical DE strategies [3].

3.3 Recombination Operator

The recombination operator is applied to create the trial vector $\vec{t}_{i,G}$. In this operator, the mutant individual, $\vec{m}_{i,G}$, is combined with the target vector $\vec{x}_{i,G}$ as follows:

$$\vec{t}_{j,i,G} = \begin{cases} \vec{m}_{j,i,G} & \text{if } (rand_{i,j}[0,1] < Cr) \vee (j = \text{Rnd}) \\ \vec{x}_{j,i,G} & \text{otherwise} \end{cases} \tag{8}$$

where Cr represents the probability of choosing the jth element of the $\vec{m}_{j,i,G}$, otherwise from the $\vec{x}_{j,i,G}$. Also, a random integer value Rnd is chosen in the interval $[1,D]$ to guarantee that at least one element is taken from $\vec{m}_{i,G}$.

3.4 Verifying Boundary Constraints

After we create $\vec{t}_{i,G}$, it is necessary to verify variable boundary constraints to avoid creating infeasible solutions. Boundary constraints are very common in real-world applications, e.g., in the ERM problem the max/min power capacity of a DG is giving as a physical parameter that cannot be violated. DE uses different penalty methods to handle boundary constraints violations, such as random initialization or bounce-back methods [4].

In this paper, we applied a simple reinitialization boundary control as follows [13]:

$$\vec{t}_{j,i,G} = \begin{cases} \vec{x}_{j,lb} & \text{if } \vec{t}_{j,i,G} < \vec{x}_{j,lb} \\ \vec{x}_{j,ub} & \text{if } \vec{t}_{j,i,G} > \vec{x}_{j,ub} \end{cases} \tag{9}$$

where $\vec{x}_{j,lb}/\vec{x}_{j,ub}$ corresponds to the lower/upper bound limit of the jth variable. This boundary control method showed to be effective for the application studied in this paper. Other repair methods can be analyzed in future work.

3.5 Fitness and Selection

The selection operator in the basic DE is a simple rule of elitist done by comparing the fitness between the trial vector $\vec{t}_{i,G}$ (originated with some of the above strategies), and the target vector $\vec{x}_{i,G}$ in the objective function:

$$\text{Pop}_{i,G+1} = \begin{cases} \vec{t}_{i,G} & \text{if } f(\vec{t}_{i,G}) \le f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} \tag{10}$$

where $\text{Pop}_{i,G+1}$ is the next generation population, 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 pseudocode of DE algorithm is presented in algorithm 1.

4 RESULTS AND DISCUSSION

The results section is divided into two parts. The first part is devoted to the analysis of the impact of DE parameters in four state-of-the-art DE strategies, namely DE/rand/1, DE/target-to-best/1, DE/rand/1 with dither and DE/rand/1/either-or algorithm. In the second part, after determining the recommended parameter setting for these strategies, extensive experimentation comparing the performance of DE strategies is presented. We also compare the results of DE strategies with some popular EAs, namely PSO, QPSO and DSA, and also a deterministic method using MINLP as a benchmark [13]. The DE strategies, EAs, and deterministic approaches were implemented using Matlab 2014b 64 bits in a computer with an

Algorithm 1 Basic DE algorithm pseudocode

```
Set the control parameters $F \in [0,1]$, $Cr \in [0,1]$ and NP.
Create randomly the initial Pop.
Evaluate fitness (i.e., Eq. (3)) of every individual.
for $G$=1:GEN do
    for $i$=1:NP do
        Select individuals from Pop for mutation
        Apply mutation strategy (any from Eq. (4-7))
        Apply recombination Eq. (8).
        Verify boundary constraints.
        if Boundary constraints are violated then
            modify the infeasible elements. Eq. 9
        end if
        Apply selection operator Eq. (10).
        Update Pop (and the best-so-far individual $\vec{x}_{best}$ for DE/target-to-best/1).
    end for
end for
```

Table 1: Available resources for 33-bus network. [13]

Resources	Price (m.u./kWh)	Capacity (kW)	No. Units
Biomass	0.09	380	4
CHP	0.06	1150	15
Fuel cell	0.15	110	7
Small hydro	0.07	70	2
PV	0.2	0-840	31
Waste-to-energy	0.1	10	1
Wind	0.15	180-890	6
Large wind	0.07	1580-1800	1
External suppliers	0.09-0.3	6200	10
V2G	0.19	0-5720	1800
ESS	0.19	900	15
DR	0.02	600-1170	32

Intel Xeon W3565 processor and 6 GB of RAM running Windows 10.

The case study considered to test the EAs consists in a 33-bus 12.66 kV distribution network adapted from [1]. Such network represents an SG operated by a VPP with projections of DG and V2G penetration levels for the year 2040, as showed in Fig. 1[2].

The 33-bus network scenario includes 67 DGs (with a large wind unit), 10 external suppliers, 15 ESS and 1800 EVs with V2G capabilities. External suppliers are modeled as a substation connected to the main grid in bus 33. Demand Response (DR) with Direct Load Control (DLC) is considered, setting DLC contracts to 0.02 m.u./kWh. The consumers receive this benefit for each unit of energy reduced, instead of paying the VPP contracted supply price of 0.14 m.u./kWh. The selling energy price is set to 0.14 m.u./kWh as well. Moreover, a fleet of 1800 EVs with V2G capabilities is considered with a total energy demand predicted for trips of 13.77 MWh and a total of 2553 trips. The discharging cost for Ev and ESS is set to 0.19 m.u./kWh. The charging/discharging efficiency of EV and ESS is set to 70% and 90 % respectively.

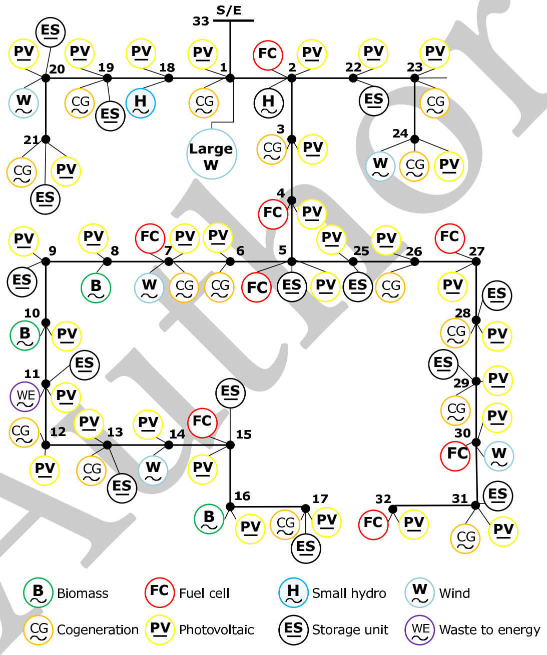


Figure 1: 33-bus network with high penetration of DER [5].

Table 1 presents the available resources of the case study. The considered prices and capacities of DG take into account the observations made in [20]. The scenario of EVs for the case study was developed using the tool presented in [12].

4.1 Tuning of Parameters

In this subsection, we provide an analysis of the impact of DE parameters, namely F, Cr and NP, for this particular case study. To this end, we carried out two experiments. The first one concerns F and Cr parameters, whereas the second is devoted to the impact of NP parameter as well as the effect of the variation in the number of generations.

For the first experiment, we fixed the number of individuals (i.e., NP = 20 for all strategies) and generations (i.e., Gen=100), and varied F and Cr parameters both in the range of $[0,1]$ with a step size of 0.1. The results are the average over ten runs of each combination setting.

Figure 2 shows heat maps plots regarding objective function values 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 fitness. It can be noticed that all the strategies are very sensitive to the selection of these two parameters. For instance, Fig. 2a shows that for DE/rand/1 strategy a value of $Cr = 0.1$ is never a good option, and a low value of F (e.g., in the range of [0,0.3]) combined with a high value of Cr (e.g., in the range of [0,0.3]) yields to poor performance. On the contrary, when F is in the range of [0.3,0.7] along with Cr in the range of [0.3,0.8] the algorithm gives acceptable fitness values.

Similar conclusions can be done for the other strategies observing Fig. 2. The important thing to point out here is that every DE strategy has a particular setting of parameters in which they perform well and others that led to a poor fitness for this case study. As a summary, Table 2 presents the recommended setting of F and Cr that yields to the best average fitness of each strategy. Table 2 also includes the standard deviation (std) and average execution time over the 10 runs.

[2] The complete data can be found in http://www.gecad.isep.ipp.pt/ies/public-data/swevo/.

In the second experiment, having found suitable parameter settings for F and Cr of each strategy (Table 2), the dependence of NP and Gen is examined. First, the number of individuals is varied in the range $10 \leq \text{NP} \leq 100$ with a step size of 10. For a fair comparison, the number of objective function evaluations (OFE) is fixed to 10000 and the number of generation adjusted accordingly such as $\text{Gen} = \lceil 10000/\text{NP} \rceil$. Ten optimization runs are done for each setting combination.

Table 2: Best DE tuning of F and Cr parameters.

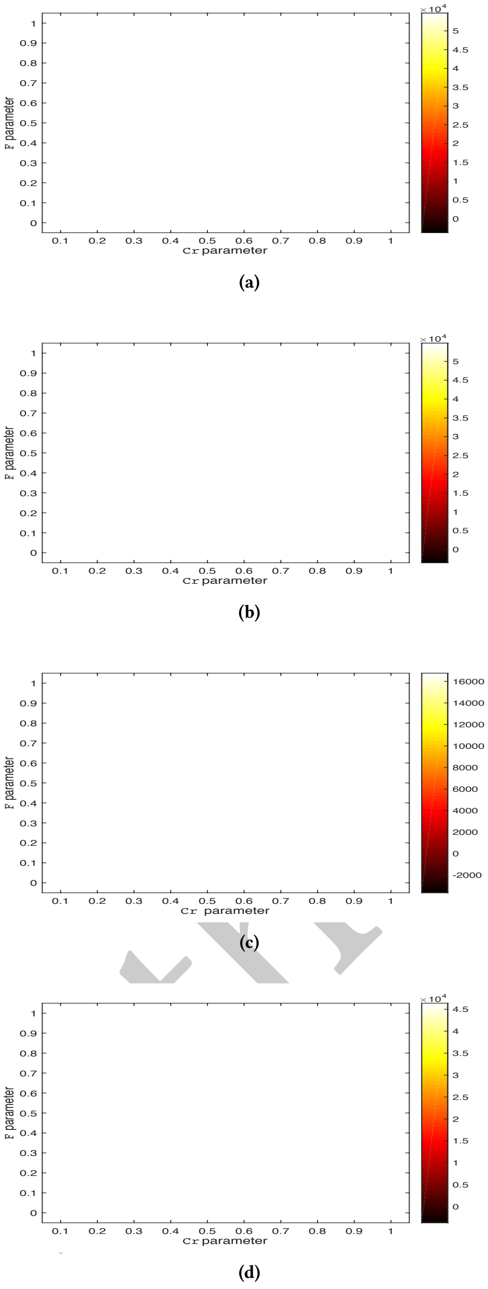
		Fitness	Time (Sec)
	(F, Cr)	**Ave. ± Std**	**Ave. ± Std**
DE/rand/1	(0.3,0.5)	-3225±90	98±2.4
DE/target-to-best/1	(0.8,0.4)	-2991±83	77±0.5
DE/rand/1 with dither	(0,0.3)	-3131±57	70±0.3
DE/rand/1/either-or	(0.4,0.2)	-3107±123	81±1.9

*All algorithms used a fix NP=20 and Gen=100 for this test.

Figure 4 shows the average fitness value in function of the NP parameter over ten runs, of each DE strategy, and also the average of all the strategies. Notice that the quality of solution improves when the parameter NP grows until a value of NP = 30. However, a value of NP > 30 does not bring any improvement to the fitness value, on the contrary, the fitness value gets worse for all strategies. Therefore, a value of NP = 30 is recommended since it presents the best performance in average for all strategies.

Regarding the number of generations, we fixed NP = 30 and varied Gen in the range $[100, 500]$ in steps of 100. 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. A value of 2000 Gen is used in this paper in Sect. 4.2, since empirically experimentation shows that with this value DE strategies return acceptable solution without excessive computational time for the considered case study.

To summarize, the best setting of F and Cr for each strategy are the one reported in Table 2. Those setting in combination with NP = 30 and Gen = 2000 were used in Sect. 4.2 to compare with other heuristics and deterministic methods.



(a)

(b)

(c)

(d)

Figure 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.

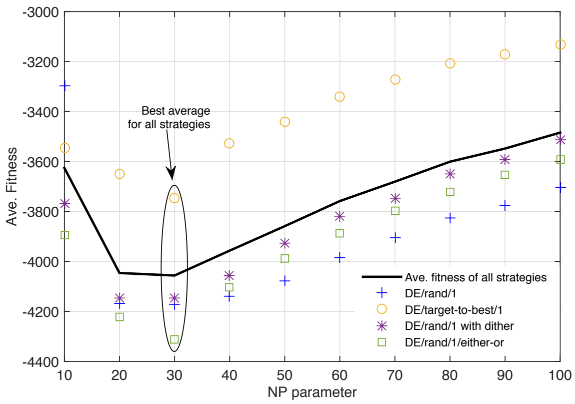


Figure 3: Fitness variation in function of the parameter NP.

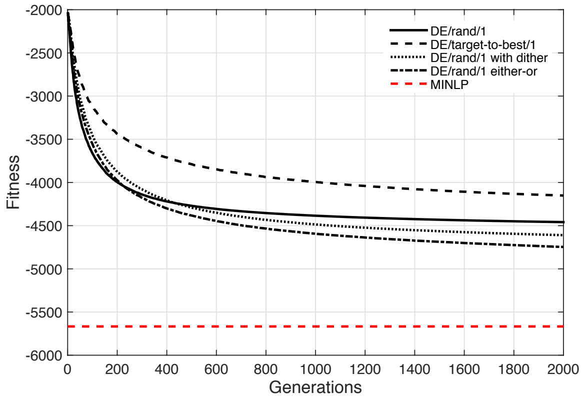


Figure 4: Average convergence over 50 runs for the four DE analyzed strategies.

4.2 Performance of DE Strategies and Comparison with Other Approaches.

In the following the average convergence obtained with the best setting found for F, Cr and NP of each DE strategy (using Gen = 2000) over 50 runs is compared.

Fig. 4 shows the average fitness convergence of the DE strategies using the best setting of parameters for each of them. It is worth noting that a negative fitness value represents a positive profit because the algorithms were setup to minimize Eq. (3). DE strategies have a similar convergence rate since they evolve fast in the first 500 generations and slow down the convergence rate onwards. Among them, DE/rand/either-or algorithm presents the best convergence properties getting a fitness around -4k in the first 150 generations and achieving a final fitness value around -4.7k. On the contrary, DE/target-to-best/1 has the worse performance, achieving the -4k value in generation 1000 and ending with a value around the -4.2k after 2000 generations. DE/rand/1/ and DE/rand/1 with dither have similar performance, with the particularity that DE/rand/1 has better convergence properties than DE/rand/1 with dither in the first 400 generations (in fact, this strategy has the best performance in the first 200 iterations), but is overtaken by DE/rand/1 with dither in the subsequent generations. This result could indicate that some DE strategies work better at the beginning of the evolution process, whereas others improve their performance in advance stages of the evolution.

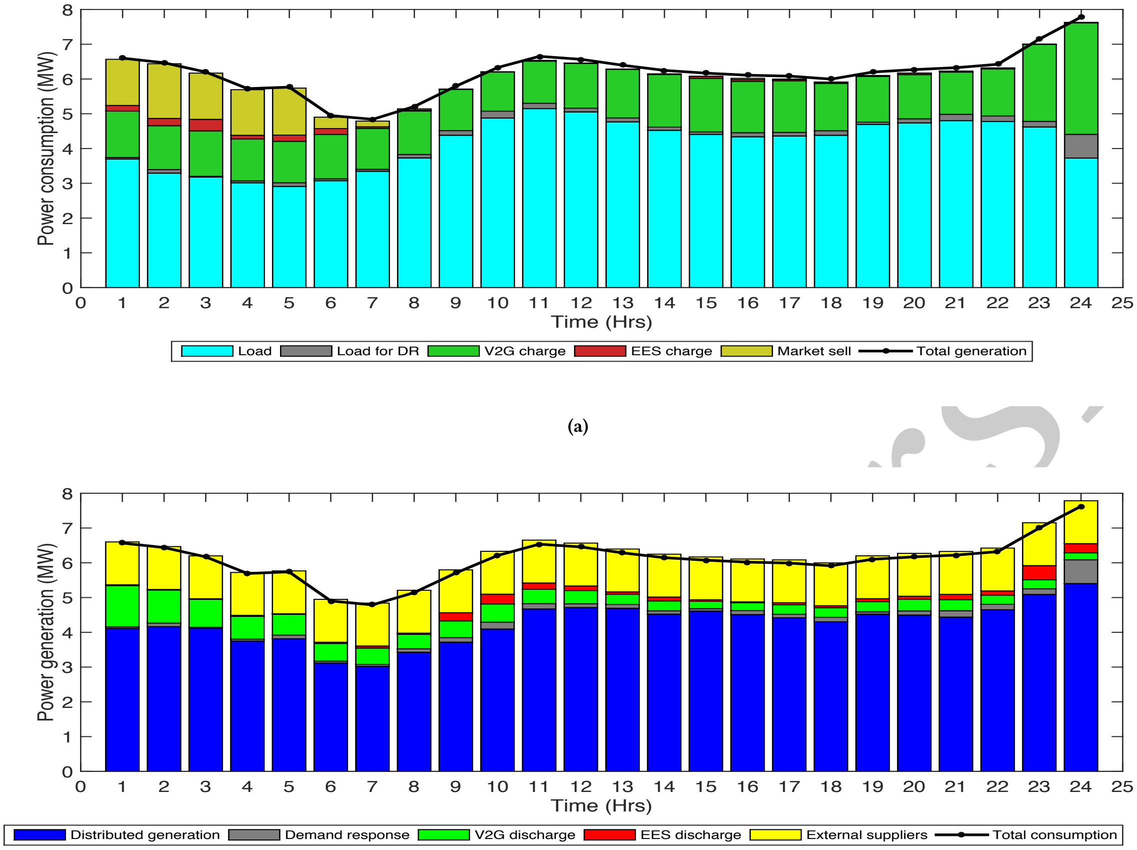
Also, DE strategies are compared with other EAs and a deterministic method based on MINLP taken from [13]. The selected EAs include a standard PSO, a QPSO with linearly decreasing alpha, and a DSA with bijective direction method. The particular parameter setting (obtained with similar experimentation as the one used in this paper) and specifications for these EAs can be found in [13]. All EAs (including DE strategies) run a power flow internally to correct violations of AC network constraints. For a fair comparison, in this paper all the algorithms use a stop criteria of 2000 generations and an equivalent population of NP = 30, (e.g., 30 particles in the case of PSO, QPSO and DSA), which leads to a fixed number of objective function evaluations (OFE) of $30 * 2000 = 60000$ for all the methods. Moreover, as a benchmark, we report the results of MINLP (using GAMS [6]) that considers the full mathematical model of [13] that include AC network equations (e.g., power losses, and voltage/thermal limits in the lines).

Table 3 presents the incomes (In), operational costs (OC), average profits, average time (in minutes), and the number of OFEs over 50 runs for the DE strategies and EAs as mentioned above. Regarding average profits, DE/rand/1/either-or algorithm found the best value, 4746.70 m.u., among the analyzed DE strategies. Moreover, DE strategies overcome all the other EAs, from which QPSO obtained profits of 3809.83 m.u., followed by DSA with 3771.85 and PSO with 3704.57. It is worth noticed that any of the EAs (including the DE strategies) was able to find a fitness as good as the ones obtained with the deterministic MINLP method. However, MINLP took around 834.3 minutes (14 hrs) to find the optimal solution, while DE strategies and the tested EAs took around 60 minutes to achieve the reported results.

Finally, Fig. 5 represents graphically one of the best solutions found with DE/rand/1 for illustrative purposes. For instance, Figs. 5a and 5b show the consumption and generation scheduling respectively. It can be seen that demand is satisfied in all periods with high participation of external suppliers and V2G/ESS discharge capabilities (see yellow, green and red bars in Fig. 5a). Overall, the scheduling found with DE has total power losses of 2.0266 MW, with an average of 0.0844 MW for period. Also notice that EVs charge (green bar in Fig. 5a) increases the consumption during night periods (e.g., period 24). This could indicate that if EVs users decide to charge EVs during night hours, they could modify typical power profiles by generating power peaks during night hours, which is not a regular behavior. However, DR programs can be used in such atypical periods to balance the power and minimize operational costs (e.g., gray bar of Fig. 5).

Table 3: 33-bus network results and comparison against other evolutionary algorithms and exact methods.

Method	IN (m.u.)	OC (m.u.)	Avg. Profits ± std	Time (min)	OFEs
DE/rand/1	19939.98	15480.99	4458.99 ± 20.48	56	60000
DE/target-to-best/1	20356.94	16205.55	4151.39 ± 28.46	58	60000
DE/rand/1 with dither	19798.25	15188.02	4610.24 ± 19.15	57	60000
DE/either-or-algorithm/1	19624.75	14878.05	**4746.70** ± 6.46	59	60000
DSA	20561.37	16789.52	3771.85 ± 109.53	53	60000
PSO	20965.90	17261.33	3704.57 ± 79.51	60	60000
QPSO	21745.39	17935.56	3809.83 ± 34.72	59	60000
MINLP	16968	11301	5667	834.3	-



(a)

(b)

Figure 5: DE/rand/1 ERM solution. a) Power consumption scheduling. b) Power generation scheduling.

5 CONCLUSION AND FUTURE WORK

In this paper, we analyzed DE strategies applied to the large-scale ERM problem in SGs. DE is a simple, yet effective EA that needs the proper selection of a few control parameters, namely *F*, *Cr*, and NP. We showed that the performance of DE strategies depends directly on the setting of such parameters and that each strategy has a different set of *F*, *Cr* and NP that leads to good performance. Overall, when a good set of parameters is selected, all DE strategies were able to find acceptable solutions when solving the ERM case study presented here. Moreover, we compared DE strategies with other well known EAs, such as PSO, QPSO, and DSA, showing that DE strategies obtain competitive results with better fitness values in acceptable times. However, it is worth to notice that the performance of DE strategies and EAs is still worse than the one obtained with an MINLP approach (even when MINLP took about 14 hrs to solve the problem). This margin for improvement suggests some future directions on the application of DE, and EAs in general, to solve the ERM problem. For instance, the incorporation of external knowledge could improve the search capabilities of DE to find solutions with better fitness. Also, the application of adaptive versions of DE that combines strategies and automatically select the best set of parameters is another step for further work. Besides, the consideration of uncertainty and robust optimization is another interesting venue for research in this topic.

A NOMENCLATURE.

Indices:

t	period
I	DG units
L	loads
J	external suppliers
K	ESS
M	EVs
N	energy buyers

Parameters:

T	number of periods
N_I	number of DG
N_L	number of loads
N_J	number of external suppliers
N_K	number of ESS
N_M	number of EVs
$C_{DG(I,t)}$	generation cost of DG I in period t (m.u.)
$C_{GCP(I,t)}$	generation curtailment power cost of DG I in period t (m.u.)
$C_{NSD(L,t)}$	non-supplied demand cost of load L in period t (m.u.)
$C_{LDR(L,t)}$	demand response program cost of load L in period t (m.u.)
$C_{Sup(J,t)}$	energy price of external supplier J in period t (m.u.)
$C_{Sdis(K,t)}$	discharging cost of ESS K in period t (m.u.)
$C_{Vdis(M,t)}$	discharging cost of EV M in period t (m.u.)
$U_{Load(L,t)}$	electricity retail price of load L in period t (m.u./kWh)
$U_{Sell(N,t)}$	electricity sell price to market N in period t (m.u./kWh)
$U_{Scha(K,t)}$	charging price of ESS K in period t (m.u./kWh)
$U_{Vcha(M,t)}$	charging price of EV M in period t (m.u./kWh)
$P_{Load(L,t)}$	day-ahead active power forecast of load L in period t (kW)

Variables:

OC_{Total}^{Day+1}	total day-ahead operation cost (m.u.)
In_{Total}^{Day+1}	total day-ahead income (m.u.)
$P_{DG(I,t)}$	active power generation of DG I in period t (kW)
$P_{GCP(I,t)}$	generation curtailment power of DG I in period t (kW)
$P_{NSD(L,t)}$	non-supplied demand power of load L in period t (kW)
$P_{LDR(L,t)}$	active power reduction of load L in period t (kW)
$P_{Sup(J,t)}$	active power flow in the branch connecting to external supplier J in period t (kW)
$P_{Sdis(K,t)}$	Power discharge of ESS K in period t (kW)
$P_{Vdis(M,t)}$	Power discharge cost of EV M in period t (kW.)
$P_{Sell(N,t)}$	electricity sell price to market N in period t (kW)
$P_{Scha(K,t)}$	Power charge of ESS K in period t (kW)
$P_{Vcha(M,t)}$	Power charge of EV M in period t (kW)

ACKNOWLEDGMENTS

The present work was done and funded in the scope of the projects:
- Sustainability Fund CONACYT-SENER by Consejo Nacional de Ciencia y Tecnología (CONACYT);
- National Center of Innovation in Energy (CEMIE-Eolico);
- H2020 DREAM-GO Project (Marie Sklodowska-Curie grant agreement No 641794);
- UID/EEA/00760/2013 funded by FEDER Funds through COMPETE program and by National Funds through FCT.

REFERENCES

[1] M. E. Baran and F. F. Wu. 1989. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery* 4, 2 (Apr 1989), 1401–1407.

[2] Pinar Civicioglu. 2012. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Computers & Geosciences* 46 (2012), 229–247.

[3] Swagatam Das, Ajith Abraham, Uday K Chakraborty, and Amit Konar. 2009. Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation* 13, 3 (2009), 526–553.

[4] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. 2011. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation* 15, 1 (2011), 4–31.

[5] Pedro Faria, Tiago Soares, Zita Vale, and Hugo Morais. 2014. Distributed generation and demand response dispatch for a virtual power player energy and reserve provision. *Renewable Energy* 66 (2014), 686–695.

[6] GAMS. 2015. http://www.gams.com. (2015).

[7] James Kennedy. 2011. Particle swarm optimization. In *Encyclopedia of machine learning*. Springer, 760–766.

[8] D. S. Kumar, D. Srinivasan, and T. Reindl. 2016. A novel hybrid method for planning and allocation of DGs in medium voltage networks considering voltage regulation and relay coordination. In *IEEE Congress on Evolutionary Computation (CEC)*. 3654–3661.

[9] T. Logenthiran, D. Srinivasan, and T. Z. Shun. 2012. Demand Side Management in Smart Grid Using Heuristic Optimization. *IEEE Transactions on Smart Grid* 3, 3 (Sept 2012), 1244–1252.

[10] Mehdi Motevasel and Ali Reza Seifi. 2014. Expert energy management of a microgrid considering wind energy uncertainty. *Energy Conversion and Management* 83 (2014), 58–72.

[11] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. 2006. *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media.

[12] João Soares, Bruno Canizes, Cristina Lobo, Zita Vale, and Hugo Morais. 2012. Electric vehicle scenario simulator tool for smart grid operators. *Energies* 5, 6 (2012), 1881–1899.

[13] João Soares, Mohammad Ali Fotouhi Ghazvini, Marco Silva, and Zita Vale. 2016. Multi-dimensional signaling method for population-based metaheuristics: Solving the large-scale scheduling problem in smart grids. *Swarm and Evolutionary Computation* 29 (2016), 13 – 32.

[14] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization* 11, 4 (1997), 341–359.

[15] Wencong Su, Jianhui Wang, and Jaehyung Roh. 2014. Stochastic energy scheduling in microgrids with intermittent renewable energy resources. *IEEE Transactions on Smart Grid* 5, 4 (2014), 1876–1883.

[16] Jun Sun, Bin Feng, and Wenbo Xu. 2004. Particle swarm optimization with particles having quantum behavior. In *IEEE Congress on Evolutionary Computation (CEC)*, Vol. 1. 325–331.

[17] Anupam Trivedi, Dipti Srinivasan, Subhodip Biswas, and Thomas Reindl. 2016. A genetic algorithm – differential evolution based hybrid framework: Case study on unit commitment scheduling problem. *Information Sciences* 354 (2016), 275 – 300.

[18] Zhongping Wan, Guangmin Wang, and Bin Sun. 2013. A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. *Swarm and Evolutionary Computation* 8 (2013), 26–32.

[19] Ye Yan, Yi Qian, Hamid Sharif, and David Tipper. 2013. A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE communications surveys & tutorials* 15, 1 (2013), 5–20.

[20] Arthouros Zervos, Christine Lins, and Josche Muth. 2010. *RE-thinking 2050: a 100% renewable energy vision for the European Union.* EREC.