





## Article

# Demand Response of Residential Houses Equipped with PV-Battery Systems: An Application Study Using Evolutionary Algorithms

Fernando Lezama <sup>1,\*</sup> , Ricardo Faia <sup>1</sup> , Pedro Faria <sup>1</sup>  and Zita Vale <sup>2</sup> 

<sup>1</sup> Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), Polytechnic of Porto (ISEP/IPP), 4200-072 Porto, Portugal; rfmfa@isep.ipp.pt (R.F.); pnf@isep.ipp.pt (P.F.)

<sup>2</sup> Polytechnic of Porto (ISEP/IPP), 4200-072 Porto, Portugal; zav@isep.ipp.pt

\* Correspondence: flz@isep.ipp.pt; Tel.: +351-228-340-511; Fax: +351-228-321-159

Received: 12 April 2020; Accepted: 9 May 2020; Published: 14 May 2020



**Abstract:** Households equipped with distributed energy resources, such as storage units and renewables, open the possibility of self-consumption of on-site generation, sell energy to the grid, or do both according to the context of operation. In this paper, a model for optimizing the energy resources of households by an energy service provider is developed. We consider houses equipped with technologies that support the actual reduction of energy bills and therefore perform demand response actions. A mathematical formulation is developed to obtain the optimal scheduling of household devices that minimizes energy bill and demand response curtailment actions. In addition to the scheduling model, the innovative approach in this paper includes evolutionary algorithms used to solve the problem under two optimization approaches: (a) the non-parallel approach combine the variables of all households at once; (b) the parallel-based approach takes advantage of the independence of variables between households using a multi-population mechanism and independent optimizations. Results show that the parallel-based approach can improve the performance of the tested evolutionary algorithms for larger instances of the problem. Thus, while increasing the size of the problem, namely increasing the number of households, the proposed methodology will be more advantageous. Overall, vortex search overcomes all other tested algorithms (including the well-known differential evolution and particle swarm optimization) achieving around 30% better fitness value in all the cases, demonstrating its effectiveness in solving the proposed problem.

**Keywords:** demand response; energy service provider; energy storage system; evolutionary algorithms; optimization; photovoltaic generation

## 1. Introduction

In the current environmental world scenario, countries are adopting a series of counter measures in what regards to the use of energy, renewable sources and DG (Distributed generator) [1]. In fact, the European Union, according to the EU (European Union) renewable energy directive (2009/28/EC), is pushing to their country members to achieve strict targets such as the of penetration of 20% of renewables into the energy mix by 2020, and increase the quantity up to by 100% by 2050. Thus, in order to achieve such ambitious targets, it is expected a systematic and elaborated transformation of the electrical grid, in line with the ambitions of the EU [2].

In this scenario, new technologies such as PV (Photovoltaic) panels and battery systems emerge as a viable solution to promote the penetration of renewables at the local level of the distribution networks. Households equipped with PV generation and storage units became small producers

(the so-called prosumers due to their condition of consumer and producer at the local level) and provide a new source of flexibility to the systems [3]. Also, prosumers allow the implementation of innovative energy management mechanisms to take advantage of DR (Demand Response) and on-site generation. The correct coordination and use of such devices, through effective management and optimization approaches, promises several benefits such as the reduction of energy bills for households and the reduction of carbon-emission footprints in general.

Different approaches have been proposed to address the optimization of households equipped with PV-battery systems. For instance, a MILP (Mixed-integer Linear Programming) problem was formulated in [4] for the management of a residential community grid with renewables, batteries, electric vehicles, and DR capabilities. This formulation searched for the minimization of purchased energy cost. In [5], a similar approach was used to minimize operation cost of a smart building considering DR and day-ahead energy resource management. In [6], the capabilities of MILP were tested again under a similar problem formulation, showing that DR can be very effective in different scenarios when a high penetration of renewables is available. On the other hand, some MINLP (Mixed-integer Non-linear Programming) have extended the mathematical formulation to include non-linearities and make the models close to real-world situations. For instance, in [7] a unit commitment problem of a microgrid is formulated to optimize the amount of load reduction and incentives given due to DR at different time intervals. Also, in [8], gas and electricity are included into the energy mix model, and the day-ahead energy scheduling is optimized for energy hubs. Some other approaches have explored the idea of an aggregator that works as an energy service provider. In this case, households can apply DR actions following incentives or responding to a direct control signals dictated by the aggregator. For instance, in [9], an aggregation of air conditioning loads is considered to perform DR actions. The study in [10] is not only limited to DR actions but also considers storage units to participate in energy and regulation markets. Also, in [11], a demand response simulator to study actions and schemes of users in distribution networks was proposed. The study took into account the technical validation of solutions including load reduction using a consumer-based price elasticity approach supported by real time pricing.

Finally, due to the complexity of the problem, EA (Evolutionary Algorithms) has been proposed in the literature trying to face issues such as scalability, memory requirements, time constraints, and other related problems that arise in the context of demand response and hybrid PV-battery systems. For instance, in [12], a bi-level formulation for optimal day-ahead price-based DR is proposed and solved by a hybrid approach in which a multi-population genetic algorithm is used for the upper level and distributed individual optimization algorithm for the lower level. Another hybrid genetic algorithm is used in [13] to consider the interaction of electricity retailers and DR. More recently, in [14] a PSO (Particle Swarm Optimization) algorithm is used for load shifting of appliances and the scheduling of PV and storage equipment using a home energy management system. In [15], the performance of evolutionary algorithms is compared solving a flexibility management model in which home appliances can perform DR actions. In addition, evolutionary algorithms have been used not only to optimize hybrid renewable energy systems [16] but also to coordinate the scheduling of PV-storage systems [17–19].

In this paper, we extend the model proposed in [20], for optimization of households equipped with PV-battery systems and DR capabilities. Different EAs, including DE (Differential Evolution, PSO, VS (Vortex Search, and other variants, are implemented to solve the optimization problem (MILP), and their performance and results are compared under two novel frameworks (one following the typical framework of EAs and another taking advantage of parallel computing). Households are provided with an independent management of resources minimizing energy bills and optimizing DR curtailment. With the objective of improving the minimization of electricity costs for households, with the support of an energy service provider, the contributions of this paper are as follows:

- An optimization framework for the optimization of PV-battery system of households minimizing energy bills and DR actions.

- A MILP formulation to optimize the resources of several households.
- Implementation of different EAs under two optimization approaches, one based on standard evolutionary computation and a second one taking advantage of parallel computing.
- Assessment of the effectiveness of EAs and the optimization framework under a case study considering up to 20 households.

The paper is organized as follows: after the introduction in Section 1; the proposed methodology and the mathematical formulation is presented in in Section 2; Evolutionary algorithms applied in this work are introduced in Section 3; Section 4 presents the two proposed optimization approaches employed with the use of EAs to make use of parallel computing; the case study and results are provided in Sections 5 and 6 respectively; and finally, the conclusions of this work are presented in Section 7.

## 2. Households Demand Response Optimization

In this section, is provided the description of the proposed optimization model, which aims to minimize the energy bill and the user discomfort. The change in the consumption pattern is considered to be a way of user discomfort. Since it is a rather complex problem to be computed at house level, the proposed methodology considers an Energy Service Provider that performs the optimization for a large set of households, and makes the results available for each one.

In each house, distributed energy resources are available, like PV generation, storage, and DR. Accordingly, each household is a prosumer (a consumer able to produce electricity), equipped with a PV and an energy storage system. Three appliances can be controlled by the optimization algorithm to reduce the consumption in periods when the electricity price is higher. For this, it is assumed that the household owns the needed control devices (e.g., plc). The PLC (Programmable Logic Controller) controller unit manages the consumption and generation resources in the houses according to the schedule received from the Energy Service Provider.

The mathematical formulation of the problem is an extension of [20] to consider up to  $I$  households (unlike the original model designed to target only one household). Thus, the formulation corresponds to a MILP model having as OF (Objective Function) Equation (1):

$$\text{Minimize } OF = \text{Energy Bill} + DR \text{ Curtailment Weight} , \quad (1)$$

where *Energy Bill* represents the costs of buying and selling electricity, while *DR Curtailment Weight* quantifies the weight of the curtailment of loads due to DR. Thus, Equation (2) represents the energy bill that households must pay due to the flow of energy exchanged with the main grid:

$$\text{Energy Bill} = \sum_{i=1}^I \left( \sum_{t=1}^T \left( P_{i,t}^{\text{Grid In}} \times C_{i,t}^{\text{Grid In}} - P_{i,t}^{\text{Grid Out}} \times C_{i,t}^{\text{Grid Out}} \right) \times \frac{1}{\Delta t} \right) + \text{Fix Cost}_i , \quad (2)$$

where  $P_{i,t}^{\text{Grid In}}$  represents the energy flow from the grid to the household,  $C_{i,t}^{\text{Grid In}}$  represents the cost of buying energy,  $P_{i,t}^{\text{Grid Out}}$  is the energy flow from household to the grid,  $C_{i,t}^{\text{Grid Out}}$  corresponds to the revenue of selling energy to the grid,  $\frac{1}{\Delta t}$  is a term that considers the modification of hourly values to another time interval (e.g., 15 min in this article),  $\text{Fix Cost}_i$  represents the fixed tariff costs pay by each household.  $i = \{1, \dots, I\}$  is used to identify households, and  $t = \{1, \dots, T\}$  for the periods. Notice that Equation (2) includes the sum of energy bill over all households. Therefore, minimizing this overall value corresponds to reduce the bill for each particular household. Moreover, the energy consumption/generation from households is independent, and thus, finding the minimum value for Equation (2) guarantees that the minimum possible bill for each household is obtained.

On the other hand, Equation (3) is used to calculate the weight of DR actions:

$$DR \text{ Curtailment Weight} = \sum_{i=1}^I \left( \sum_{t=1}^T \left( \sum_{l=1}^L \left( P_{i,t,l}^{\text{Cut}} \times X_{i,t,l}^{\text{Cut}} \times W_{i,t,l}^{\text{Cut}} \right) \right) \right) , \quad (3)$$

where  $P_{i,t,l}^{\text{Cut}}$  represents the energy load cuts,  $X_{i,t,l}^{\text{Cut}}$  are binary decision variables indicating a DR action,  $W_{i,t,l}^{\text{Cut}}$  represents the weight of energy cuts,  $l = \{1, \dots, L\}$  is used to represent loads available for DR.

It is important to point out, as explained in [20] that the energy bill (first term) and DR curtailment (second term) can be seen as opposite objectives in Equation (1). This is because the curtailment of loads reduces energy bills, but at the same time affects user comfort in different ways depending on user preferences. In this work, however, we decided to select the DR weights of energy cuts following a trend contrary to the buy from grid tariff to promote the use of DR when the price of energy is higher. Other assumptions and targets can be explored in future work.

Equation (4) represents the energy balance at each period:

$$P_{i,t}^{\text{Grid}} = P_{i,t}^{\text{Load}} + P_{i,t}^{\text{Bat}} - \sum_{l=1}^L \left( P_{i,t,l}^{\text{Cut}} \times X_{i,t,l}^{\text{Cut}} \right) - P_{i,t}^{\text{PV}}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \quad (4)$$

where  $P_{i,t}^{\text{Grid}}$  represents the energy flow between grid and household,  $P_{i,t}^{\text{Load}}$  represents consumption from non-controllable loads,  $P_{i,t}^{\text{Bat}}$  corresponds to energy charge/discharge of batteries (charge or discharge) and  $P_{i,t}^{\text{PV}}$  represents the energy generated by PV panels.

Equation (5) is applied to obtain the flow of energy between the grid to household:

$$P_{i,t}^{\text{Grid In}} = P_{i,t}^{\text{Grid}} \geq 0, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}. \quad (5)$$

Equation (6) is applied to obtain the energy flow from households to the grid (exported energy):

$$P_{i,t}^{\text{Grid Out}} = P_{i,t}^{\text{Grid}} < 0, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}. \quad (6)$$

Equation (7) represents the balance of the batteries for all households at all periods:

$$E_{i,t}^{\text{Bat}} = E_{i,t-1}^{\text{Bat}} + P_{i,t}^{\text{Bat}} \times \frac{1}{\Delta t}, \forall i \in \{1, \dots, I\}, \forall t \in \{2, \dots, T\}, \quad (7)$$

where  $E_{i,t}^{\text{Bat}}$  is the state of the battery of household  $i$  at period  $t$ , and  $E_{i,t-1}^{\text{Bat}}$  represents the previous state of the battery of household  $i$  at period  $t - 1$ . Equation (7) is applied from the second to the last period of optimization, while  $E_{i,1}^{\text{Bat}}$  is an input parameter of the case study.

Equation (8) is used to represent the bounds of  $P_{i,t}^{\text{Grid}}$  variable:

$$P_{i,t}^{\text{Gridmin}} \leq P_{i,t}^{\text{Grid}} \leq P_{i,t}^{\text{Gridmax}}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \quad (8)$$

where  $P_{i,t}^{\text{Gridmin}}$  corresponds to the lower bound and  $P_{i,t}^{\text{Gridmax}}$  to the upper bound values of  $P_{i,t}^{\text{Grid}}$ .

Equation (9) represents the upper bound (maximum cut capacity) for the variable  $P_{i,t,l}^{\text{Cut}}$ :

$$P_{i,t,l}^{\text{Cut}} = P_{i,t,l}^{\text{Cutmax}}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \forall l \in \{1, \dots, L\}. \quad (9)$$

Equation (10) presents the bounds for the variable  $E_{i,t}^{\text{Bat}}$ .

$$0 \leq E_{i,t}^{\text{Bat}} \leq E_{i,t}^{\text{Batmax}}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \quad (10)$$

where  $E_{i,t}^{\text{Batmax}}$  are the upper bound of variables  $E_{i,t}^{\text{Bat}}$ .

Equation (11) presents the bound for the variable  $P_{i,t}^{\text{Bat}}$ :

$$-P_{i,t}^{\text{Batdch}} \leq P_{i,t}^{\text{Bat}} \leq P_{i,t}^{\text{Batch}}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \quad (11)$$

where  $-P_{i,t}^{\text{Batdch}}$  and  $P_{i,t}^{\text{Batch}}$  are the lower and upper bounds of the variable  $P_{i,t}^{\text{Bat}}$ .

Equation (12) represents the bounds for the variable  $X_{i,t,l}^{Cut}$ .

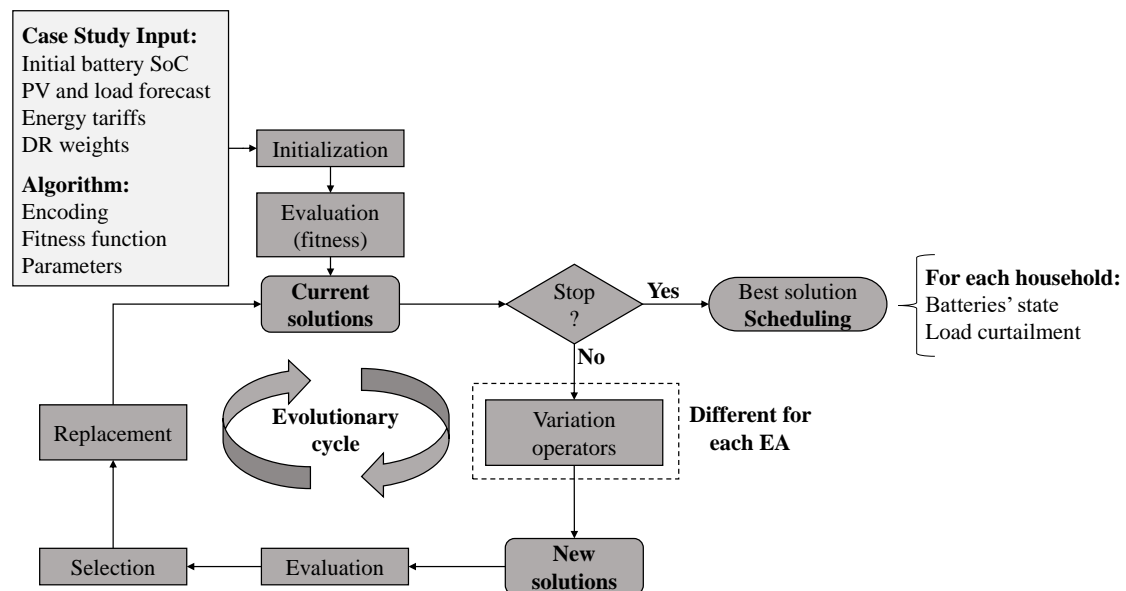
$$X_{i,t,l}^{Cut} = \begin{cases} 1 \\ 0 \end{cases}, \forall i \in \{1, \dots, I\}, \forall t \in \{1, \dots, T\}, \forall l \in \{1, \dots, L\}, \quad (12)$$

where variable  $X_{i,t,l}^{Cut}$  can take the value of '1' when the cut is active and the '0' when the cut is not active.

### 3. Evolutionary Computation

EC (Evolutionary Computation) is one of the three pillars of computational intelligence (along with artificial neural networks and fuzzy systems). EC includes a set of algorithms for optimization inspired in biological and evolutionary processes [21]. In fact, there are in the literature now a huge set of algorithms available for optimization, but in general, they can be grouped in some popular categories such as EA, SI (Swarm Intelligence), nature-inspired algorithms, natural computation, etc.

In this paper, we focus our attention in a class of algorithms that share some common mechanisms. This choice eases the experimental analysis since a fair comparison can be performed between the algorithms. Figure 1 illustrates the evolutionary mechanism employed by the selected EAs. Thus, in a first stage, an encoding of solutions and a fitness function are defined for a particular problem. The EAs act over an initial set of candidate solutions encoded as vectors (i.e., a population) that is iteratively updated through generations. The way in which new solutions are created from the initial population is what distinguishes each EA (i.e., each of the selected EAs has its own variation operator). Solutions' performance is measured by the fitness function, and at each generation, solutions with inferior performance are replaced by the new solutions with better performance. It is empirically proved that by the principles of natural selection (or artificial selection in this case), the population will gradually evolve towards an optimal fitness value.



**Figure 1.** Typical optimization scheme of evolutionary algorithms. All the evolutionary algorithms used in this work follow this scheme.

We describe the solution encoding and fitness function shared by the selected EAs in Section 3.1. After that, a brief description of the chosen algorithms is provided in Section 3.2.

### 3.1. Solution Encoding and Fitness

The optimization problem searches for the optimal scheduling of charging and discharging cycles of batteries and the choice of which loads are used for DR curtailment, for each user (as stated in Section 2).

Therefore, the selected encoding should include all the information to validate a solution, and it is very similar to that used in [20], but generalized for  $I$  users. Figure 2 shows the structure of a given solution in our framework. The solution first include continues variables representing the charging/discharging state (positive for charging, and negative for discharging) of the users' battery, at all periods  $t$ , for each user  $i$ . Therefore, this set includes  $T \times I$  variables. Then, a second set of binary variables is used to indicate a cut action in all load  $l$  ('1' if load  $l$  is curtailed, and '0' if not), at all periods  $t$ , for each user  $i$ . Therefore, this second set includes  $L \times T \times I$  binary variables. In general, a complete given solution to the problem is of dimension  $D = T \times I \times (1 + L)$ . The variables are bounded by:

$$\vec{lb} = \{-P_{i,t}^{\text{dchmax}}, X_{i,l,t}^{\text{cutmin}}\}, \quad i = \{1, \dots, I\}, t = \{1, \dots, T\}, l = \{1, \dots, L\}, \quad (13)$$

$$\vec{ub} = \{P_{i,t}^{\text{chmax}}, X_{i,l,t}^{\text{cutmax}}\}, \quad i = \{1, \dots, I\}, t = \{1, \dots, T\}, l = \{1, \dots, L\}. \quad (14)$$

Thus, the EAs can generate initial populations with random candidate solutions between those bounds using a random function such as:

$$\vec{x}_j = \text{rand}(\vec{lb}, \vec{ub}), \quad j = \{1, \dots, N_{\text{sol}}\}, \quad (15)$$

where  $\text{rand}(\vec{lb}, \vec{ub})$  generates a random solution between the bounds defined in Equations (13) and (14), and  $N_{\text{sol}}$  is the size of the population defined by the user.

Since the formulation includes constraints that can be difficult to optimize by the algorithms, we apply some direct repair techniques to ease the generation of feasible solutions. Equation (16) presents the direct repair mechanism employed to keep variables  $E_{i,t}^{\text{Bat}}$  into the allowed limits:

$$E_{i,t}^{\text{Bat}} = \begin{cases} 0, & \text{if } E_{i,t}^{\text{Bat}} < 0 \\ E_{i,t}^{\text{Batmax}}, & \text{if } E_{i,t}^{\text{Bat}} > E_{i,t}^{\text{Batmax}} \\ P_{i,t}^{\text{Bat}}, & \text{otherwise} \end{cases}, \quad \forall i \in \{1, \dots, I\}, \forall t \in \{2, \dots, T\}, \quad (16)$$

where variable  $E_{i,t}^{\text{Bat}}$  represent the energy state of charge of the battery.  $E_{i,t}^{\text{Bat}} < 0$  represents a discharge state greater than the allowed one, so that the variable is fixed to its minimum value. When  $E_{i,t}^{\text{Bat}} > E_{i,t}^{\text{Batmax}}$ , the battery has charged more energy than the allowed, thus, the value of maximum energy in the battery is fixed the maximum allowed value. After repairing the state of charge, variables  $P_{i,t}^{\text{Bat}}$  should also being repaired as:

$$P_{i,t}^{\text{Bat}} = \begin{cases} E_{i,t}^{\text{Bat}} - E_{i,t-1}^{\text{Bat}}, & \text{if } E_{i,t}^{\text{Bat}} < 0 \\ E_{i,t}^{\text{Bat}} - E_{i,t-1}^{\text{Bat}}, & \text{if } E_{i,t}^{\text{Bat}} > E_{i,t}^{\text{Batmax}} \\ P_{i,t}^{\text{Bat}}, & \text{otherwise} \end{cases}, \quad \forall i \in \{1, \dots, I\}, \forall t \in \{2, \dots, T\}. \quad (17)$$

Notice that variable  $P_{i,t}^{\text{Bat}}$  is repaired following the same conditions of Equation (16). This procedure guarantees feasible solutions, helping in the iterative process of the EAs.

Since the encoding has been designed to include all information needed to evaluate the mathematical model from Section 2, the fitness function is taken directly from Equation (1) including penalties due to the possibility of generate infeasible solutions. Therefore, the fitness value includes the energy bill (costs and revenues), fixed costs, and DR curtailment weight off all users plus penalties:

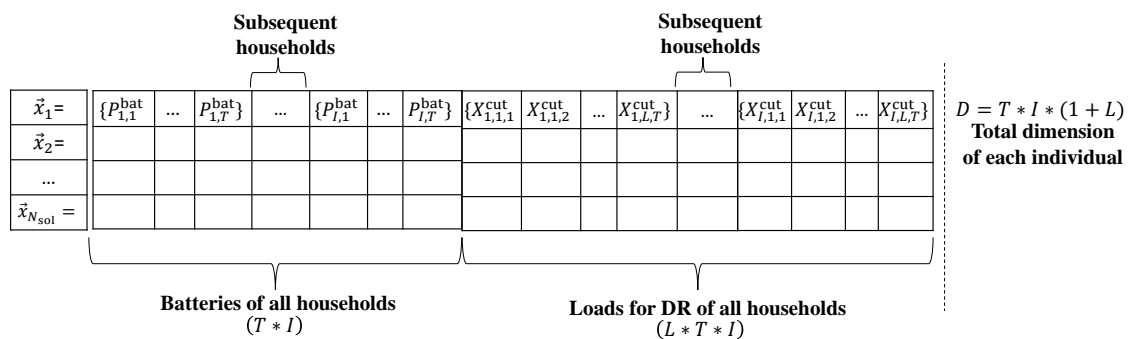
$$\text{Fitness}(\vec{x}_j) = f(\vec{x}_j) + p_{\text{function}}(\vec{x}_j), \quad (18)$$

where  $f(\vec{x}_j)$  is equivalent to Equation (1), and  $p_{\text{function}}(\vec{x}_j)$  is a function that returns a penalty value associated with the violation of the limits of variable  $P_{i,t}^{\text{Grid}}$  for each user  $i$  at each time  $t$ , defined as:

$$p_{\text{function}}(\vec{x}_j) = \sum_{i=1}^I \sum_{t=1}^T \rho_{i,t}, \quad (19)$$

$$\rho_{i,t} = \begin{cases} 0 - P_{i,t}^{\text{Grid}}, & \text{if } (P_{i,t}^{\text{Grid}} < 0) \\ P_{i,t}^{\text{Grid}} - P_{i,t}^{\text{Gridmax}}, & \text{if } (P_{i,t}^{\text{Grid}} > P_{i,t}^{\text{Gridmin}}) \\ 0 & \text{otherwise} \end{cases}, \quad (20)$$

where  $\rho_{i,t}$  is a penalty factor related to the limits of variable  $P_{i,t}^{\text{Grid}}$ . Notice that direct repair methods are used to avoid as much as possible violations of constraints (see direct repair methods in [20]), yet, due to the stochastic nature of EAs, infeasible solutions may arise for large instances (as the result section shows).



**Figure 2.** Solution encoding. The individual structure used by the EAs include all information needed to evaluate a solution.

### 3.2. EAs Used for DR of Households

Now that we defined the encoding of individuals and the fitness function, we apply EAs following the scheme from Figure 1 to solve the problem. In this paper, we apply DE and two of its variants hyde and HyDE (Hybrid Differential Evolution) (due to its success in many applications and easy implementation [22]), an improved version of the well-known PSO, and the vs [23]. In the following subsections, we provide a brief description of the algorithms, and their variation operators that distinguish each of them.

#### 3.2.1. Differential Evolution

DE uses a Pop (Population) of individuals  $\vec{x}_{j,i,G} = [x_{1,i,G}, \dots, x_{D,i,G}]$ , where  $G$  is the number of iterations,  $i = [1, \dots, NP]$  is the index of individuals in the population, and  $j = [1, \dots, D]$  is the index for the variables to optimize. In the initialization stage,  $NP$  solutions are generated randomly within lower and upper ranges  $\vec{x}_{lb}$  and  $\vec{x}_{ub}$ . In the standard form of DE, the so-called DE/rand/1 algorithm, new solutions are created applying a mutation and recombination operator defined by:

$$\vec{m}_{i,G} = \vec{x}_{r1,G} + F(\vec{x}_{r2,G} - \vec{x}_{r3,G}), \quad (21)$$

$$\vec{t}_{j,i,G} = \begin{cases} \vec{m}_{i,G} & \text{if } (\text{rand}_{i,j}[0, 1] < Cr) \vee (j = \text{Rnd}) \\ \vec{x}_{j,i,G} & \text{otherwise} \end{cases}, \quad (22)$$

where  $\vec{x}_{r1,G}, \vec{x}_{r2,G}, \vec{x}_{r3,G} \in \text{Pop}$  are three random individuals from the Pop, mutually different from each other.  $F$  and  $Cr$  are the mutation and recombination parameters of DE, usually set in the range  $[0, 1]$ . The fitness function, (i.e., Equation (18)), is used to evaluate the performance of new individuals.

An elitist selection procedure is applied in DE by replacing solution with worse performance than the new generated ones. A detailed explanation of DE can be found in [24,25].

### 3.2.2. Hybrid Adaptive DE

HyDE is a new self-adaptive version of DE proposed in [25]. The distinguish HyDE variation operator, known as “DE/target-to-perturbed<sub>best</sub>/1”, modifies the well-known DE/target-to-best/1 strategy [22] perturbing the best individual (similar to the evolutionary PSO [26]). HyDE also integrate a self-adaptive parameter mechanism (taken from the jDE (Self-Adaptive Differential Evolution algorithm [27])). The main operator of HyDE is defined as follows:

$$\vec{m}_{i,G} = \vec{x}_{i,G} + F_i^1(\epsilon \cdot \vec{x}_{\text{best}} - \vec{x}_{i,G}) + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}), \quad (23)$$

where  $F_i^1$  and  $F_i^2$ , are scale factors in the range  $[0, 1]$  independent for each individual  $i$ , and  $\epsilon = \mathcal{N}(F_i^3, 1)$  is a random perturbation factor following a normal distribution with mean  $F_i^3$  (random number in the range  $[0, 1]$ ) and standard deviation 1.  $F_i^1$ ,  $F_i^2$  and  $F_i^3$  are updated at each iteration with the same rule of jDE algorithm (see Section III.B of [25]).

### 3.2.3. Hybrid Adaptive DE with Decay Function

HyDE-DF is an improved version of HyDE used for function optimization [28]. The main different in its operation is the incorporation of a decay function that allows a transition in the iterative process from the main operator of HyDE (Equation (23)) to the basic operator of DE (Equation (21)):

$$\vec{m}_{i,G} = \vec{x}_{i,G} + \delta_G \cdot [F_i^1(\epsilon \cdot \vec{x}_{\text{best}} - \vec{x}_{i,G})] + F_i^2(\vec{x}_{r1,G} - \vec{x}_{r2,G}), \quad (24)$$

where  $\delta_G$  is a decreasing function (from  $1 \rightarrow 0$ ) that gradually mitigates the influence towards  $x_{\text{best}}$ , and takes advantage of the inherent DE exploitation capabilities in later stages of the evolutionary process. The decay factor at each generation  $G$  is calculated as:

$$\delta_G = e^{1-1/a^2}; \quad \text{with} \quad a = (GEN - G)/GEN. \quad (25)$$

$\delta_G$  factor alleviate the premature convergence effect towards the best individual in the population (i.e., due to the term  $F_i^1(\epsilon \cdot \vec{x}_{\text{best}} - \vec{x}_{i,G})$ ). Such transition also allows an enhance exploration phase in the early stages of evolution, and improves exploitation in later stages of the optimization. HyDE-DF achieved third place (out of 36 algorithms) in the 100-digit challenge at CEC/GECCO 2019 [29].

### 3.2.4. PSO-LVS

PSO [20] belongs to the class of SI, in which particles (solutions to the problem) coordinate their actions by modifying their position towards the optimum value. Particles are evaluated in the fitness function and improve their position in each iteration using the following variation operator:

$$x_{j,i,G} = x_{j,i,G-1} + v_{j,i,G}, \quad (26)$$

$$v_{j,i,G} = w_G \cdot v_{j,i,G-1} + c1_G \cdot \text{rand}() \cdot (P_i^{\text{best}} - x_{j,i,G-1}) + c2_G \cdot \text{rand}() \cdot (G^{\text{best}} - x_{j,i,G-1}), \quad (27)$$

where  $v_{j,i,G}$  represents the velocity vector,  $w_G$  is the inertia weight,  $c1_G$  and  $c2_G$  are the acceleration coefficients for personal and global component,  $\text{rand}()$  is a uniformly distributed random number,  $P_i^{\text{best}}$  is the historical best position obtained by particle  $i$  while  $G^{\text{best}}$  is the population historical best position obtained by the swarm.

PSO-LVS (PSO with Local Vortex Search) (Equation (28)) is a variant of PSO developed by the authors that includes a local search based on the VS algorithm. The movement of PSO-LVS is therefore obtained by following equation:

$$x_{j,i,G} = \begin{cases} x_{j,i,G-1} + v_{j,i,G} & \text{if, } \text{rand}() \leq p^{\text{PSO}_G} \\ p(\vec{m}/\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (\vec{x} - \mu)^G \Sigma^{-1} (\vec{x} - \mu) \right\} & \text{otherwise,} \end{cases} \quad (28)$$

where  $p^{\text{PSO}_G}$  is a probability factor that switch between PSO standard equation and VS. Another difference is that  $\mu$  in Equation (29) is replaced by  $G^{\text{best}}$ . In addition,  $p^{\text{PSO}_G} = 0.9 \frac{G}{G}$  is a probability that decreases in function of the number of generations. With this method, it is expected the execution of LVS (Local Vortex Search) in later stages of the iterative process.

### 3.2.5. Vortex Search

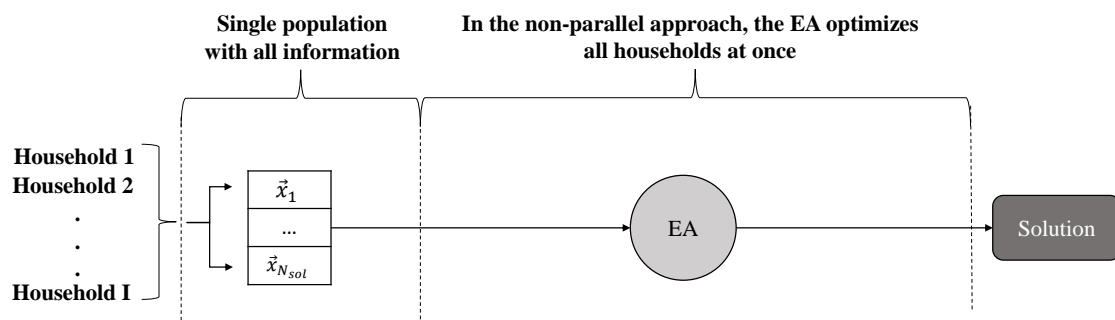
VS is classified as a single-solution-based metaheuristic, although it has an analogous framework to the EAs selected for this study. In each iteration,  $N$  given number of neighbor solutions are generated using a multivariate Gaussian distribution around the initial solution using:

$$p(\vec{m}/\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left\{ -\frac{1}{2} (\vec{x} - \mu)^G \Sigma^{-1} (\vec{x} - \mu) \right\}, \quad (29)$$

where  $d$  represents the dimension,  $\vec{x}$  is the  $d \times 1$  vector of a random variable,  $\mu$  is the  $d \times 1$  vector of sample mean (center), and  $\Sigma$  is the covariance matrix. The  $N$  solutions generated with this function are evaluated in the fitness function, and the best solution replaces the initial single-solution. The radius of search is gradually reduced during the iterative process, favoring exploitation capabilities in the final iterations. This process is iterative repeated until a stop criterion is achieved [23].

## 4. Non-Parallel and Parallel-Based Approaches

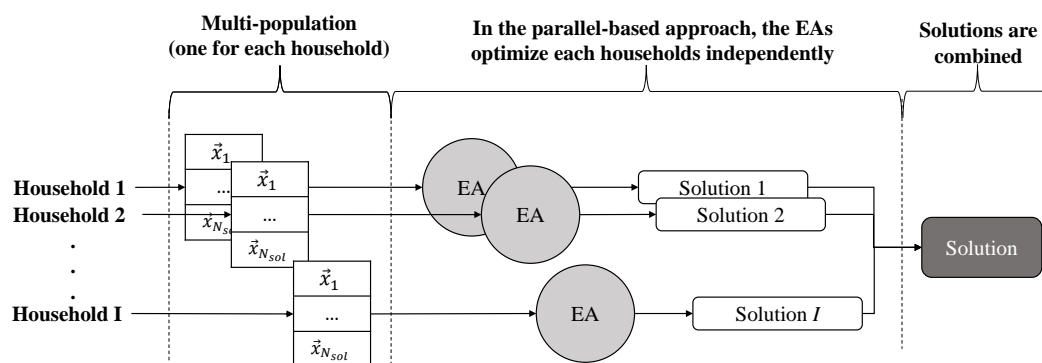
In this paper, given the nature of the mathematical formulation, and the independence of variables between households, we propose two approaches to use the EAs. In the first approach illustrated in Figure 3, so-called non-parallel approach, all variables are combined in a single encoding (explained in Section 3.1). Thus, the EAs use their variation operators over the whole set of variables, until a stop criterion is achieved. This is the typical form in which an EA is applied to solve a given problem.



**Figure 3.** The non-parallel approach optimizes all variables in one population. This is in line with the typical mechanism of EAs.

However, the problem formulation assume that each household scheduling is independent from each other, since their resources are individual and not shared among them. Thus, in the second approach illustrated in Figure 4, variables are divided in groups corresponding to each household. After that, multi-populations are generated and optimized independently by a chosen EA.

The independent solutions are combined in a post-optimization stage, to calculate the total costs of all households. While the solution returned by both approaches is equivalent, results show that breaking the groups of variables into sets corresponding to each household in fact improves the performance of the EAs. In addition, the parallel-based approach can make use of distributed computing, running in parallel different EAs and improving convergence times.



**Figure 4.** The proposed parallel-based approach breaks the solution into parts corresponding to the variables of each household. After that, each EA optimizes the variables and a post-optimization procedure combines the solutions into a single solution.

## 5. Case Study

We design a case study to evaluate our framework considering households representing prosumers complying with actual Portuguese legislation, which allows small producers (consumers with local generation) to use their energy to satisfy their own load needs, and inject excess of energy to the grid. We assume that households are equipped with PV panels with a maximum power capacity of 7.5 kW and a battery unit belonging to one of the four models showed in Table 1 (distributed randomly within the households). In addition, households equipped with controllable loads can reduce 10% on average of their total consumption.

**Table 1.** Battery models used for the case study, taken from [30].

	$p_{chmax}$	$-p_{dchmax}$	$E_{Batmax}$
Laboratory battery used in [20]	1.5 kW	−1.5 kW	12 kWh
Tesla Powerwall	5 kW	−5 kW	13.5 kWh
Alpha Smile	2.87 kW	−2.87 kW	14.5 kWh
Sonnen	3.3 kW	−3.3 kW	15 kWh

For consumption and PV generation, two sample power profiles were used to generate data of residential households. The profiles were built using real open datasets available at PES ISS website [31]. With these base profiles, up to 20 households' data was generated using a randomized function with a uniform distribution,  $\pm 25\%$  around the base profiles.

Figure 5 shows the retail tariffs and PV generation of the base profiles. We assume that households have a power supply contract with a given retailer of 11 kW characterized by three different periods: peak (0.33 EUR/kWh), intermediate (0.16 EUR/kWh), and off-peak (0.093 EUR/kWh). We also consider a feed-in tariff of 0.095 EUR/kWh and a DR weight with a trend contrary to the buy from grid tariff, i.e., promoting the use of dr when the price of energy is higher (these weights are applied to the second term of Equation (1)). Tariffs are based on real values of a Portuguese retailer.

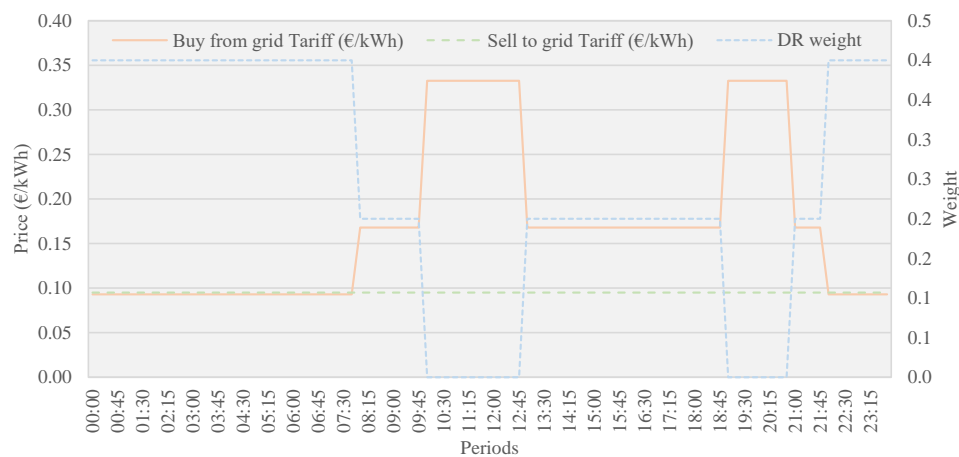
Figures 6 shows the aggregated consumption profiles of 20 households. Notice that the aggregated profile correspond to a typical profile since data from households is generated following base profiles, which in practice might not be the case. However, this paper is focused on the performance of EAs rather than the impact in the diversity of consumers. Further studies can be done considering

households with diverse characteristics and their impact in costs and DR curtailment. Figure 7 the total aggregated consumption and generation of 20 households. Finally, input values of variables for each household are summarized in Table 2

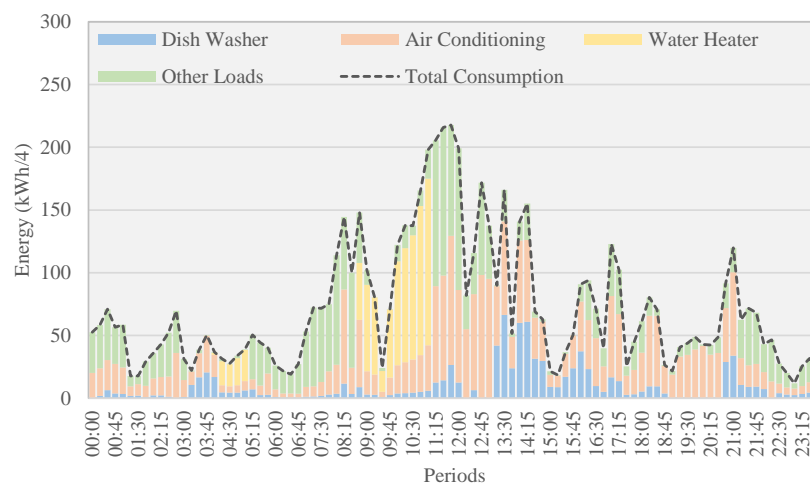
**Table 2.** Input variables of the problem. Values are applied to each household.

Parameter	Variable	Value	Units
Maximum power injected to the grid	$p_{i,t}^{\text{Gridmax}}$	5.1	kW
Maximum power required from grid	$p_{i,t}^{\text{Gridmin}}$	1000	kW
Maximum battery capacity	$E_{i,t}^{\text{Batmax}}$	12, 13.5, 14.5, 15	kW
Battery charge/discharge rate	$B_{i,t}^{\text{BatCh}}/B_{i,t}^{\text{BatDch}}$	1.5, 5, 2.87, 3.3	kWh
Initial state of charge of batteries	$E_{i,1}^{\text{Batmax}}$	0	kWh
PV maximum generation capacity	$p_{i,t}^{\text{PV}}$	7.5	kW
Total Periods	$T$	96	-
Total of controllable loads	$L$	3	-
Total of batteries	$B$	1	-
Total of PV units	-	1	-
Adjust parameter *	$\Delta t$	4	-

\* 1The factor of 4 is used since there are four 15-min periods in an hour.



**Figure 5.** Considered tariffs and PV generation base profile.



**Figure 6.** Aggregated loads and total consumption.

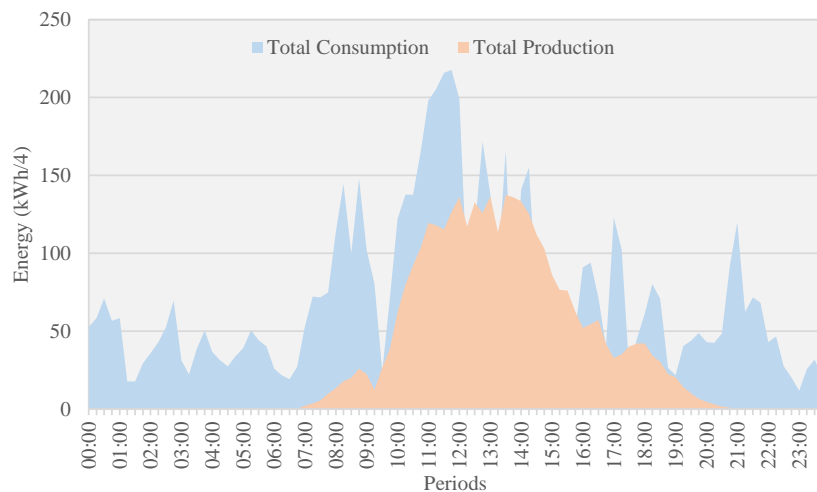


Figure 7. Aggregated consumption and production.

## 6. Results

We present the results of our methodology applied to the case study of Section 5. The experiments were implemented using MATLAB2018a, in a computer with Intel Xeon(R) E5-2620v2@2.1 GHz processor with 16GB of RAM running Windows 10. All the algorithms were run for 30 times (given the stochastic nature of eas), so the reported results correspond to the average of those runs.

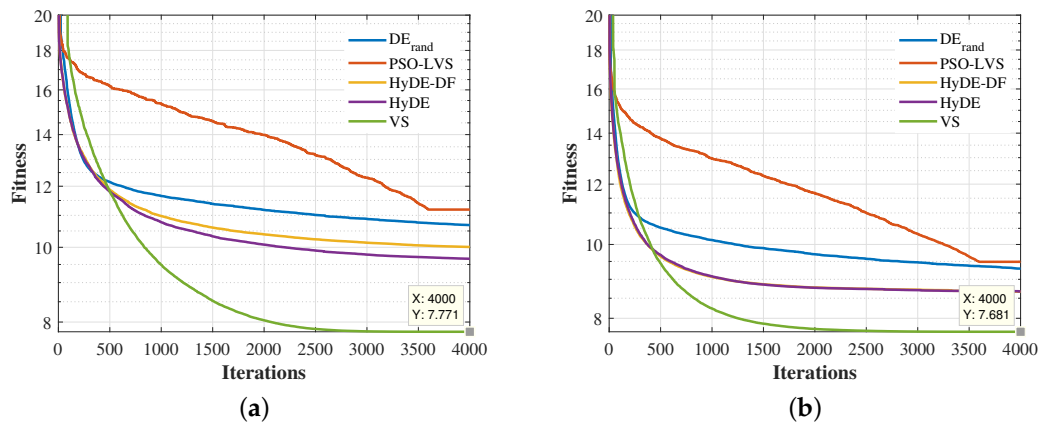
We perform four different experiments based on the number of households and the ea optimization approach. Table 3 show the four cases, identified by the letter C1–C4, related to the experiments. C1 is designed to assess the selected eas under the non-parallel approach considering two households. C2 also considers two households but under the parallel-based approach. C3 and C4 assess eas under non-parallel and parallel-based approaches respectively, but considering 20 households.

Table 3. Available equipment in houses for analyzing the impact of storage and dr.

Case	Two Households	20 Households	Non-Parallel	Parallel-Based
C1	✓		✓	
C2	✓			✓
C3		✓	✓	
C4		✓		✓

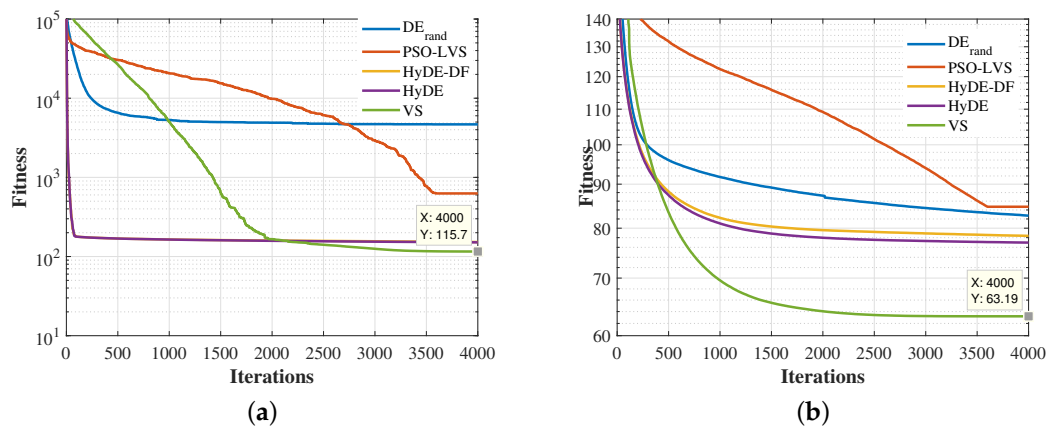
The parameters for each tested ea were selected following the recommendation of previous studies. Therefore, for de, the mutation factor and recombination constant ( $F$  and  $Cr$ ) were set to 0.5 and 0.9 respectively [32]. hyde and HyDE-DF [25] are a self-adaptive parameter versions but initial values for  $F^i$  and  $Cr$  where set to 0.5. For PSO-LVS the  $w_G$  inertia weight is linearly decreasing with the number of iterations between 0.9 and 0.4 [33]. The constants  $c1_G$  is set 0.5 and  $c2_G$  set 1.8. For variables boundary control Bounce Back method is used. VS algorithm does not have any parameter to configure [23]. To make a fair comparison, all the algorithms used a population of 20 initial solutions and run for  $4e3$  iterations.

Figure 8 shows the convergence of the tested algorithms considering the two players and the non-parallel and parallel-based approaches (C1 and C2 cases). It can seem that VS presents the best convergence performance in both cases. HyDE-DF and HyDE show similar performance (in fact, convergence lines are overlapped in Figure 8b, which indicates that the improvements incorporated in HyDE-DF (that showed remarkable performance in the 100-digit challenge [29]) have almost no impact solving the proposed problem. Overall, the parallel-based approach seems to slightly improve the performance of all algorithms, without modifying the overall ranking of them. In fact, VS algorithm obtains a similar final valor.



**Figure 8.** Average convergence of the tested EA considering two players under: (a) non-parallel approach (case C1); (b) parallel-based approach (case C2).

Figure 9 shows the results when increasing the number of players to 20 (C3 and C4 cases). In this case, while the non-parallel approach degrades the convergence performance of all EAs, the parallel-based approach keeps the convergence performance and increases only the final fitness value related with the cost of more households (see for instance Figures 8b and 9b). In summary, the parallel-based approach can help EAs in finding quality solutions for even large instances of the problem. Also, notice that  $DE_{rand}$  and PSO-LVS, apart from showing the worse performance, switch their convergence behavior when the non-parallel approach is used and the number of players increases (see for instance Figures 8a and 9a). That result shows evidence of their lack of robustness, since their performance should not be affected by the increase of the number of players



**Figure 9.** Average convergence of the tested EA considering 20 players under: (a) non-parallel approach (case C3); (b) parallel-based approach (case C4).

We also analyze the average fitness and associated costs/revenues obtained by the EAs in all the cases. Tables 4–7 present the average values of fitness, time, daily bill and DR curtailment, as well as the calculation of fitness percentage improvement, taking as reference the worst fitness value in each case. Table 4 shows the average results obtained in the case C1. First thing to observe is that VS presents the lower fitness value, but also the higher optimization time. However, all EAs present similar optimization times (ranging from 1.15 to 1.5 min). Regarding costs/revenues, it is interesting to note that despite VS obtains the best fitness value, its daily costs (Daily Bill column in the table) is slightly higher than that obtained by DE. This is explained by column DR curtailment, which shows that DE activates DR curtailment in a higher degree than the other algorithms. While this is

beneficial for the energy bill, it also represents a higher number of interruption of loads during the day, which can impact user comfort in some degree. Notice that DR curtailment in the formulation is not a monetary cost, but rather a weight associated with the interruption of loads. Future work can study the multi-objective nature of the formulation to optimize both terms in Equation (1) simultaneously. Finally, VS achieved the best performance with an improvement of around 30 % compared to PSO (worst algorithm for case C1).

Table 5 presents the results corresponding to case C2. It can seem that the general trends, as reported in case C1 results, are followed by the EAs when low number of households are considered. Mean Fitness and overall daily bills are slightly improved. Optimization times are equivalent, but it should be taken into account that column Time reflects the sum of the independent optimization of both households. Such optimizations can be done in parallel since are independent, reducing the optimization time by half, while obtaining better results regarding fitness and daily bills. In case C2, VS again achieved the best performance with an improvement of around 20 % compared to PSO (worst algorithm for case C2).

**Table 4.** Case C1: Fitness value and associated costs considering two households and non-parallel EAs.

	Fitness		Time (min)	Costs (€)	Revenues (€)	Fixed Costs (€)	Daily Bill * (€)	Monthly Bill ** (€)	DR	Imp (%)
	mean	std								
DE	10.69	0.46	1.16	9.29	−2.75	1.02	7.56	226.94	1.68	4.52
PSO	11.19	0.44	1.14	10.29	−2.11	1.02	9.21	276.25	0.75	0.00
HyDE	9.94	0.78	1.43	10.13	−1.21	1.02	9.95	298.43	0.00	11.23
HyDE-DF	9.59	0.83	1.42	9.78	−1.18	1.02	9.62	288.74	0.00	14.35
VS	7.77	0.08	1.48	9.09	−2.43	1.02	7.69	230.65	0.08	30.57

\* Daily bills are calculated as buy Costs − sell Revenues + fixed Costs. \*\* Monthly bill on average considering 30 days.

**Table 5.** Case C2: Fitness value and associated costs considering two households and parallel-based EAs.

	Fitness		Time (min)	Costs (€)	Revenues (€)	Fixed Costs (€)	Daily Bill * (€)	Monthly Bill ** (€)	DR	Imp (%)
	mean	std								
DE	9.30	0.34	1.03	9.34	−2.80	1.02	7.56	226.84	0.54	2.03
PSO	9.49	0.22	1.05	9.79	−2.04	1.02	8.77	263.21	0.28	0.00
HyDE	8.64	0.41	1.31	9.66	−1.97	1.02	8.72	261.55	0.00	8.97
HyDE-DF	8.67	0.46	1.30	9.53	−1.67	1.02	8.88	266.46	0.00	8.68
VS	7.68	0.03	1.36	9.27	−2.62	1.02	7.67	230.18	0.00	19.06

\* Daily bills are calculated as buy Costs − sell Revenues + fixed Costs. \*\* Monthly bill on average considering 30 days.

When the number of households increases, different conclusions are achieved. Tables 6 and 7 present the results corresponding to cases C3 and C4. The first thing to remark are the high fitness value reported by DE and PSO-LVS in case C3. Such high values are associated with the inability of both algorithms to find feasible solutions (i.e., the solutions include penalties explained in Equation (19)). Therefore, it is confirmed that these two algorithms are very sensitive to the increase in the number of households when the non-parallel approach is used. Such situation is corrected by the parallel-based approach, as Table 7 shows. In fact, the advantage of using this approach is stressed concerning fitness and daily bill values when the number of households is increased. Notice that since optimization times in the parallel-based approach correspond to the sum of independent optimizations, increasing the number of households affect notably the optimization times (see column Time of Table 7). However, such independent optimization can be performed in parallel reducing the time considerable depending the available parallel computing capacity. For instance, in MATLAB using four workers in the parallel pool (four parallel optimizations), the optimization time can be reduced by a factor of 5. Overall, VS achieved the best performance in both cases, with an improvement of around 22% compared to HyDE in case 3 (worst algorithm without considering DE and PSO due to reported infeasible solutions) and around 25% in case C4 compared to PSO.

**Table 6.** Case C3: Fitness value and associated costs considering 20 households and non-parallel EAs.

	Fitness		Time (min)	Costs (€)	Revenues (€)	Fixed Costs (€)	Daily Bill * (€)	Monthly Bill ** (€)	DR	Imp (%)
	mean	std								
DE	4672.93	2362.42	3.98	104.67	−35.33	10.24	79.57	2387.22	4.31	-
PSO	626.70	602.42	3.70	147.50	−19.79	10.24	137.94	4138.31	2.03	-
HyDE	149.45	3.86	4.93	150.40	−11.17	10.24	149.47	4484.04	0.02	0.00
HyDE-DF	148.58	4.26	4.93	150.59	−11.99	10.24	148.84	4465.34	0.03	0.58
VS	115.69	3.95	5.51	83.17	−33.48	10.24	59.93	1797.81	2.57	22.59

\* Daily bills are calculated as Buy costs − Sell Revenues + Fixed Costs. \*\* Monthly bill on average considering 30 days.

**Table 7.** Case C4: Fitness value and associated costs considering 20 households and parallel-based EAs.

	Fitness		Time (min)	Costs (€)	Revenues (€)	Fixed Costs (€)	Daily Bill * (€)	Monthly Bill ** (€)	DR	Imp (%)
	mean	std								
DE	82.72	2.24	9.69	86.54	−31.95	10.24	64.83	1944.86	0.82	2.36
PSO	84.72	0.90	9.60	92.32	−25.01	10.24	77.55	2326.41	0.39	0.00
HyDE	77.83	2.40	11.87	89.79	−21.38	10.24	78.65	2359.42	0.02	8.13
HyDE-DF	76.77	2.24	11.77	89.21	−20.02	10.24	79.43	2382.86	0.00	9.38
VS	63.19	0.26	12.31	84.29	−31.53	10.24	63.00	1889.92	0.01	25.41

\* Daily bills are calculated as Buy costs − Sell Revenues + Fixed Costs. \*\* Monthly bill on average considering 30 days.

## 7. Conclusions

In this paper, a different EAs are used to solve an optimization problem considering households with PV-battery systems and DR. Taking advantage of the independence of variables between households, two optimization approaches, non-parallel and parallel-based, are proposed. Results showed that EAs using the parallel-based approach provide solutions with better fitness value when the number of households increases. It was demonstrated that the direct application of an EA to larger instances of the problem (the non-parallel approach) has poor convergence capabilities (despite being very efficient when applied to one or two households). On the other hand, the proposed parallel-based approach showed excellent performance even when increasing the number of households. It is important to notice that the parallel-based approach is only valid for a framework as the one assumed in this work (which is actually a very likely real scenario due to the possible resistance of households to share data or equipment between peers), so changing such conditions might require a hybrid non-parallel and parallel approach. Overall, VS algorithm overcomes other tested EAs when using both optimization approaches. In fact, improvements of 30.57 % for case C1, 19.06 % for case C2, 22.59 % for case C3, and 25.41 % for case C4, were achieved with VS (best performance) compared to PSO (worst performance). Another advantage of the parallel-based approach is related to the possibility of using parallel computing to reduce optimization times while obtaining solutions with good quality. From the practical point of view, in this work we have envisaged the involvement of an Energy Service Provider that performs the optimization of households equipped with distributed energy resources (like PV generation, storage, and demand response) and the needed control devices. In this way, several business models can be explored by the Energy Service Provider within this framework. For instance, the service provider can charge a fee or commission from the total bill reduction achieved by the households, or receive incentives from upper level actors (such as the DSO) for the reduction of peak demand through DR. These two options, and other business model possibilities exploring the use of available infrastructure for practical implementations can be explored in future work. Another line of research for future work is related to the mathematical model. In this work, energy bill and DRdr curtailment are combined into a single objective formulation despite being terms that can be optimized in function of user preferences. Thus, multi-objective optimization versions of EAs can be employed to find Pareto optimal solutions. Moreover, a relation between DR curtailment and user comfort was not explicitly defined in this study, so another line of research can be followed concerning the modelling of user comfort. Finally, the practical implementation of EAs is also worth

to be explored in future works. The parallel-based approach uses a multi-population similar to that used by coevolutionary algorithms, so testing those kinds of algorithms and their performance in this problem since a good research avenue. In addition, in this study the parallel-based approach was implemented sequentially, so optimization times reflect the sum of all independent optimizations. In a future study, the implementation of an actual parallel platform can be proposed to handle larger instances of the problem and assess the reaches and scalability of the approach.

**Author Contributions:** Conceptualization, F.L., P.F. and Z.V.; methodology, F.L. and R.F.; software, F.L. and R.F.; validation, F.L., R.F. and P.F.; writing—original draft preparation, F.L. and R.F.; writing—review and editing, P.F. and Z.V.; supervision, P.F. and Z.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has received funding from FEDER Funds through COMPETE program and from National Funds through (FCT) under the projects UID/EEA/00760/2019, and grants CEECIND/02887/2017 and SFRH/BD/133086/2017. This work has received funding from H2020 in scope of DOMINOES project.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

DE	Differential Evolution
DG	Distributed generator
DR	Demand Response
EA	Evolutionary Algorithms
EC	Evolutionary Computation
EU	European Union
HyDE	Hybrid Differential Evolution
HyDE-DF	HyDE with Decay function
jDE	Self-Adaptive Differential Evolution
LVS	Local Vortex Search
MILP	Mixed-integer Linear Programming
MINLP	Mixed-integer Non-linear Programming
OF	Objective Function
PLC	Programmable Logic Controller
Pop	Population
PSO	Particle Swarm Optimization
PSO-LVS	PSO with Local Vortex Search
PV	Photovoltaic
SI	Swarm Intelligence
VS	Vortex Search

## Nomenclature

### Indices

$l$	Controllable load
$i$	Household
$t$	Period

### Parameters

$p_{i,t}^{\text{Load}}$	Consumption from non-controllable loads
$C_{i,t}^{\text{Grid In}}$	Cost of buying energy
$p_{i,t}^{\text{PV}}$	Energy generated by PV panels.
$p_{i,t,l}^{\text{Cut}}$	Energy load cuts
$\text{Fix Cost}_i$	Fixed tariff costs
$p_{i,t}^{\text{Gridmin}}$	Lower bond for buying energy
$p_{i,t}^{\text{Batdch}}$	Lower bound for discharge the battery
$C_{i,t}^{\text{Grid Out}}$	Revenue of selling energy
$p_{i,t}^{\text{Gridmin}}$	Lower bond for buying energy

$p_{i,t}^{\text{Batdch}}$	Lower bound for discharge the battery
$C_{i,t}^{\text{Grid Out}}$	Revenue of selling energy
$\Delta t$	Time adjust parameter
$L$	Total of controllable loads
$I$	Total of households
$T$	Total of periods
$E_{i,t}^{\text{Batmax}}$	Upper bound battery energy level
$p_{i,t}^{\text{Batch}}$	Upper bound for charge the battery
$p_{i,t}^{\text{Gridmax}}$	Upper bound for selling energy
$W_{i,t,l}^{\text{Cut}}$	Weight of energy cuts

#### Variables

$X_{i,t,l}^{\text{Cut}}$	Binary decision variables for DR action
$p_{i,t}^{\text{Bat}}$	Energy charge/discharge of batteries
$p_{i,t}^{\text{Grid}}$	Energy flow
$p_{i,t}^{\text{Grid Out}}$	Energy flow from household to the grid
$p_{i,t}^{\text{Grid In}}$	Energy flow from grid to the household
$E_{i,t}^{\text{Bat}}$	State of the battery

#### References

1. Gasparatos, A.; Doll, C.N.; Esteban, M.; Ahmed, A.; Olang, T.A. Renewable energy and biodiversity: Implications for transitioning to a Green Economy. *Renew. Sustain. Energy Rev.* **2017**, *70*, 161–184. [\[CrossRef\]](#)
2. Connolly, D.; Lund, H.; Mathiesen, B. Smart Energy Europe: The technical and economic impact of one potential 100% renewable energy scenario for the European Union. *Renew. Sustain. Energy Rev.* **2016**, *60*, 1634–1653. [\[CrossRef\]](#)
3. Morais, H.; Kádár, P.; Faria, P.; Vale, Z.A.; Khodr, H. Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renew. Energy* **2010**, *35*, 151–156. [\[CrossRef\]](#)
4. Nan, S.; Zhou, M.; Li, G. Optimal residential community demand response scheduling in smart grid. *Appl. Energy* **2018**, *210*, 1280–1289. [\[CrossRef\]](#)
5. Zhang, D.; Shah, N.; Papageorgiou, L.G. Efficient energy consumption and operation management in a smart building with microgrid. *Energy Convers. Manag.* **2013**, *74*, 209–222. [\[CrossRef\]](#)
6. Pina, A.; Silva, C.; Ferrão, P. The impact of demand side management strategies in the penetration of renewable electricity. *Energy* **2012**, *41*, 128–137. [\[CrossRef\]](#)
7. Nwulu, N.I.; Xia, X. Optimal dispatch for a microgrid incorporating renewables and demand response. *Renew. Energy* **2017**, *101*, 16–28. [\[CrossRef\]](#)
8. Alipour, M.; Zare, K.; Abapour, M. MINLP Probabilistic Scheduling Model for Demand Response Programs Integrated Energy Hubs. *IEEE Trans. Ind. Inform.* **2018**, *14*, 79–88. [\[CrossRef\]](#)
9. Zhou, X.; Shi, J.; Tang, Y.; Li, Y.; Li, S.; Gong, K. Aggregate Control Strategy for Thermostatically Controlled Loads with Demand Response. *Energies* **2019**, *12*, 683. [\[CrossRef\]](#)
10. Yao, Y.; Zhang, P.; Chen, S. Aggregating Large-Scale Generalized Energy Storages to Participate in the Energy and Regulation Market. *Energies* **2019**, *12*, 1024. [\[CrossRef\]](#)
11. Faria, P.; Vale, Z. Demand response in electrical energy supply: An optimal real time pricing approach. *Energy* **2011**, *36*, 5374–5384. [\[CrossRef\]](#)
12. Meng, F.L.; Zeng, X.J. A bilevel optimization approach to demand response management for the smart grid. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 287–294.
13. Alves, M.J.; Antunes, C.H.; Carrasqueira, P. A hybrid genetic algorithm for the interaction of electricity retailers with demand response. In Proceedings of the European Conference on the Applications of Evolutionary Computation, Porto, Portugal, 30 March–1 April 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 459–474.
14. Hussain, B.; Khan, A.; Javaid, N.; Hasan, Q.U.; A Malik, S.; Ahmad, O.; Dar, A.H.; Kazmi, A. A Weighted-Sum PSO Algorithm for HEMS: A New Approach for the Design and Diversified Performance Analysis. *Electronics* **2019**, *8*, 180. [\[CrossRef\]](#)

15. Lezama, F.; Soares, J.; Canizes, B.; Vale, Z. Flexibility management model of home appliances to support DSO requests in smart grids. *Sustain. Cities Soc.* **2020**, *55*, 102048. [\[CrossRef\]](#)
16. Fadaee, M.; Radzi, M. Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review. *Renew. Sustain. Energy Rev.* **2012**, *16*, 3364–3369. [\[CrossRef\]](#)
17. Teng, J.H.; Luan, S.W.; Lee, D.J.; Huang, Y.Q. Optimal charging/discharging scheduling of battery storage systems for distribution systems interconnected with sizeable PV generation systems. *IEEE Trans. Power Syst.* **2012**, *28*, 1425–1433. [\[CrossRef\]](#)
18. Ismail, M.S.; Moghavvemi, M.; Mahlia, T. Genetic algorithm based optimization on modeling and design of hybrid renewable energy systems. *Energy Convers. Manag.* **2014**, *85*, 120–130. [\[CrossRef\]](#)
19. Ghorbani, N.; Kasaeian, A.; Toopshekan, A.; Bahrami, L.; Maghami, A. Optimizing a hybrid wind-PV-battery system using GA-PSO and MOPSO for reducing cost and increasing reliability. *Energy* **2018**, *154*, 581–591. [\[CrossRef\]](#)
20. Faia, R.; Faria, P.; Vale, Z.; Spinola, J. Demand Response Optimization Using Particle Swarm Algorithm Considering Optimum Battery Energy Storage Schedule in a Residential House. *Energies* **2019**, *12*, 1645. [\[CrossRef\]](#)
21. Lezama, F.; Soares, J.; Vale, Z.; Rueda, J.; Rivera, S.; Elrich, I. 2017 IEEE competition on modern heuristic optimizers for smart grid operation: Testbeds and results. *Swarm Evol. Comput.* **2019**, *44*, 420–427. [\[CrossRef\]](#)
22. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [\[CrossRef\]](#)
23. Dogan, B.; Olmez, T. A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Inform. Sci.* **2015**, *293*, 125–145. [\[CrossRef\]](#)
24. Das, S.; Abraham, A.; Chakraborty, U.K.; Konar, A. Differential evolution using a neighborhood-based mutation operator. *IEEE Trans. Evol. Comput.* **2009**, *13*, 526–553. [\[CrossRef\]](#)
25. Lezama, F.; Soares, J.; Faia, R.; Pinto, T.; Vale, Z. A New Hybrid-Adaptive Differential Evolution for a Smart Grid Application Under Uncertainty. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [\[CrossRef\]](#)
26. Miranda, V.; Fonseca, N. EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems. In Proceedings of the IEEE/PES Transmission and Distribution Conference and Exhibition, Yokohama, Japan, 6–10 October 2002; Volume 2, pp. 745–750. [\[CrossRef\]](#)
27. Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Trans. Evol. Comput.* **2006**, *10*, 646–657. [\[CrossRef\]](#)
28. Lezama, F.; Soares, J.A.; Faia, R.; Vale, Z. Hybrid-adaptive Differential Evolution with Decay Function (HyDE-DF) Applied to the 100-digit Challenge Competition on Single Objective Numerical Optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion GECCO '19, Prague, Czech Republic, 13–17 July 2019; ACM: New York, NY, USA, 2019; pp. 7–8.
29. Price, K.; Awad, N.H.; Ali, M.Z.; Suganthan, P. *The 2019 100-Digit Challenge on Real-Parameter, Single Objective Optimization: Analysis of Results*; Technical Report 2019. Available online: [https://www.ntu.edu.sg/home/epnsugan/index\\_files/CEC2019/CEC2019.htm](https://www.ntu.edu.sg/home/epnsugan/index_files/CEC2019/CEC2019.htm) (accessed on 12 April 2020).
30. Naked Solar Ltd. *Solar Batteries & Storage*; Naked Solar Ltd.: Newquay, UK, 2020.
31. IEEE PES ISS Open Data Sets. Available online: <https://site.ieee.org/pes-iss/data-sets/> (accessed on 12 April 2020).
32. Price, K.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
33. Faia, R.; Pinto, T.; Vale, Z.; Corchado, J.M. Strategic Particle Swarm Inertia Selection for Electricity Markets Participation Portfolio Optimization. *Applied Artif. Intell.* **2018**, *32*, 745–767. [\[CrossRef\]](#)

