Bi-level Particle Swarm Optimization and Evolutionary Algorithm Approaches for Residential Demand Response with Different User Profiles

Pedro Carrasqueira[3], Maria João Alves[1,3], Carlos Henggeler Antunes[2,3]

[1]Faculty of Economics, University of Coimbra, Av. Dias da Silva 165, 3000 Coimbra, Portugal

[2]Dept. of Electrical and Computer Engineering, University of Coimbra, Polo 2, 3030-290 Coimbra, Portugal

[3]INESC Coimbra, DEEC, University of Coimbra, Polo 2,3030-290 Coimbra, Portugal

pmcarrasqueira@net.sapo.pt; mjalves@fe.uc.pt; ch@deec.uc.pt;

Abstract

The deregulation of electricity retail markets requires the development of new modeling approaches for the optimal setting of dynamic tariffs, in which consumers' responses according to their flexibility to schedule demand are considered. Retailers and consumers have conflicting goals: the former aim to maximize profits and the latter aim to reduce electricity bills. Also, there is a hierarchical relation between them, as retailers (upper-level decision makers) determine the pricing strategy and consumers (lower-level decision makers) react by scheduling their loads according to price signals and comfort requirements. This is a bi-level optimization problem. In this paper, typical residential loads are considered and three scenarios of feasible windows of appliance operation are established. Two new population-based approaches, an evolutionary algorithm and a particle swarm optimization algorithm, are developed to solve the bi-level problem. The results obtained are then compared with a hybrid algorithm that solves the lower-level problem exactly.

Keywords: Bi-level optimization; Particle swarm optimization; Evolutionary algorithms; Demand response; Electricity retail markets.

1. Introduction

In liberalized electricity markets, retailers procure energy in, for instance, intraday, day-ahead or future markets, with some dependence on purchase time and peak demand. Then retailers sell electricity to their clients who buy the necessary amount to satisfy demand, which may be adjusted according to budgetary constraints and comfort requirements. The further deregulation of the retail market requires the development of new modeling approaches for the optimal setting of dynamic tariffs, in which consumers' reactions in the framework of demand response (DR) programs are considered. Consumers aim to minimize their electricity bill by using their flexibility to schedule demand of shiftable appliances and the settings of thermostatically-controlled loads in the face of dynamic (time-of-use) tariffs established by the retailer.

Modern communication technologies play a central role in smart grids, which are characterized by bi-directional interaction between retailers and consumers. Smart grids provide the technological basis to implement effective DR programs, enabling consumers to react to price. The management of residential loads has many beneficial effects on the grid. It contributes to improving use efficiency of available infrastructure capacity, decreasing peak load demand, reducing GHG emissions levels (by increasing the share of supply from renewable energy sources under "load follows supply" strategies), providing ancillary services and enhancing overall grid sustainability.

Models using time-variant pricing schemes have been proposed to induce a shift in the peak load and smooth the load diagram, offering the system operator additional means for demand-supply balancing and constraint management purposes. A strategy often used is day-ahead pricing, in which the consumer receives tariff information one day or some hours before. The consumer may then react by scheduling load operation, i.e., changing the load profile by (optimally) deferring the use of some appliances, to set a trade-off between minimizing their electricity bill and maximizing their welfare in terms of comfort associated with the energy services provided (hot water, laundry, electric vehicle charging, etc.).

Some authors use bi-level programming models to address the interaction between retailers and consumers aiming to find an equilibrium solution. In a bi-level problem, the upper-level decision maker (leader) decides first. The goal of the leader is to optimize his/her objective function while considering the reaction of the lower-level decision maker (follower), since the follower's decision affects the leader's optimal solution. Meng and Zeng [7] converted the bi-level problem into a single-objective constrained model using Karush-Kuhn-Tucker conditions, solved with a branch-and-bound algorithm. Zugno et al. [16] proposed a bi-level model in which the retailer's objective is the maximization of profits and the consumer's objective is the minimization of cost and discomfort. In the upper level (retailer), stochastic variables are used to determine wholesale prices, whereas in the lower level (consumer), flexible and inflexible load appliances are considered and discomfort is measured against a range of indoor temperatures considered comfortable. Zhang et al. [15] presented a bi-level model with multiple objective functions at the upper level aimed at maximizing supply company profits. The consumers can choose the supply company to minimize their electricity bill.

Sekizaki et al. [10] proposed a bi-level model that included the network tariff cost in the evaluation of expected retailer profit and disutility cost for the consumer, which is proportional to the suppressed load of the corresponding ideal level (i.e., the one that maximizes their utility). Each type of consumer (residential, commercial, industrial and a mixed type resulting from the combination of these) has an objective function resulting from a weighted sum of the cost of purchasing power from the retailer plus the disutility associated with load reduction. Different scenarios to address the uncertainty of spot prices are also considered by Sekizaki et al. [10].

Meng and Zeng [8] considered non-shiftable, interruptible, non-interruptible and curtailable loads. The relation between one retailer and N consumers is modeled by a bi-level multi-follower problem. The retailer's problem is solved by a genetic algorithm. The consumer's problem is linear and solved exactly by a linear solver. Meng and Zeng [9] split the customer load into interruptible and non-interruptible appliances. A learning model is developed to identify the usage pattern of each appliance. Customers may solve their problem automatically by means of a smart meter or making energy decisions themselves.

Bu et al. [3] proposed a game-theoretic decision-making scheme in which the retailer purchases energy from two distinct sources, choosing the amount of energy to buy from each option considering different scenarios. The consumer's objective is to maximize a utility function based on the amount of energy purchased and the price paid. Yang et al. [14] developed a game-theoretic approach in which the consumer's objective is a satisfaction function relative to the difference between effective and nominal load, defining a time-of-use pricing strategy and comparing results with flat and hourly pricing strategies.

A survey of multi-level optimization problems related to hierarchical decision-making in many fields is presented by Lu et al. [6]. The survey includes the description of different types of bi-level problems and practical applications, including strategic bidding in electricity markets.

To obtain a more realistic load characterization, the working cycles of appliances should be defined, which was not done in the bi-level models referred to above. The most common household appliances are characterized by Soares et al. [12], presenting typical load cycles for each appliance and establishing its potential to be controlled. Soares et al. [13] used this information to develop a multi-objective model for domestic load scheduling.

In the present paper, the interaction between retailers (which establish dynamic tariffs) and consumers (who adjust consumption by changing habitual load scheduling) is formulated as a bi-level programming problem. In the upper level, the objective function is the maximization of retailer profit, while the objective function at the lower level is the minimization of the consumer's electricity bill. The lower-level problem is based on the model proposed by Soares et al. [13]. Two novel bi-level population-based algorithms are proposed, one based on an evolutionary algorithm (BLEA) and the other on particle swarm optimization (BLPSO), both of which make the most of the problem structure since generic bi-level algorithms do not give acceptable solutions. Given that consumers have different profiles concerning the acceptance of potential discomfort associated with load shifting, depending on the appliance, feasible working windows are defined for each appliance. Three profiles are considered for this purpose. These profiles aim to model the willingness of different types of consumers to engage in DR programs, which is reflected in more or less stringent constraints of the lower-level problem regarding allowable load operation time slots, and thus in the difficulty of computing optimal solutions to this problem (i.e., feasible solutions to the bi-level problem). The performance of the algorithms is assessed under these distinct conditions. A base profile represents the typical operational periods of each appliance. The two remaining profiles consider more restricted and more extended periods of operation. In addition, we develop two other scenarios, which displace the windows of appliance operation defined in each profile along the planning period, so that there is a higher simultaneity. The results are compared with the solutions obtained by a hybrid algorithm presented by Alves et al. [1], in which the lower-level problem is formulated by means of a mixed integer linear programming (MILP) model. No other approaches have been found in the literature to deal with a model with these characteristics.

The structure of the paper is as follows. In section 2, bi-level programming is introduced and the bi-level model for the interaction between retailers and consumers is presented. In section 3, the BLEA and BLPSO algorithms are described. Experimental results and their discussion are presented in section 4. In section 5, we draw the main conclusions.

2. A bi-level model for the electricity retail market problem

The electricity supply chain encompasses producers, transmission and distribution network operators, and retailers that supply consumers. Retailers buy electricity in the wholesale market and then sell it to end-users to power a diversity of appliances, fulfilling the need for energy services (hot water, acclimatization, entertainment, etc.). In general, wholesale market prices are variable and set by auctions or bilateral contracts. Wholesale electricity markets may include forward markets for future generation and acquisition, on-demand (spot, day-ahead) markets with a daily contracting component and an adjusted intraday component, and (real-time) ancillary services markets dealing with balancing electricity generation and consumption.

Electricity prices as seen by consumers may be static or dynamic. Static pricing includes flat rates (uniform price €/kWh, sometimes with a contracted power component €/kW), in which the retailer takes on the risk of market price uncertainty, and time-of-use rates, in which prices are established (long) in advance for specific time periods, thus already offering some economic signals to consumers. Dynamic pricing involves price schemes that can change on short notice, including critical peak pricing in which the retailer can sporadically establish a very high retail price for a limited period of time. Thus, it is possible to send economic signals to consumers in periods of generation shortage and/or network congestion, offering real-time tariffs reflecting the actual hourly price during that period. Time-of-use, critical peak pricing and real-time pricing schemes require advanced electricity meters, whose massive deployment is expected in the implementation of smart grids.

It is expected that dynamic tariffs will become the relevant pricing scheme in smart grids. Residential consumers are expected to react to the prices sent by the retailer by adjusting the operation cycle of some loads to time periods with more favorable tariffs within a planning period, considering their comfort requirements.

We have modeled the relation between retailers and consumers as a bi-level problem where the retailer is the leader and the consumer is the follower.

2.1 Bi-level programming

A bi-level programming problem consists of two embedded optimization problems, which may arise in many real-life situations where two decision-making entities are hierarchically related and have different goals.

The general bi-level programming problem, with the objective function of the upper level to be maximized and the objective function of the lower level to be minimized, can be stated as follows:

$$\begin{aligned} \max_{x_u \in X} \quad & F(x_u, x_l) \\ s.t. \quad & G(x_u, x_l) \le 0 \\ & x_l \in \arg\min_{x'_l \in Y} \left\{ f(x_u, x'_l) : g(x_u, x'_l) \le 0 \right\} \end{aligned} \tag{1}$$

where $X \subset \mathbb{R}^{n_1}$ (n_1 being the number of upper-level variables) and $Y \subset \mathbb{R}^{n_2}$ (n_2 being the number of lower-level variables) are closed sets. The upper-level decision maker (the leader) controls variables x_u while the lower-level decision maker (the follower) controls x_l. $F(x_u, x_l)$ and $f(x_u, x_l)$are the leader's and the follower's objective functions, respectively. Since the follower optimizes $f(x_u, x_l)$ after x_u has been selected, x_u is a constant vector whenever $f(x_u, x_l)$ is optimized.

For fixed $x_u \in X$, the set $Y(x_u) = \{x_l \in Y : g(x_u, x_l) \le 0\}$ is the *feasible set* of the follower. The set $\Psi(x_u) = \left\{ x_l \in Y : x_l \in \arg \min_{x'_l \in Y(x_u)} f(x_u, x'_l) \right\}$ is called the follower's *rational reaction set* to a given x_u. The feasible set of Problem (1), also called the *induced region*, is $IR = \{(x_u, x_l) : x_u \in X, G(x_u, x_l) \le 0, x_l \in \Psi(x_u)\}$.

Bi-level programming problems are very difficult to solve due to their inherent non-convexity. Even the linear bi-level problem is NP-hard [4]. Theory and methodology on bi-level programming can be found in [2,4].

2.2 Model

In the bi-level model for the electricity retail market problem, the planning period is divided into sub-periods P_i $(i = 1, \cdots, I)$ and the retailer should determine the price of electricity x_i(€/kWh) for each sub-period i. To guarantee competitiveness, prices set by the retailer are limited to minimum $(\underline{x_i})$ and maximum $(\overline{x_i})$ values in each sub-period and an average price(x^{AVG}) relative to the entire planning period is imposed [16]. The consumer has a base load, corresponding to non-controllable appliances, and a set of shiftable loads (appliances), whose cycles of operation can be set within acceptable time slots specified according to consumer preferences, habits and comfort requirements regarding the provision of energy services. Reacting to energy prices, the consumer wants to determine the time (z_j) each appliance j must start its work cycle to minimize the electricity bill, ensuring that the entire work cycle is within the corresponding comfort time slot T_j.

Notation:

T = number of intervals (minutes, quarter-hour, half-hour or other period of time) the planning period is divided into (t =1,…,T). Let T = {1,…, T }.

J = number of shiftable loads to be scheduled by the consumer's energy management system (j =1,..,J).

C_t = contracted power at time t of the planning period (kW).

π_t = energy price on the spot market at time t of the planning period (€/kW$\hbar$), where $\hbar$ is the duration of one interval (to accommodate the most convenient discretization of the planning period in face of the problem at hand).

b_t = non-controllable base load at time t of the planning period (kW).

d_j = duration ($\hbar$) of the operation cycle of load j.

$g_j(r)$ = power requested by load j at time r of its work cycle (r =1,…,d_j) (kW).

$T_j = [T1_j, T2_j] \subseteq$ T: time slot in which load j is allowed to operate.

I = number of sub-periods of time$P_i \subset$ Tin which different electricity prices (time-of-use tariffs) are charged by the retailer to the consumer (i=1,…,I).

$P1_i$, $P2_i$: time intervals delimiting each sub-period$P_i, i = 1, \cdots, I$, such that $P_i = [P1_i, P2_i]$ and $\bigcup_{i=1}^{I} P_i =$ T. $\overline{P_i}$ denotes the amplitude of P_i, i.e. $\overline{P_i} = P2_i - P1_i + 1$.

$\overline{x_i}$= maximum price charged to the consumer in sub-period P_i (€/kW$\hbar$).

$\underline{x_i}$= minimum price charged to the consumer in sub-period P_i (€/kW$\hbar$).

x^{AVG}= average price charged to the consumer in the planning period(€/kW$\hbar$).

Upper-level decision variables:

x_i= price charged to the consumer during sub-period P_i (€/kW$\hbar$); i=1,…,I.

Lower-level decision variables:

z_j = starting time of the working cycle of load j; j =1,..,J.

Auxiliary lower-level variables:

u_{jt}= binary variable representing whether the working cycle of load j is "on" or "off" at time t of the planning period; j =1,..,J; t =1,…,T.

p_{jt} = power requested from the grid by load j at time t of the planning period (kW); j =1,..,J; t =1,…,T.

The objective function of the upper-level decision maker in Eq. (2) is the maximization of the retailer's profit (revenues from selling electricity to consumers minus cost of purchasing the electricity in the spot market). The x and z variables, upper- and lower-level decision variables in this model, correspond to the x_u and $x_l$variables in the general bi-level formulation in Section 2.1, respectively.

Eq. (3) – (4) define the upper and lower bounds for the electricity prices charged to the consumer in each sub-period P_i. The constraint in Eq. (5) sets an average price in planning period T.

Eq.(6) is the objective function of the consumer, who aims to minimize the electricity bill (to supply non-controllable and shiftable loads).

Eq. (7) defines the value of the auxiliary binary variables u_{jt} as a function of decision variables z_j regarding the operation period of load j. Variables u_{jt} are then used in Eq. (8) to obtain the value of the power requested from the grid by load j at each time t of the planning period, i.e., time t-z_j+1 of the load operation cycle initiated in z_j. Constraints in Eq. (9) ensure that the contracted power is never exceeded. Constraints in Eq.(10) establish the time slots allowed for the operation of each load j according to the consumer's preferences.

$$\max F = \sum_{i=1}^{I}\sum_{t\in P_i} x_i\left(b_t + \sum_{j=1}^{J} p_{jt}\right) - \sum_{t=1}^{T}\pi_t\left(b_t + \sum_{j=1}^{J} p_{jt}\right) \tag{2}$$

s.t.

$$x_i \le \overline{x}_i \quad i=1,\dots,I \tag{3}$$

$$x_i \ge \underline{x}_i \quad i=1,\dots,I \tag{4}$$

$$\frac{1}{T}\sum_{i=1}^{I} \bar{P}_i x_i = x^{AVG} \tag{5}$$

$$\min f = \sum_{i=1}^{I}\sum_{t\in P_i} x_i\left(b_t + \sum_{j=1}^{J} p_{jt}\right) \tag{6}$$

s.t.

$$u_{jt} = \begin{cases} 1 & \text{if} z_j \le t \le z_j + d_j - 1 \\ 0 & \text{otherwise} \end{cases} \quad j=1,\dots,J;\ t=1,\dots,T \tag{7}$$

$$p_{jt} = g_j(t - z_j + 1)u_{jt} \quad j=1,\dots,J;\ t=1,\dots,T \tag{8}$$

$$\sum_{j=1}^{J} p_{jt} + b_t \le C_t \qquad t=1,\dots,T \tag{9}$$

$$T1_j \le z_j \le T2_j - d_j + 1 \quad j=1,\dots,J \tag{10}$$

Note: If there are different possible disjoint slots for the operation of load j, constraints in Eq. (10) should be replaced by disjunctive constraints such as $(T1_j^1 \le z_j \le T2_j^1 - d_j + 1) \vee (T1_j^2 \le z_j \le T2_j^2 - d_j + 1) \vee \dots$, where $[T1_j^k, T2_j^k], k = 1, 2, \dots$, are the different time slots in which load j is allowed to operate.

3. Population-based algorithms

Two algorithms are proposed to solve the bi-level model presented in Section 2 using population-based meta-heuristics at both upper and lower levels. Population-based algorithms allow for the simultaneous exploration of different electricity prices to charge to the consumer (upper-level solutions) and, for each of them, different scheduling plans for shiftable appliances (lower-level solutions). The proposed approaches are a bi-level evolutionary algorithm (BLEA) and a bi-level particle swarm optimization algorithm (BLPSO). Alves et al. [1] proposed a hybrid algorithm (H-BLEA) to solve a model equivalent to the one in Eq. (2) – (10), in which the lower-level problem in Eq. (6) – (10) is transformed into a MILP model. The upper-level problem is solved using a genetic algorithm and a MILP solver is used to obtain the (exact) optimal solution to the lower-level problem for each upper-level setting (electricity prices charged to the consumer along the planning period). However, transforming the lower-level problem into a MILP requires a very high number of binary variables, which leads H-BLEA to become impracticable for large-scale problems. Therefore, BLEA and BLPSO are also aimed at overcoming this limitation of hybrid approaches integrating an exact solver.

The proposed algorithms use the same structure to represent the populations. In both cases, the population consists of N individuals split into n_s sub-populations, each one containing N_l individuals (see Fig. 1). All individuals (x, z) of a sub-population have the same values for the upper level variables (i.e., same x). The upper level real-valued vector $x = (x_1, x_2, \cdots, x_I)$ represents an electricity price setting and the lower level integer-valued vector $z = (z_1, z_2, \cdots, z_J)$ represents the time interval in which each load starts its working cycle. Both algorithms are described below.

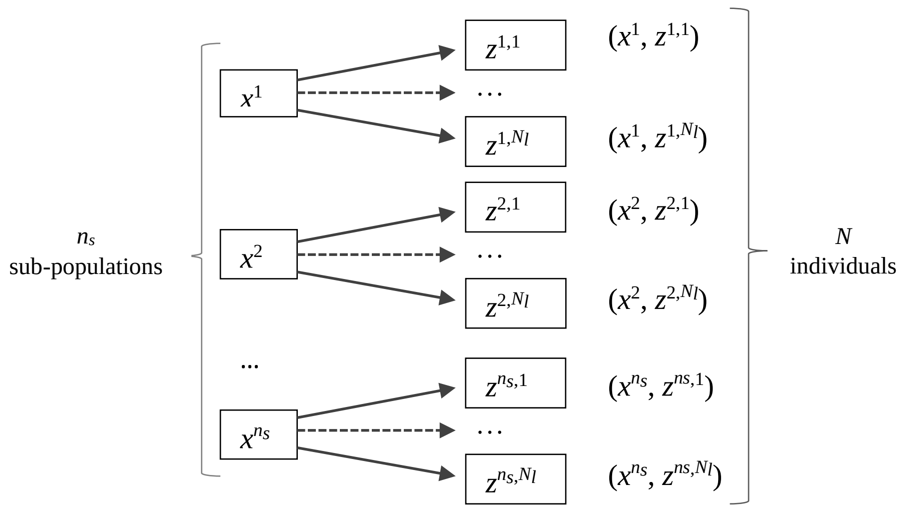

Fig.1 – Structure of the population: N individuals split into n_s sub-populations with N_l individuals each; x^s, s= 1,…,n_s, denotes the s^{th} upper-level component and z^{s,n_l},n_l=1,..., N_l denotes the n_l th lower-level component of each individual in the sub-population s.

3.1. Bi-level Evolutionary Algorithm (BLEA)

The BLEA solves the bi-level problem by applying genetic operators to individuals of the population at both upper- and lower-level sub-problems. The algorithm consists of three steps (see BLEA pseudocode below). In step 1, a new population *Pop*1 (as represented in Fig. 1) is created through selection, crossover and mutation operators. The population individuals are initially created randomly within the pre-defined ranges; first, the values of the upper-level variables are generated, and then the values of the lower-level variables are obtained. In order to obtain a feasible price setting x, the vector generated undergoes the repairing procedure described below to satisfy the constraint in Eq. (5) regarding average price compliance during the planning period.

Repairing procedure(x)

$S = \sum_{i=1}^{I} \bar{P}_i x_i$

Set $A = \{1,2,\cdots,I\}$, the set of indices i of x that are allowed to be changed

While ($S \neq Tx^{AVG}$) and ($A \neq \emptyset$)do

  $\Delta = Tx^{AVG} - S$

  $P = \sum_{i \in A} \bar{P}_i$

  For each $x_i,\ i \in A$

    $x_i \leftarrow x_i + \Delta/P$

  End For

  For $i = 1$ to I

    If $x_i < \underline{x_i}$ then

      $x_i \leftarrow \underline{x_i}$

      $A = A \backslash \{i\}$

    Else If $x_i > \overline{x}_i$ then

      $x_i \leftarrow \overline{x}_i$

      $A = A \setminus \{i\}$

    End If

  End For

  $S = \sum_{i=1}^{I} \bar{P}_i x_i$

End While

If $S = Tx^{AVG}$ then Return x

Else Discard x /* $A = \emptyset$ */

End If

If it is not possible to obtain a feasible x vector, a new x vector is generated and the repairing procedure is reinitiated. After obtaining a feasible x vector, the lower-level vector z is generated, satisfying the bounds in Eq. (10). If it does not satisfy constraints in Eq. (9), which occurs when the contracted power is exceeded during some interval of time, a new z vector is generated. Steps 2 and 3 of BLEA constitute the main cycle of the algorithm.

In step 2, a new population *Pop*2 is created using the selection, crossover and mutation operators to obtain the upper- and lower-level solution vectors. For each sub-population of *Pop*2, the upper-level vector x is firstly created, which is the same for all individuals of that sub-population, and then the lower-level vector z for each individual is obtained.

To obtain the upper-level vector x, two parents from different sub-populations of *Pop*1are selected: one is obtained by binary tournament, selecting the solution with the best value for the upper-level objective function, and the other one is randomly chosen. A one-point crossover operator is applied to generate vector x^c, with an equal chance of each parent vector giving the first or the second part of x^c.

Then, the mutation operator is applied to each variable x_i^c with probability P_m, consisting of adding or subtracting a positive perturbation randomly generated in the range $[0, 0.2(\overline{x}_i - \underline{x_i})]$ to generate x_i^m. If any variable of the vector x^m is outside the bounds imposed by Eq. (3) and Eq. (4), then it is pushed to the closest bound. The index of this variable is excluded from set A of indexes that correspond to variables allowed to be changed in the repairing procedure. If the individual obtained is feasible, then it is accepted as an x vector of *Pop*2; otherwise, the process is repeated to achieve a feasible individual.

To obtain the optimal lower-level vector z for the x vector of each sub-population of *Pop*2, the lower-level algorithmic operations are performed during k_l iterations, evolving individuals of the sub-population through selection, crossover and mutation. These operators are performed in a similar manner to those of the upper-level. After obtaining z^c by one-point crossover, the mutation operator is applied to each variable z_j^c with probability P_m, consisting of adding or subtracting a positive integer perturbation randomly generated in the range $[0, 0.2(T2_j - d_j + 1 - T1_j)]$ to generate z_j^m. If any variable of the z^m vector is out of bounds concerning the admissible time slots for load operation defined by Eq. (10), it is pushed to the closest bound. If z^m leads to the violation of constraints in Eq. (9), due to Eq. (7) and Eq. (8), the process is repeated, generating a new z vector until a feasible individual is obtained.

In each lower-level iteration, this process is repeated for the N_l individuals of each sub-population, thus generating the *Offspring*. The *Offspring* will compete with the current *Pop*2 to generate the new population *Pop*2f or the next lower-level iteration.

In step 3, populations *Pop*1 and *Pop*2 are merged and the best lower-level solution for each sub-population is identified. Then, among the solutions identified, the best solution according to the upper-level objective function is selected, and the sub-population containing this solution is included in the new *Pop*1 for the next iteration, thus accounting for some elitist pressure. The remaining $n_s - 1$ sub-populations are then chosen by binary tournament, without replacement, between *Pop1* and *Pop2*. The binary tournaments are performed between the best solutions identified in each sub-population according to the upper-level objective function.

The pseudocode of BLEA is presented below.

BLEA pseudocode

```
(Step 1) Initialize Pop1 containing n_s sub-populations Pop1_s with N_l individuals each:
         randomly generate n_s feasible x and, for each one, randomly generate N_l feasible
         z vectors.
         Assess each population individual (x, z) evaluating F(x, z) and f(x, z).
For k = 1, ..., K
  (Step 2)
        For s = 1, ..., n_s
        Repeat
                Select x^a and x^b from Pop1, one randomly and the other by
                binary tournament
                x^cross ← Crossover of x^a and x^b
                x^mut ← Mutation of x^cross
                Repairing procedure (x^mut)
        Until x^mut is feasible for the upper level
         Create Pop2_s with x^mut and all z in Pop1_s:
                Pop2_s = {(x^mut, z^{s,1}), (x^mut, z^{s,2}), ..., (x^mut, z^{s,Nl})}
        For k_l = 1, ..., K_l do
                Offspring_s = ∅
                For n_l = 1, ..., N_l
                Repeat
                        Select z^{s,a} and z^{s,b} from Pop2_s, one randomly and
                        the other by binary tournament
                        z^{s,cross} ← Crossover of z^{s,a} and z^{s,b}
                        z^{s,mut} ← Mutation of z^{s,cross}
                Until z^{s,mut} is feasible for the lower level
                 Insert (x^mut, z^{s,mut}) as (x^s, z^{s,n_l}) into the sub-population Offspring_s
                End For n_l
                Select the best N_l individuals from Pop2_s and Offspring_s according to f
                to create the new Pop2_s for the next iteration (k_l + 1).
        End For k_l
      End For s
  /* select the population for the next generation */
  (Step 3) Update Pop1 for generation k + 1 by copying the sub-population of
      Pop1 ∪ Pop2 containing the best solution according to F found so far, and
      performing n_s − 1 binary tournaments without replacement between the sub-
      populations of Pop1 and Pop2.
End For k
```

3.2. Bi-level Particle Swarm Optimization Algorithm(BLPSO)

While many evolutionary algorithms create a new population in each generation, in PSO, the individuals (particles) of the population are moved from one iteration to the next, changing their positions. In each iteration of a PSO algorithm, the individuals are moved according to Eq. (11) and Eq. (12).

$$v^q \leftarrow wv^q + c_1 rand(\quad)(pbest^q \quad - y^q) + c_2 rand(\quad)(gbest - y^q) \qquad (11)$$

$$y^q \leftarrow y^q + v^q,\ q = 1, \dots, n \qquad (12)$$

In these equations, y denotes the position of the particle (solution vector), v is the velocity vector, w is the inertia weight, c_1 and c_2 are the cognitive and social parameters, $rand(\,)$ is a random uniform value in the interval $[0, 1]$, q is the index of the particle and n is the number of particles in the population. The $pbest^q$ represents the best fitness position (solution) achieved by individual q, and $gbest$ represents the best position achieved by all individuals of the population during the iterations performed so far. The w parameter is set linearly decreasing, as presented in Shi and Eberhart [11]. This aims at increasing the exploration ability of the algorithm at earlier iterations and the exploitation ability at later iterations. The BLPSO algorithm is developed using the PSO approach to solve the bi-level problem (see BLPSO pseudocode below).

The algorithm starts by initializing a population of N individuals split into n_s sub-populations, each one with $N_l = \frac{N}{n_s}$ individuals (see Fig. 1). Each individual has two components, the upper level x and the lower level z . For each sub-population, the upper-level vector is the same and randomly generated; the repairing procedure described for the BLEA to guarantee that x satisfies the constraint in Eq. (5) is applied. If the repairing procedure fails to yield a feasible x, the process is repeated until a feasible upper-level vector is obtained. Then, the lower-level vectors z for the individuals of the sub-population are also randomly generated. If the z vector of an individual does not satisfy constraints in Eq. (9), a new attempt is made and the process is repeated until a feasible z is obtained.

The best position achieved by an individual in the main (upper-level) cycle, $pbest1$, is initialized with its current position and the best solution found in the population is saved in the temporary vector pre_gbest, to be used in the first iteration of the algorithm. This solution is provisional because the lower-level optimization phase has not yet been executed.

The main cycle of the algorithm is executed for K iterations. Firstly, the upper-level part x of each individual is updated using the PSO operators in Eq. (11)-(12) with $y = x$ and the bounds in Eq. (3)-(4) and constraint (5) are checked to ensure the feasibility of the resulting individual. If any of the bounds is violated, the corresponding variable is pushed to the closest bound; for the constraint in Eq. (5), the repairing procedure referred to above is performed.

After obtaining the upper-level vector of each sub-population, the lower-level algorithmic operations are executed for k_l iterations aiming at obtaining the lower-level optimal solution for each sub-population. The lower-level search uses $gbest^s$ (the best solution of each sub-population s according to *f*) and a local personal best $pbest2$ is defined for each individual. At each lower-level iteration, the lower-level vectors of the N_l individuals are also updated using Eq. (11)-(12) with $y = z$ and the feasibility of each individual is checked. If the updated individual does not satisfy the bounds established in Eq. (10) and/or violates constraints in Eq. (9), a repairing process is performed during a maximum of L trials. This lower-level repairing process consists of pushing the particle to the closest bounds defined in Eq. (10), and it is moved according to Eq. (11)-(12). Then the constraints in Eq. (9) are checked again. If the particle remains infeasible for more than L trials, it is randomly reinitialized and the update process is repeated until a feasible position is achieved. After concluding the update process of lower-level individuals, $pbest2$ of each individual and $gbest^s$ are updated according to *f*.

At the end of the lower-level optimization stage, $pbest1$ of each individual is updated according to F in the main cycle. This step also updates the best solution found so far ($gbest$) considering the entire population. It should be noted that, after the first iteration, the best solution of the entire population replaces the temporary pre_gbest solution. The $pbest1$ and $gbest$ update process starts by analyzing the value associated with each particle position regarding the upper level objective function F and, if the new position is better, it replaces the former. Otherwise, if both positions have the same F value, then the comparison is performed relative to the lower-level objective function *f*; if the new position has a better value according to f, it replaces the former. It is worth noting that only the individuals that hold as sub-population $gbest^s$ for at least r_0 lower-level iterations are tested to replace the $gbest$ solution of the entire population. In the experiments performed, this parameter has been found to be of great relevance for improving the ability of the algorithm to achieve better solutions to the problem. This process imposes some elitist pressure, as a $gbest^s$ of a sub-population s must prove to be the best for some iterations until it can be elected as the global $gbest$ for the entire population.

The pseudocode of the BLPSO algorithm as well as the procedure to update $gbest$ and $pbest1$ are presented below.

BLPSO algorithm pseudocode

Initialize Pop containing n_s sub-populations Pop_s with N_l individuals each: randomly generate n_s feasible x vectors and, for each one, randomly generate N_l feasible z vectors.
Assess each population individual (x, z) evaluating $F(x, z)$ and $f(x, z)$.
Initialize the best position of each individual: $pbest1^{s,n_l} = (pbest1^s_x, pbest1^{s,n_l}_z) = (x^s, z^{s,n_l}), s = 1, \dots, n_s,\ n_l = 1, \dots, N_l$.
Assign to pre_gbest the best individual of Pop according to F, to be used as $gbest$ in the first iteration.

```
For k = 1, ⋯, K
    For s = 1, …, n_s
    Repeat
        Update the component x^s of the individual using Eq. (11) and (12) with
        y = x^s, pbest1_x^s and gbest_x
        Repairing procedure (x^s)
    Until x^s is feasible for the upper level
    Initialize the best solution gbest^s of Pop_s according to f
    Initialize pbest2^{s,n_l} = (pbest2_x^s, pbest2_z^{s,n_l}) = (x^s, z^{s,n_l}), n_l = 1, …, N_l
    For k_l = 1, ⋯, K_l
            For n_l = 1, …, N_l
            l=0
            Repeat
            l ← l + 1
                Update the component z^{s,nl} of each individual using Eq. (11) and (12)
                with y = z^{s,nl}, pbest2_z^{s,n_l} and gbest_z^s
                If z^{s,nl} does not satisfy the bounds in Eq. (10), then push it to the
                closest bounds
                If z^{s,nl} does not satisfy constraint in Eq. (9) and l > L, then z^{s,nl} is
                reinitialized and l=0
            Until z^{s,nl} is feasible for the lower level
              Update the best position pbest2^{s,n_l} achieved by individual n_l of Pop_s
              according to f: if f(x^s, z^{s,nl}) < f(pbest2^{s,n_l}) then pbest2_z^{s,n_l} ← z^{s,nl}
            End For n_l
            Update the best individual gbest^s of Pop_s according to f
    End For k_l
    End For s
    Update procedure of gbest and pbest1
End For k
```

Update $gbest$ and $pbest1$

For $s = 1, \dots, n_s$
/* update $pbest1$ of each individual */
For $n_l = 1, \dots, N_l$
If $[F(x^s, z^{s,nl}) > F(pbest1^{s,n_l})]$ or
$[F(x^s, z^{s,nl}) = F(pbest1^{s,n_l})$ and $f(x^s, z^{s,nl}) < f(pbest1^{s,n_l})]$ then
$pbest1^{s,n_l} \leftarrow (x^s, z^{s,nl})$
End For n_l
/* update $gbest$ */
Let r be the number of times $gbest^s$ was kept in Pop_s
If $r > r_0$
Update the $gbest$ of the whole population *Pop* using $gbest^s$ according to *F*
End For s

4. Comparison of algorithms - Experimental results and discussion

4.1. Problem data and parameter setting

To perform a comparative analysis between BLPSO, BLEA and the hybrid genetic algorithm (H-BLEA) proposed by Alves et al. [1], a case study with three profiles for the windows of appliance operation is considered. Most data were obtained from actual audit information and some values were estimated.

A planning period of 24 hours split into 15-minute intervals is considered, leading to T={1,…,96}. The load is composed of a base load (depicted in Fig. 2; see also the table in Appendix), which is associated with appliances that cannot be controlled, and of five shiftable appliances (J=5): laundry machine, dishwasher, electric water heater (EWH), drying machine and electric vehicle. The work cycles of these appliances are displayed in Fig. 3, which shows the power $g_j(r)$ requested by load j at time r of its work cycle (r =1,…,d_j).

The contracted power C_t is 4.6 kW for t =28,…,84 and 3 kW for the other $t \in$ T.

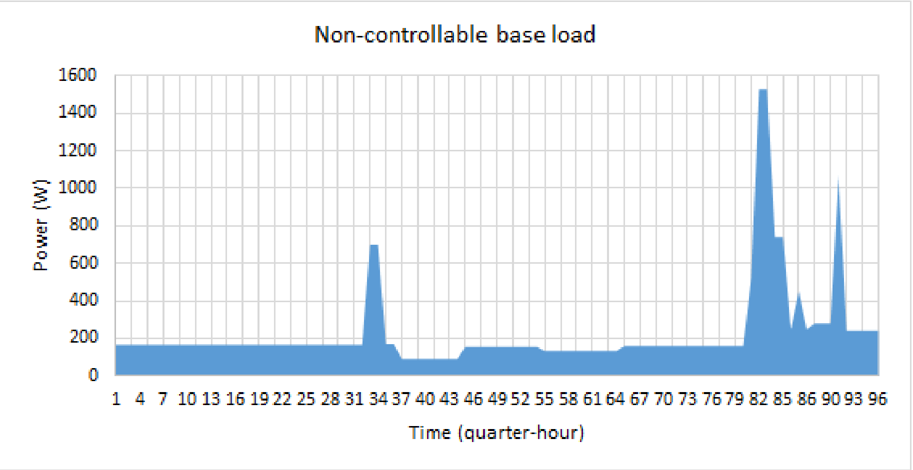

Fig. 2. Power requested from the grid by the base (non-controllable) load.

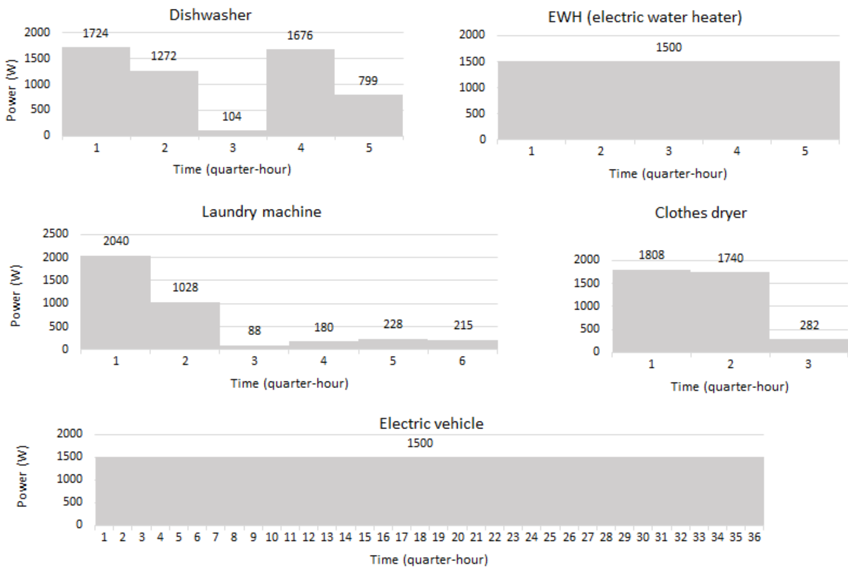

Fig. 3 – Operation cycles of the shiftable appliances

The minimum and maximum electricity prices that can be charged to the consumer in each of the seven sub-periods of time (I=7) are displayed in Table 1. The average energy price for the overall planning period is set to x^{AVG}=0.116 €/kWh. The electricity prices seen by the retailer at the spot market are displayed in Fig. 4. All prices are in €/kWh, so they were then converted to periods of quarter-hours ($\hbar = {}^{1}/_{4}$ h) to feed the model.

Table 1.Minimum ($\underline{x}_i$) and maximum ($\overline{x}_i$)electricity prices that can be charged to the consumer in each sub-period

Sub-period	**Start interval** (t)	**End interval** (t)	**Minimum price** (€/kWh)	**Maximum price** (€/kWh)
1	1	28	0.04	0.10
2	29	38	0.08	0.24
3	39	44	0.03	0.12
4	45	60	0.10	0.28
5	61	76	0.03	0.12
6	77	84	0.08	0.24
7	85	96	0.04	0.10

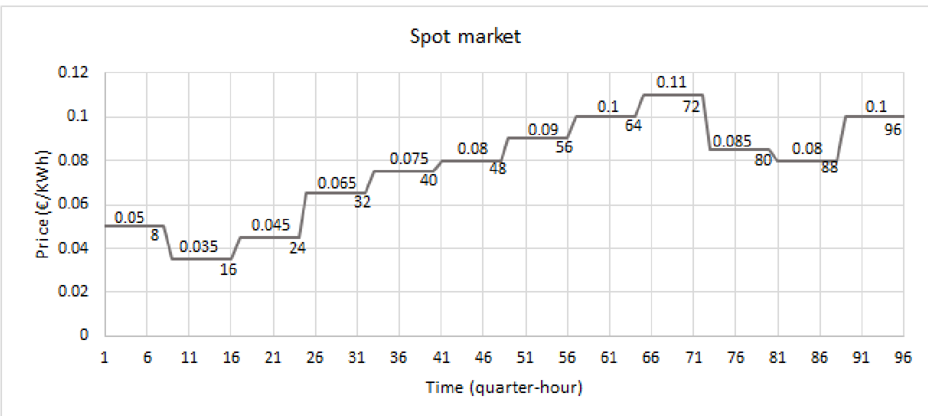

Fig. 4.Prices at spot market seen by the retailer.

The work cycle of the five shiftable appliances can be scheduled within the planning period, i.e., their habitual operational slot can be shifted according to electricity prices and consumer preferences. Three different profiles of consumer behavior are considered regarding the width of the time slots in which load operation is allowed. In addition to a base profile, reflecting the most habitual load scheduling pattern, a restricted profile with a lower tolerance for load shifting, and an extended profile with a higher tolerance for complying with load operation in larger time slots, are considered. These profiles are associated with more or less stringent search spaces regarding constraints in Eq. (10) and therefore with the number of potential lower-level solutions. The time slots allowed for appliance operation relative to each profile are shown in Table 2.

Table 2.Comfort time slots [earliest interval $T1_j$– latest interval$T2_j$] allowed for the operation of each appliance in the base, restricted and extended profiles.

	Appliances				
Profile	**Dishwasher**	**Laundry machine**	**EWH**	**Electric vehicle**	**Clothes dryer**
Base	[1 – 36]	[32 – 60]	[24 – 40]	[1 – 48]	[76 – 96]
Restricted	[1 – 34]	[32 – 50]	[24 – 36]	[1 – 45]	[70 – 82]
Extended	[1 – 44]	[28 – 65]	[24 – 45]	[1 – 48]	[70 – 96]

In BLEA, a population of $N = 240$ individuals split into $n_s = 12$ sub-populations, each containing $N_l = 20$ individuals, was considered. BLEA was run for $T = 100$ generations, each performing the lower-level optimization task for $T_l = 40$ iterations. In the BLPSO algorithm, the parameters $N = 240$, $N_l = 30$, $T = 100$ and $T_l = 60$ were considered and the $r0$ parameter was set to 4. The parameter w was set to decrease linearly from 0.9 (at the beginning of the algorithm) to 0.4 (at the end of the algorithm).

The algorithm presented by Alves et al. [1], H-BLEA, was also implemented for comparison with BLEA and BLPSO algorithms. All algorithms were implemented in Matlab running on an Intel Core i7 3.2 GHz 32 GB RAM machine. The upper-level search of H-BLEA is similar to that of BLEA, and the lower-level problem is exactly solved using Cplex for each instantiation of x. Thus, the lower-level optimal solutions obtained by H-BLEA enable us to assess the ability of BLEA and BLPSO algorithms to determine good quality solutions to the problem. It is worth mentioning that a feasible solution to the bi-level problem must be an optimal solution to the lower-level problem.

In H-BLEA, a population of $N = 30$ individuals (in this case $n_s = N$ because a single lower-level solution is associated with each upper-level solution) was considered. H-BLEA was run for $T = 100$ generations.

4.2. Results

The problem was solved for the three profiles shown in Table 2 using BLEA, BLPSO and H-BLEA algorithms. The experimentation consisted of running the three algorithms to obtain 60 valid solutions for each profile. At the end of each run, solutions are checked for validity. A solution is valid if it is truly optimal to the lower-level problem for the price configuration set in the upper level. Therefore, in H-BLEA, 60 runs were required, since the lower level is exactly solved. In BLPSO, just 60 runs were necessary in the extended profile; in the other profiles, a few more runs (76 in the base profile and 75 in the restricted profile) were required to obtain 60 valid solutions. BLEA displayed difficulties in obtaining optimal lower-level solutions in the extended profile, in which on average only 1 solution was valid in every 8 runs. BLEA only needed 60 runs in the restricted profile and it required 64 runs in the base profile. In the extended profile, the search space is larger, and BLEA had difficulty in escaping from sub-optimal solutions to the lower-level problem.

Information about the 60 valid solutions obtained for the base, restricted and extended profiles is presented in Tables 3-5 and in Fig. 5. Table 3 displays the best (maximum), worst, mean and standard deviation for retailer profit (F); it also shows the consumer's electricity bill (f) in the best solution, i.e., with maximum F. The consumer's electricity bill and retailer profit are both expressed in €. The results refer to a period of 24 hours and correspond to a cluster of 1,000 consumers with similar energy consumption and demand response patterns. Fig. 5 shows box plots for retailer profit (F), providing a comparison of the median and interquartile ranges of F in valid solutions obtained with the three algorithms in the three profiles. Table 4 displays the energy prices (upper-level variables) and Table 5 displays the ranges of working intervals (lower-level solution) in the best solution obtained with each algorithm for each profile.

In comparison with the base profile, the main difficulty in the restricted profile is obtaining feasible lower-level solutions, whereas the major difficulty in the extended profile relates to the existence of local optimal lower-level solutions. Note also that the contracted power constraint in Eq. (9) and the characteristics of the load operation cycles (Fig. 3) introduce additional difficulties in determining optimal solutions.

BLPSO obtains the best solutions in all profiles regarding the maximum and the median values of *F*, as well as the best mean of *F* in the base and extended profiles (*cf.* Table 3 and Fig. 5). BLPSO is the algorithm that displays the highest variability across runs, as measured by standard deviation. This variability is mainly due the presence of outliers (see Fig. 5), which justifies why the mean of *F* computed by BLPSO is not always the best in all profiles. The largest variation of the maximum *F* given by any algorithm with respect to the maximum *F* of all algorithms is just 0.50% in the base profile, 0.41% in the restricted profile and 1.13% in the extended profile. The largest variation in the mean of *F* is 0.68% across all profiles. For the 60 valid solutions, the deviation between the maximum *F* and the worst *F* across the 3 profiles ranges between 8.8% and 9.7% in BLPSO, 2.5%-4.5% in BLEA and 2.3%-4.9% in H-BLEA. These values denote the robustness of the algorithms to obtain consistently good solutions.

H-BLEA gives the best standard deviation of *F* in the base and restricted profiles and is not far from the best in the extended profile, in which BLEA displays the best value.

The Kruskal-Wallis statistical test was performed to assess the significance of the differences in retailer profit *F* across the three algorithms. At the level of significance $\alpha=0.05$, these differences are statistically significant for all profiles.

Table 3. Retailer profit and customer cost (for the best solution according to *F*) for 60 valid solutions obtained in the base, restricted and extended profile models.

Profile	*Algorithm*	*Maximum F*	*Worst F*	*Mean F*	*Std dev F*	*f*
Base	BLPSO	**1827.639**	1649.959	**1803.529**	50.922	3357.584
	BLEA	1826.810	1743.808	1796.915	18.268	3356.755
	H-BLEA	1818.490	1757.000	1801.506	13.337	3351.680
Restricted	BLPSO	**1825.732**	1656.247	1805.484	29.504	3378.776
	BLEA	1818.161	1771.928	1803.582	12.713	3329.201
	H-BLEA	1818.500	1776.000	**1812.874**	6.882	3329.500
Extended	BLPSO	**1493.347**	1361.399	**1469.164**	26.726	3047.809
	BLEA	1476.473	1437.628	1459.160	11.108	3039.473
	H-BLEA	1481.960	1409.000	1462.371	14.769	3040.170

Table 4**.** Energy prices x_i (€/kWh) for each sub-period in the best solution obtained for the base, restricted and extended profile models.

		Pricing sub-periods						
Profile	*Algorithm*	P_1 [1, 28]	P_2 [29, 38]	P_3 [39, 44]	P_4 [45, 60]	P_5 [61, 76]	P_6 [77, 84]	P_7 [85, 96]
Base	BLPSO	0.10	0.24	0.12	0.101	0.03	0.24	0.10
	BLEA	0.10	0.24	0.12	0.100103	0.030897	0.24	0.10
	H-BLEA	0.099843	0.239843	0.119835	0.101761	0.031761	0.235828	0.10
Restricted	BLPSO	0.10	0.24	0.12	0.10	0.066648	0.24	0.052470
	BLEA	0.10	0.24	0.12	0.120143	0.048983	0.24	0.049166
	H-BLEA	0.09999	0.23999	0.11999	0.120020	0.051141	0.23999	0.046493
Extended	BLPSO	0.10	0.24	0.12	0.10	0.060571	0.24	0.060571
	BLEA	0.10	0.24	0.12	0.100642	0.058904	0.24	0.061939
	H-BLEA	0.099989	0.239989	0.119950	0.100067	0.060634	0.239926	0.060507

Table 5. Appliance work intervals in the best solution obtained for the base, restricted and extended profile models. The first value of each interval for each appliance is z_j (starting time of load j).

		Appliances				
Profile	**Algorithm**	**Dishwasher**	**Laundry machine**	**EWH**	**Electric vehicle**	**Clothes dryer**
Base	BLPSO	1-5	45-50	36-40	5-40	85-87
	BLEA	1-5	45-50	36-40	5-40	85-87
	H-BLEA	1-5	48-53	36-40	5-40	85-87
Restricted	BLPSO	5-9	39-44	28-32	9-44	74-76
	BLEA	1-5	39-44	28-32	5-40	73-75
	H-BLEA	1-5	39-44	28-32	5-40	74-76
Extended	BLPSO	39-43	60-65	39-43	1-36	85-87
	BLEA	39-43	60-65	41-45	1-36	74-76
	H-BLEA	39-43	60-65	41-45	1-36	85-87

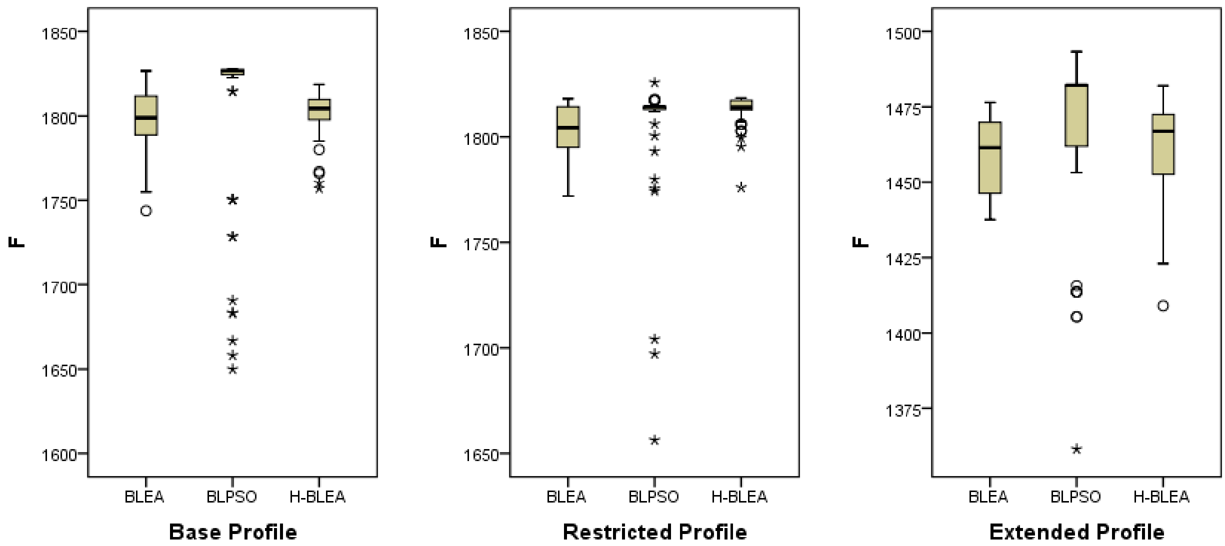

Fig. 5. Box plots of retailer profit (F) for the base, restricted and extended profile models (the symbols "o" and "*" denote outliers, i.e., values beyond 1.5 and 3 times the interquartile range, respectively).

In the base profile, the lower-level optimal solutions obtained by the three algorithms only differ in the time range the laundry machine operates, although this occurs in the same pricing sub-period P_4. In the restricted profile, the lower-level solutions differ in the operation time range of three appliances; for the dishwasher and clothes dryer, the different schedules are within the same pricing sub-periods (P_1 and P_7, respectively) and the operation schedule for the electric vehicle spans three pricing sub-periods (P_1 to P_3) with distinct intersection degrees. As expected, the algorithms obtain different lower-level solutions in the extended profile because the search space is larger due to the wider time slots for load operation. Notably, the clothes dryer operates in P_5 in the solution obtained by BLEA and in P_7 in the solutions obtained by BLPSO and H-BLEA. This is because the price set by BLEA in sub-period P_7 is higher than the price in sub-period P_5 and the reverse relation holds for H-BLEA, being equal for BLPSO.

Indicators of run times (in seconds) of each algorithm over 60 runs, considering the base profile, are presented in Table 6. The run times of the two algorithms proposed in this paper, BLPSO and BLEA, are similar, although BLEA takes slightly more time to solve the problem. H-BLEA takes more time because it calls on an external solver to compute the optimal solution to the lower-level problem. The computational effort of each algorithm is measured and balanced in terms of run time instead of the number of function evaluations because the latter is difficult to obtain for H-BLEA. The relations between the computational times of the three algorithms are similar for the remaining profiles.

Table 6. Run time (in seconds) of the algorithms for the base profile model.

Algorithm	Maximum	Median	Mean	std. deviation
BLPSO	552.90	528.10	530.20	9.77
BLEA	639.47	627.32	627.67	3.99
H-BLEA	1133.52	1107.16	1109.25	10.43

The load diagrams associated with the best solution for each profile are depicted in Figs. 6-8 in stacked area charts. All these solutions were obtained by BLPSO. In the best solution for the restricted profile, peak consumption is higher due to the reduction of the available operation range for the laundry machine, which is the appliance with the highest maximum consumption, although this is partially compensated by shifting the operation of the EWH to a period with lower consumption. In the extended profile, the higher flexibility in load operation leads to more diversified shifts to obtain better lower-level solutions in face of the prices set by the retailer.

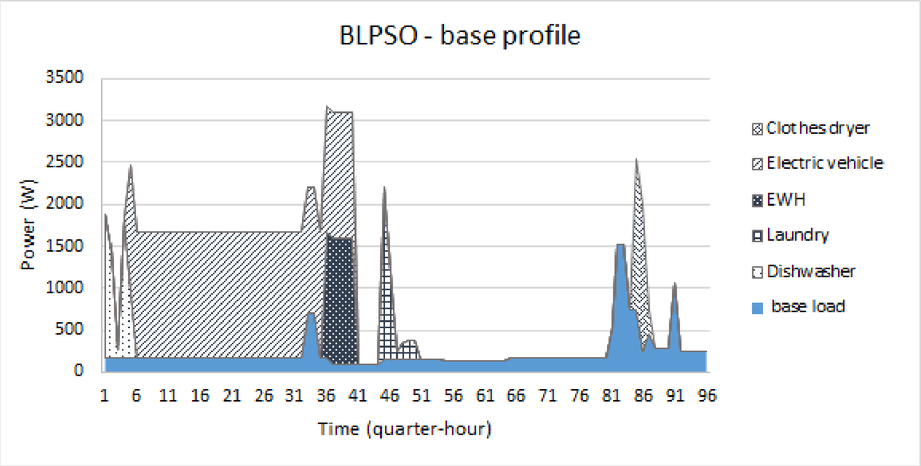

Fig. 6. Load diagram of the best solution for the base profile model, obtained by BLPSO

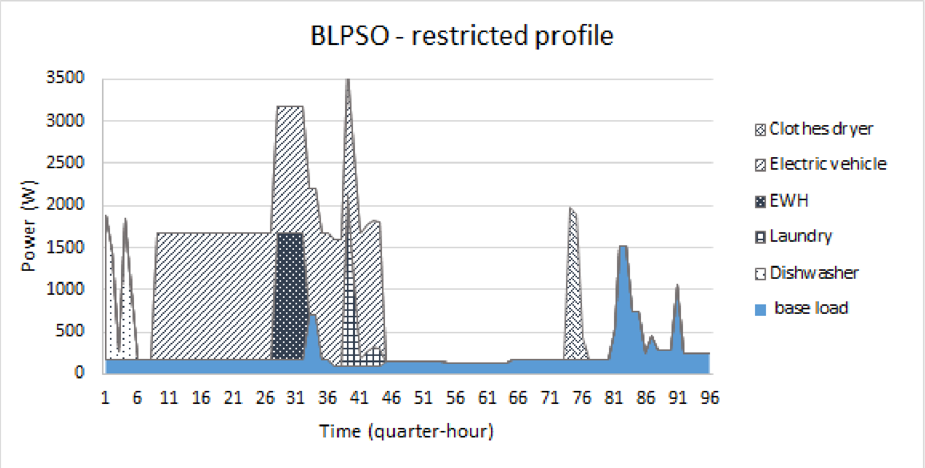

Fig. 7. Load diagram of the best solution for the restricted profile model, obtained by BLPSO

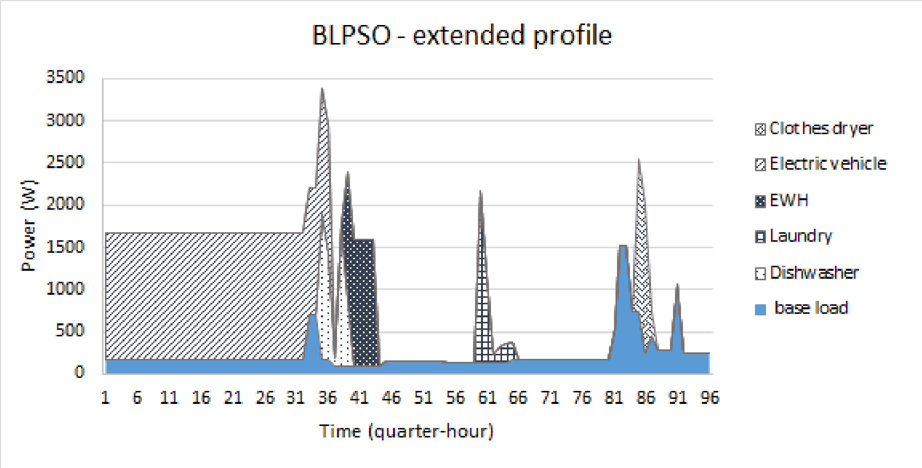

Fig. 8. Load diagram of the best solution for the extended profile model, obtained by BLPSO

Two other scenarios have been tested, which consider the same data but change the allowable time slots for load operation in each profile (base, restricted and extended), maintaining the width of the time slots. Therefore, six additional problems were tested. In both scenarios, a higher simultaneity of slots was considered, which is more challenging for the algorithms. In scenario 1, the allowable time slots were more concentrated in the middle of the planning period, in which the lower/upper bounds of prices charged to the consumer are higher. In scenario 2, the allowable time slots were more concentrated at the end of the planning period, in which the uncontrollable load is higher and the spot market prices seen by the retailer are also higher. Scenario 2 is less favorable for retailer profit.

In scenario 1, BLPSO displays the best values of F for all profiles, while the best mean value is obtained by BLEA for all profiles. BLEA gives the best mean value of F and H-BLEA gives the best standard deviation for all profiles. In scenario 2, BLPSO only presents the best value of F for the extended profile, while BLEA gives the best F value for the restricted profile and H-BLEA for the base profile. BLEA gives the best mean and standard deviation values for all profiles. In both scenarios 1 and 2, BLPSO presents the highest standard deviation for all profiles. H-BLEA presents the worst value for the maximum F for all profiles in scenario 1. However, the largest variation of the maximum F given by any algorithm with respect to the maximum F of all algorithms is 2.11% (given by H-BLEA) across all profiles, in the two scenarios.

The analysis of the nine problem instances (3 profiles × 3 scenarios) leads to the following main conclusions. BLPSO found the best solution (maximum F) in 7 of the 9 instances, whereas BLEA and H-BLEA computed the best solution in just one instance each; yet in the 2 instances BLPSO does not give the maximum F, it never deviates more than 0.47% from it. BLEA experiences some difficulties in obtaining lower-level optimal solutions for the extended profiles. H-BLEA and BLEA display the worst of the maximum F in 4 of the 9 instances each. H-BLEA shows the lowest standard deviation in 5 out of 9 instances, and BLEA is in this position in the remaining 4 instances. BLPSO shows a larger variability than BLEA and H-BLEA, as measured by the standard deviation in the 9 instances, i.e., it obtains very good solutions in some runs but also worse solutions than those obtained by the other algorithms in other runs. This variability seems to result from the higher repair rate required by infeasible lower-level solutions due to the specificities of particle movement in particle swarm optimization, in comparison with the other algorithms.

We also attempted to solve this model with a recent generic bi-level optimization algorithm, Bilevel Evolutionary Algorithm based on Quadratic Approximations – BLEAQ-II, for which a Matlab code is publicly available [17]. Sixty runs of BLEAQ-II have been carried out for the base profile model. BLEAQ-II could not find feasible solutions to the upper-level problem due to the constraint in Eq. 5 in the mathematical model, which imposes an average price in the planning period. Moreover, BLEAQ-II was not able to find true optimal solutions to the lower level problem for any setting of the upper level variables. This justifies the need to develop algorithms for bi-level optimization models with specific features, such as the ones presented in section 2.2, making the most of the model characteristics to design efficient algorithms able to deal effectively with the computational difficulties arising in bi-level optimization, even for simple problems. BLPSO, BLEA and H-BLEA algorithms were endowed with a customized repairing procedure to deal with the upper-level equality constraint. In addition, these algorithms were able to guarantee the optimality of the lower-level solutions in almost all cases, except for BLEA in the extended profile.

5. Conclusions

Two bi-level population-based algorithms were proposed in this paper, one based on an evolutionary algorithm and the other on particle swarm optimization, to determine the optimal electricity prices a retailer can set to maximize profit in the face of demand response strategies set by consumers minimizing their electricity bills. The load scheduling of appliances depends on the prices set by the retailer and consumer comfort requirements, established by more or less stringent time slots for load operation. This bi-level model was also solved by the hybrid algorithm H-BLEA. This algorithm uses an exact solver to obtain an optimal solution to the lower-level problem, formulated as a mixed-integer linear programming problem. Since this model generally has multiple alternative optimal solutions to the lower-level problem for a given upper-level solution, the use of population-based meta-heuristics enables the algorithm to explore them. In addition, H-BLEA becomes impracticable for higher dimensional problems, for which BLPSO and BLEA approaches present a good computational performance.

To assess the ability of the algorithms to obtain practical solutions, three profiles based on consumer willingness to shift load operation were considered, which are reflected on more or less stringent constraints of the lower-level problem. Two additional scenarios were also considered, which were more challenging due to changes in the allowable load operation time slots in each profile leading to a higher simultaneity of slots. BLPSO performed better than BLEA and H-BLEA in most cases in terms of providing the best solution within 60 valid solutions obtained from independent runs of the 9 instances, but it also displayed higher variability. The capability of BLPSO and BLEA to exploit multiple alternative optimal solutions to the lower-level problem, which may present different retailer profit values for the same consumer cost, justifies why these algorithms generally obtain better values for retailer profit than H-BLEA regarding the maximum value computed in all runs.

Further work will consider an explicit objective function in the lower-level problem, assessing consumer discomfort associated with scheduling loads outside their habitual time slots. This increases the complexity of the model, which becomes a semi-vectorial bi-level problem, enabling us to assess the trade-offs between economic and quality of service axes of evaluation. Future work will also include different clusters of consumers, leading to a bi-level multi-follower decision problem [6].

Acknowledgments

This work was partially supported by Projects MITP-TB/CS/0026/2013, UID/MULTI/00308/2013 and by the European Regional Development Fund through the COMPETE 2020 Programme and FCT - Fundação para a Ciência e a Tecnologia within project SAICTPAC/0004/2015- POCI-01-0145-FEDER-016434.

References

[1] Alves, M.J., Antunes, C. H., Carrasqueira, P. A hybrid genetic algorithm for the interaction of electricity retailers with demand response. In: G. Squillero and P. Burelli (Eds.), EvoApplications 2016, Part I, LNCS 9597, 459–474, Springer, 2016

[2] Bard, J.F.: Practical Bilevel Optimization: Algorithms and Applications, Springer Science + Business Media, Dordrecht, 1998.

[3] Bu, S., Yu, F. R., Liu, P. X.: A game-theoretical decision-making scheme for electricity retailers in the smart grid with demand-side management. 2011 IEEE International Conference on Smart Grid Communications - SmartGridComm, 387-391, 2011.

[4] Dempe, S.: Foundations of Bilevel Programming, Kluwer Academic Publishers, Dordrecht, 2002.

[5] Lu, J., Han, J., Hu, Y., Zhang, G.: Multilevel decision-making: A survey. Information Sciences 346 –347, 463–487, 2016.

[6] Lu, J., Shi, C., Zhang, G.: On bilevel multi-follower decision making: General framework and solutions. Information Sciences 176 (11), 1607-1627, 2006.

[7] Meng, F.-L., Zeng, X.-J.: An optimal real-time pricing algorithm for the smart grid: a bilevel programming approach. Imperial College Computing Student Workshop - ICCSW`13, 81-88, 2013.

[8] Meng, F.L., Zeng, X.J.: An optimal real-time pricing for demand-side management: A stackelberg game and genetic algorithm approach. In: 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 1703–1710, 2014.

[9] Meng, F.L., Zeng, X.J.: Appliance level demand modeling and pricing optimization for demand response management in smart grid. In: 2015 International Joint Conference on Neural Networks (IJCNN), 1–8. IEEE, 2015.

[10] Sekizaki, S., Nishizaki, I., Hayashida, T.: Electricity retail market model with flexible price settings and elastic price-based demand responses by consumers in distribution network. Electric Power and Energy Systems, 81, 371-386, 2016.

[11] Shi, Y., & Eberhart, R.: Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, 1945-1950, 1999.

[12] Soares, A., Gomes, A., Antunes, C. H.: Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions, Renewable and Sustainable Energy Reviews 30, 490–503, 2014a.

[13] Soares, A., Antunes, C.H., Oliveira, C., Gomes, A.: A multi-objective genetic approach to domestic load scheduling in an energy management system. Energy 77, 144–152, 2014b.

[14] Yang, P., Tang, G., Nehorai, A.: A game-theoretic approach for optimal time-of-use electricity pricing. IEEE Transactions on Power Systems 8(2), 884-892, 2012.

[15] Zhang, G., Zhang, G., Gao, Y., Lu, J.: Competitive Strategic Bidding Optimization in Electricity Markets Using Bilevel Programming and Swarm Technique. IEEE Transactions on Industrial Electronics 58, 2138-2146, 2011.

[16] Zugno, M., Morales, J. M., Pinson, P., Madsen, H.: A bilevel model for electricity retailers' participation in a demand response market environment. Energy Economics 36, 182-197, 2013.

[17] http://bilevel.org/ [Accessed March 2017.]

Appendix – Base load data

Table A.1 - Power requested from the grid by the base (non-controllable) load (cf. Fig. 2)

t	b(t) (W)	t	b(t) (W)
1-32	166	82-83	1528
33-34	700	84-85	742
35-36	170	86	249
37-44	92	87	452
45-54	156	88-90	280
55-64	133	91	1064
65-80	159	92-96	241
81	522		