Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

An explicit evolutionary approach for multiobjective energy consumption planning considering user preferences in smart homes

Sergio Nesmachnow[a*], Diego Gabriel Rossit[b], Jamal Toutouh[c] and Francisco Luna[d]

[a]*Universidad de la República, Uruguay*
[b]*Departamento de Ingeniería, INMABB, Universidad Nacional del Sur (UNS)-CONICET, Argentina*
[c]*Massachusetts Institute of Technology, United States*
[d]*Instituto de Tecnologías e Ingeniería del Software, Universidad de Málaga, Spain*

CHRONICLE

Article history:
Received April 1 2021
Received in Revised Format
May 12 2021
Accepted May 12 2021
Available online
May, 12 2021

Keywords:
Smart cities
Energy consumption planning problem
User preferences
Multiobjective optimization
Evolutionary algorithm
Greedy algorithms

ABSTRACT

Modern Smart Cities are highly dependent on an efficient energy service since electricity is used in an increasing number of urban activities. In this regard, Time-of-Use prices for electricity is a widely implemented policy that has been successful to balance electricity consumption along the day and, thus, diminish the stress and risk of shortcuts of electric grids in peak hours. Indeed, residential customers may now schedule the use of deferrable electrical appliances in their smart homes in off-peak hours to reduce the electricity bill. In this context, this work aims to develop an automatic planning tool that accounts for minimizing the electricity costs and enhancing user satisfaction, allowing them to make more efficient usage of the energy consumed. The household energy consumption planning problem is addressed with a multiobjective evolutionary algorithm, for which problem-specific operators are devised, and a set of state-of-the-art greedy algorithms aim to optimize different criteria. The proposed resolution algorithms are tested over a set of realistic instances built using real-world energy consumption data, Time-of-Use prices from an electricity company, and user preferences estimated from historical information and sensor data. The results show that the evolutionary algorithm is able to improve upon the greedy algorithms both in terms of the electricity costs and user satisfaction and largely outperforms to a large extent the current strategy without planning implemented by users.

1. Introduction

Power supply is of paramount importance in modern societies, which have become increasingly dependent on electricity for a growing number of commercial, industrial and residential activities. In this regard, it is not surprising that a big effort has been invested by electricity companies, regulation agencies, and governmental decision-makers to take advantage of computer-aid tools to enhance efficiency of energy usage (Calvillo et al., 2016). Among the diverse types of electricity users in modern societies, companies and industries have led the process of implementing sustainable and efficient energy management policies for several decades, since they had to adapt to an increasingly competitive world aiming at reducing budgetary expenses and responding to environmental regulations (Turner & Doty, 2007). However, households are also another relevant user since they are responsible for a large proportion of the consumed energy. For example, in 2018 the United States Energy Information Administration reported that the average household energy consumption was 12,906 KWh, with an annual growing rate of approximately 0.9% (U.S. Energy Information Administration EIA). A similar situation occurs in Europe. Despite being less than 10,000 KWh, the residential consumption of energy in the European Union is also significant since it represents 27.2% of the total energy consumed (Eurostat Statistic Explained). These values are in line with the average worldwide situation about the relevance of the residential users' consumption for the electric systems (Rabbani et al, 2018).

* Corresponding author
E-mail: sergion@fing.edu.uy (S. Nesmachnow)

2021 Growing Science Ltd.
doi: 10.5267/j.ijiec.2021.5.005

However, not only the total amount of consumed energy has an impact on the stress of the electric system but also the instant consumption in peak hours. Largely unbalanced consumptions along the day can lead to poor operative conditions of the electric grid, raising the risk of shortcuts (Yang et al, 2012). In this sense, the implementation of Time-of-Use (TOU) prices has been an asset for encouraging citizens to shift the usage of deferrable appliances, i.e., those devices which usage can be postponed with little impact on the Quality of Service (QoS) provided to the users such as dishwashers or washing machines, towards off-peak hours (Soares et al., 2013, 2014a). Thus, accurately planning the usage of deferrable appliances can enhance energy efficiency in the household, allowing users to take advantage of using periods of lower electricity prices. However, the continuous monitoring of household energy consumption and appliance usage is too demanding for residential users because of the diversity of decisions that are involved. Therefore, the deployment of automated support tools is essential to optimize the integrated management of energy resources (Bilil et al., 2016; Antunes et al., 2017).

In this context, the main goal of this article is to study the application of multiobjective algorithms for household energy consumption planning considering user preferences. The proposed algorithms aim to provide a recommendation system to help residential users to make efficient decisions concerning the use of deferrable appliances in a given planning period (e.g., daily, weekly, etc.). This system considers two crucial aspects: the economic cost of their final electricity bill (considering TOU prices) and the user satisfaction (in terms of QoS). Thus, the underlying optimization problem, i. e., the household energy consumption planning problem, consists of scheduling the use of different deferrable household appliances to minimize the electric bill, taking into account the TOU prices of electricity and the maximum contracted power, and maximizing the user satisfaction, which is a function of the user preferences. To solve this problem, a Multiobjective Evolutionary Algorithm (MOEA) and a family of greedy algorithms are devised. These algorithms are compared with the Business-as-Usual (BaU) strategy of residential users. The computational tests are performed on a set of instances that was built from real data. Therefore, the novel contributions of this paper are the following:

- The development of a MOEA with problem-specific operators and a family of greedy algorithms to solve the household energy consumption planning problem.
- The construction of a set of real-world instances of the household energy consumption planning problem based on historical information and sensor data.

The research has been performed in the context of project "Cloud Computing for Smart Energy Management (CC-SEM)" (Lujan et al., 2018; Lujan et al., 2020), which aims at designing a smart platform that contributes to the decision-making process of energy management in urban scenarios, and project "Computational intelligence to characterize the use of electric energy in residential customers" (Chavat et al, 2020a), which studies the application of computational intelligence techniques to characterize and plan household electricity consumption. Overall, these two projects are research efforts to provide a set of tools for controlling home devices, planning/simulating scenarios of energy generation, and to advance in the communication infrastructure for transmitting the generated data.

The article is organized as follows. The next section presents the addressed household energy consumption planning problem and revises the main related works. Section 3 describes the proposed resolution algorithms. Section 4 introduces the real-world instances that were constructed and presents the computational experimentation. Finally, Section 5 presents the conclusions and the main lines of future work.

2. Problem Description

This section presents the formulation of the household energy consumption planning problem and the main related work.

2.1. Problem formulation

The multiobjective version of the household energy planning problem addressed in this article considers the following elements:

- a set of users $U = (u_1, \dots, u_{|U|})$, each user represents a household;
- a set of time slots $T = (t_1, \dots, t_{|T|})$ in the planning period;
- a set of domestic appliances $L = (l_1, \dots, l_{|L|})$ for each user;
- a function $E: U \to N$, where $E(u)$ indicates the maximum electric power contracted by user u;
- a penalty term ρ applied to those users that surpass the maximum electric power contracted

- a function $D: L \times U \rightarrow N$, where $D(l, u)$ indicates the average time of utilization of appliance l for user u;
- a function $C: T \rightarrow N$, where $C(t)$ indicates the utilization cost (per kW) of the energy in time t;
- a function $P: L \rightarrow N$, where $P(l)$ indicates the power (in kWh) consumed by appliance l;
- and a function $UP: U \times L \times T \rightarrow N$, where $UP(u, l, t)$ indicates the preference of user u to use the appliance l at time t

Consider the binary variable x_{lt}^{u} that indicates if appliance l of user u is switched ON at time t; and function $y(x_{lt}^{u})$ that indicates the number of time slots in which appliance l of user u is switched ON continuously (without intermediate switch off) counting from time t. This is, $y(x_{lt}^{u}) = m - j$ with $m = max\, r / \forall h \in (t, r)\, x_{lt}^{u} = 1$.

The problem proposes finding a planning function $X = \{x\}$ for the use of each household appliance that simultaneously maximizes the user satisfaction or Quality of Service (QoS) defined in Eq. (1) and minimizes the price of the total energy consumed defined in Eq. 2.

$$f(X) = \sum_{u\in U}\sum_{t\in T}\sum_{l\in L} UP(u, l, t) \times \delta_{lt}^{u} \text{ with } \delta_{lt}^{u} = \begin{cases} 1\; if\; y(x_{lt}^{u}) \geq D(l, u) \\ 0\; otherwise \end{cases} \tag{1}$$

$$g(X) = \sum_{u\in U}\sum_{t\in T}\sum_{l\in L} x_{lt}^{u} \times C(t) \times P(l) + \rho \times \psi_{lt}^{u}$$
$$with\; \psi_{lt}^{u} = \begin{cases} P(l) - E(u)\; if\; P(l) - E(u) \geq 0 \\ 0\; otherwise \end{cases} \tag{2}$$

In Eq. (1), functions $UP(\cdot)$ and $D(\cdot)$ are defined considering the energy consumption pattern of each user. This process is explained in Section 4.1 with the construction of instances. Eq. (2) considers two costs for each combination of time slot, user and appliance, i.e., the cost of the energy consumed and the cost of penalties if the consumption exceeds the maximum power contracted. The penalization discourages the model from building planning in which consumption would exceed the power consumption limit for several time intervals in order to reduce the risk of generating a shortcut.

2.2. Related work

The household energy consumption planning problem has been addressed in several articles in the related literature. This section presents the most relevant works that apply metaheuristics to this problem. Soares et al. (2015) provides a comprehensive review of the application of metaheuristics to this problem. Barbaro & Capone (2014) and Makhadmeh et al. (2019) present more general reviews of the energy consumption planning in households.

Among the main studies in literature, Soares et al. (2013) proposed a single-objective genetic algorithm (GA) that minimizes the cost of the planning. The cost of planning includes the price of the energy and other three penalty cost terms. Two terms are associated with violations of the user preferences in which a specific appliance should be switched ON and how close the household's consumption to the maximum electric power contracted (as a proxy of the risk of shortcuts), respectively. The third cost term is related to energy availability to better deal with possible unexpected variations of the base (uncontrollable) load. Later, Soares et al. (2014a) proposed a multiobjective approach that minimizes the electricity cost and maximizes user satisfaction related to two aspects: the preferred time slots for appliance usage and the risk of shortcuts. However, no trade-off solutions that explore the compromising solutions among these objectives were computed. Another feature that was not addressed is building-like instances or using real sensor data to build the instances. Both previous articles used the categorization of household appliances proposed by Soares et al. (2014b), according to the usage and the management strategies that can be applied to each appliance.

Our previous work (Orsi & Nesmachnow, 2017) presented a hardware and software platform for smart monitoring and planning of energy loads in households, integrating a hardware controller to measure electricity consumption in real-time, a communication protocol for transmitting collected data, and a specific software module for optimizing the appliances usage and operation. The scheduling was performed by a greedy algorithm that takes into account user preferences and the maximum allowed power consumption. Muhsen et al. (2019) solved a similar problem with a multiobjective optimization differential evolution algorithm that aims to minimize the energy cost and the instant consumption in the peak hour and maximize users' comfort. After constructing a set of compromising Pareto optimal solutions, a compromising solution is selected with multi-criteria techniques based on the opinions of experts.

A recent article by Pamulapati et al. (2020) proposed three MOEAs to solve a household energy consumption planning problem to minimize electric cost and user dissatisfaction. In the proposed model, the user can customize the dissatisfaction function by setting the priority weights assigned to each appliance. For example, a large weight implies that the associated appliance is relevant for user comfortability and, thus, the algorithm will prioritize scheduling its inconvenient usage hours. Similarly, the usage of less important appliances (small weight) can be displaced to off-peak hours with a smaller energy cost.

Our previous work (Colacurcio et al., 2019) addressed the household energy consumption problem with a single-objective genetic algorithm that considers user satisfaction and energy cost in a weighted sum in the fitness function. This article extends that research line by considering an explicit multiobjective optimization problem addressed by a MOEA. It also expands the experimentation considering larger instances of the problem. The new instances include building-like scenarios, i.e., scenarios that consist of a unique building that has inside several housing units or users. In these instances, two different levels of penalization for excessive consumption are applied. One level of penalization is specific for each user, i.e., an extra cost is applied if the instant consumption is larger than the maximum allowable power stated in the contract. The other level of penalization applies to the whole building: an additional fee is charged to all the users if the sum of energy consumption (of all the users) is larger than a certain maximum value. These building-like instances are useful for representing real-world cases in which several nearby housing units share some electric infrastructure of a microgrid, such as apartments in modern smart buildings, and, thus, users have to coordinate to avoid the deficient operation of the microgrid.

Theanalysis of the related works concluded that there is room tocontribute with solutions focused on the development of recommendation systemsfor household electricity demand management, through the integration of IoTtechnologies and computational intelligence algorithms that explicitly considermultiobjective formulations of the problem.

3. The proposed algorithms for the household energy planning

This section describes the proposed algorithms to solve the household energy planning problem.

3.1. Greedy algorithms

Greedy algorithms are heuristics that aim to construct a good solution by taking locally optimal choices repeatedly (Cormen et al., 2009). By taking these decisions at each step, the algorithms seek to obtain a good-quality solution. Although greedy methods can compute the optimal solution in some well-known problems –e.g., Prim's algorithm for the minimal spanning tree problem or the Dijkstra's algorithm for finding the shortest path in a graph (Cormen et al., 2009)-, they do not lead to an optimal solution in the general case. However, a greedy algorithm may produce a sufficiently good solution with a relatively small computing effort since they follow a straightforward strategy.

In this article, three greedy algorithms were conceived and implemented. These algorithms differ in the optimization criteria they pursued while constructing a solution, either user satisfaction, energy cost or the total energy consumed. The three algorithms consider two main constraints: i) if a certain appliance is switched on, it must remain on for a period of time equal to or larger than the expected average time of utilization for that appliance (function $D(\cdot)$ defined in Section 4.1); ii) the instant consumption of energy must not exceed the maximum power contracted by the in any interval of time. The three greedy algorithms are described next.

Greedy-QoS algorithm. The procedure of this algorithm is presented in Algorithm 1. The algorithm aims at maximizing the user preferences or QoS according to Eq. (1). For each user and for each appliance, the algorithm seeks the time slot in which the preference of the user to switch ON the appliance is maximum (regarded that the maximum electrical energy contracted by the user is not exceeded) and, thus, set the appliance as switched ON starting from that time slot (up to the time slot in which expected duration is fulfilled). Within each user, appliances are processed in descending order of power consumption (parameter $P\,(l)$).

Algorithm 1. Greedy-QoS algorithm

```
procedure searchintervalmaxpref($u_i$, $t_m$, $l_k$,$\mathbf{X}$)
    pref ← 0
    duration ← 0
    for ($t_n$=$t_m$; $t_n$< $t_{|T|}$; $t_n$++) do
        if duration < $D(d,u)$ then
            if $\sum_{l\in L}(x_{lt}^{u} * P\,(l\,)) + P\,(l_k) < E(u_i)$ then
                pref += $UP(u_i, t_n, l_k)$
                duration += $(t_{n-1} - t_n)$
            else
```

```
                pref ← 0
                duration ← 0
            end if
        else
            return ($t_n$,pref)
        end if
    end for
return ($t_{|T|}$–D($l_k$, $u_i$), pref)
end procedure
X ← $\overleftarrow{0}$                                          ► Vector of variables $x_{lt}^u$
for ($u_i$=$u_0$; $u_i$≤$u_{|U|}$; $u_i$++) do                   ► For each user
    for ($l_k$=$l_0$; $l_k$≤$l_{|L|}$; $l_k$++) do               ► For each appliance
        bestPref ← 0
        pref ← 0
        $t_{\text{bestMin}}$ ← $t_0$
        for ($t_m$= $t_0$; $t_m$<$t_{|T|}$–D($l_k$, $u_i$); $t_m$++) do
            $t_m$, pref = searchintervalmaxpref($u_i$, $t_m$, $l_k$,**X**)
            if pref > bestPref then
                bestPref ← pref
                $t_{\text{bestMin}}$ ← $t_m$
            end if
        end for
        for ($t_m$= $t_{\text{bestMin}}$ − D($l_k$, $u_i$); $t_m$≤$t_{\text{bestMin}}$; $t_m$++) do
            $x_{lt}^u$ ← 1                                       ► Switch ON an appliance
        end for
    end for
end for
```

Greedy-cost algorithm. The algorithm is presented in Algorithm 2. The main goal of this algorithm is to build low-cost solutions (according to Eq. (2)). It also considers a threshold level of user satisfaction ($0 < \pi < 1$) that must be fulfilled. For this reason, the function *searchintervalmaxpref*(), defined for Algorithm 1, is used in Algorithm 2. Thus, the Greedy-cost algorithm seeks the solution that minimizes the cost given that the user satisfaction is not smaller than a percentage of the maximum user satisfaction for the same user and appliance. For each user and for each appliance, the algorithm seeks the time slot in which the cost of energy is minimal (regarded that the maximum electrical energy contracted by the user is not exceeded and a certain threshold level of user satisfaction is achieved) and, thus, set the appliance as switched ON starting from that time slot (up to the time slot in which expected duration is fulfilled). Within each user, appliances are processed in descending order of power consumption.

Algorithm 2. Greedy-cost algorithm

```
procedure searchintervalmincost($u_i$, $t_m$, $l_k$,X)
    cost ← 0
    duration ← 0
    pref ← 0
    for ($t_n$= $t_m$; $t_n$< $t_{|T|}$; $t_n$++) do
        if duration < D($l_k$, $u_i$) then
            if $\sum_{l \in L}(x_{lt}^u * P\,(l)) + P\,(l_k) < E(u_i)$ then
                cost += C(min)
                duration += ($t_{n-1} - t_n$)
            else
                cost ← 0
                duration ← 0
                pref ← 0
            end if
        else
            return ($t_n$,cost,pref)
        end if
    end for
return ($t_{|T|}$–D($l_k$, $u_i$), cost ← MaxInt, pref)
end procedure
```

```
X ← $\overleftarrow{0}$                                        ► Vector of variables $x_{lt}^{u}$
minSatUser ← π                                  ► User preferences level
for ($u_i$=$u_0$; $u_i$≤$u_{|U|}$; $u_i$++) do                    ► For each user
    for ($l_k$=$l_0$; $l_k$≤$l_{|L|}$; $l_k$++) do                 ► For each appliance
        bestPref ← 0
        pref ← 0
        for ($t_m$= $t_0$; $t_m$<$t_{|T|}$−D($l_k$, $u_i$); $t_m$++) do
            $t_m$, pref = searchintervalmaxpref($u_i$, $t_m$, $l_k$,X)
            if pref > bestPref then
                bestPref ← pref
            end if
        end for
        bestCost ← Max INT
        cost ← Max INT
        $t_{\text{bestMin}}$ ← $t_0$
        prefAux = 0
        for ($t_m$= $t_0$; $t_m$<$t_{|T|}$−D($l_k$, $u_i$); $t_m$++) do
            $t_m$,cost,prefAux = searchintervalmincost($u_i$, $t_m$, $l_k$,X)
            if cost < bestCost && pref * minSatUser > prefAux then
                bestCost ← cost
                $t_{\text{bestMin}}$ ← $t_m$
            end if
        end for
        for ($t_m$= $t_{\text{bestMin}}$ − D($l_k$, $u_i$); $t_m$≤$t_{\text{bestMin}}$; $t_m$++) do
            $x_{lt}^{u}$ ← 1                               ► Switch ON an appliance
        end for
    end for
end for
```

Greedy-energy algorithm. This algorithm seeks to minimize the energy consumed. The total energy consumed of given solution $X$is computed by applying Eq (3).

$$h(X) = \sum_{u\in U}\sum_{t\in T}\sum_{l\in L} x_{lt}^{u} \times P(l) \tag{3}$$

For each user and for each appliance, the algorithm seeks an interval to switch ON the appliance, so that the overall energy consumed in that interval is minimal, and the maximum electrical energy contracted by the user is not exceeded. Within each user, appliances are processed in descending order of power consumption. For the sake of simplicity, the pseudo-code of the Greedy-energy algorithm is not presented since it is similar to the Greedy-QoS algorithm (Algorithm 1) replacing variable *pref* with a variable representing the total energy consumed.

3.2. Proposed Multiobjective Evolutionary Algorithm

Evolutionary Algorithms (EA) are stochastic methods for optimization through a search that emulates the evolution of species in Nature. An EA is an iterative technique that starts the search with a set of potential solutions to the problem, generated by a random procedure or using a specific heuristic. Each solution of the pool (a.k.a., individual) has a fitness value that determines its suitability to solve the problem. In each iteration (generation), probabilistic operators are applied to selected individuals to improve their fitness. The basic evolutionary operators are the recombination of parts of two individuals and the random mutation of individuals. The new individuals built by the evolutionary operators replace part of the previous population, guiding the EA toward tentative solutions of higher quality. The stopping condition usually is based on a fixed number of generations or computing time, a certain quality level on the fitness of the best individual or detecting stagnation. EAs have been applied for solving complex real-world problems in multiple application areas (Nesmachnow, 2014).

This article devises a NSGA-II evolutionary algorithm for solving the household energy problem in an explicitly multiobjective fashion. NSGA-II is a state-of-the-art MOEA that is characterized by a set of particular features: an evolutionary search using a non-dominated elitist ordering that reduces the complexity of the dominance check during the optimization process, a crowding distance technique that aims at pre- serving diversity, and, finally, a fitness assignment technique that considers dominance ranks and crowding distance values in order to achieve a spread Pareto front of the problem (Deb, 2001). The generic pseudo-code of NSGA-II is presented in Algorithm 3.

Algorithm 3. Pseudo-code of the NSGA-II algorithm.

```
t←0                                              ► Generation counter
offspring ← ∅
initialize(P(0))                                 ► Initialize population
while not stopping criterion do
    evaluate(P(t))                                   ► Evaluate population
    R ← P(t) ∪ offspring
    fronts ← non-dominated sorting(R))           ► Dominance checking
    P(t+1) ← ∅; i ← 1
    while |P(t+1)| + |fronts(i)| ≤ N do
        crowding distance(fronts(i))
        P(t+1) ← P(t+1) ∪ fronts(i)
        i ← i+1
    sorting by distance (fronts(i))              ► Crowding distance for diversity
    P(t+1) ← P(t+1) ∪ fronts(i)[1:(N-|P(t+1)|)]
    selected ← selection(P(t+1))
    offspring ← evolutionary operators(selected) ► Evolutionary search
    t ← t+1
return computed Pareto front
```

The main features of the proposed MOEA for household appliances planning are described next.

Solution encoding. A problem-specific encoding is used to represent the solutions. The proposed encoding considers for each user a vector $X = (x_o, x_1, \ldots, x_{|T|})$, where $|T|$ is the total number of timesteps -in this case, minutes- in the planning period. Each element x_j in the encoding is a vector of binary values $x_j = (b_o, b_1, \ldots, b_{|L|})$, where $|L|$ is the number of appliances considered in the planning and each value b_i indicates if the appliance is ON timestep j. Fig. 1 presents an example of solution encoding considering five users and five appliances for user 2. In this example, at timestep i, user 2 has appliances #1, #2, and #5 switched ON and appliances #3 and #4 switched off.

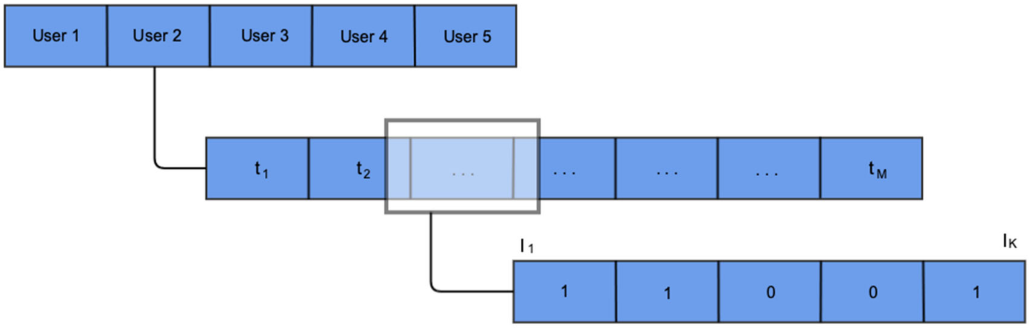


Fig. 1. Example of Solution encoding for NSGA-II.

Initialization. The initial population is built with a randomized procedure that assigns to each appliance a probability $\gamma = 0.6$ (value tuned in preliminary experiments) to be ON at each timestep. From preliminary tests, it was found that assigning a slightly larger probability to each appliance to be ON than to be OFF contributes to the diversification of the algorithm exploring a larger number of solutions.

Selection. The standard tournament selection is applied. After a preliminary configuration analysis, the size of the tournament was set to two individuals, and the individual with the best fitness was chosen.

Evolutionary operators. Ad-hoc evolutionary operators were conceived to provide efficacy and diversity to the search, working with the proposed solution encoding. The proposed evolutionary operators are:

- *Recombination*. An ad-hoc version of the Single Point Crossover operator was devised to recombine solutions. A cutting point is selected for each user, and a new planning is created for each user, using information from the first parent (before the cutting point) and the second parent (after the cutting point). Fig. 2 presents an example of the application of the proposed recombination operator between two solutions for a problem instance with three users and six appliances for each user.

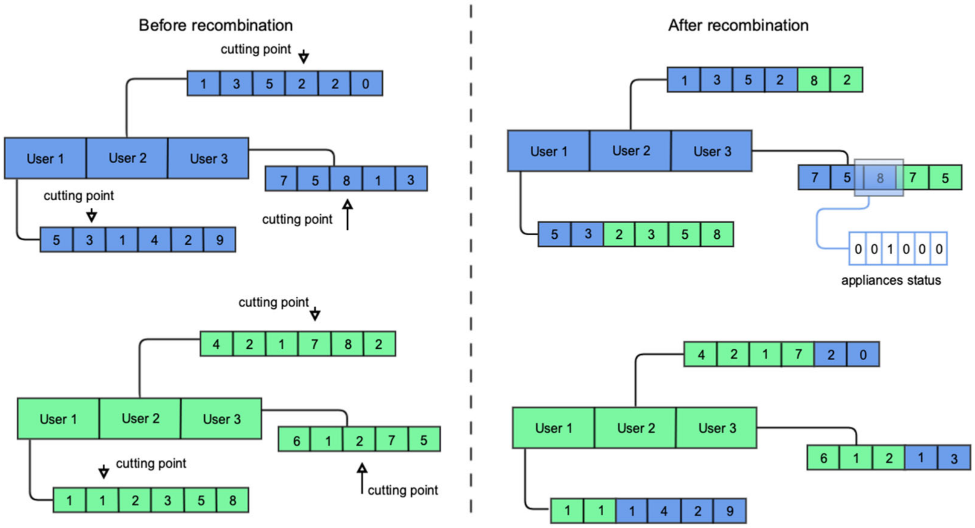


Fig. 2. Schema of the recombination operator

- *Mutation*. The mutation operator modifies the current state of an appliance. First, a specific time interval is randomly selected for every user, according to a uniform distribution. An appliance is then randomly selected (applying a uniform distribution) from all belonging to that user, and its state is changed (ON/OFF or vice-versa). Fig. 3 presents an example of the mutation operator.

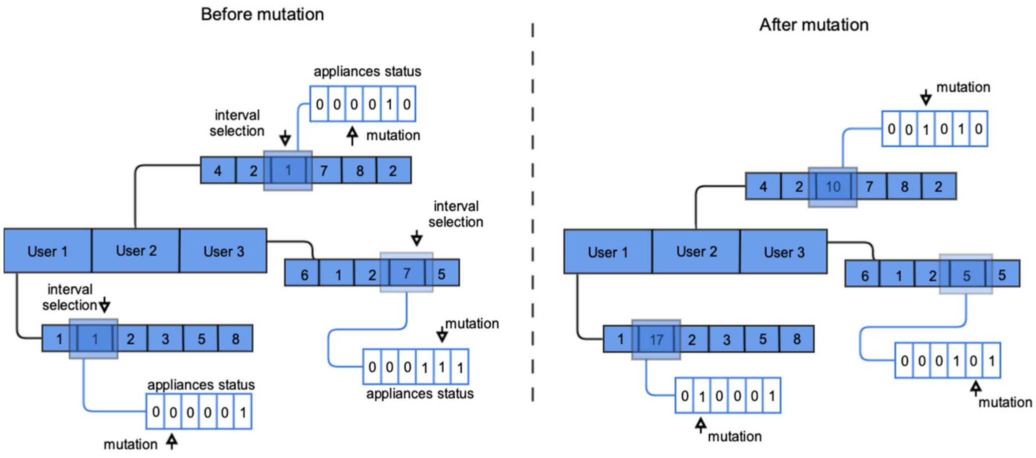


Fig. 3. Schema of the mutation operator

3.3. Business as Usual (BaU) strategy

The proposed algorithms are compared with a Business as Usual (BaU) strategy, which represents a baseline case. BaU proposes switching ON each appliance without planning, considering only user satisfaction. In general, these schedules have good user preference values but suboptimal cost values. A pseudo-code of this strategy is presented in Algorithm 3.

Algorithm 4. Business as Usual (BaU) strategy

```
$\mathbf{X} \leftarrow \bar{0}$                                   ► Vector of variables $x_{lt}^{u}$
for ($u_i = u_0$; $u_i \leq u_{|U|}$; $u_i$++)  do                ► For each user
```

```
    for ($l_k$=$l_0$; $l_k$≤$l_{|L|}$; $l_k$++) do                        ► For each appliance
        for ($t_m$= $t_0$; $t_m$<$t_{|T|}$−D($l_k$, $u_i$); $t_m$++) do
            if UP($u_i$, $t_m$, $l_k$) then
                $x_{lt}^u \leftarrow 1$                        ► Switch ON an appliance
            end if
        end for
    end for
end for
```

3.4. Development and execution platform

The algorithms were implemented using Java. Particularly, the evolutionary approach was developed in the ECJ library (Scott & Luke, 2019). ECJ is a Java-based evolutionary computation framework developed at George Mason University (cs.gmu.edu/~eclab/projects/ecj). ECJ includes easily modifiable classes for solving optimization problems. The experimental evaluation was performed in Dell PowerEdge servers with a Quad-core Xeon E5430 processor at 2.66GHz and 8 GB RAM from National Supercomputing Center, Uruguay (Cluster-UY) (Nesmachnow & Iturriaga, 2019).

4. Experimental analysis

This section presents the experimental analysis of the proposed algorithms for household energy consumption planning in order to evaluate the capability of developing an automated recommendation system for users. As aforementioned, this is a relevant problem for both energy companies and citizens under the novel smart city paradigm.

4.1. Problem instances

The experimental evaluation was performed on realistic problem instances built considering real data from the REDD dataset (Kolter and Johnson, 2011) and information from the National Electricity Company (UTE) of Uruguay (Chavat et al., 2019, 2020b).

A specific methodology was applied to determine a set of representative deferrable appliances with a high impact on household power consumption. First, household appliances were classified according to their operating profiles and purposes, following the categorization proposed by Soares et al. (2013). A set of deferrable appliances was selected and studied by computing the estimated average and maximum time of use for each of them, using the real data from the REDD dataset energy household consumption data. Appliances that contribute the most to the overall energy consumption were selected. Other relevant statistical values were computed as secondary parameters, including the number of times that the appliance is switched in a given period, the number of households where each appliance is present, the frequency of activation of each appliance, etc. These secondary parameters are used to improve the characterization.

As a result, a set of six deferrable appliances were selected for the study, namely dishwasher, microwave, dryer, air conditioning, oven, and refrigerator. The corresponding energy consumption data for these appliances were retrieved from the REDD dataset. Average values were considered to discretize the problem data in time intervals with a length of one minute, for which the users' preferences were computed. User preferences were defined according to the following procedure: records in the REDD dataset corresponding to each appliance and each one-minute time interval were counted for each day. The analysis allowed detecting significantly different utilization patterns of all appliances on weekdays and weekends. Thus, different problem instances were created to account for working days and weekends. The user preference to use the appliance in each interval was defined as directly proportional to the count for each day. In turn, the average length of the utilization period was also computed from the REDD dataset (i.e., the number of consecutive one-minute intervals in which the corresponding appliance was used). The considered problem instances also include real energy price data and real values for the maximum contractible electric power by a household, according to Uruguayan regulations. This information was provided by the National Electricity Company (UTE), Uruguay.

Regarding constant values included in the problem formulation, the penalty term ρ has two values that define two different penalty levels. If the energy consumed exceeds the maximum power contracted for less than 30%, a small penalty of $\rho = 0.3$ is applied whereas if the energy consumed surpasses this limit a larger penalty with $\rho = 1$ is used. This model is used in order to represent the requirements of the electric grid, which might be able to tolerate small over consumptions in individual users but large massive deviations should be discouraged for not stressing the grid (Yang et al, 2012). The planning period was set to one day for all instances, meaning that a one-day ahead planning is performed. The set of considered problem instances includes three different sizes for households (small, medium, and large houses), and a larger size to model a building with several households. Small instances include two users with two appliances each, medium size instances include four users with four appliances each, and large and building sized ones include six users with six appliances each.

Four pairs of instances –accounting for weekday and weekend profile– were built considering different sizes: small size (two users with two appliances each), medium size (four users with four appliances each), and large and building size (six users with six appliances each). The difference between the large and the building size is that the building size also has a maximum joint energy contracted, which means that the global consumption of all the users should not exceed a given threshold value. Instances considered in the experimental evaluation are described in Table 1.

Table 1
Details of the proposed problem instances.

#	Instances name	Users	Appliances in the in the instances	Consumption pattern
1	small.1 (s1)	2	(2,3)	weekday
2	small.2 (s2)	2	(2,3)	weekend
3	medium.1 (m1)	4	(4,4,3,2)	weekday
4	medium.2 (m2)	4	(4,4,3,2)	weekend
5	large.1 (l1)	6	(5,5,4,4,3,2)	weekday
6	large.2 (l2)	6	(5,5,4,4,3,2)	weekend
7	building.1 (l1)	6	(5,5,4,4,3,2)	weekday
8	building.2 (l2)	6	(5,5,4,4,3,2)	weekend

4.2. Parametric configuration analysis of the proposed MOEA

MOEAs are rather sensitive to misconfigured parameterizations, so a preliminary experimentation was undertaken to find a well-performing set of parameter values of the NSGA-II algorithm. Different values for the crossover probability (p_c), the mutation probability (p_m) and the number of generations (G) were considered. Specifically, the tested values were: $p_c = 0.1, 0.25$ and 0.5; $p_m = 0.01, 0.05$ and 0.1 and $G = 2500, 5000$ and 10000. The experiments considered three different problem instances from the one described in the previous section to avoid bias. Thirty independent runs were performed for each testing instance. Table 2 presents the results of the mean and standard deviation of the relative hypervolume (RHV) for each configuration and each test instance of the parameter calibration. The RHV is a well-known quality indicator in the context of multiobjective optimization. RHV is defined as the ratio between the volumes covered by the approximated Pareto fronts and the true Pareto front of the problem, and it measures both convergence and how well distributed are non-dominated solution along the approximated front. The ideal value for RHV is therefore 1. As the true Pareto fronts are unknown for the problem instances addressed, approximations are built by aggregating all non-dominated solutions found for each instance in all the executions (Coello et al., 2002).

Table 2
RHV of the test instances for different configurations of parameter calibration.

Name	Configuration (G, $p_{m,}$,p_c)	Instance 1	Instance 2	Instance 3
#0	(2500,0.01,0.1)	0.08 ± 0.02	0.01 ± 0.00	0.02 ± 0.02
#1	(5000,0.01,0.1)	0.16 ± 0.02	0.07 ± 0.02	0.03 ± 0.02
#2	(10000,0.01,0.1)	0.27 ± 0.01	0.22 ± 0.04	0.16 ± 0.03
#3	(2500,0.01,0.5)	0.08 ± 0.02	0.01 ± 0.00	0.01 ± 0.02
#4	(5000,0.01,0.5)	0.16 ± 0.0	0.08 ± 0.03	0.03 ± 0.02
#5	(10000,0.01,0.5)	0.27 ± 0.02	0.22 ± 0.03	0.15 ± 0.03
#6	(2500,0.01,0.25)	0.08 ± 0.02	0.01 ± 0.01	0.02 ± 0.02
#7	(5000,0.01,0.25)	0.16 ± 0.02	0.07 ± 0.02	0.02 ± 0.02
#8	(10000,0.01,0.25)	0.27 ± 0.01	0.21 ± 0.03	0.16 ± 0.03
#9	(2500,0.1,0.1)	0.47 ± 0.07	0.40 ± 0.02	0.33 ± 0.02
#10	(5000,0.1,0.1)	0.78 ± 0.06	0.66 ± 0.07	0.66 ± 0.06
#11	(10000,0.1,0.1)	0.92 ± 0.01	0.90 ± 0.01	0.92 ± 0.02
#12	(2500,0.1,0.5)	0.45 ± 0.06	0.39 ± 0.01	0.33 ± 0.02
#13	(5000,0.1,0.5)	0.77 ± 0.07	0.66 ± 0.07	0.63 ± 0.07
#14	(10000,0.1,0.5)	0.92 ± 0.01	0.91 ± 0.01	0.92 ± 0.02
#15	(2500,0.1,0.25)	0.45 ± 0.69	0.40 ± 0.02	0.33 ± 0.03
#16	(5000,0.1,0.25)	0.77 ± 0.07	0.66 ± 0.06	0.64 ± 0.08
#17	(10000,0.1,0.25)	0.92 ± 0.13	0.90 ± 0.01	0.91 ± 0.02
#18	(2500,0.05,0.1)	0.30 ± 0.02	0.24 ± 0.02	0.16 ± 0.03
#19	(5000,0.05,0.1)	0.46 ± 0.06	0.43 ± 0.03	0.38 ± 0.03
#20	(10000,0.05,0.1)	0.78 ± 0.06	0.70 ± 0.07	0.69 ± 0.07
#21	(2500,0.05,0.5)	0.30 ± 0.02	0.23 ± 0.02	0.15 ± 0.03
#22	(5000,0.05,0.5)	0.48 ± 0.08	0.43 ± 0.03	0.38 ± 0.04
#23	(10000,0.05,0.5)	0.78 ± 0.7	0.72 ± 0.06	0.68 ± 0.07
#24	(2500,0.05,0.25)	0.30 ± 0.02	0.24 ± 0.03	0.15 ± 0.02
#25	(5000,0.05,0.25)	0.46 ± 0.08	0.42 ± 0.02	0.38 ± 0.04

The final configuration was chosen with the following procedure: the Shapiro-Wilk normality test was applied to determine that the results do not follow a normal distribution, and, thus, the Friedman Rank Test was applied to establish that the differences among the parametric configurations were significant and chose the best configuration. The results of this test are reported in Fig. 4.

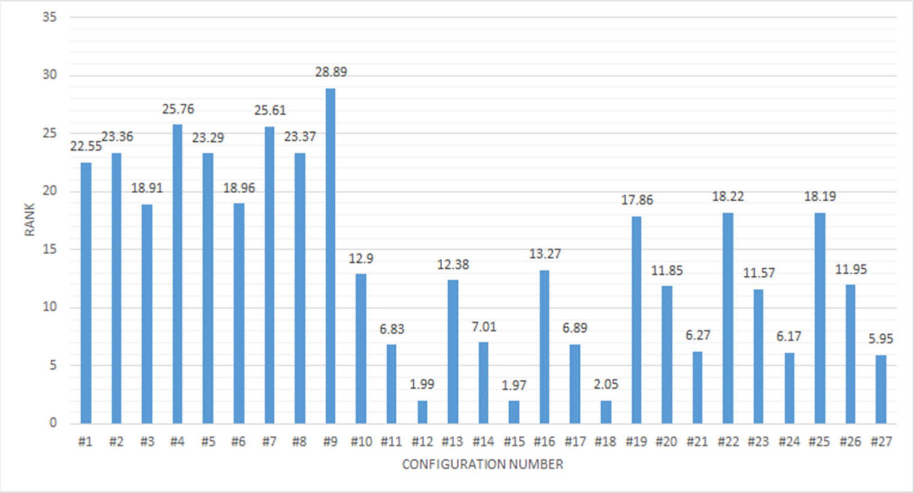


Fig. 4. Friedman Rank Test results for the parameter calibration

From the Friedman Rank test the chosen parameter configuration is #14 which has a crossover probability (p_c) equal to 0.5, a mutation probability (p_m) equal to 0.1 and a number of generations (G) equal to 10000. Additionally, in preliminary tests the population size was set to 150 individuals in preliminary studies.

4.3. Experimental results

This section presents the results obtained in the experimental evaluation. For NSGA-II, 50 independent executions of the algorithm were performed over each instance.

4.3.1. Computed Pareto fronts

Fig. 5, Fig. 6 and Fig. 7 display representatives approximated Pareto fronts of NSGA-II taken from these independent runs for the small, large, and building instances (respectively), for weekdays, and the solutions obtained with the greedy algorithms and BaU. As the Greedy-cost algorithm requires the definition of a user minimum satisfaction, π, several solutions are presented, varying π from 0.0 to 1.0 with a step of 0.1. The solutions of the greedy algorithms are reported as the baseline for the comparison.

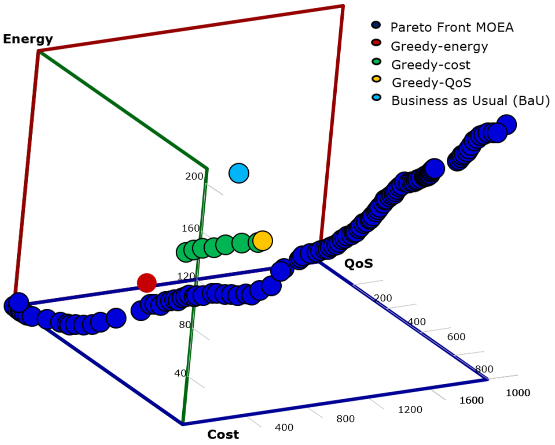


Fig. 5. Solutions of NSGA-II (approximated Pareto Front), greedy algorithms and BaU strategy for the s.1 instance

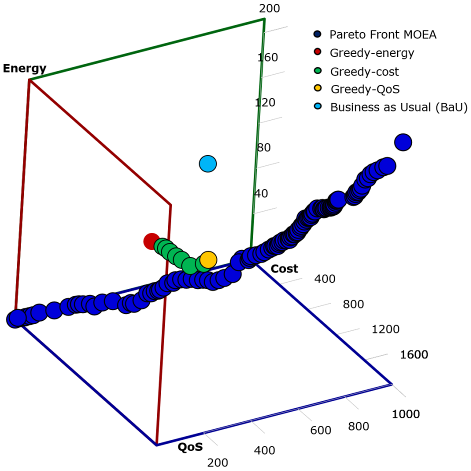


Fig. 6. Solutions of NSGA-II (approximated Pareto Front), greedy algorithms and BaU strategy for the l.1 instance

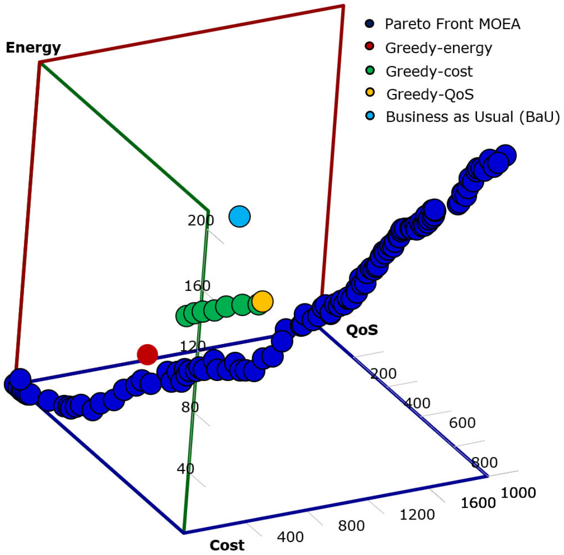


Fig. 7. Solutions of NSGA-II (approximated Pareto Front), greedy algorithms and BaU strategy for the b.1 instance

The analysis of Pareto fronts in Figs. (5-7) indicate that the NSGA-II is able to sample the search space presenting a large and spread set of solutions with different trade-offs among objectives. Conversely, the Greedy-cost –with the variations of π parameters- presents a much-limited sample of the search space. Additionally, greedy solutions are clearly suboptimal, being dominated by the NSGA-II in terms of each specific objective. The BaU solution is far from the Pareto front, remaining clearly suboptimal.

4.3.2. Improvements over baseline planning strategies

Comparison with greedy algorithms. Since NSGA-II provides the decision-maker with a set of non-dominated solutions, and the greedy algorithms obtain just a single one, the comparison between these two types of approaches have been undertaken by picking up the non-dominated solution of the approximated Pareto front reached by NSGA-II that has the smallest distance in terms of the QoS function $f(\cdot)$, as detailed next.

Consider the set of solutions P computed by NSGA-II and x^{QoS} the solution of the Greedy-QoS. Then $x' \in P$ is chosen with $x' = \underset{x\in P}{\operatorname{argmin}}|f(x) - f(x^{QoS})|$. Fig. 8 illustrates the relation between x' and x^{QoS}, being x' the solution in set P that has the smallest distance in the QoS objective. Recall that this criterion is the fairest possible as the approximated fronts may reach much better solutions on each of the objectives, but compromising the remaining ones.

Similarly, the comparison with the the solutions reached by Greedy-cost and Greedy-energy, x^c and x^e, respectively, the solutions $x'' = \underset{x\in P}{\operatorname{argmin}}|g(x) - g(x^c)|)$ and $x''' = \underset{x\in P}{argmin}|h(x) - h(x^e)|$ are chosen correspondingly. In the case of the Greedy-cost algorithm, the solution x^c chosen is that with the user satisfaction threshold level $\pi = 0.8$. Table 3 presents percentage of improvement of NSGA-II solution with respect that of the greedy algorithms.

Table 3
Comparison of NSGA-II with the greedy algorithms

	Greedy-QoS		**Greedy-cost ($\pi = 0.8$)**		**Greedy-energy**	
Instance	Cost	Energy	QoS	Energy	QoS	Cost
s1	76.79%	58.18%	38.72%	17.21%	52.16%	29.52%
s2	49.23%	63.74%	51.79%	27.78%	61.05%	26.98%
m1	46.21%	38.80%	43.34%	5.22%	47.27%	3.37%
m2	36.46%	33.40%	51.64%	3.79%	43.34%	1.17%
l1	49.16%	36.17%	25.36%	27.89%	42.03%	41.68%
l2	55.16%	35.57%	20.64%	20.05%	30.66%	24.12%
b1	46.13%	38.00%	43.37%	4.62%	49.98%	8.04%
b2	43.93%	35.19%	42.61%	0.21%	42.87%	3.66%

The first clear conclusion that can be drawn is that the NSGA-II algorithm is capable of improving the solution quality over the solutions calculated by the Greedy-QoS algorithm in all studied instances. In the best case, NSGA-II improves the cost by 76.79% (50.38% on average) and the energy consumed by 38.12% (42.38% on average), maintaining the quality of service.

A similar situation occurs when compared with the Greedy-Cost algorithm. The non-dominated solution chosen from the approximated Pareto front of NSGA-II improves up to 51.79% (39.68% on average) in QoS and up to 27.78% (13.34% on average) in energy consumed over the baseline solutions. When comparing to solutions computed by the Greedy-Energy algorithm, the proposed MOEA improves up to 26.98% in cost in the best case (17.31% on average) and up to 61.05% in the best case (46.17% on average) regarding QoS. These results clearly show that the NSGA-II algorithm is a highly promising approach to address this problem.

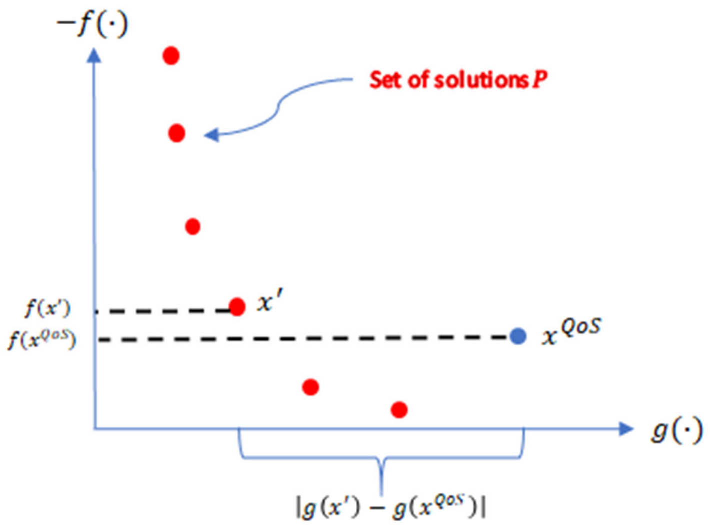
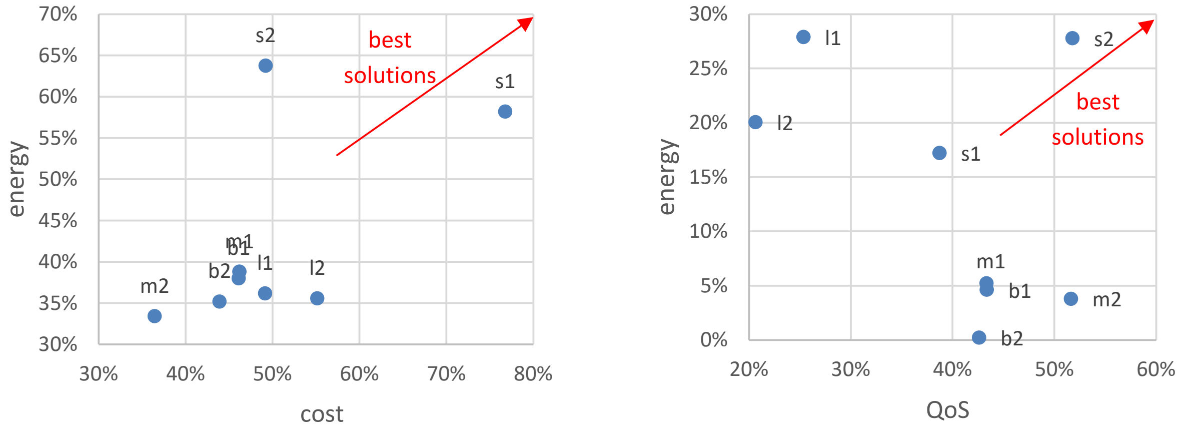


Fig. 8. Example of the method applied to select a representative solution computed by NSGA-II for comparison with solutions of greedy algorithms and BaU strategy

Fig. 9 graphically presents the improvements outlined in Table 3: percentage improvement of NSGA-II compromise solution over the Greedy-cost (a) and Greedy-QoS (b) solutions, respectively. In both comparisons, the largest improvements of NSGA-II over the greedy strategies are the small solutions (s1 and s2). In the case of the Greedy-cost, the solutions of medium, large and building sizes have a similar percentage of improvements for both objectives. In the case of Greedy-QoS, the large instances again have a homogeneous percentage of improvement. On the other hand, the medium and building instances obtain larger percentages of improvement of QoS than in energy consumed.



(a) Improvements over Greedy-cost

(b) Improvements over Greedy-QoS

Fig. 9. Improvements of the best trade-off solution computed by NSGA-II over Greedy-cost and Greedy-QoS

Comparison with Business-as-Usual. Regarding the comparison with the BaU method, two representative solutions are chosen from the Pareto front of NSGA-II: the most similar energy consumption value solution and the most similar cost solution. The results are included in Table 4.

Table 4
Comparison of NSGA-II with the BaU strategy

	Solution with equal energy consumed		Solution with equal cost	
Instance	**QoS**	**Cost**	**QoS**	**Energy**
s1	46.22%	8.46%	45.39%	1.32%
s2	270.21%	6.84%	268.76%	8.86%
m1	208.86%	4.70%	209.15%	1.60%
m2	213.17%	1.48%	215.85%	2.42%
l1	363.63%	3.73%	364.26%	4.41%
l2	206.97%	8.48%	203.18%	2.68%
b1	203.25%	2.88%	207.83%	2.66%
b2	255.82%	1.15%	255.82%	1.84%

When compared to a solution of equal consumed energy, in the best case the cost is improved in 8.48% (4.71% average) and the QoS in 363.63% (228.81% on average). Compared to a solution of the same energy cost, NSGA-II achieves solutions with up to 364.26% (221.18% average) in QoS and up to 4.41% (3.22% average) on consumed energy. Again, NSGA-II is clearly able to improve upon the BaU solution for all the instances addressed. This improvement is remarkable in the QoS objective, and it is of major relevance as BaU is designed to follow the users' preferences at every time step. This feature of BaU clearly limits its ability to reach high user satisfaction in the whole time frame. An additional result is that the costs of the solutions computed by NSGA-II and BaU (when compared to the solution with the same energy consumed as that of NSGA-II) are very tight. The same holds with respect to the energy consumed (and compared to the solution with the same cost reached by NSGA-II). Fig. 10 graphically presents the percentage improvement of NSGA-II compromising solution over the BaU strategy.

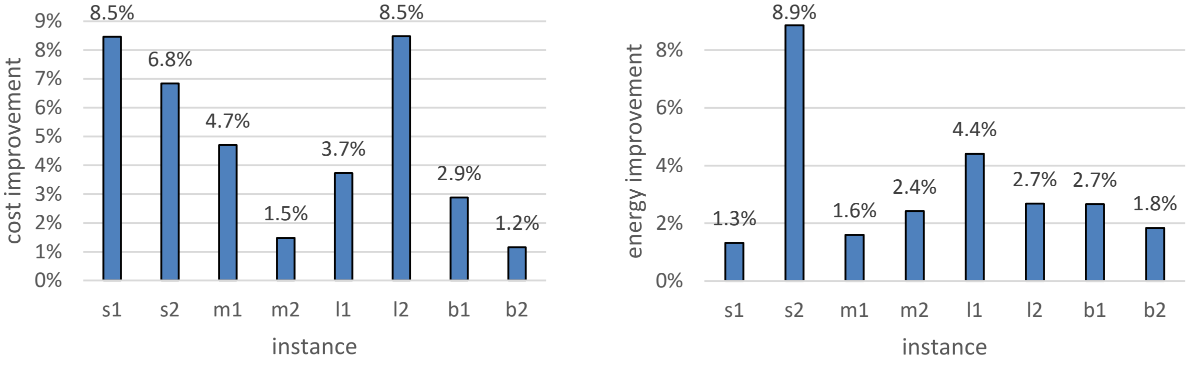


Fig. 10. Improvements of the best trade-off solution computed by NSGA-II over the BaU strategy

Two-dimensional projection examples. To better illustrate the relation between the solutions obtained by the different algorithms, Fig. 11 to Fig. 13 present 2D projections of the approximated Pareto front obtained by NSGA-II and the results computed by the greedy algorithms and BaU strategy for the medium size instance m.1.

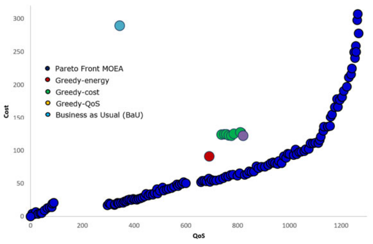


Fig. 11. 2D cut QoS-Cost of the Pareto front of instance m.1

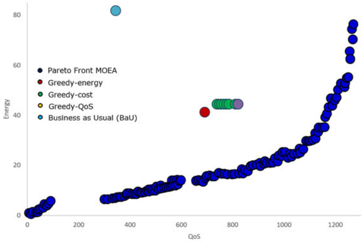


Fig. 9. 2D cut QoS-Energy consumed of the Pareto front of instance m.1

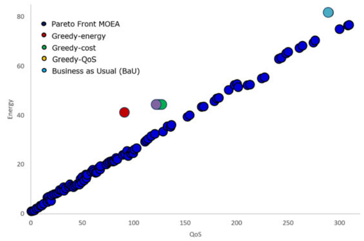


Fig. 10. 2D cut Cost-Enegy consumed of the Pareto front of instance m.1

As a relevant example, Table 5 reports the comparison of the best trade-off solutions computed by NSGA-II with the greedy algorithms and the BaU strategy, for m.1 instance. Th best trade-off is the solution x that minimizes $\sum_{k\in K}\left(\frac{v_k(x)-v_k^{best}}{v_k^{best}}\right)^2$, where $v_k()$ is the function that returns the value of criteria $k \in K$ and v_k^{best} is the best value that criteria in the Pareto front. Best trade-off solutions are highly regarded by decision-makers, as the offer the most balanced compromise between the problem objectives. Thus, they are frequently used to evaluate the benefits of explicit multiobjective approaches, especially when using evolutionary algorithms (Coello et al., 2002).

Table 5
Improvements achieved by the compromise solution of NSGA-II versus the greedy algorithms and BaU strategy for instance m.1

Solution	QoS	Cost	Consumed Energy
Greedy-cost ($\pi = 0.8$)	31%	20%	42%
Greedy-QoS	29%	21%	41%
Greedy-energy	54%	12%	36%
BaU	211%	65%	69%

Results in Table 5 indicate that NSGA-II is able to improve upon all the criteria with respect to all the other approaches used. The largest improvements are reached over the BaU strategy (up to 211% for the QoS). In comparison to the greedy algorithms: NSGA-II obtains non-dominated solutions with 20% lower cost than the Greedy-cost, 29% higher QoS than the Greedy-QoS, and 36% smaller consumed energy than the Greedy-energy. It has been therefore shown that the non-dominated solution with a trade-off between the problem objectives is also able to improve upon the other approaches used. From the point of view of multiobjective optimization, the aforementioned results are confirmed by computing relevant metrics, such as hypervolume, on the 2D reported projections. The analysis of the sample case (instance m.1) indicates that the hypervolume of the Pareto front computed by the proposed NSGA-II is 73.4% of the (hypothetical) total area, while the Greedy cost heuristic has a hypervolume of just 37,1% of the (hypothetical) total area. The reference nadir point for the hypervolume calculation is (0.0, 310.0), according to the corresponding objective function values. These results clearly demonstrate the superiority of the explicit multiobjective evolutionary approach by evaluating the dominated area of the computed Pareto front, which doubles the one covered by Greedy-Cost.

5. Conclusion

In order to achieve a sustainable usage of energy and reduce the consumption of peak hours, smart cities decision-makers have implemented several strategies to work on the demand side –residential consumers-. In this line of work, this article proposed an optimization algorithm capable of providing accurate plans for a recommendation system for users to improve electricity utilization. These algorithms aim to reduce the cost of the energy bill of households while also maximizing the satisfaction that users might get from this usage. Particularly, a MOEA based on NSGA-II was devised with specific evolutionary operators and solution encoding for the planning problem. Realistic problem instances were built using real-data from historical information and sensor data of representative households, which include both individual housing units and building-like households in which all the residents have to respect a maximum energy consumption limit for the whole building. The proposed MOEA was compared with a family of greedy algorithms –specially devised for this problem– and a business as usual strategy that implies the common practice without planning. The MOEA is able to outperform the rest of the strategies significantly, also obtaining representative approximated Pareto Fronts for the solved instances.

The main future line of research is considering uncertainty in user's preferences. This would lead to a further analysis of real-data in order to assess the probabilistic distribution of the behaviour of users. Additionally, robust optimization algorithms will have to be designed for dealing with this stochastic optimization problem. Another research line is to devise an online recommendation system for users. Given the proposed schedule for the day and sensor data of the actual usage of devices in real-time as inputs, this should be able to detect actual or potential deviations from the original planning and reoptimize on-the-fly the solution for the remaining part of the day for diminishing the risk of surpassing the maximum energy contracted and/or the energy cost.

Acknowledgments

Part of the research was developed by Giovanni Colacurcio in his graduate project ``Evolutionary algorithms for household energy planning'' (advisor: S.~Nesmachnow). Part of the research was supported by project ``Computational intelligence for characterization of electric energy consumption in residential households'' (CSIC UdelaR-UTE, Uruguay).

Regarding funds, S. Nesmachnow is partly supported by ANII and PEDECIBA, Uruguay. D. Rossit is supported by CONICET and Universidad Nacional del Sur. J. Toutouh is partly funded by the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement No 799078, the Junta de Andalucía UMA18-FEDERJA-003, and the University of Málaga, Andalucía Tech. F. Luna is partly supported by MICINN, under grants TIN2016-75097-P, RTI2018-102002-A-I00 and EQC2018-004571-P.

References

Antunes, C. H., Soares, A., & Gomes, Á. (2017). An Integrated Building Energy Management System. In *Mediterranean Green Buildings & Renewable Energy* (pp. 191-199). Springer, Cham.

Barbato, A., & Capone, A. (2014). Optimization models and methods for demand-side management of residential users: A survey. *Energies, 7*(9), 5787-5824..

Bilil, H., Aniba, G., & Gharavi, H. (2016). Dynamic appliances scheduling in collaborative microgrids system. *IEEE Transactions on Power Systems, 32*(3), 2276-2287.

Calvillo, C. F., Sánchez-Miralles, A., & Villar, J. (2016). Energy management and planning in smart cities. *Renewable and Sustainable Energy Reviews*, *55*, 273-287.

Chavat, J. P., Graneri, J., & Nesmachnow, S. (2019). *Household energy disaggregation based on pattern consumption similarities*. In Ibero-American Congress on Information Management and Big Data (pp. 54-69). Springer, Cham.

Chavat, J. P., Graneri, J., & Nesmachnow, S. (2020a). Nonintrusive energy disaggregation by detecting similarities in consumption patterns. *Revista Facultad de Ingeniería Universidad de Antioquía*, *98*, 27-46.

Chavat, J. P., Graneri, J., Alvez, G., & Nesmachnow, S. (2020b). ECD-UY: Detailed household electricity consumption dataset of Uruguay. *Scientific Data*, submitted manuscript.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2002). *Evolutionary algorithms for solving multi-objective problems*. Springer.

Colacurcio, G., Nesmachnow, S., Toutouh, J., Luna, F., & Rossit, D. (2019). *Multiobjective household energy planning using evolutionary algorithms*. In Ibero-American Congress on Information Management and Big Data (pp. 269-284). Springer, Cham.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. 3rd ed. MIT press.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. John Wiley & Sons.

Eurostat Statistic Explained. Energy consumption in households. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households (Accessed: 12th June 2020).

Kolter, J. Z., & Johnson, M. J. (2011, August). REDD: A public data set for energy disaggregation research. *In Workshop on data mining applications in sustainability (SIGKDD), San Diego*, CA (Vol. 25, No. Citeseer, pp. 59-62).

Luján, E., Otero, A., Valenzuela, S., Mocskos, E., Steffenel, L. A., & Nesmachnow, S. (2018). *Cloud computing for smart energy management (CC-SEM project)*. In: Ibero-American Congress on Information Management and Big Data (pp. 116-131). Springer, Cham.

Lujan, E., Otero, A., Valenzuela, S., Mocskos, E., Steffenel, L., & Nesmachnow, S. (2020). An integrated platform for smart energy management: the CC-SEM project. *Revista Facultad de Ingeniería Universidad de Antioquía, 97*, 41–55.

Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Abasi, A. K., & Alyasseri, Z. A. A. (2019). Optimization methods for power scheduling problems in smart home: Survey. *Renewable and Sustainable Energy Reviews*, *115*, 109362.

Muhsen, D. H., Haider, H. T., Al-Nidawi, Y. M., & Khatib, T. (2019). Domestic load management based on integration of MODE and AHP-TOPSIS decision making methods. *Sustainable Cities and Society*, *50*, 101651.

Nesmachnow, S. (2014). An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, *3*, 320–347.

Nesmachnow, S., Baña, S., & Massobrio, R. (2017). A distributed platform for big data analysis in smart cities: combining intelligent transportation systems and socioeconomic data for Montevideo, Uruguay. *EAI Endorsed Transactions on Smart Cities, 2*(5), 1–18.

Nesmachnow, S., & Iturriaga, S. (2019). *Cluster-UY: Collaborative scientific high performance computing in Uruguay*. In: International Conference on Supercomputing in Mexico (pp. 188-202). Series Communications in Computer and Information Science, Springer, Cham.

Orsi, E., & Nesmachnow, S. (2017). *Smart home energy planning using IoT and the cloud*. In 2017 IEEE URUCON (pp. 1-4). IEEE.

Pamulapati, T., Mallipeddi, R., & Lee, M. (2020). Multi-objective home appliance scheduling with implicit and interactive user satisfaction modelling. *Applied Energy*, *267*, 114690.

Rabbani, M., Mohammadi, S., & Mobini, M. (2018). Optimum design of a CCHP system based on Economical, energy and environmental considerations using GA and PSO. *International Journal of Industrial Engineering Computations*, *9*(1), 99-122.

Scott, E. O., & Luke, S. (2019). *ECJ at 20: toward a general metaheuristics toolkit*. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, 1391-1398.

Soares, A., Antunes, C., Oliveira, & C., Gomes, A. (2014a). A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy* , *77*, 144–152.

Soares, A., Gomes, A., & Antunes, C. (2014b). 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.

Soares, A., Gomes, Á., & Antunes, C. (2015). *Integrated management of energy resources in the residential sector using evolutionary computation: a Review*. In Soft Computing Applications for Renewable Energy and Energy Efficiency (pp. 320-347). IGI Global.

Soares, A., Gomes, Á., Antunes, C. H., & Cardoso, H. (2013, April). Domestic load scheduling using genetic algorithms. *In European Conference on the Applications of Evolutionary Computation* (pp. 142-151). Springer, Berlin, Heidelberg.

Turner, W., & Doty, S. (2007). Energy management handbook. The Fairmont Press.

U.S. Energy Information Administration (EIA), Energy use in homes. https://www.eia.gov/ (Accessed: 12th June 2020).

Yang, P., Tang, G., & Nehorai, A. (2012). A game-theoretic approach for optimal time-of-use electricity pricing. *IEEE Transactions on Power Systems*, *28*(2), 884-892.