On the optimal demand-side management in microgrids through polygonal composition

A.O. Topa[a], N.C. Cruz[b], J.D. Álvarez[a,*], J.L. Torres[c]

[a] *Department of Computer Engineering, Automation and Robotics, CIESOL—ceiA3, Ctra. Sacramento s/n, La Cañada de San Urbano, University of Almería, 04120 Almería, Spain; atg511@inlumine.ual.es and jhervas@ual.es*

[b] *Department of Computer Architecture and Technology, University of Granada. Journalist Daniel Saucedo Street, 18014 Granada, Spain; ncalvocruz@ugr.es*

[c] *Department of Engineering, CIESOL—ceiA3, Ctra. Sacramento s/n, La Cañada de San Urbano, University of Almería, 04120 Almería, Spain: jltmoreno@ual.es*

Abstract

This article presents a novel methodology for energy management in microgrids focused on the demand side. It is inspired by the Tangram puzzle. The energy demand and production profiles are represented by polygons and managed through computational geometry. Therefore, an optimization problem is defined to place n energy demand profiles (pieces) to cover the total energy production profile (target shape). The optimization problem is addressed with a genetic algorithm. It tries to calculate the optimal positions of the polygons of the demands covering the maximum energy production. Since the referred production comes from renewable energy sources in the microgrid, this method allows reducing both the consumption of fossil fuels and energy bills.

Keywords: demand-side management, microgrid, optimization.

1. Introduction

A large part of the energy consumption in smart bioclimatic buildings is carried out through microgrids (MGs) made up of control systems, advanced detection technologies, communication infrastructures, and smart meters [1, 2]. Over

*Corresponding author

Email address: jhervas@ual.es (J.D. Álvarez)

time, great challenges have been encountered related to environmental problems, security, and energy management of the public grid. Facing them requires an intelligent energy generation system to obtain an MG featuring maximum renewable generation, reliability, and intelligence, known as Smart Grid [3, 4, 5]. An MG offers a bidirectional energy flow and information between the energy provider and the customer. For this purpose, an energy management system (EMS) is necessary to guarantee the load demand and the commercialization of electric energy. EMSs are classified into supply-side management (SSM) and demand-side management (DSM). These strategies help minimize energy and operating costs and CO_2 emissions and maximize energy production while efficiently managing energy consumption [1, 6, 7].

Although SSM guarantees efficient energy supply, satisfying energy demand and reducing polluting emissions and costs, it is affected by market price volatility. Hence, DSM becomes more attractive and allows the active participation of users, who can take load demand management decisions affecting the energy usage patterns. The aim is to optimize energy consumption, which allows reducing the maximum load demand and maintains the stability of the MG [1, 7, 8, 9].

DSM strategies consist of: (i) energetically efficient controllable devices with different consumption patterns, (ii) control systems that allow load demand conformation, (iii) ON/OFF controllers or actuators to turn on and off the devices, and (iv) communication link for users and external agents [10]. The objective of DSM is to change energy demand based on energy production, which directly relates to users' consumption patterns. Several DSM strategies have been developed recently, most based on moving energy demands. These displacements consider aspects such as energy availability, on-peak to off-peak electricity tariff hours, and improving energy performance [11].

Recent studies have developed different optimization approaches that aim to approximate the energy consumption curve with the original consumption one. For example, Djeudjo et al. [12] use a multi-objective particle swarm optimization model for performing a techno-economic analysis to respond to energy demand in communities in the Sub-Saharan African region. The authors of

[13, 14, 15, 16] focus on demand areas, such as residential, commercial, and industrial ones, considering controllable loads. They use several optimization models to satisfy the demand efficiently in energy and economic terms. Additionally, the authors of [17, 18] proposed an innovate algorithm based on Grey Wolf Optimization. Its main goal is to reduce energy bills and the peak demand of residential, commercial, and industrial microgrids. Alternatively, [19, 20] use blockchain-connected smart controllers. They aim to improve DSM, energy efficiency of buildings, and comfort level while reducing CO_2 emissions.

Although recent methods, such as particle swarm and Grey Wolf optimization, have been used in DSM optimization, genetic algorithms (GAs) are arguably one the most used population-based optimizers [21, 22]. For instance, the authors of [23, 24, 25] propose strategies based on GAs achieving substantial savings, reducing the energy demand, and motivating users to shift their loads to off-peak hours. The complexity of the resulting problems and the lack of mathematically exploitable properties, such as linearity and convexity, explain the popularity of Evolutionary Algorithms (EA), including GAs [10, 26]. These optimizers are inspired by the Darwinian theory of evolution. They define a generic global search strategy in which every solution is treated as an individual subject to the biological processes of sexual reproduction, mutation, and selective pressure to survive. As individuals evolve, the corresponding solutions improve [24, 27, 28].

This article focuses on EMS by displacing energy demands over time. Energy production is supposed to include renewable sources, so it is fixed in time and shape. The aim is to minimize electricity costs, carbon emissions, and user intervention. The main contribution is conceptualizing energy management as a shape composition problem in which the energy production and demand profiles are handled as polygons. In this context, a genetic algorithm seeks the optimal position of the demand profiles to fit the production one, which results in a schedule for using the available devices. This planning allows maximizing the use of renewable energy instead of the public grid. Therefore, the contributions of this work are three: Firstly, it describes a methodology to handle the inherently intermittent availability of renewable energy resources. Secondly, it confirms the aptitude of GAs to let an EMS adapt its configuration to arbitrary production profiles despite using a new problem representation. Thirdly and last, the referred representation conceptually simplifies the underlying optimization problem of covering the energy production profile with the demands as shape composition. It allows users to assimilate and face a non-linear optimization problem in a simple way. Several case studies have been included to test the effectiveness of the proposal. Although the first examples are didactic, there are realistic cases that the methodology also successfully addresses. For this purpose, data from a bioclimatic building, the CIESOL research center of the University of Almería (Spain), have been used.

The rest of the article is structured as follows: Section 2 presents the proposed methodology for MG demand-side energy management. Then, Section 3 describes the experimentation and the results obtained. Finally, Section 4 shows the conclusions and some ideas for future work.

2. Methodology

As introduced, this work focuses on minimizing the cost of electricity and the associated environmental impact by managing the energy demand in time. This section explains the proposal, starting with modeling the energy demand management as a polygonal shape replication problem. After that, an approach to evaluate and compare different candidate solutions is exposed, which allows facing energy demand management as an optimization problem. Finally, the section ends with a description of the method used for solving the resulting optimization problem.

2.1. Problem representation

The main idea of this work is that, in practical terms, DSM resembles the ancient Chinese puzzle known as Tangram [29]. This logic game consists in composing desired shapes, such as a house, using only its predefined set of pieces. Figure 1 shows the different parts of a Tangram puzzle on the left, some simple target shapes in the center, and how to achieve them on the right.

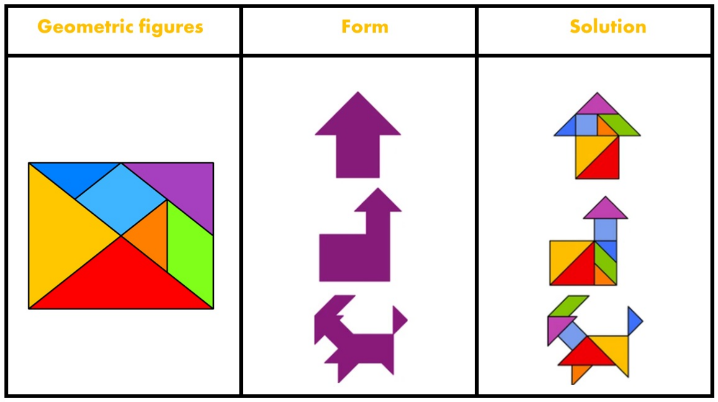

Figure 1: Representation of Tangram puzzle game.

For the problem at hand, the energy demand profile of each device can be represented by a small polygon in two dimensions, i.e., time and energy consumption. The same occurs with the production profile, which results in a larger polygonal shape. Both kinds of polygons are considered in a 2D coordinate system in which the vertical axis is power, in kW, and the horizontal one is the time of the day, in hours. Accordingly, the total energy expressed in kWh is the area of the resulting polygon. Figure 2 shows a simulation scenario determined through the polygons, both energy consumption and production. In this context, the methodology proposed tries to form the big polygon, i.e., the production, by combining the smaller ones, i.e., the consumption profiles. In contrast to the Tangram game, perfect replication might not always be possible in this case, but the conceptual similarity of the proposed approach is obvious.

Accordingly, the problem statement consists of n consumption profiles and the target energy production. They and the candidate solutions will be represented by polygons to be handled through computational geometry.

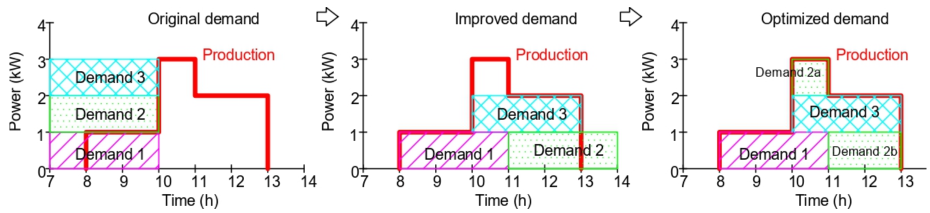

Figure 2: Sample problem context.

2.2. Problem formulation

Having expressed the problem at hand in terms of composing a target polygon by combining n different ones, addressing it as an optimization problem comes naturally. The fundamental aspect is to define how to encode and compare solutions, which allows us to decide if a given configuration is better than any other one.

As stated, n polygons represent the demand profile of n devices, and they can be seen as Tangram pieces to place to cover the energy production. Thus, any candidate solution consists of a vector that assigns a particular position to each demand profile. Defining those positions, which are the decision or design variables, is left to the selected optimization method. However, the strategy for decoding and assessing each possible solution is decoupled from it and explained next.

The space in which the polygons are considered has two dimensions, the total energy and the time of day. They are placed and shifted in both at optimization. Accordingly, each candidate solution has two components per demand profile, i.e., $2n$ variables. For evaluating any particular distribution or candidate solution, it is first necessary to put each demand profile where encoded. Then, the resulting shape must be compared to the production polygon.

Let D_i^y be the vertical dimension (total power, in kW) and D_i^x the horizontal one (time of day, in hours). The first stage, i.e., demand polygon placement, can be defined as in Eq. (1), where the achieved profile, D_T, results from reading the position of each demand polygon, D_i for $i = 1, \cdots, n$, and putting them appropriately in the energy and time axes. Since the polygons can only be translated, they are identified by a single reference point. By convention, the bottom-left point is considered. The position of the reference point of the i-th demand polygon D_i is labeled as $(D_i^x,\ D_i^y)$, where the first component refers to the first dimension, i.e., time, and the second is linked to the second one, i.e., power. The abstract function '*translate*' is responsible for placing the demand polygon that is part of the problem input in the position proposed for its reference point. The first argument is the polygon to place, and the second is the position of its reference point defined by its coordinates in both dimensions of interest. These positions will be ultimately adjusted through optimization. The displaced polygons form a total demand polygon DT using the logical union operation, represented by $\bigcup$.

$$D_T = \bigcup_{i=1}^{n} translate(D_i, (D_i^x, D_i^y)) \tag{1}$$

Regarding the second and last stage, i.e., polygon comparison, it follows Eq. (2). F_{DSM} is the area of the difference between the energy production polygon, P_E, and the one composed by the different demand profiles, D_T. Hence, it is a real number in the range $[0, \infty)$. The nearer it is to 0, the better the shape replication is, so this is the value to minimize for addressing the problem. Symbol $\oplus$ represents the exclusive OR (XOR) operation between the area of both polygons involved. Function '*area*' is an abstract function taking as input a polygon and computing its area.

$$F_{DSM}(D_T) = area(P_E \oplus D_T) \tag{2}$$

The optimization problem can be formulated according to Eq. (3). The aim is to find the position of each demand polygon so that the objective function, i.e., the difference between the target and the composed polygon, is minimized. The constraints require each demand polygon to stay in the region of interest. Namely, they limit the coordinates of reference points for the arbitrary bounds x_{min} and y_{max}, which refer to the dimension of power, and t_{min} and t_{max}, linked to that of time.

$$\begin{cases} \min\limits_{D_1^x, D_1^y, \cdots, D_n^x, D_n^y} & F_{DSM}(D_T) \\ \text{s.t.} & x_{min} \leq D_i^x \leq x_{max} \forall i \in \{1, \cdots, n\} \\ & y_{min} \leq D_i^y \leq y_{max} \forall i \in \{1, \cdots, n\} \end{cases} \tag{3}$$

The previous definitions are mainly conceptual. In practical terms, F_{DSM} is computed according to Algorithm (1). Notice that P_E results from combining all the renewable and non-renewable energy production profiles available. In this work, P_E might consist of photovoltaic energy production, wind energy production, battery supply, electric vehicle energy production, and public grid supply.

Algorithm 1 Objective function computation

Require: $\{D_1 \dots D_n\}, P_E, (D_1^x, D_1^y, \dots, D_n^x, D_n^y)$

$D_T = \emptyset$
for $i = 1 : n$ **do**
    $D_T = (D_T \bigcup translate(D_i, [D_i^x, D_i^y]))$ ▷ See Eq.(1)
end for
$[I_D, I_D^y] = PolygonIntersect(D_i, D_T)$
$D_T = D_T \bigcup translate(I_D, [0, I_D^y])$
return $area(P_E \oplus D_T)$ ▷ See Eq.(2)

The *PolygonIntersect* function referred to in Algorithm (1) is highly relevant for comparing solutions. Conceptually, it aims to identify and correct the overlappings of polygons considering their real meaning, i.e., energy consumption profiles. Its outputs, I_D and I_D^y, are the intersected polygon and the vertical displacement of this polygon, respectively. Computationally, it implements Algorithm (2). This process handles the overlapping of energy demand profiles. The reason is that they cannot absorb each other in the problem context, as standard boolean operations over polygons suggest. This situation arises while the optimization algorithm studies different placements of the energy profiles considered in time (hours) and magnitude (kW). Algorithm (2) identifies overlapping, and the intersected energy amount is displaced only in magnitude (kW). This approach does not alter the energy consumption timing, which avoids inconvenient pauses in the resulting schedule. Hence, notice that despite the plain geometric representation, the solution assessment logic must ultimately parse the different cases in terms of the underlying problem.

Algorithm 2 Function *PolygonIntersect*

Require: $\{D_1 \dots D_n\}, D_T$.

$Tolerance = 0.08, Displacement = 0.05, I_D^y = 0, A_{I_D} = \infty$
$Control = 0$
for $i = 1 : n - 1$ **do**
  for $j = 2 + Control : n$ **do**
    $I_D = D_i \bigcap D_j$ ▷ Compute intersection
    if $area(I_D) > Tolerance$ **then**
      while $A_{I_D} \geq Tolerance$ **do**
        $I_D^y = I_D^y + Displacement$
        $I_D = translate(I_D, [I_D^x, I_D^y])$ ▷ Move intersection up only
        $A_{I_D} = area(D_T \bigcap I_D)$
      end while
    end if
  end for
  $Control = Control + 1$
end for
return I_D, I_D^y

Additionally, it is relevant to mention that the theoretical conception of the objective function allows modifying its practical implementation, which could implicitly allow prioritizing demand profiles. In other words, provided an optimization method that focuses on comparing objective function values, its decisions are directly affected by the definition and behavior of the objective function. Similarly, notice that some devices might represent divisible demand profiles, such as a washing machine executing several processes. Its stages could be provided as input as different polygons, but the evaluation of solutions should promote (or require, if possible) that they appear in the appropriate order. Regardless, an exhaustive analysis of these extensions of the proposed formulation is out of the scope of the present paper.

2.3. Optimization Method

The objective function of the previous optimization problem does not feature a closed analytical expression with known mathematical properties to exploit, such as linearity and convexity [30]. In this situation, nature-inspired meta-heuristic optimization algorithms are valuable tools. They allow obtaining acceptable solutions despite the lack of certainty of optimality [30, 22]. Evolutionary algorithms stand out from them as highly-adaptable methods with outstanding exploration capabilities. They use a population of candidate solutions or individuals that interact with each other in a simulated context of biological evolution and randomness [21, 31, 32]. Genetic algorithms [33, 34, 35] are arguably their most visible exponent due to their high performance, simplicity, and adaptability.

In this work, the GA shipped with the Global Optimization Toolbox of MATLAB has been used with its default configuration [36]. However, the reader should notice that the present methodology is not linked to the optimization algorithm chosen for implementing the proposal. Instead, any other general-purpose optimization engine, like one of the plethora of evolutionary methods [21, 22], could be used within the same proposed polygonal context. The only requirement is that they focus on computing and comparing values of the objective function defined, which could encapsulate any comparison and prioritization criteria, as previously mentioned.

Centering our attention on the selected GA and according to its official documentation, the algorithm starts by initializing a population of candidate solutions. More specifically, it randomly creates a user-defined number of solution vectors within the bounds of the search space. They are evaluated according to the objective function, i.e., Eq. (3).

After initialization, the algorithm executes its main loop. The aim is to create new individuals and evolve them to produce better solutions after several iterations. The main loop consists of these genetic operators: Selection, Generation of offspring, and Replacement. It also has an elitist component to ensure that the best results continue in the active population [33]. The selection operator, which starts every evolutionary loop, chooses some individuals from the population to become the parents of a new generation of candidate solutions. As in nature, every individual can become a parent, but better solutions are more likely to be selected. In terms of implementation, according to the documentation, the algorithm implements a stochastic uniform selection procedure. It represents all the candidate solutions in a common segment. The section length of each one depends on its quality as a solution, so the better value, the longer portion. Then, the algorithm moves along the segment taking steps of equal size and selecting the individual linked to the portion reached every time. Although individuals can be selected more than once, this approach avoids limiting to the best individuals and enhances dispersion in the search space [22, 21]. The step size is randomly determined by the algorithm.

The generation of offspring creates the new individuals that will form the population of the next iteration. It consists of elite selection, crossover, and mutation. Elite selection directly takes the best individuals for the next population. This quantity is defined by a parameter whose default value is 5% of the total population size.

Aside from the previous individuals, the optimizer executes a crossover process to combine the contents of different progenitors and create potentially better candidate solutions as their descendants. More specifically, the algorithm makes pairs of progenitors and obtains a descendant from each. Every descendant is defined by randomly selecting the value of one of its parents for each component as a solution, i.e., the coordinates of the reference position of every demand polygon. The number of individuals to create in this way is set by a user-defined parameter that is a percentage of the population size without considering the elite size. By default, the percentage is 0.8.

Regarding mutation, it is launched when the combination of the individuals in the elite and the descendants results in fewer individuals than the current population size, which must be kept constant. In this situation, the algorithm changes the required number of parents by adding a perturbation vector to each one. Every component follows a Gaussian distribution with mean 0 and standard deviation scaled by considering the range of each variable.

It is relevant to highlight that the new solutions resulting from these steps must be evaluated. This consideration includes altered or mutated individuals, which become new solutions in practical terms. The exception is the set of individuals forming the elite. They do not vary, and it is unnecessary to re-evaluate them.

The replacement ends the main loop of the method by establishing the set of individuals coming from the generation of offspring, i.e., elite, crossover, and mutation, as the current population.

The GA iterates until one of the following stopping conditions is met. The first one is after executing the maximum number of iterations, which is set to 100 times the number of variables of the optimization problem by default. The second one is to complete a given number of iterations with the average relative change in the best fitness function value being less than or equal to a given threshold. It is also possible to define other conditions, such as a maximum time or a particular value for the best solution. The interested reader can access the official documentation for further information.

3. Experimentation and results

The proposed DSM strategy has been implemented in MATLAB using its built-in functions for polygon handling (*polyshape*) and the GA provided by its Global Optimization Toolbox [36]. It has been tested in seven different situations to test its effectiveness. The first four have been chosen because they are easy to understand and solve. Conversely, the fifth example shows a more realistic scenario, which includes multiple demands of different shapes and is harder to solve. Finally, the last two use real data from a bioclimatic building, the CIESOL research center of the University of Almería. The main aim is to take as much energy as possible from the production profile, i.e., to cover it with the demand profiles. The section ends with the computational cost of addressing each case.

The interested reader can find the source code used at the following link: `https://github.com/ual-arm/DSMoptimizer`.

3.1. Simulation setup

The proposed methodology expects as input the demand and production profiles presented as polygons, as described in Section 2.2. For simplicity and without loss of generality, the fundamental experimentation considers three energy demands of the same area to move in $[D_i^y, D_i^x]$. Figure 3 shows them. As can be seen, all the energy demand profiles have the same shape representing 1 kW during three hours, which results in a total energy consumption of 3 kWh. Accordingly, the GA will see optimization problems of six variables. It has been configured with the parameters shown in Table 1, which were tuned after preliminary experimentation.

GA parameters				
Population	Tolerance	Optimization parameters	Time max (s)	Generations
50	0.05	6	10800	36

Table 1: Parameters for GA simulation.

3.2. Simple case 1: Energy production is equal to energy demand without overlap

In the first case, Fig. 4 shows that the energy production starts with 2 kW during the first 3 hours and ends with 1 kW during the last 3 hours. The total energy production is 9 kWh, which is the sum of the three demands from Fig. 3.

As shown in Fig. 5, the developed DSM strategy can cover the whole energy production by moving the energy demands without overlapping. The optimiza-

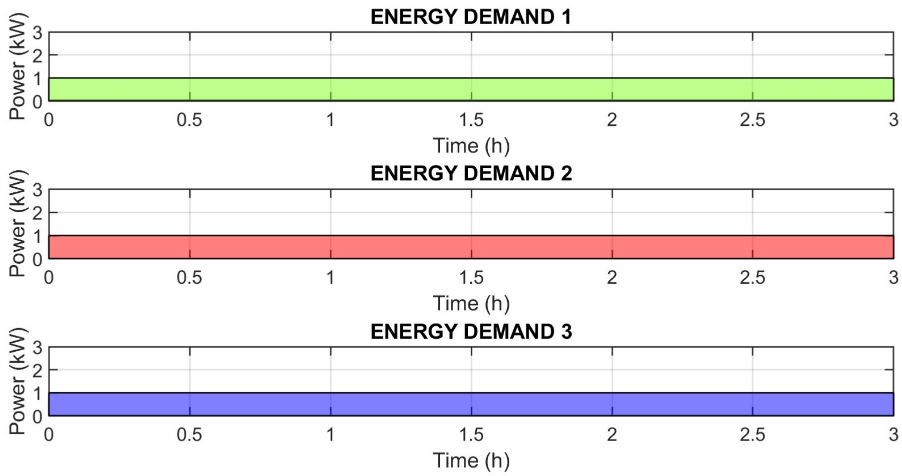

Figure 3: Representation of the polygons of energy demands.

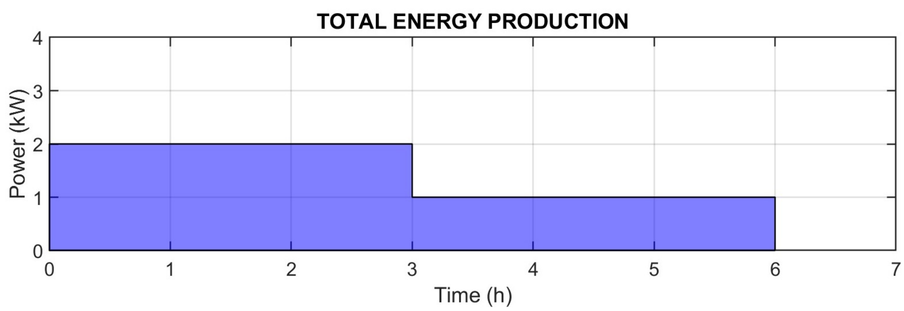

Figure 4: Polygon of total energy production.

tion algorithm optimally moved the energy demand. Hence, the total energy demand consumes all the production profile, which comes from renewable resources and avoids using the public grid. The lower graph shows a negligible error of the GA in the excess energy, but it is due to numerical precision.

3.3. Simple case 2: Energy production is equal to energy demand with overlap

In the second case, Fig. 6 shows that the energy production starts with 1 kW during the first 2 hours. Then, the energy production increases its power to 3 kW for 1 hour. Finally, it decreases to 2 kW during the last 2 hours. The total energy production is 9 kWh. It is the same as in the previous case, which

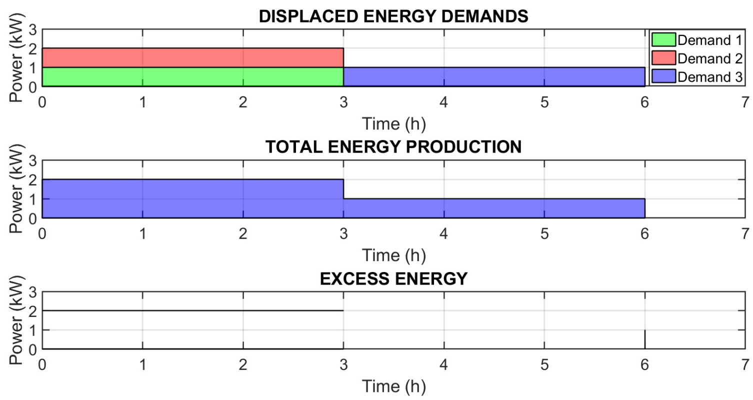

Figure 5: Results for the first case when energy production and demand are equal without overlap.

is equal to the sum of the three demands in Fig. 3. However, in this case, it is impossible to fit any of the demand profiles in the upper part of the production. Thus, some demand profiles must be split to cover the energy production profile.

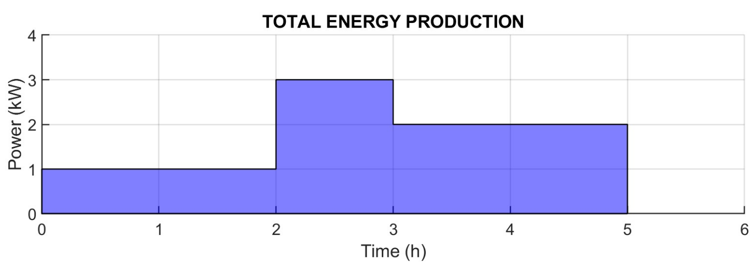

Figure 6: Representation of the polygon of total energy production.

Figure 7 shows the results obtained by the proposed DSM strategy when energy production equals energy demand with overlap. The optimization algorithm can move the energy demands optimally. The overlap between them is displaced by the GA so that the energy demand shapes cover the production profile. Thus, it takes all the available energy generated through renewable resources without consuming it from the public grid. As in the previous case, the lines shown in the lower graph are due to numerical precision errors.

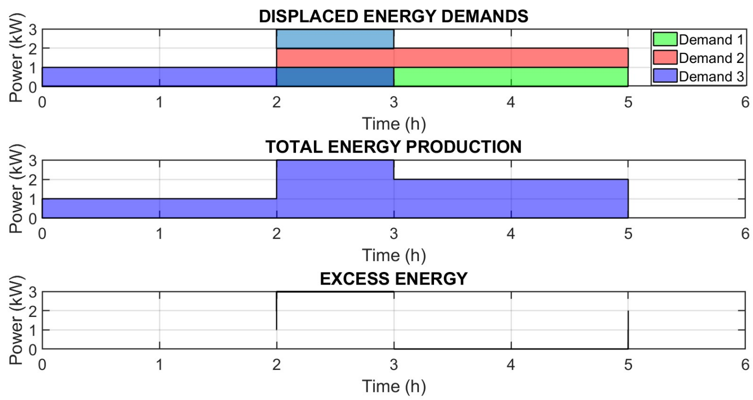

Figure 7: Results for the second case when energy production and demand are equal with overlap.

3.4. Simple case 3: Energy production below energy demand with overlap

For the third test scenario, the energy production starts with 1 kW during the first 2 hours and ends with 2 kW in the last 3 hours. The total energy production is 8 kWh, as shown in Fig. 8. It is less than the sum of the three demand profiles from Fig. 3.

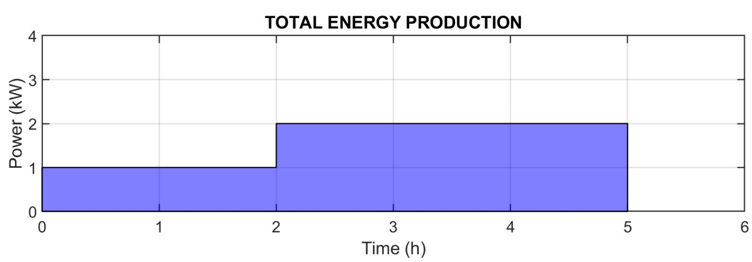

Figure 8: Representation of the polygon of total energy production.

Figure 9 shows the results of the proposed DSM strategy for the third case. The overlap between demands is displaced by the GA outside the production profile because, as previously pointed out, the total energy demand exceeds the energy production from renewable sources. Thus, a part of this demand must be covered using the public grid. However, the energy consumption from the public grid is minimal.

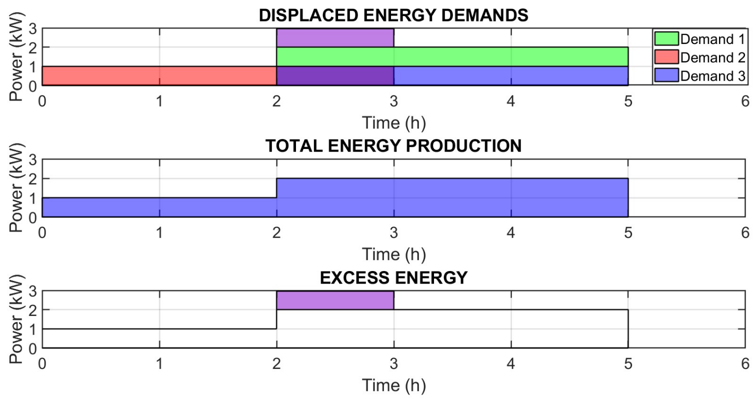

Figure 9: Results for the third case when energy production is below demand with overlap.

3.5. Simple case 4: Energy production higher than energy demand

In this case, the energy production is 3 kW during the 5 hours. As shown in Fig. 10, the total energy production is 15 kWh, higher than the sum of the three demands from Fig. 3.

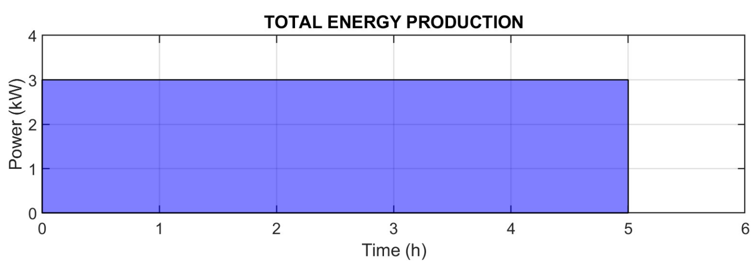

Figure 10: Representation of the polygon of total energy production.

Figure 11 shows the results of the proposed DSM strategy for the fourth case. As the energy production is greater than the sum of the energy demands, the placement of the latter is irrelevant as long as they stay in the production profile. For this reason, the total energy demand consumes part of the energy of the production profile without needing the public grid. The excess of energy is significant, and it could be either stored in a power supply system or sold to electric companies.

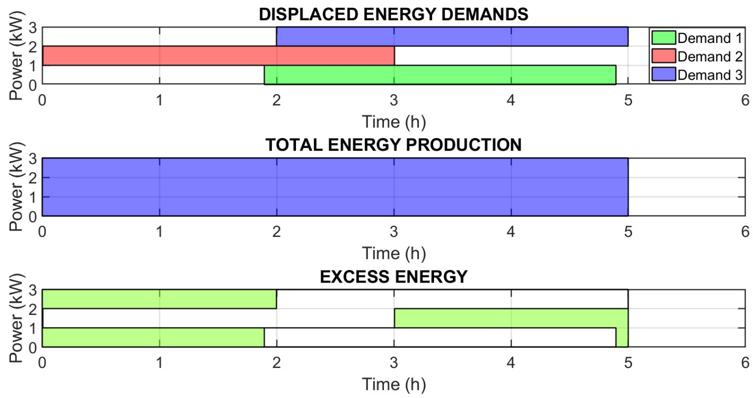

Figure 11: Results for the fourth case when energy production is higher than energy demand.

3.6. Complex case

The proposed methodology has also been tested in a more complex scenario to demonstrate its applicability to reality. Namely, the production profile used is more sophisticated, as shown in Fig. 12. It tries to reproduce the energy production from renewable sources where a constant wind energy source can produce 3 kW of power during the day. In the middle hours, this production is complemented by the power of a photovoltaic plant with a maximum production peak of 5 kW. It is worth mentioning that this energy production profile is the polygonal approximation of a real one. In general, any profile can be approximated by a polygon of N sides.

Aside from sophisticating the production profile, up to six demands will be used in this example, as shown in Fig. 13. Besides, in contrast to the previous cases, the demand polygons have different shapes, such as rectangular, triangular, and trapezoidal, and areas. Some of them can be only put in one place of the energy production profile, e.g., the triangular demands two and three, while others can be placed in several locations, such as the rectangular demands five

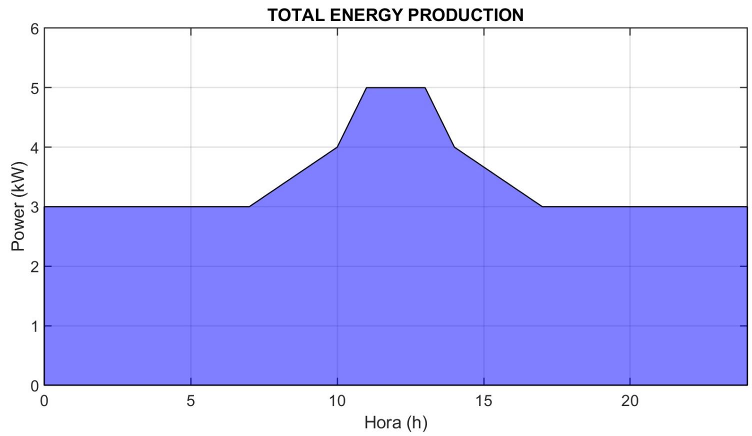

Figure 12: Energy production profile for the complex case.

and six.

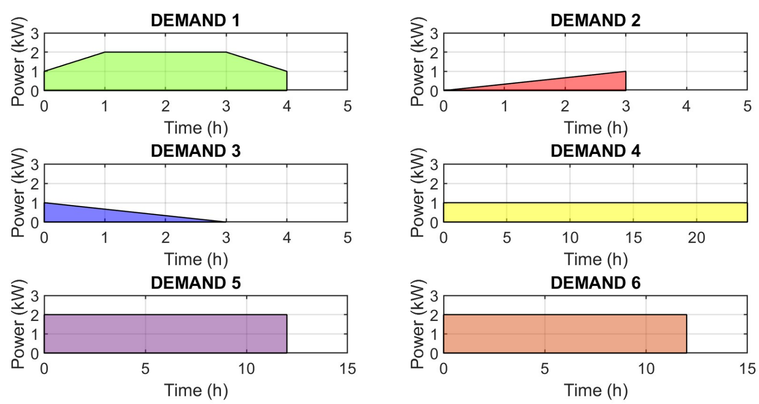

Figure 13: Energy demand profiles for the complex case.

Although it is difficult to appreciate it from Fig. 12 and Fig. 13, this case is similar to the first one, where the energy production equals the sum of energy demands without overlap. This fact can be seen in Fig. 14. As shown, the proposed methodology to manage energy demands in microgrids puts each one in its optimal place to cover all the energy production. Therefore, this example confirms that the presented methodology can be successfully applied even with complex production profiles and demands that differ in shape and size.

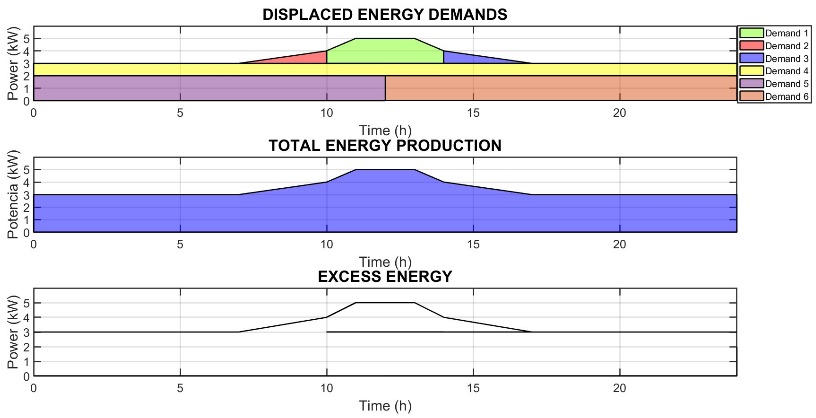

Figure 14: Results for the complex case.

3.7. Real cases

Aside from the previous theoretical examples, the proposed methodology has also been executed with real data to demonstrate its applicability.

To this aim, data from a bioclimatic building, the CIESOL research center, placed at the campus of the University of Almería, Spain, are used, see Fig. 15. It is a bioclimatic building with several energy systems for self-consumption, such as flat solar collectors for hot water and a photovoltaic plant for electricity generation. The CIESOL has a wide sensor network to monitor hundreds of variables, which includes power meters to measure the energy consumed or produced for their subsystems. Thus, the real energy production of the photovoltaic plant during a sunny day is presented together with the energy consumption of one lab of the building.

3.7.1. Real case 1

This case relies on data obtained from the photovoltaic plant of the CIESOL building. The upper graph of Fig. 16 shows its total energy production with a sampling time of 1 hour, dotted line. The observed shape corresponds to a

Figure 15: CIESOL research center.

typical sunny day in summer, when the photovoltaic plant can reach a maximum peak of 3 kW, approximately. Moreover, it shows the total energy demand too, green area. As it occurs with the energy production, it has been sampled in intervals of 1 hour. After that, it has been split into four different irregular polygons, as shown in the lower graph of Fig. 16. It represents the energy demand of different devices running at one of the laboratories of CIESOL. It is worth mentioning that these devices do not depend on each other. Thus, none of them must wait for any other to start or end.

As shown in Fig. 17, the proposed methodology successfully manages the energy demand of the laboratory. More specifically, it moves the demand profiles inside the 'bell' corresponding to the energy production of the photovoltaic plant. The profile of the total energy demand once the individual four demands are moved is drawn by a red dotted line. Thus, the use of renewable energy improves. Moreover, as energy production exceeds the sum of the four demands, other systems could benefit from the excess.

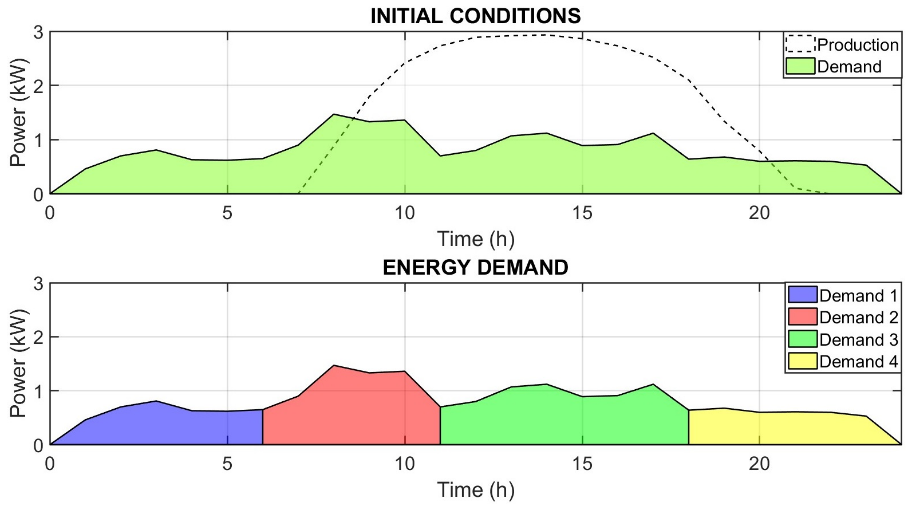

Figure 16: Total energy demand and production (upper graph), and its division into four irregular demand profiles (lower graph).

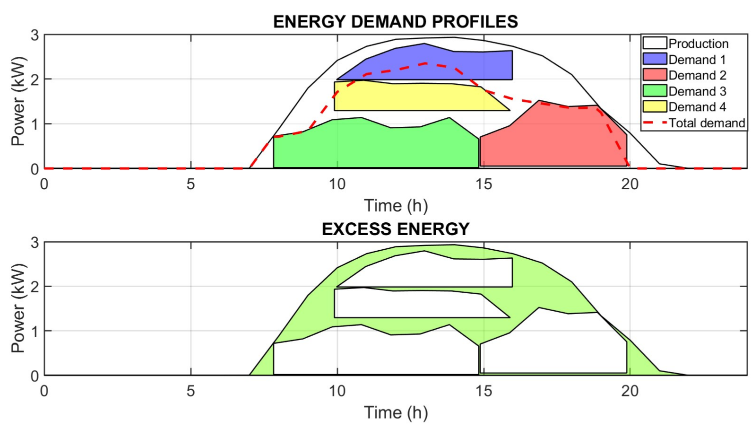

Figure 17: Results for the real case 1.

3.7.2. Real case 2

The second case with real data uses the production profile shown in the upper graph of Fig. 18. It corresponds to the production of the photovoltaic system on a typical sunny day in winter. As in the previous case, the energy production polygon has been built with radiation data sampled at intervals of 1 hour. In the same graph, the green polygon contains the total energy demand considered for this example. Again, sampling intervals are of 1 hour each. However, this time the total profile has been split into the three demands depicted in the lower graph of Fig. 18. Notice that one of them simulates a device that is always working. For instance, it could correspond to a lamp that is always on or a computer executing a program uninterruptedly.

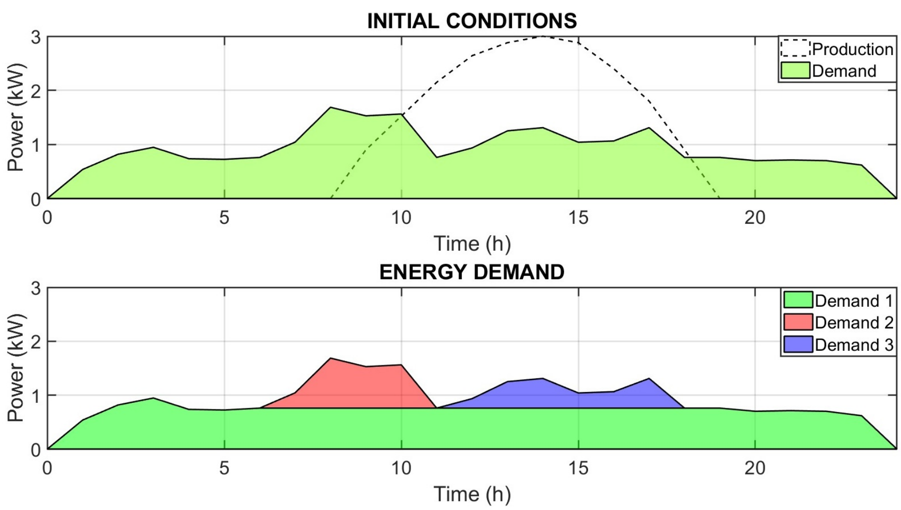

Figure 18: Representation of the total energy demand polygon.

The energy production is enough to supply all the demand, but one of the demand profiles cannot be split. For this reason, there will be a deficit of energy at the beginning and the end of the day. Although the algorithm can move the other demands into the production profile, it cannot do anything with the bigger one. As in the previous case, the total energy demand after moving the individual four ones is drawn by a red dotted line. However, the results obtained show an improvement over the initial conditions. Therefore, the proposed methodology demonstrates that it can be applied successfully to realistic situations involving complex and irregular production profiles.

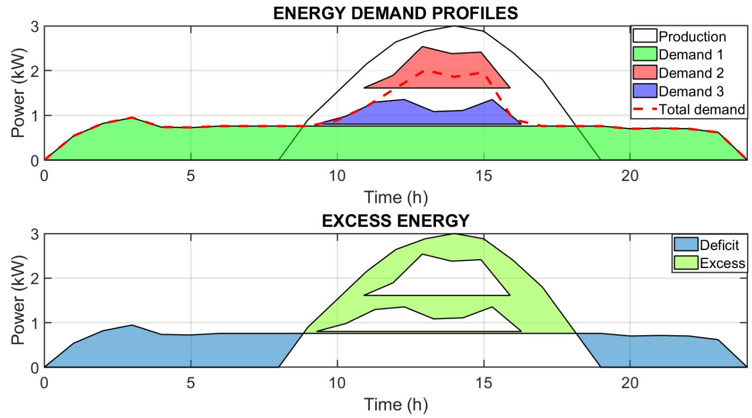

Figure 19: Results for the real case 2.

3.8. Computational cost

This section provides the reader with an overview of the execution time taken by our sample implementation for each case. Table 2 contains the times. They have been measured in a non-dedicated personal laptop featuring 11th Gen Intel(R) Core(TM) i5-11400H, 2.70GHz, RAM=16GB and MATLAB version: Matlab2022b. As can be seen, the execution time T_{exe} is directly proportional to the number of polygons and overlap cases. Nevertheless, the time records remain compatible with realistic use, especially considering the lack of real-time requirements. This planning should be executed offline and rely on predictions and recorded consumption patterns. Moreover, the MATLAB implementation used is a prototype that could be profiled to speed up its execution, if needed, or even ported to a non-interpreted language, such as C.

4. Conclusions

Successfully implementing microgrids in the current electricity market requires defining strategies with algorithms that optimize the available energy from renewable resources. Including these algorithms transforms microgrids into smart grids since they become able to manage their energy sources. The

Study case	**Demand polygons**	**T$_{exc}$(min)**
Simple case 1	3	5.6748
Simple case 2	3	5.6765
Simple case 3	3	6.2863
Simple case 4	3	1.6851
Complex case	6	12.4117
Real case 1	4	7.8370
Real case 2	3	3.9734

Table 2: Execution times of the proposed DSM methodology.

optimization algorithms can be on the production side, the demand one, or both.

The main objective of this work is to present a demand-side management methodology that optimizes the energy consumption profiles of a microgrid. The DSM strategy is based on the Tangram puzzle since demand profiles are represented as polygons to be combined to form the production profile for each target case. For this reason, this paper proposes an optimization problem focused on composing the production profile using the demand ones. The representation and operations with polygons have been implemented with the built-in *polyshape* functions of MATLAB. The optimization method used is the genetic algorithm included in the Global Optimization Toolbox of MATLAB. This optimization algorithm calculates the optimal positions of each energy demand to fill the production shape.

The results obtained in five different scenarios show that the proposed methodology can manage several energy demand profiles, with and without overlap, to fill an arbitrary production profile. Thus, as energy production comes from renewable sources, consumption from the public grid is minimized, as well as energy bills and polluting emissions. Although the first test scenarios include a few rectangular energy demands and a simple production profile for better understanding, the proposed methodology can deal with multiple energy demands and irregular shapes. These capabilities are demonstrated in the complex experiment, where only the computational cost of the search increases acceptably, and the quality of solutions remains high.

In future works, more sophisticated problem formulations will be studied. For instance, they could include shifting and fixed loads, energy prices, prioritization, and divisible demands, i.e., devices that can be paused. Besides, increasing the complexity of the problem might require considering different optimization algorithms.

Acknowledgements

This work has been financed by the COMMIT4.0EB (ref. PID2021-126889OB-I00) funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe". N.C. Cruz is supported by the Ministry of Economic Transformation, Industry, Knowledge and Universities from the Andalusian Government.

References

[1] E. Sarker, P. Halder, M. Seyedmahmoudian, E. Jamei, B. Horan, S. Mekhilef, A. Stojcevski, Progress on the demand side management in smart grid and optimization approaches, International Journal of Energy Research 45 (1) (2021) 36–64. doi:https://doi.org/10.1002/er.5631.

[2] D. Mariano-Hernández, L. Hernández-Callejo, A. Zorita-Lamadrid, O. Duque-Pérez, F. S. García, A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis, Journal of Building Engineering 33 (2021) 101692.
URL https://linkinghub.elsevier.com/retrieve/pii/S2352710220310627

[3] Z. Zhu, J. Tang, S. Lambotharan, W. H. Chin, Z. Fan, An integer linear programming based optimization for home demand-side management in smart grid, in: 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), 2012, pp. 1–5. doi:10.1109/ISGT.2012.6175785.

[4] I. Khan, Energy-saving behaviour as a demand-side management strategy in the developing world: the case of bangladesh, International Journal of Energy and Environmental Engineering 10 (4) (2019) 493–510. doi:https://doi.org/10.1007/s40095-019-0302-3.

[5] C.-S. Tai, J.-H. Hong, D.-Y. Hong, L.-C. Fu, A real-time demand-side management system considering user preference with adaptive deep q learning in home area network, Sustainable Energy, Grids and Networks 29 (2022) 100572. doi:https://doi.org/10.1016/j.segan.2021.100572.
URL https://www.sciencedirect.com/science/article/pii/S2352467721001387

[6] D. Li, W.-Y. Chiu, H. Sun, H. V. Poor, Multiobjective optimization for demand side management program in smart grid, IEEE Transactions on Industrial Informatics 14 (4) (2018) 1482–1490. doi:10.1109/TII.2017.2776104.

[7] S. Sharda, M. Singh, K. Sharma, Demand side management through load shifting in iot based hems: Overview, challenges and opportunities, Sustainable Cities and Society 65 (2021) 102517. doi:https://doi.org/10.1016/j.scs.2020.102517.
URL https://www.sciencedirect.com/science/article/pii/S2210670720307344

[8] H. J. Jabir, J. Teh, D. Ishak, H. Abunima, Impacts of demand-side management on electrical power systems: A review, Energies 11 (5). doi:10.3390/en11051050.
URL https://www.mdpi.com/1996-1073/11/5/1050

[9] K. Dassa, A. Recioui, Demand side management and dynamic economic dispatch using genetic algorithms, Engineering Proceedings 14 (1) (2022) 12. doi:10.3390/engproc2022014012.
URL https://www.mdpi.com/2673-4591/14/1/12

[10] A. Arteconi, N. Hewitt, F. Polonara, Domestic demand-side management (dsm): Role of heat pumps and thermal energy storage (tes) systems, Applied Thermal Engineering 51 (1) (2013) 155–165. doi:https://doi.org/10.1016/j.applthermaleng.2012.09.023.
URL https://www.sciencedirect.com/science/article/pii/S1359431112006357

[11] J. Lizana, D. Friedrich, R. Renaldi, R. Chacartegui, Energy flexible building through smart demand-side management and latent heat storage, Applied Energy 230 (2018) 471–485. doi:https://doi.org/10.1016/j.apenergy.2018.08.065.
URL https://www.sciencedirect.com/science/article/pii/S0306261918312170

[12] D. Temene Hermann, N. Donatien, T. Konchou Franck Armel, T. René, Techno-economic and environmental feasibility study with demand-side management of photovoltaic/wind/hydroelectricity/battery/diesel: A case study in sub-saharan africa, Energy Conversion and Management 258 (2022) 115494. doi:https://doi.org/10.1016/j.enconman.2022.115494.
URL https://www.sciencedirect.com/science/article/pii/S0196890422002904

[13] M. Zeeshan, M. Jamil, Adaptive moth flame optimization based load shifting technique for demand side management in smart grid, IETE Journal of Research 68 (1) (2022) 778–789. doi:10.1080/03772063.2021.1886607.
URL https://doi.org/10.1080/03772063.2021.1886607

[14] T. N. Qureshi, N. Javaid, A. Naz, W. Ahmad, M. Imran, Z. A. Khan,

A novel meta-heuristic hybrid enhanced differential harmony wind driven (edhwdo) optimization technique for demand side management in smart grid, in: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2018, pp. 454–461. doi:10.1109/WAINA.2018.00128.

[15] H. Wasim Khan, M. Usman, G. Hafeez, F. R. Albogamy, I. Khan, Z. Shafiq, M. Usman Ali Khan, H. I. Alkhammash, Intelligent optimization framework for efficient demand-side management in renewable energy integrated smart grid, IEEE Access 9 (2021) 124235–124252. doi:10.1109/ACCESS.2021.3109136.

[16] H. M. Hussain, N. Javaid, S. Iqbal, Q. U. Hasan, K. Aurangzeb, M. Alhussein, An efficient demand side management system with a new optimized home energy management controller in smart grid, Energies 11 (1). doi:10.3390/en11010190.
URL https://www.mdpi.com/1996-1073/11/1/190

[17] G. R. Hemanth, S. C. Raja, J. J. D. Nesamalar, J. S. Kumar, Cost effective energy consumption in a residential building by implementing demand side management in the presence of different classes of power loads, Advances in Building Energy Research 16 (2) (2022) 145–170. doi:10.1080/17512549.2020.1752799.
URL https://doi.org/10.1080/17512549.2020.1752799

[18] J. Ebrahimi, M. Abedini, A two-stage framework for demand-side management and energy savings of various buildings in multi smart grid using robust optimization algorithms, Journal of Building Engineering 53 (2022) 104486. doi:https://doi.org/10.1016/j.jobe.2022.104486.
URL https://www.sciencedirect.com/science/article/pii/S2352710222004995

[19] M. Afzal, Q. Huang, W. Amin, K. Umer, A. Raza, M. Naeem, Blockchain enabled distributed demand side management in community energy system

with smart homes, IEEE Access 8 (2020) 37428–37439. doi:10.1109/ACCESS.2020.2975233.

[20] L. Raju, V. Balaji, Internet of things and blockchain-based demand side management of smart microgrids, in: P. K. Mallick, A. K. Bhoi, G. Marques, V. Hugo C. de Albuquerque (Eds.), Cognitive Informatics and Soft Computing, Springer Singapore, Singapore, 2021, pp. 773–784. doi:https://doi.org/10.1007/978-981-16-1056-1_61.

[21] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Information Sciences 237 (2013) 82–117, prediction, Control and Diagnosis using Advanced Neural Computations. doi:https://doi.org/10.1016/j.ins.2013.02.041.
URL https://www.sciencedirect.com/science/article/pii/S0020025513001588

[22] G. Lindfield, J. Penny, Introduction to nature-inspired optimization, Academic Press, 2017.

[23] T. Logenthiran, D. Srinivasan, T. Z. Shun, Demand side management in smart grid using heuristic optimization, IEEE Transactions on Smart Grid 3 (3) (2012) 1244–1252. doi:10.1109/TSG.2012.2195686.

[24] A. Sardar, S. U. Khan, M. A. Hassan, I. M. Qureshi, A demand side management scheme for optimal power scheduling of industrial loads, Energy Systems (2022) 1–22doi:https://doi.org/10.1007/s12667-022-00510-x.

[25] C. Bharathi, D. Rekha, V. Vijayakumar, Genetic algorithm based demand side management for smart grid, Wireless personal communications 93 (2017) 481–502. doi:10.1007/s11277-017-3959-z.

[26] B. Lokeshgupta, S. Sivasubramani, Multi-objective dynamic economic and emission dispatch with demand side management, International Journal of Electrical Power & Energy Systems 97 (2018) 334–343.

doi:https://doi.org/10.1016/j.ijepes.2017.11.020.
URL https://www.sciencedirect.com/science/article/pii/S0142061517317350

[27] C. Roy, D. K. Das, A hybrid genetic algorithm (ga)–particle swarm optimization (pso) algorithm for demand side management in smart grid considering wind power for cost optimization, Sādhanā 46 (2) (2021) 1–12. doi:https://doi.org/10.1007/s12046-021-01626-z.

[28] M. A. Khan, N. Javaid, A. Mahmood, Z. A. Khan, N. Alrajeh, A generic demand-side management model for smart grid, International Journal of Energy Research 39 (7) (2015) 954–964. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/er.3304, doi:https://doi.org/10.1002/er.3304.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/er.3304

[29] F. T. Wang, C.-C. Hsiung, A theorem on the tangram, The American Mathematical Monthly 49 (9) (1942) 596–599. arXiv:https://doi.org/10.1080/00029890.1942.11991289, doi:10.1080/00029890.1942.11991289.
URL https://doi.org/10.1080/00029890.1942.11991289

[30] N. Cruz, J. L. Redondo, E. Ortigosa, P. Ortigosa, On the design of a new stochastic meta-heuristic for derivative-free optimization, in: International Conference on Computational Science and Its Applications, Springer, 2022, pp. 188–200.

[31] A. Rajabi, C. Witt, Self-adjusting evolutionary algorithms for multimodal optimization, Algorithmica 84 (2022) 1694–1723. doi:https://doi.org/10.1007/s00453-022-00933-z.

[32] X. Zhang, X. Fan, S. Yu, A. Shan, S. Fan, Y. Xiao, F. Dang, Intersection signal timing optimization: A multi-objective evolutionary algorithm, Sustainability 14 (3). doi:10.3390/su14031506.
URL https://www.mdpi.com/2071-1050/14/3/1506

[33] N. C. Cruz, S. Salhi, J. L. Redondo, J. D. Álvarez, M. Berenguel, P. M. Ortigosa, Design of a parallel genetic algorithm for continuous and pattern-free heliostat field optimization, The Journal of Supercomputing 75 (3) (2019) 1268–1283. doi:https://doi.org/10.1007/s11227-018-2404-8.

[34] X. Lü, Y. Wu, J. Lian, Y. Zhang, C. Chen, P. Wang, L. Meng, Energy management of hybrid electric vehicles: A review of energy optimization of fuel cell hybrid power system based on genetic algorithm, Energy Conversion and Management 205 (2020) 112474. doi:https://doi.org/10.1016/j.enconman.2020.112474. URL https://www.sciencedirect.com/science/article/pii/S0196890420300108

[35] A. Lambora, K. Gupta, K. Chopra, Genetic algorithm- a literature review, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 380–384. doi:10.1109/COMITCon.2019.8862255.

[36] Mathworks, Global Optimization Toolbox - MATLAB, https://es.mathworks.com/products/global-optimization.html, Accessed on: 21 July 2022 (2022).