

Exploring Multi-Objective Metaheuristics to Optimise Resource Allocation in Energy Communities

Daniel Vallejo España¹, Maria Ruxandra Cojocaru¹, David Criado Ramón¹, Luis Gonzaga Baca Ruiz², María del Carmen Pegalajar Jiménez^{1,*}

¹Department of Computer Science and Artificial Intelligence, University of Granada, Granada, 18014, Andalucía, Spain; dvalessp@ugr.es, ruxandracojocaru@ugr.es, dcariado@ugr.es, mcarmen@decsai.ugr.es

²Department of Software Engineering, University of Granada, Granada, 18014, Andalucía, Spain; bacaruiz@ugr.es

✉ Corresponding author: María del Carmen Pegalajar Jiménez (mcarmen@decsai.ugr.es)

Abstract. In the transition to a sustainable urban model, renewable energy communities where participants act as both producers and consumers have gained substantial importance. In this context, the main challenge lies in the optimal sizing of energy generation and storage units to achieve a balance between economic profitability and independence from the conventional grid. In this study, we define the resource management problem for the initial setup of an energy community, optimising the number of solar panels, wind turbines, and battery capacity per member. Besides, the objective functions considered include the levelized cost of energy (LCOE), self-sufficiency, and self-consumption, evaluated under four different interaction models, from full independence to shared battery use. While metaheuristics are commonly employed in energy management problems, the increasing variety of algorithms complicates the selection of an effective optimisation method. We implemented several multi-objective metaheuristics, including Particle Swarm Optimisation, Marine Predators Algorithm, Whale Optimisation Algorithm, and Equilibrium Optimiser. Additionally, we introduce the Political Optimizer, specifically adapted for multi-objective optimisation, which provides an additional tool for this kind of problems. Results from simulations demonstrate that the Multi-Objective Equilibrium Optimiser with Archive Evolution Path achieved the lowest LCOE, 0.0736 \$/kWh, and the highest self-sufficiency rate, 0.8873, with shared excess energy production. When implementing shared storage, the best configurations reached an LCOE of 0.0725 \$/kWh, a self-consumption rate of 1.0000 and a self-sufficiency rate of 0.9509. Finally, the largest emission reductions were also observed with shared storage, and reached up to 38.1%.

Keywords: multi-objective optimisation, metaheuristics, renewable energy community, optimal sizing

1 Introduction

The continuous growth of urban areas makes current energy practices increasingly unsustainable as they are still largely reliant on fossil fuel systems. About 81.5% of the energy consumed worldwide comes from fossil fuels [1], contributing to greenhouse gas emissions and climate change. This dependency has negative consequences, including environmental pollution that affects public health, and the rising frequency of extreme weather events that threaten urban infrastructure and quality of life [2]. Renewable energy models are being adopted to mitigate these issues, with global renewable energy production increasing by over 59% in the past decade [3]. However, renewable energy production often depends on uncontrollable factors, such as weather variability. As a consequence, designing and managing local renewable energy communities to ensure economic viability remains a significant challenge.

Renewable energy communities are groups of citizens, companies, or local entities collaborating to produce and share energy [4, 5]. They have been proposed as a solution towards carbon neutrality in urban environments [6, 7]. These communities operate as microgrids that combine distributed generation and storage resources, and they usually require an energy management system to optimise resource allocation and energy flow [8]. As a result, diverse optimisation techniques have been explored in this context, including linear programming [9], neural networks [10], fuzzy logic [11] and game theory [12]. Although studies such as [13-15] successfully address self-consumption and self-sufficiency optimisation, these models can be rigid or problem-specific, because they limit adaptability to changing production and consumption patterns. As illustrated in [16-18], metaheuristics offer flexibility and can better handle multi-objective optimisation. Nevertheless, the inherent difficulty remains, in other words, the problem's search space is large, objectives often are contradictory, and renewable production is highly variable. Metaheuristic methods have been applied successfully in various other energy-related optimisation problems, including the siting of solar panels and wind turbines to maximise generation output [19], as well as optimisation in electric vehicle systems and energy distribution networks [20-22]. Although these studies do not focus on energy communities directly, they demonstrate the versatility of metaheuristics in solving complex optimisation tasks regarding energy distribution.

In the metaheuristics field, there are many recent algorithms that can potentially solve complex problems such as the design of energy systems. For example, in [23] the African Vulture Optimization Algorithm (AVOA) is specifically used to design a Hybrid Solar PV/Wind/Hydrogen/Lithium Battery energy system, making it suitable for optimizing energy configurations in community settings. Moreover, an improved hybrid variant, the hybrid aquila optimizer and African vultures optimisation algorithm [24], has been proposed and tested on 23 classical benchmark functions, the IEEE CEC2019 test suite and 5 engineering problems, obtaining effective balancing of exploration and exploitation and opening up new opportunities in the field. Other relevant novelties include the artificial lemming algorithm [25] which demonstrated superior results in IEEE CEC2017 and CEC2022 benchmark tests and outperformed 17 other

metaheuristics, and the multi-strategy boosted snow ablation optimizer [26] tested on six realistic constrained engineering design problems, which proposes an enhanced search mechanism that integrates strategies like initialization of good point sets, greedy selection, differential evolution, and reverse lens learning to improve optimization efficiency.

According to the No Free Lunch theorem, no single optimisation algorithm can consistently outperform others across all problems. As [27] suggests, the future of metaheuristics lies in hybridisation and in applying them to real-world problems. In this context, we propose applying several state-of-the-art multi-objective metaheuristics to the optimal allocation of renewable production and storage resources in local energy communities. To do so, we implemented Particle Swarm Optimisation (PSO) [28], Marine Predators Algorithm (MPA) [29], Whale Optimisation Algorithm (WOA) [30] and Equilibrium Optimiser (EO) [31]. In addition, we introduce and evaluate a novel multi-objective adaptation of the Political Optimiser (PO) [32]. Our goal was to exploit the algorithm's capacity for dynamic adjustments and adaptability to more effectively address the highly variable nature of renewable energy production and demand.

To assess and validate our approach, we designed and simulated a residential renewable energy community using data from the Typical Meteorological Year 3 (TMY3) dataset [33]. The optimisation problem considers three objectives reflecting economic and operational goals: the Levelized Cost of Energy (LCOE), the Self-Consumption Rate (SCR), and the Self-Sufficiency Rate (SSR). Furthermore, four different energy management scenarios proposed by [34] were analysed in order to capture different energy flow priorities among community members. The metaheuristics were evaluated using metrics such as spacing, hypervolume and the epsilon indicator.

Our experimental results confirm that applying multi-objective metaheuristics leads to viable and efficient community designs. Furthermore, our adaptation of the PO also showed competitive performance compared to existing algorithms, which validates its potential as an effective tool for real-world problems such as the one at hand. Thus, the main contributions of this work are:

- Formulation of the optimal sizing of photovoltaic, wind, and storage units in an urban renewable energy community as a multi-objective optimisation problem.
- Evaluation of three objectives (Levelized Cost of Energy, Self-Consumption Rate, and Self-Sufficiency Rate) under four energy management scenarios.
- Implementation and comparison of five state-of-the-art multi-objective metaheuristics, including a novel adaptation of the Political Optimiser for this context.
- Thorough evaluation of algorithm performance using established quality indicators and analysis of representative energy allocation solutions.

The following sections of this manuscript are organised as follows. In Section 2, we provide a description of the proposed methodology. It discusses the dataset employed, the methods implemented, and the metrics utilised in the research. Subsequently, Section 3 details the experiments carried out along with the corresponding parameters. The

experimental results are presented in Section 4. Finally, we conclude this research with Section 5 by defining the conclusions and future work.

2 Methodology

This section presents the methodology used in this study, as depicted in Figure 1. It consists of several key components. First, we detail the datasets and sources, including the TMY3 dataset for meteorological data and load profiles from the Open Energy Data Initiative. Next, we describe the community simulation model, which includes PV and wind generation models, a battery storage simulation with a charge/discharge strategy, and four energy management scenarios. We then formulate the problem by defining decision variables such as the number of PV panels, wind turbines, and battery capacity, along with objectives related to the LCOE, SCR and SSR, taking into consideration pertinent constraints. The next step in our methodology is the metaheuristic optimisation phase, and finally, we assess the quality of the optimisation results using appropriate metrics for this problem.

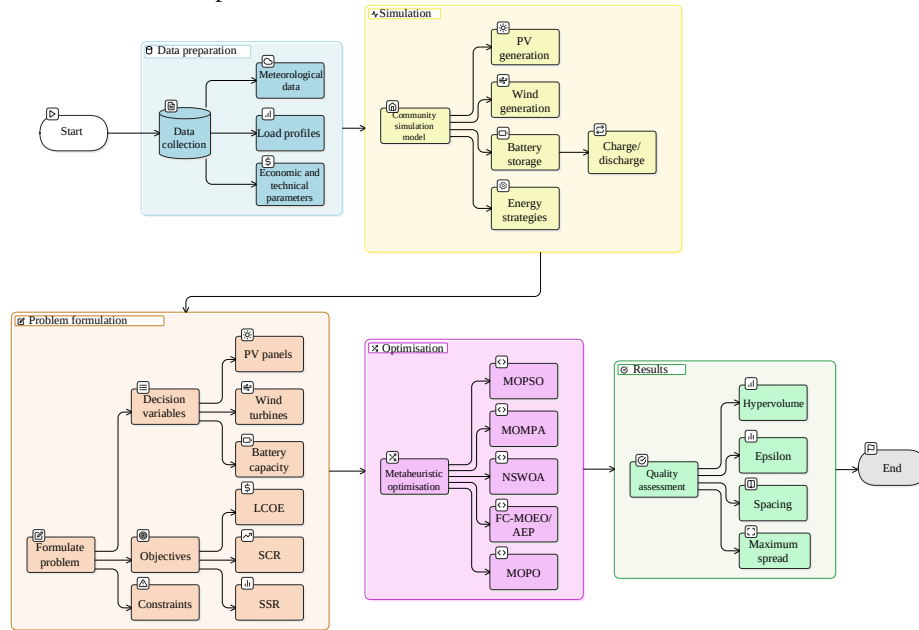


Figure 1. General overview of the proposed methodology.

2.1 Dataset

The data used in this study are divided into two categories: residential load profiles [33] and meteorological data [35]. The load profile data consist of information about

the electricity consumption of individual households. On the other hand, the meteorological data include variables such as temperature, humidity, wind speed, and precipitation, recorded hourly from various weather stations. These data are essential for simulating the community's energy generation process.

Load profiles represent a building's energy consumption over time. The Open Energy Data Initiative (OEDI) provides simulated load profiles for commercial and residential buildings at each TMY3 location. These profiles are generated based on the characteristics of various building prototypes and the climate zone in which they are located. They represent electricity and gas consumption in kilowatt-hours (kWh) at hourly intervals, subdivided into categories such as lighting and heating.

In the first place, the load profiles used in this study correspond to three single-family homes located in Alabama and the neighbouring state of Mississippi. Specifically, these profiles are associated with the following TMY3 locations: Birmingham-Shuttlesworth International Airport, Anniston Regional Airport, and McComb-Pike County Airport. These locations were chosen to ensure they are within the same climate zone as the selected meteorological station, but with different consumption patterns, making the internal interactions characteristic of an energy community necessary. Each participant in the energy community is assigned one of the load profiles. Besides, this study only considers the total electricity consumption of the households.

Secondly, the TMY3 dataset contains weather-related data for hundreds of locations in the United States over a typical year. This typical year is constructed for each location using data collected at a weather station at hourly intervals over at least ten years between 1976 and 2005. TMY3 employs a method that selects individual months from different years within the data recording period. For example, the first month of each year is compared, and the month considered most typical according to various criteria is selected [36]. This process is repeated for all twelve months, which are then concatenated to form a complete typical year. Since adjacent months in the typical year may come from different years, we can find discontinuities between months. We therefore applied smoothing techniques to address these discontinuities.

TMY3 includes a wide range of weather variables, such as temperature, precipitation, solar radiation, wind speed and direction, humidity and visibility. Among these, the variables needed to simulate the renewable energy generation of the community are air temperature ($^{\circ}\text{C}$), wind speed (m/s) and global horizontal irradiance (W/m^2), i.e., the total amount of solar radiation incident on a horizontal surface during each hourly period.

For this study, we selected the meteorological station at Birmingham-Shuttlesworth International Airport in Alabama which is located at an elevation of 189 meters above sea level and has geographic coordinates of 33.567° N, 86.750° W. The data recording period for this location is 24 years.

It is important to take into account that our study utilises synthetic datasets, which, while standard in many research contexts, may not fully capture the real behavioural variability of actual communities. The use of synthetic datasets is often imposed by the challenges associated with obtaining publicly available real data, which is frequently limited in scope or involves a small number of buildings. However, we would like to note that the most recent and validated version of the «End-Use Load Profiles for the

U.S. Building Stock» serves as the foundation for our analysis and it is a good choice since this dataset is based on simulations of approximately 900.000 modelled buildings [37], both residential and commercial. Although the resulting profiles are synthetic, they have been extensively calibrated and validated against measured data where available.

2.2 Problem formulation

For each participant i , three decision variables were defined: 1) the number of photovoltaic modules (N_{PV_i}), 2) the number of wind turbines (N_{WT_i}) and 3) the battery capacity in kWh (C_{bat_i}). For a community of p participants, the problem involves $3p$ decision variables, and a solution is represented as:

$$(N_{PV_1}, N_{WT_1}, C_{bat_1}, N_{PV_2}, N_{WT_2}, C_{bat_2}, \dots, N_{PV_p}, N_{WT_p}, C_{bat_p}) \quad (1)$$

Since the community is small-scale, the size of the participants' facilities is subject to the constraints given by equation (2).

$$\begin{cases} 0 < N_{PV_i} \leq 50 \\ 0 \leq N_{WT_i} \leq 10 \\ 0 < C_{bat_i} \leq 25 \end{cases} \quad (2)$$

To determine optimal dimensions for installations in a renewable energy community, several factors must be considered. Beyond economic aspects, maximizing the community's internal production is crucial due to the intermittent nature of renewable energy and the need to minimize environmental impacts from the electrical grid.

The objectives to be optimized in the community model adopted in this work are three: 1) the LCOE as an indicator of the project's economic feasibility, 2) the SSR and 3) SCR. These metrics are computed based on a one-year simulation of the energy community.

The first objective, the LCOE, is an economic metric for comparing energy generation costs over their lifespan, reflecting the average cost per unit of energy, including construction, operation and maintenance expenses. In this study, LCOE is expressed in \$/kWh and calculated using the next equation.

$$LCOE = \frac{(C_{ini} + C_m) \cdot CRF(i, N)}{E_{gen}} \quad (3)$$

Where C_{ini} is the initial capital cost of the installation, C_m is the operational and maintenance cost, and E_{gen} is the energy generated by the installation in one year. CRF is the capital recovery factor, calculated based on the interest rate i and the project lifetime (N years):

$$\text{CRF}(i, N) = \frac{i(1+i)^N}{(1+i)^N - 1} \quad (4)$$

The initial capital cost of the installation depends solely on the total number of panels, turbines, and batteries installed, as well as their individual costs:

$$C_{ini} = \sum_{i=1}^p (N_{PV_i} \cdot c_{PV} + N_{WT_i} \cdot c_{WT} + C_{bat_i} \cdot c_{bat}) \quad (5)$$

where p is the number of participants in the community, and c_{PV} , c_{WT} and c_{bat} are the cost of a solar panel, a wind turbine and one kWh of battery storage, respectively.

The second and third objective are the SCR and SSR. They can be calculated from an individual or collective perspective. However, since renewable energy communities seek to reduce grid reliance, the optimisation metrics we adopted are based on the collective perspective. The SCR is defined as the proportion of energy generated by the community that is directly consumed within the community, rather than exported to the electrical grid. Intuitively, smaller installations result in higher self-consumption rates, as the likelihood of producing excess energy decreases. It is defined as follows:

$$\text{SCR} = \frac{E_{concom}}{E_{gen}} = \frac{E_{gen} - E_{exp}}{E_{gen}} \quad (6)$$

The SSR, on the other hand, represents the percentage of total energy demand met by the energy generated within the community. A high self-sufficiency rate indicates a reduced need to import energy from the grid and is typically associated with larger generation units. Both rates are bounded in the interval $[0, 1]$ and is calculated as follows:

$$\text{SSR} = \frac{E_{concom}}{E_{contotal}} = \frac{E_{gen} - E_{exp}}{E_{gen} - E_{exp} + E_{imp}} \quad (7)$$

E_{concom} is the energy consumed from the community's production, $E_{contotal}$ is the total energy consumed, E_{gen} is the total energy generated by the community, E_{exp} is the energy exported to the grid, and E_{imp} is the energy imported from the grid. These energy quantities were considered over a one-year period.

To conclude, we must note that hourly renewable generation was estimated using established photovoltaic and wind power models parameterized by the TMY3 meteorological dataset. Photovoltaic production was computed based on irradiance and temperature data, and they account for panel efficiency and surface area, while wind power generation was derived from wind speed measurements and turbine power curves. Battery operation was simulated as a simple state-of-charge model, where energy is stored or discharged depending on production-demand balance, subject to capacity limits and

round-trip efficiency losses. Finally, energy flows within the community were simulated under four management strategies adapted from [34]. In the first scenario, members prioritize self-consumption of their own generated energy before sharing or exporting surplus to the grid. The second scenario prioritizes maximizing the community's collective self-consumption; in doing so, it encourages local sharing of extra energy. The third scenario focuses on reducing reliance on external sources by prioritizing community self-sufficiency, by using batteries and shared resources to cover demand internally. The fourth scenario seeks to minimize costs by dynamically deciding between local consumption and grid exchange based on real-time production, demand and economic factors. These simulations are the input data required to compute our objective functions.

2.3 Multi-objective metaheuristics

A metaheuristic is an iterative optimisation method that starts with an initial population of solutions and utilises characteristics of the entire population at each iteration to guide the search in subsequent iterations.

The problem at hand is a multi-objective optimisation problem, where multiple objective functions are optimised simultaneously. It results in a Pareto set of solutions rather than a single best solution. For further reference, the basic concepts associated with this term are defined in [38].

The algorithms implemented in this study were the Multi-Objective PSO (MOPSO), Multi-Objective MPA (MOMPA), Non-dominated Sorting WOA (NSWOA) and Fast Convergence Multi-Objective EO with Archive Evolution Path (FC-MOEO/AEP). These metaheuristics are inspired by different biological and physical processes. In contrast, the PO, introduced in [32] has its inspiration from politics. Since we recognise the potential of this unique approach, we also propose an adaptation of this metaheuristic for multi-objective optimisation problems (referred to from now on as MOPO).

Multi-Objective Political Optimiser

The PO draws inspiration from the human-centric process of politics [32]. It models key phases of democratic systems, including party formation, constituency allocation, election campaigns, voting and parliamentary affairs. In this context, each solution represents a candidate for public office, assigned to a specific political party and constituency, which corresponds to an electoral division. Each constituency is represented by a single candidate from each political party. The total population size is determined by the number of parties and areas. This algorithm was selected for this study due to its ability to capture the dynamics of negotiation and coordination observed in energy communities. In this analogy, political parties can be used to represent community members, each controlling a distinct set of resources such as solar generation, wind generation and batteries. This reflects the presence of diverse stakeholders with individual objectives, comparable to political actors with their own agendas, who must nonetheless cooperate to achieve beneficial collective outcomes. The optimisation process operates on community-wide metrics such as the LCOE, SSR and SCR. Through iterative

negotiation, the algorithm structure emulates the process that is essential for aligning the interests of all participants in a shared energy system.

In this study, we propose a novel approach by introducing its multi-objective version, MOPO. We specifically designed it to tackle the optimisation challenges faced by energy communities. Unlike the original PO, which requires selecting the best solution from each party and area, MOPO employs a more sophisticated ranking mechanism. This mechanism utilises non-dominated sorting and a reference point technique on a hyperplane [39]. The reason for this is to enhance diversity and efficiency. The pseudocode for this algorithm can be found below.

Algorithm 1: MOPO pseudocode

Input: Maximum number of iterations T , number of parties P , number of areas A , archive capacity N_A , maximum party-switching probability λ_{max}

Output: Archive of non-dominated solutions $\mathbf{A}$

$N \leftarrow P \times A$

Randomly initialize the population $\mathbf{X}$ and evaluate it

Update the archive $\mathbf{A}$ using *non-dominated sorting* and reference points

Find the leader of each party and the winner of each area

for $t = 1$ **to** T **do**

 Update λ

for $i = 1$ **to** N **do**

 Update X_i according to the party leader, area winner and archive

$r \leftarrow$ random number between 0 and 1

if $r < \lambda$ **then**

 Swap X_i with the worst member of a random party p

end

 Evaluate the new solution X_i

end

 Update the archive $\mathbf{A}$ using *non-dominated sorting* and reference points

 Find area winners (who form the parliament) and party leaders

for $i = 1$ **to** A **do**

 Compute new solution P'_i from parliament member P_i

if P_i does not dominate P'_i **then**

 Replace P_i in the population with P'_i

end

end

end

In our algorithm, each individual in the initial population is linked to a political party and an area. Leaders are selected from each party through non-dominated sorting, with a focus on the least dense reference points. Similarly, winners are chosen from each area, resulting in a set of party leaders and area winners.

During the electoral campaign phase, the positions of individuals are updated based on their previous election results, the influence of their party leader, and the area winner. In doing so, it is expected that the algorithm explores promising regions more effectively. Moreover, individuals are encouraged to approach solutions from an archive, which represents the approximate Pareto set, thereby accelerating convergence.

Following the electoral campaign, a party-switching phase occurs, where individuals have the opportunity to replace less successful members of other parties. The objectives of each solution are evaluated, reflecting voter support. New solutions are added to the archive, and dominated solutions are removed to maintain quality. If the archive reaches its capacity, a density criterion is applied to eliminate excess solutions, so only the most relevant candidates remain.

Finally, the winners from each area form a parliament, where members adjust their positions based on interactions with randomly selected colleagues. This adjustment adds new positions to the population if they show enhanced performance, which promotes a competitive and dynamic optimisation environment.

Multi-Objective Particle Swarm Optimisation

The MOPSO algorithm [40] is an adaptation of the classic PSO [28] designed for multi-objective optimization problems. PSO mimics the social behaviour of animal swarms, where each particle represents a candidate solution that adjusts its movement based on personal and collective experiences. In each iteration, particles update their velocities based on inertia, their best-known position, and the global best position found by the swarm.

MOPSO modifies this approach to handle multiple objectives by replacing the global best position with one from a non-dominated solution archive, selected randomly with probabilities influenced by a grid mechanism that splits the objective space into hypercubes. This promotes exploration of less populated areas. After updating velocities and positions, a particle's best position is only updated if the new position dominates the previous one. At the end of each iteration, non-dominated solutions are stored in the archive, which uses the grid mechanism to remove solutions from crowded regions if it exceeds capacity.

Multi-Objective Marine Predators Algorithm

The third metaheuristic is inspired by the hunting strategies of marine predators and their interactions with prey, the MPA [29]. It operates on the principle of «survival of the fittest», where the most promising solutions act as predators guiding the search for optimal solutions. The algorithm utilises an elite solution selected from a prey matrix, which represents all potential solutions. To balance exploration and exploitation of the search space, MPA alternates between Lévy flights and Brownian motion phases.

In the multi-objective version, MOMPA [41], the initial population is a set of randomly generated solutions, with an elite solution chosen from this prey matrix. The algorithm progresses through a series of iterations. It prioritises exploration in the first third by allowing prey to move faster than predators using Brownian motion. In the second third, the first half of the population focuses on exploitation through small movements inspired by Lévy flights, while the second half continues to explore. As iterations progress, the influence of the elite position increases. In the final phase, the predator moves faster than the prey, emphasizing exploitation through Lévy flights. MOMPA also incorporates a mutation mechanism to modify predator behaviour and employs a diversity technique to enhance exploration by utilizing reference points on a

hyperplane in the objective space. In doing so, the algorithm provides a well-balanced selection of elite solutions.

Non-dominated Sorting Whale Optimization Algorithm

The original WOA [30] is a metaheuristic inspired by the hunting strategies of humpback whales, particularly the bubble-net technique. In this algorithm, solutions represent the positions of whales, which are updated in each iteration based on various hunting actions: encircling the prey (the best solution), attacking in a shrinking spiral, or searching for prey by moving away from other whales.

In NSWOA [42], the position of the fittest whale is determined at the start of each iteration using a roulette-wheel selection mechanism, where the selection probability is inversely related to the front number from non-dominated sorting. At each iteration, the hunting strategy for each whale adjusts parameters, with a specified probability of entering an exploration phase, in which whales move towards randomly chosen targets rather than the prey. If exploration does not occur, solutions update their positions through either an encircling phase, moving toward the best solution, or a bubble-net attack phase, where they approach the best solution in a spiral trajectory. Moreover, an adaptive parameter guarantees that exploitation becomes more localised as iterations progress. After evaluating the objective functions of the new solutions, the new population is formed by merging previous and newly generated solutions, from which the best solutions are selected based on non-dominated sorting and crowding distance criteria.

Fast Convergence Multi-Objective Equilibrium Optimiser with Archive Evolution Path

The last optimisation algorithm we implemented is a multi-objective version of the EO [31] with an Archive Evolution Path (AEP) mechanism, which utilises the trajectory of non-dominated solutions [43]. EO is inspired by mass balance models and updates the population in each iteration by adjusting the positions of solutions based on the principles of equilibrium and dynamic states. On the other hand, AEP enhances convergence toward the Pareto set by generating new candidate solutions based on the evolution of the archive from the second iteration onward. Besides, it also employs a diversity technique based on the maximin function.

In the algorithm, solutions obtained from EO and AEP are merged, and their objective values are calculated to identify non-dominated solutions for the archive. The archive is sorted using the maximin metric, which measures the diversity of the Pareto front, favouring solutions in less dense regions of the objective space. If the archive exceeds its capacity, the solution with the largest maximin value is removed. The population is then reduced to the desired size by selecting solutions with the smallest maximin values. After this, new solutions are generated from the population, and the candidates are selected from the archive based on a roulette-wheel method that prioritises those with smaller maximin values. Furthermore, AEP generates new solutions independently from the archive, guided by the archive's centre and its evolution taking into account recent iterations.

2.4 Performance metrics

In order to evaluate the implemented algorithms, we must define some metrics to assess their effectiveness. This section defines the four adopted metrics: Maximum spread, spacing, ϵ indicator, and hypervolume.

The first metric is Maximum Spread (MS) [44]. It measures the extent of the solution set across the objective space and is defined based on the range of each objective in the set of m -objective vectors corresponding to the N solutions obtained by an algorithm, A . Larger MS values indicate a greater spread of the obtained Pareto front, so MS should be maximised. With W and W' representing individual solutions, MS is defined as follows:

$$MS(A) = \sqrt{\sum_{j=1}^m \max_{W, W' \in A} (W_j - W'_j)^2} \quad (8)$$

The spacing (SP) metric calculates the variation in distances between solutions. SP is an indicator of uniformity, taking non-negative values, with smaller values indicating greater uniformity. A zero SP value corresponds to a set of equidistant solutions, in terms of the Manhattan distance, in the objective space. Its mathematical formula is:

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\bar{d} - dist(W_i, A \setminus \{W_i\}))^2} \quad (9)$$

Where $dist(W_i, A)$ represents the Manhattan distance from W to the set A , and $\bar{d}$ is the mean of the minimum distances calculated between one solution and all others.

For our problem, we only require only one solution that represents the generation and storage resources. Therefore, an algorithm with limited exploration capacity could still obtain the most suitable configuration for installation. As a consequence, dominance relations between solution sets were used to compare our metaheuristics. Given two sets of objectives, A and B , the ϵ indicator [38] quantifies the dominance relationship between them. It is a binary indicator that represents a pair of values (ϵ_+, ϵ_x) , which stand for the smallest number that must be added to or multiplied by the objective set A for it to dominate B .

$$\begin{aligned} \epsilon_+(A, B) &= \max_{W \in B} \min_{V \in A} \max_{i \leq j \leq m} (V_j - W_j) \\ \epsilon_x(A, B) &= \max_{W \in B} \min_{V \in A} \max_{i \leq j \leq m} \left(\frac{V_j}{W_j} \right) \end{aligned} \quad (10)$$

The last metric adopted is the hypervolume (HV) [45] which provides a measure of the space covered by the obtained Pareto front A , with respect to a reference point R . Formally, it is defined as the volume of the union of the hyperrectangles with one vertex

at R and the opposite vertex at the point defined by the objective values of a solution, where X and Y are elements within the set A and the solution space, respectively. $\mathcal{S}$ signifies the solution space itself, which encompasses all possible solutions. Finally, the symbol λ represents the Lebesgue measure.

$$HV(A) = \lambda \left(\bigcup_{X \in A} \{Y \in \mathcal{S} | X < Y < R\} \right) \quad (11)$$

Unlike the ϵ indicator, HV is not binary, making it easier to interpret. However, the choice of the reference point is complex in the optimisation problem at hand, due to the lack of knowledge about the true Pareto front. In this case, the reference point for the hypervolume was calculated using the worst possible value for each objective. Since the reference point significantly influences the HV calculation, it has been implemented alongside the epsilon indicator. The use of both indicators can contribute to more informed decision-making, as both perform similarly when used to compare algorithms [46].

3 Experiments

In this section, we first present an exploratory analysis of the experiments conducted, followed by a detailed overview of the parameter settings used for each metaheuristic. The experiments were carried out on an HP Pavilion x360 14-dy1xxx computer using Python 3.9.

3.1 Exploratory analysis

Before running the optimisation experiments, an exploratory analysis was conducted to better understand the characteristics of our datasets. Table 1 summarises the key statistics of the economic parameters and technical specifications used in the simulation model. The skewness and kurtosis coefficients calculated are Fisher's, which means that the reference value for a normal distribution is zero.

Table 1. Descriptive statistics of the dataset.

Statistic	Irradiance	Temperature	Wind speed	Load 1	Load 2	Load 3
Unit	W/m ²	°C	m/s	kWh	kWh	kWh
Mean	186.404	16.972	2.943	0.789	1.498	1.713
Std. Dev.	263.347	9.097	1.993	0.287	0.642	0.877
Skewness	1.281	-0.406	0.436	0.369	0.526	1.439
Kurtosis	0.392	-0.560	0.410	-0.744	-0.310	3.319
Min	0.000	-12.200	0.000	0.328	0.496	0.480
Q1	0.000	10.575	1.500	0.564	1.027	1.103

Q2	12.000	18.300	2.600	0.754	1.414	1.543
Q3	324.000	23.900	4.100	0.996	1.934	2.149
Max	1013.000	35.000	14.400	1.693	3.825	8.007

As shown in Figure 2, all three participants experience increased consumption in winter and summer, with the winter load being more pronounced for the third participant. Similarly, two daily spikes are observed at 8:00 AM and 8:00 PM, with the latter being higher. The lowest consumption occurs during the early morning hours for all three participants.

The first participant requires less daily energy than the others, with an average hourly consumption below 1 kWh and little variation throughout the year. Participants 2 and 3 have similar energy needs, except in December, January and February, when the third participant's consumption is significantly higher.

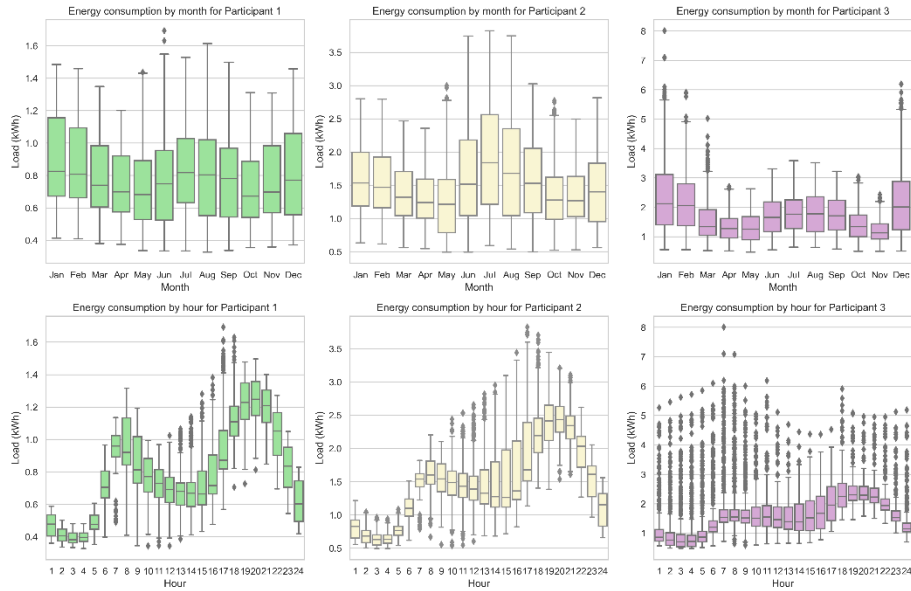


Figure 2. Boxplots of the three load profiles by month and time of the day.

Temperature exhibits a wider range during winter months (see Figure 3). During July and August, temperatures never fall below 20°C, indicating warm summers both day and night. Due to the fact that there is no irradiance between 7:00 PM and 4:00 AM, no solar energy production occurs during this time, so it will require reliance on wind energy or stored energy in batteries. In contrast, wind speed is highly inconsistent, with numerous outliers observed across most months and hours. This inconsistency highlights the intermittent nature of wind energy generation and the advantages of combining it with solar energy.

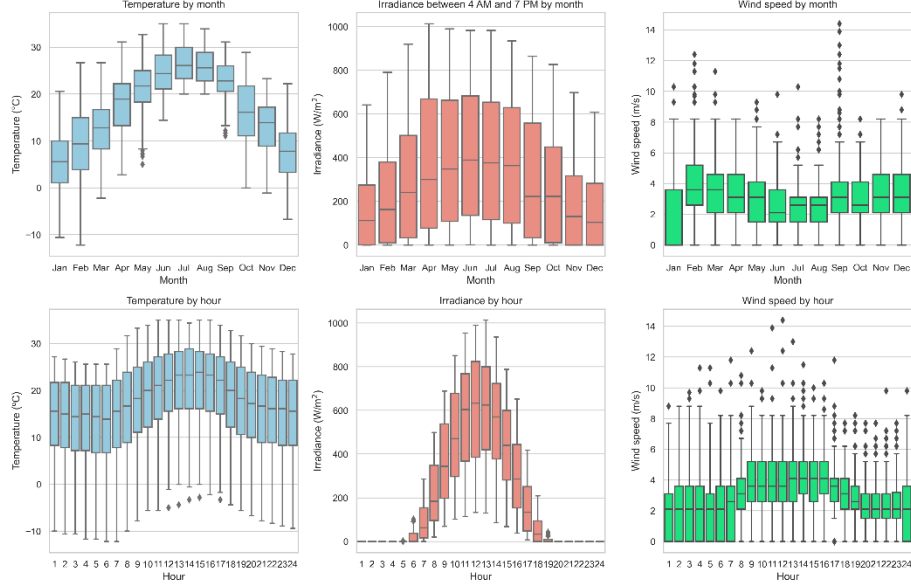


Figure 3. Boxplots of the variables in the weather dataset by month and time of the day.

3.2 Parameter settings

The optimisation problem we intend to solve has four distinct versions, referred to as Scenarios 1 to 4. The first scenario represents a model where participants operate independently. They consume only the energy produced or stored in their own installations and import energy from the grid, when necessary, with any surplus energy exported to the grid. In contrast, Scenario 2 introduces internal interactions, which allows participants to import surplus energy from others if their own generation is insufficient, while still retaining grid interactions as a last resort. Scenario 3 maintains the same structure as Scenario 2 but prioritises community interactions over energy storage during each time interval. Lastly, Scenario 4 incorporates shared storage systems, that enables participants to charge and discharge the batteries of others after updating their own storage units.

We applied the five aforementioned algorithms to these versions of the optimisation problem to compare the results. The algorithms were developed in Python 3.9, and all experiments were conducted on a laptop equipped with an Intel Core i5-1155G7 processor running at 2.50 GHz and 16 GB of RAM. To execute the algorithms, it was necessary to assign values to the parameters defining the components of the energy community simulation, including the characteristics and costs of the energy generation modules. These values are summarised in the next Table 2. Note that maintenance costs are assumed to be 10% of the initial capital cost.

Table 2. Parameters for the energy community simulation.

Component	Parameter	Name	Value
Solar energy	μ	Efficiency	0.95
	P_{STC}	Maximum power under standard conditions	250 W
	G_{STC}	Solar irradiance under standard conditions	1,000 W/m ²
	γ	Temperature coefficient under standard conditions	-0.0044 V/°C
	T_{STC}	Temperature under standard conditions	25 °C
	$NOCT$	Nominal operating cell temperature	47.5 °C
	G_{NOCT}	Solar irradiance under NOCT conditions	800 W/m ²
	T_{NOCT}	Temperature under NOCT conditions	20 °C
Wind energy	h_{WT}	Rotor height	20 m
	h	Anemometer height	10 m
	α	Power law exponent	0.142
Batteries	c	Fraction of available charge	0.271
	k	Flow rate between tanks	0.38
LCOE	C_m	Maintenance costs	10%*
	i	Nominal interest rate	0.05
	N	Project lifetime	20 years
	C_{PV}	Cost per photovoltaic module	\$170
	C_{WT}	Cost per wind turbine	\$3500
	C_{bat}	Cost per kWh of battery storage	\$250

The photovoltaic panel and wind turbine models used as references for renewable energy generation simulation are the Sharp ND-R250A5 and the Bergey BWC Excel 1, respectively.

Since the three objectives are computed using the amounts of energy generated, imported, and exported, their calculation depends on the parameter values. LCOE is an objective to be minimised, while SCR and SSR are objectives to be maximised. To simplify the implementation of experiments and interpreting the results, SCR and SSR are redefined as negative values. In this way, we will seek to maximise in the optimisation process.

Furthermore, in order to choose the optimal combination of parameters for each algorithm, including the population size, we followed the next strategy. The selection criteria were the ϵ and HV indicators. A grid search was conducted twice for each algorithm, refining the parameter values based on the results of the previous search to increase their granularity. The chosen parameter configurations are listed in Table 3. The archive size was set to 50, while the population size was 100 for MOPSO, MOMPA, NSWOA, and FC-MOEO/AEP. For MOPO, the population size was calculated as the product of the number of parties and the number of areas, resulting in 80.

Table 3. Metaheuristic parameters.

Metaheuristic	Parameter	Name	Value
MOPO	P	Number of parties	8
	A	Number of areas	10
	λ_{max}	Maximum party switch rate	0.65
MOPSO	w	Inertia parameter	0.7
	k	Number of axis divisions in the grid	7
MOMPA	p	Step size	1

NSWOA FC-MOEO/AEP	p_{FADs}	FADs probability	0.6
	b	Spiral parameter	0.85
	a_1	Exploration parameter in EO	2
	a_2	Exploitation parameter in EO	0.5
	p_G	Generation probability in EO	0.75
	σ	Success rate threshold in AEP	0.5
	ω	Parameter for updating α in AEP	1.5
	k	Number of stored archives in AEP	4

Since the algorithms differ in their structures, using iterations for comparison is not ideal in this case. Therefore, we chose a fixed limit of 5.000 objective function evaluations to ensure a fair comparison. Besides, the quality of the obtained solutions is evaluated from two perspectives: external indicators and internal feasibility. External evaluation involves calculating maximum spread, spacing, epsilon, and hypervolume indicators, while internal assessment focuses on the feasibility of installations based on LCOE, SCR, and SSR at both individual and collective levels across different scenarios.

For each of the four scenarios, the evaluation procedure includes executing each algorithm 30 times, calculating the mean and standard deviation for execution time and all indicators except epsilon, and computing the epsilon indicator as the mean of values from the Pareto fronts of paired algorithms.

In this study, the constraints applied were limited to the upper and lower bounds of the decision variables. The dataset did not contain specific technical or economic limits such as panel or battery sizing. While it would be possible to introduce assumptions, for example to prevent one participant from taking a disproportionate share of resources, these restrictions were not implemented directly in order not to bias the outcomes. Nevertheless, the Pareto front solutions obtained are inherently balanced due to the optimisation method, which seeks the best compromise between the individual objective functions.

4 Results and discussion

Once the parameters are set, this section presents the results from applying the five multi-objective metaheuristics to the optimisation problem across the four scenarios. The results are presented on two perspectives: the performance of the metaheuristics in solving the problem and the characteristics of the solutions obtained.

The performance of the algorithms in the four Scenarios was examined using the quality indicators are presented in Table 4. The ϵ values were calculated as the average of the ϵ indicators between the Pareto fronts of the evaluated algorithm and those of the other algorithms. According to the table, the ability of the algorithms to find a Pareto front that spans a wide region of the objective space has been quantified using the maximum spread indicator. MOMPA stands out in this regard, while MOPSO sometimes struggles to find solutions with extreme values for individual objectives, as shown by its MS values. The uniformity of the solutions, measured by SP, is higher for MOPSO

and MOPO. NSWOA demonstrated a more limited capacity to explore the interior regions of the space, which results in gaps that manifest as high spacing values.

The solutions obtained by FC-MOEO/AEP require smaller translations to dominate the Pareto fronts of other algorithms, as indicated by both additive and multiplicative ϵ indicators. MOPSO ranks second in this regard. While HV can compare dominance relationships, its reliance on an estimated nadir point makes it less reliable as a sole criterion. In this context, MOPSO performs well, followed by MOMPA and FC-MOEO/AEP. The differences in dominance relationships are more evident in the first three scenarios, likely due to closer objective values in Scenario 4. Pairwise epsilon indicators show that FC-MOEO/AEP slightly outperforms MOPSO across all scenarios. Additionally, FC-MOEO/AEP is the fastest algorithm in terms of execution time.

If we focus on our proposed algorithm, MOPO, we can observe that its HV values are generally competitive, often falling within a close range of the best-performing algorithms, MOPSO, MOMPA and FC-MOEO/AEP. This suggests that MOPO is capable of finding solutions that effectively cover a significant portion of the objective space. In terms of other performance metrics, MOPO's performance exhibits certain variability. It is often neither the best nor the worst performer.

Table 4. Quality indicators and execution time for each metaheuristic in each scenario.

Metaheuristic	MS	σ	SP	σ	ϵ_+	σ	ϵ_-	σ	HV	σ	Time (s)	σ
Scenario 1												
MOPSO	0.950	0.13	0.065	0.01	0.029	0.02	1.175	0.09	1.422	0.00	436	1.70
MOMPA	1.080	0.06	0.104	0.03	0.049	0.03	1.264	0.14	1.410	0.01	531	0.74
NSWOA	0.928	0.12	0.169	0.07	0.086	0.04	1.560	0.28	1.347	0.06	971	9.80
FC-MOEO/AEP	1.020	0.03	0.095	0.02	0.022	0.01	1.161	0.08	1.407	0.01	331	0.61
MOPO	0.992	0.13	0.086	0.02	0.049	0.03	1.290	0.15	1.404	0.01	846	2.53
Scenario 2												
MOPSO	0.898	0.13	0.061	0.01	0.027	0.01	1.184	0.01	1.422	0.00	450	2.38
MOMPA	1.020	0.06	0.100	0.03	0.050	0.03	1.294	0.16	1.406	0.01	553	0.90
NSWOA	0.942	0.10	0.163	0.04	0.094	0.05	1.601	0.31	1.332	0.06	992	2.44
FC-MOEO/AEP	0.992	0.03	0.084	0.02	0.023	0.01	1.168	0.09	1.403	0.01	354	1.47
MOPO	1.040	0.12	0.087	0.03	0.065	0.04	1.396	0.22	1.383	0.02	846	0.61
Scenario 3												
MOPSO	0.901	0.11	0.065	0.01	0.030	0.02	1.188	9.01	1.427	0.00	499	2.70
MOMPA	1.050	0.05	0.100	0.03	0.056	0.03	1.303	0.16	1.409	0.02	615	1.13
NSWOA	0.927	0.09	0.147	0.05	0.100	0.05	1.648	0.33	1.334	0.05	1060	3.13
FC-MOEO/AEP	0.991	0.03	0.090	0.02	0.023	0.01	1.166	0.09	1.406	0.02	411	0.60
MOPO	1.020	0.10	0.088	0.02	0.062	0.03	1.364	0.20	1.395	0.02	937	1.77
Scenario 4												
MOPSO	0.697	0.25	0.047	0.01	0.017	0.01	1.068	0.03	1.505	0.00	1750	5.36
MOMPA	0.969	0.10	0.079	0.03	0.043	0.02	1.170	0.09	1.502	0.01	2180	8.29
NSWOA	0.940	0.08	0.101	0.03	0.071	0.04	1.284	0.14	1.497	0.01	2630	4.05
FC-MOEO/AEP	0.961	0.03	0.062	0.01	0.009	0.00	1.060	0.03	1.504	0.00	1210	1.47
MOPO	0.737	0.29	0.087	0.06	0.038	0.02	1.146	0.08	1.502	0.04	3560	20.00

Now we present Table 5, a statistical analysis for the optimization capability of the algorithms. Across 30 executions, the maximum spread, spacing, hypervolume and the execution time are extracted for each algorithm and scenario. Pairwise statistical significance was assessed using the Wilcoxon signed-rank test with a Holm correction to control the family-wise error rate.

According to Table 5, the results confirm several of the descriptive observations. For MS, MOMPA significantly outperforms all other algorithms in Scenario 1 and remains among the top performers in Scenarios 2 and 3. In these two scenarios, MOPO performs similarly to MOMPA, showing competitive results. In Scenario 4, MOMPA loses its

advantage to NSWOA and FC-MOEO/AEP, while MOPO maintains a moderate performance across all scenarios, i.e., never the best but consistently competitive. From the point of view of spacing, MOPSO is significantly better than all other algorithms in all scenarios, followed by MOPO, MOMPA and FC-MOEO/AEP, which show statistically similar performance and generally outperform NSWOA. The high HV values obtained by MOPSO are statistically higher than those of any other algorithm, making MOPSO the winner in this regard. FC-MOEO/AEP shows similar performance to both MOMPA and MOPO, while NSWOA is statistically outperformed by all. Finally, the time execution is different for all algorithms, with the best ranking one being FC-MOEO/AEP.

Table 5. Comparison between algorithms according to Wilcoxon signed-rank test with Holm correction, in every scenario. Font colours indicate the statistical significance of the p-values. Bold corresponds to highly significant results ($p < 0.01$), italic to significant results including the marginally significant ($0.01 \leq p < 0.10$) and grey to non-significant results ($p \geq 0.10$).

Scenario	Metric	MOPSO vs MOMPA	MOPSO vs NSWOA	MOPSO vs FC- MOEO/AEP	MOPSO vs MOPO	MOMPA vs NSWOA	MOMPA vs FC- MOEO/AEP	MOMPA vs MOPO	NSWOA vs FC- MOEO/AEP	NSWOA vs MOPO	FC- MOEO/AEP vs MOPO
1	MS	2.42E-04	1.00E+00	2.25E-01	4.21E-01	9.22E-05	1.22E-03	4.27E-02	1.79E-03	3.07E-01	1.00E+00
	SP	1.68E-07	3.73E-08	1.68E-07	1.43E-06	1.84E-03	1.84E-01	1.28E-01	2.48E-04	2.83E-06	1.40E-01
	HV	3.35E-08	1.86E-08	6.52E-08	7.82E-08	4.47E-08	2.48E-01	1.73E-02	4.00E-07	1.54E-06	3.18E-01
	Time	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05	1.36E-05
2	MS	1.40E-04	2.81E-01	5.59E-03	2.08E-03	5.71E-03	1.87E-02	2.81E-01	1.39E-01	4.04E-03	4.62E-02
	SP	1.86E-08	1.86E-08	3.20E-05	2.57E-05	6.52E-08	1.15E-01	3.96E-01	1.86E-08	9.83E-07	1.00E+00
	HV	1.30E-07	1.86E-08	1.86E-08	1.86E-08	1.86E-08	3.60E-01	5.59E-05	1.12E-07	4.59E-04	1.74E-03
	Time	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05	1.31E-05
3	MS	2.83E-06	5.29E-01	4.40E-04	2.07E-03	6.15E-07	2.21E-05	4.41E-01	2.07E-03	2.07E-03	2.76E-01
	SP	2.04E-06	3.73E-08	1.83E-04	2.20E-04	1.52E-03	6.37E-01	2.88E-01	1.17E-06	6.99E-06	6.55E-01
	HV	9.13E-08	1.86E-08	1.86E-08	1.86E-08	9.13E-08	3.28E-01	1.38E-03	9.31E-08	2.29E-06	1.40E-01
	Time	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05	1.42E-05
4	MS	2.83E-06	9.75E-05	2.05E-06	6.32E-01	6.32E-01	6.32E-01	7.92E-03	6.32E-01	6.79E-03	9.62E-04
	SP	1.33E-05	1.86E-08	1.42E-04	5.60E-05	2.17E-02	6.98E-02	5.84E-01	1.84E-06	3.15E-01	3.41E-01
	HV	6.08E-01	4.42E-05	6.08E-01	3.77E-02	4.27E-02	6.17E-01	6.17E-01	4.42E-05	1.75E-02	3.03E-01
	Time	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05	1.70E-05

Additionally, we performed a Friedman test to compare all algorithms simultaneously, providing mean ranks that summarise their overall performance. It can be seen in Table 6. This table confirms the presence of at least one statistically significant difference among the algorithms for each metric in every scenario. Examining the mean rankings across scenarios in the table and the overall rankings in Table 7, MOMPA and MOPO consistently achieve top positions in most scenarios for MS. NSWOA generally ranks lowest, while FC-MOEO/AEP occupies middle positions, i.e., competitive but without consistent dominance. For SP, the final ranking highlights MOPSO, MOPO, and FC-MOEO/AEP as the best-performing algorithms, followed by MOMPA and NSWOA. The HV metric also supports MOPSO's superiority, followed by FC-MOEO/AEP and MOMPA, whereas execution time clearly favours FC-MOEO/AEP.

Table 6. Friedman test results and ranks of algorithms per metric and scenario.

Scenario	Metric	χ^2	p	Rank				
				MOPSO	MOMPA	NSWOA	FC-MOEO/AEP	MOPO
1	MS	33.52	0	3.5333	1.7	3.9	2.9333	2.9333
	SP	65.36	0	1.2333	3.4	4.4333	3.2	2.7333
	HV	83.84	0	1.1	2.6667	4.7667	3.1	3.3667
	Time	119.2267	0	2	3	4.9667	1	4.0333
2	MS	37.2533	0	3.9667	2.3333	3.6333	3.2	1.8667
	SP	78.2133	0	1.2667	3.2667	4.8333	2.7333	2.9
	HV	87.28	0	1.0667	2.6	4.6667	2.9	3.7667
	Time	120	0	2	3	5	1	4
3	MS	47.9733	0	4.1	1.8333	3.9333	2.8667	2.2667
	SP	53.4133	0	1.6	3.4333	4.4667	2.8667	2.6333

4	HV	88.7733	0	1.0667	2.6333	4.8	2.9667	3.5333
	Time	119.2267	0	2	3	4.9667	1	4.0333
	MS	34.5867	0	4.2	2	2.7	2.6333	3.4667
	SP	51.8933	0	1.5667	3.2667	4.4333	2.6333	3.1
	HV	26.48	0	2.4	2.7	4.2333	2.5333	3.1333
	Time	120	0	2	3	4	1	5

Table 7. Average algorithm ranks across scenarios, for each metric.

Metric	Rank				
	MOPSO	MOMPA	NSWOA	FC-MOEO/AEP	MOPO
MS	3.95	1.9666	3.5416	2.9083	2.6334
SP	1.4167	3.3417	4.5417	2.8583	2.8416
HV	1.4084	2.65	4.6167	2.875	3.45
Time	2	3	4.7334	1	4.2666

MOPSO performs very well in spacing and hypervolume with a moderate execution time. Its leader selection based on crowding distance and velocity updates tends to maintain a well-distributed and dense Pareto front, offering low spacing and high hypervolume. However, it captures extreme solutions less effectively, which explains its lower maximum spread.

MOMPA’s foraging phases, Lévy movements and adaptive changes, seem to favour exploration of the edges of the objective space, leading to higher spread and hypervolume, but do not provide the most uniform spacing.

In our experiments, NSWOA might perform worse because of how it moves (encircling and spiralling around prey solutions). In a smooth, gently sloped problem like sizing PV, wind and batteries, neighbouring solutions do not differ a lot and it can be more difficult to spot the promising directions. This encircling behaviour can make the population cluster in the wrong zones, lowering spread and hypervolume.

In the case of MOPO, our implementation maintains a relatively large set of candidate parties exploring the solution space, with 8 parties and 10 areas. This can preserve diversity, helping spread and spacing. Additionally, the repeated party-switching and population updates explain the relatively high execution time.

Finally, FC-MOEO/AEP uses the AEP technique to generate additional solutions at each iteration of FC-MOEO/AEP. It exploits the monotonic behaviour of the SCR and SSR. Unlike traditional approaches that focus on objective values, AEP considers the differences between iterations in the mean values of the decision variables to identify new candidate solutions. For instance, when an increase in a decision variable leads to an improvement in the objectives, it is likely that further increasing this variable will continue to yield better solutions. This alignment between the AEP mechanism and the characteristics of these two objectives may explain the algorithm’s superior performance.

We can now shift our focus to the analysis of solutions, specifically the representative solutions, where some overall trends can be identified, see Table 8. The representative solution was chosen as the one best trade-off between individual LCOE, SCR, and SSR, to prevent heavy reliance of any participant on energy produced by the rest [47]. The best trade-off among objectives was typically achieved when storage units have a large capacity, around 25 kWh. Additionally, the wind turbine components never

exceed a value of 2. It therefore indicates that this type of generator is not cost-effective in any of the scenarios. Also, the number of photovoltaic modules was consistently greater than 20 for all participants, with this value being higher for participants 2 and 3. This may be attributed to the lower energy demand of participant 1. In Scenario 4, all algorithms tended to converge toward installations that maximize the number of photovoltaic modules. The flexibility offered by shared storage in Scenario 4 enables larger system sizes without raising the levelized cost of energy, and in certain cases, it can actually be decreased. Consequently, Scenario 4 outperforms the others in terms of community-level LCOE, SCR, and SSR.

Table 8. Representative solution for each algorithm in each scenario.

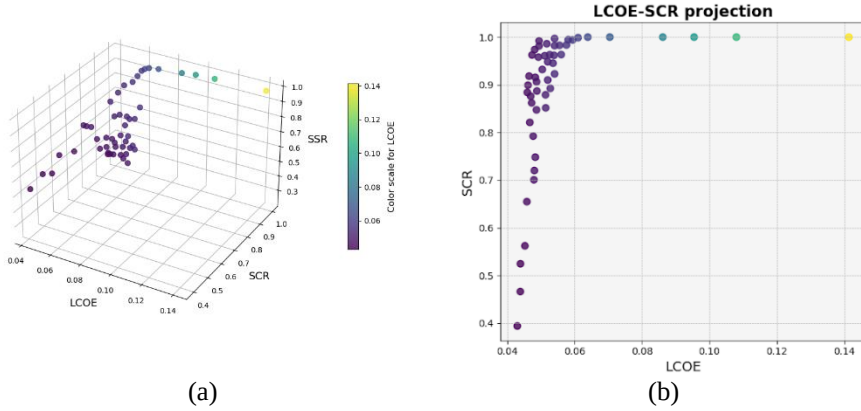
Metaheuristic	Representative solution	LCOE	SCR	SSR
Scenario 1				
MOPSO	(37,0,25,42,0,25,44,0,25)	0.07925	0.9263	0.8608
MOMPA	(30,0,25,35,0,24,40,0,25)	0.08509	0.9640	0.8227
NSWOA	(26,0,25,30,0,25,37,0,25)	0.09134	0.9824	0.7883
FC-MOEO/AEP	(39,0,25,50,0,25,47,0,25)	0.07567	0.8896	0.8802
MOPO	(29,0,23,35,0,25,32,0,25)	0.08851	0.981	0.7957
Scenario 2				
MOPSO	(39,0,25,34,0,25,44,0,25)	0.08117	0.9362	0.8434
MOMPA	(26,0,25,35,0,25,34,0,25)	0.09029	0.9825	0.7975
NSWOA	(24,0,20,32,0,25,35,1,25)	0.09659	0.9811	0.7921
FC-MOEO/AEP	(41,0,25,50,0,25,50,0,25)	0.07447	0.8762	0.8859
MOPO	(28,2,25,35,2,25,29,2,25)	0.13240	0.9832	0.8339
Scenario 3				
MOPSO	(36,0,25,44,0,25,41,0,25)	0.07987	0.9311	0.8592
MOMPA	(31,0,25,50,0,25,38,1,25)	0.08643	0.9238	0.8634
NSWOA	(29,0,22,27,0,22,29,0,22)	0.08949	0.9840	0.7384
FC-MOEO/AEP	(49,0,25,50,0,25,46,0,25)	0.07357	0.8647	0.8873
MOPO	(36,0,25,40,0,25,41,0,25)	0.08117	0.9422	0.8514
Scenario 4				
MOPSO	(50,0,25,50,0,25,50,0,25)	0.07251	1	0.9509
MOMPA	(50,0,25,50,0,25,50,0,25)	0.07251	1	0.9509
NSWOA	(50,0,25,50,0,25,50,0,25)	0.07251	1	0.9509
FC-MOEO/AEP	(50,0,25,45,1,25,50,2,25)	0.08807	1	0.9590
MOPO	(50,0,25,50,0,25,50,0,25)	0.07251	1	0.9509

In the following part, we present the percentages of emissions observed in our study. CO₂ emission reductions from renewable energy generation were assessed relative to exclusive dependence on conventional grid electricity. We will use the representative solution obtained by FC-MOEO/AEP as it was the best-performing algorithm for the calculation (see Table 9). Emission levels were set at 0.225 kgCO₂/kWh for solar energy, 0.008 kgCO₂/kWh for wind energy, and 0.028 kgCO₂/kWh for stored energy [18], while the grid electricity average emission factor was 0.389 kgCO₂/kWh [48]. Greater independence from the grid was associated with progressively larger emission reductions, increasing from 34.0% in Scenario 1 to 34.4% in Scenario 2, 34.6% in Scenario 3, and 38.1% in Scenario 4.

Table 9. Summary of the energy community simulation provided by the solution with the best balance of LCOE, SCR, and SSR at the participant level obtained by FC-MOEO/AEP.

Variable	Scenario 1	Scenario 2	Scenario 3	Scenario 4
<i>Costs (\$)</i>				
Installation	41870	42720	43400	53900
Imports	978	881	835	571
<i>Energy (kWh)</i>				
Generated	48641	50636	52073	54019
Imported	5915	5713	5718	2597
Exported	5394	6269	7046	0
Shared	-	0	541	476
Shared storage	-	-	-	8608
<i>Emissions (%)</i>				
Reduction	34.0	34.4	34.6	38.1

As a final result, we present Figure 4. It shows the Pareto front distribution obtained by one of the algorithms in the fourth scenario. In the LCOE–SCR projection (Figure 4b), most solutions cluster at low values of LCOE with high SCR, while only a few points extend towards higher LCOE and lower SCR. In the LCOE–SSR projection (Figure 4c), a similar pattern is observed, with many solutions concentrated at low LCOE and high SSR and fewer solutions spreading towards less favourable combinations. In the last graph, Figure 4d, the distribution is more dispersed, with several solutions located in the region of high SCR and high SSR, although a number of points also appear at lower levels, which illustrates the trade-offs between these two objectives.



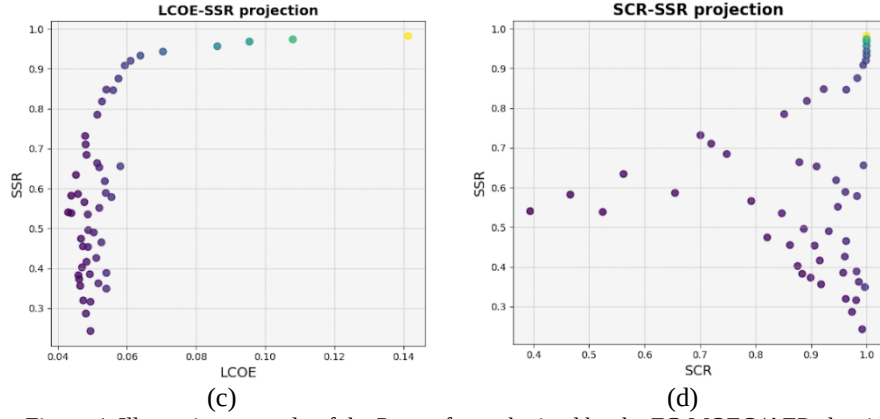


Figure 4. Illustrative example of the Pareto front obtained by the FC-MOEO/AEP algorithm for the fourth scenario.

5 Conclusions

In this study, the optimal sizing of photovoltaic, wind, and storage units for a small-scale urban renewable energy community was formulated as a multi-objective optimisation problem. The cost of energy served as the economic metric, while self-consumption and self-sufficiency rates measured energy flexibility and grid independence. Four scenarios were considered, from full independence among participants to shared energy generation and storage. To solve this problem, five metaheuristics were applied and evaluated using quality indicators.

Among these algorithms, FC-MOEO/AEP, which utilises a solution generation mechanism based on the evolution of the archive of non-dominated solutions, demonstrates rapid convergence and superior consistency in finding dominant solutions. The MOPSO algorithm emerges as a viable alternative, excelling in the uniformity of the Pareto fronts obtained. Additionally, our proposed adaptation, MOPO, shows generally competitive HV values, often falling within a close range of the best-performing algorithms, MOPSO and FC-MOEO/AEP. The results indicate that MOPO can discover solutions that adequately cover a substantial area of the search space, although additional refinement could be beneficial for MOPO to achieve the optimal outcome.

The strong performance of FC-MOEO/AEP may be attributed to the monotonic nature of the self-consumption and self-sufficiency rates set as objectives. Thus, simulations based on FC-MOEO/AEP solutions indicated that allowing participants to use surplus energy generated by other members reduces grid interactions and associated emissions, especially when the energy storage is shared.

These findings highlight the need for optimisation approaches that align with the specific characteristics of resource management in energy communities, particularly as the adoption of renewable energy in residential areas continues to grow.

A key limitation of this study is the restricted size of the dataset, which includes only three participants. While our objective is to work with larger communities, the specific type of detailed energy data required is insufficient. To our knowledge, no publicly available dataset meets these requirements at a larger scale. Expanding the model to communities with more participants would be ideal to capture a wider range of energy consumption patterns and improve the model's capabilities.

Our proposed adaptation, MOPO, shows generally competitive HV values, often falling within a close range of the best-performing algorithms, MOPSO and FC-MOEO/AEP. While MOPO can discover solutions that adequately cover a substantial area of the search space, its exploitation capability is currently limited compared to MOPSO and FC-MOEO/AEP, indicating that additional refinement could further improve its performance and help achieve the optimal outcome.

Future work might focus on introducing constraints to limit the decision space during the optimisation process and incorporating additional objectives to avoid infeasible solutions while effectively managing the trade-off between grid independence and the autonomy of individual participants. Furthermore, we suggest as future research considering new competitive methods, such as the adapted and proposed versions of PO, will be essential in enhancing the effectiveness of optimisation strategies in these contexts. Finally, we consider that future research could explore the integration of real-world data. Such an approach would require careful consideration of the data's characteristics and variability. It would lead to an interesting understanding of community behaviours and energy usage particularities that are only present in real data.

6 Abbreviations

EO	Equilibrium Optimiser
FC-MOEO/AEP	Fast Convergence Multi-Objective Equilibrium Optimiser with Archive Evolution Path
HV	Hypervolume
LCOE	Levelized Cost of Energy
MOMPA	Multi-Objective Marine Predators Algorithm
MOPO	Multi-Objective Political Optimiser
MOPSO	Multi-Objective Particle Swarm Optimisation
MPA	Marine Predators Algorithm
MS	Maximum Spread
NSWOA	Non-dominated Sorting Whale Optimisation Algorithm
PO	Political Optimiser
PSO	Particle Swarm Optimisation
SCR	Self-Consumption Rate
SP	Spacing
SSR	Self-Sufficiency Rate
TMY3	Typical Meteorological Year 3
WOA	Whale Optimisation Algorithm

7 Data availability statement

The data used in this study are publicly accessible and can be obtained from [\[33\]](#) and [\[35\]](#).

8 Competing interests

The authors declare no conflict of interest.

9 Author contributions

All authors contributed equally to this work.

10 Acknowledgements

We acknowledge financial support from Ministerio de Ciencia e Innovación (Spain) (Grant PID2020-112495RB-C21 funded by MCIN/AEI/10.13039/ 501100011033).

11 Bibliography

1. Energy, I., *Statistical Review of World Energy 2024*. 2024. [<https://www.energyinst.org/statistical-review>]
2. World Health, O., *COP26 Special Report on Climate Change and Health: The Health Argument for Climate Action*. 2021. [<https://apps.who.int/iris/bitstream/handle/10665/346168/9789240036727-eng.pdf>]
3. International Renewable Energy, A., *Renewable Energy Statistics 2024*. 2024. [<https://www.irena.org/Publications/2024/Jul/Renewable-energy-statistics-2024>]
4. Schram, W., A. Louwen, I. Lampropoulos, and W. van Sark, *Comparison of the Greenhouse Gas Emission Reduction Potential of Energy Communities*. *Energies*, 2019. **12**(23): p. 4440. [<http://dx.doi.org/10.3390/en12234440>]
5. Bauwens, T., D. Schraven, E. Drewing, J. Radtke, L. Holstenkamp, B. Gotchev, and Ö. Yildiz, *Conceptualizing community in energy systems: A systematic review of 183 definitions*. *Renewable and Sustainable Energy Reviews*, 2022. **156**: p. 111999. [<http://dx.doi.org/10.1016/j.rser.2021.111999>]
6. Gržanić, M., T. Capuder, N. Zhang, and W. Huang, *Prosumers as active market participants: A systematic review of evolution of opportunities, models and challenges*. *Renewable and Sustainable Energy Reviews*, 2022. **154**: p. 111859. [<http://dx.doi.org/10.1016/j.rser.2021.111859>]
7. Chen, L., G. Msigwa, M. Yang, A.I. Osman, S. Fawzy, D.W. Rooney, and P.-S. Yap, *Strategies to achieve a carbon neutral society: a review*. *Environmental Chemistry Letters*, 2022. **20**(4): p. 2277 – 2310. [<http://dx.doi.org/10.1007/s10311-022-01435-8>]
8. Battula, A.R., S. Vuddanti, and S.R. Salkuti, *Review of Energy Management System Approaches in Microgrids*. *Energies*, 2021. **14**(17). [<http://dx.doi.org/10.3390/en14175459>]

9. Gilani, M.A., A. Kazemi, and M. Ghasemi, *Distribution system resilience enhancement by microgrid formation considering distributed energy resources*. Energy, 2020. **191**. [<http://dx.doi.org/10.1016/j.energy.2019.116442>]
10. Rao, S.N.V.B., V.P.K. Yellapragada, K. Padma, D.J. Pradeep, C.P. Reddy, M. Amir, and S.S. Refaat, *Day-Ahead Load Demand Forecasting in Urban Community Cluster Microgrids Using Machine Learning Methods*. Energies, 2022. **15**(17). [<http://dx.doi.org/10.3390/en15176124>]
11. Dong, W., Q. Yang, X. Fang, and W. Ruan, *Adaptive optimal fuzzy logic based energy management in multi-energy microgrid considering operational uncertainties*. Applied Soft Computing, 2021. **98**: p. 106882. [<http://dx.doi.org/10.1016/j.asoc.2020.106882>]
12. Anoh, K., S. Maharjan, A. Ikpehai, Y. Zhang, and B. Adebisi, *Energy Peer-to-Peer Trading in Virtual Microgrids in Smart Grids: A Game-Theoretic Approach*. IEEE Transactions on Smart Grid, 2020. **11**(2): p. 1264 – 1275. [<http://dx.doi.org/10.1109/TSG.2019.2934830>]
13. Cielo, A., P. Margiaria, P. Lazzeroni, I. Mariuzzo, and M. Repetto, *Renewable Energy Communities business models under the 2020 Italian regulation*. Journal of Cleaner Production, 2021. **316**. [<http://dx.doi.org/10.1016/j.jclepro.2021.128217>]
14. Vecchi, F., U. Berardi, and G. Mutani, *Self-Sufficiency Building Energy Modelling from Urban to Block-Scale with PV Technology*. International Journal of Sustainable Development and Planning, 2023. **18**(8): p. 2309-2318. [<http://dx.doi.org/10.18280/ijstdp.180801>]
15. Todeschi, V., P. Marocco, G. Mutani, A. Lanzini, and M. Santarelli, *Towards Energy Self-Consumption and Self-Sufficiency in Urban Energy Communities*. International Journal of Heat and Technology, 2021. **39**(1): p. 1-11. [<http://dx.doi.org/10.18280/ijht.390101>]
16. Faria, J., J. Fermeiro, J. Pombo, M. Calado, and S. Mariano, *Proportional Resonant Current Control and Output-Filter Design Optimization for Grid-Tied Inverters Using Grey Wolf Optimizer*. Energies, 2020. **13**(8). [<http://dx.doi.org/10.3390/en13081923>]
17. Suresh, V., P. Janik, M. Jasinski, J.M. Guerrero, and Z. Leonowicz, *Microgrid energy management using metaheuristic optimization algorithms*. Applied Soft Computing, 2023. **134**: p. 109981. [<http://dx.doi.org/10.1016/j.asoc.2022.109981>]
18. Faria, J., C. Marques, J. Pombo, S. Mariano, and M.d.R. Calado, *Optimal Sizing of Renewable Energy Communities: A Multiple Swarms Multi-Objective Particle Swarm Optimization Approach*. Energies, 2023. **16**(21). [<http://dx.doi.org/10.3390/en16217227>]
19. Singh, D., N. Elgeberi, M. Aljaidi, R. Kumar, R.-E.-A. Mamlook, and M.-K. Singla, *Optimal Location of Renewable Energy Generators in Transmission and Distribution System of Deregulated Power Sector: A Review*. Energy Engineering, 2025. **122**(3): p. 823--859. [<http://dx.doi.org/10.32604/ee.2025.059309>]

20. Aljaidi, M., G. Samara, M.K. Singla, A. Alsarhan, M. Hassan, M. Safaraliev, P. Matrenin, and A. Tavlintsev, *A particle swarm optimizer-based optimization approach for locating electric vehicles charging stations in smart cities*. International Journal of Hydrogen Energy, 2024. **87**: p. 1047-1055. [<http://dx.doi.org/10.1016/j.ijhydene.2024.09.029>]
21. Bairappaka, S.K., A. Ghosh, O. Kaiwartya, M. Aljaidi, Y. Cao, and R. Kharel, *A novel design of broadband circularly polarized rectenna with enhanced gain for energy harvesting*. IEEE Access, 2024. **12**: p. 65583-65594. [<http://dx.doi.org/10.1109/ACCESS.2024.3397016>]
22. Aljaidi, M., N. Aslam, X. Chen, O. Kaiwartya, and M. Khalid. *An energy efficient strategy for assignment of electric vehicles to charging stations in urban environments*. in 2020 11th international conference on information and communication systems (ICICS). 2020. IEEE.
23. Amoussou, I., E. Tanyi, L. Fatma, T.F. Agajie, I. Boulkaibet, N. Khezami, A. Ali, and B. Khan, *The optimal design of a hybrid solar PV/wind/hydrogen/lithium battery for the replacement of a heavy fuel oil thermal power plant*. Sustainability, 2023. **15**(15): p. 11510. [<http://dx.doi.org/10.3390/su151511510>]
24. Xiao, Y., Y. Guo, H. Cui, Y. Wang, J. Li, and Y. Zhang, *IHAOAVOA: An improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems*. Mathematical Biosciences and Engineering, 2022. **19**(11): p. 10963-11017. [<http://dx.doi.org/10.3934/mbe.2022512>]
25. Xiao, Y., H. Cui, R.A. Khurma, and P.A. Castillo, *Artificial lemming algorithm: a novel bionic meta-heuristic technique for solving real-world engineering optimization problems*. Artificial Intelligence Review, 2025. **58**(3): p. 84. [<http://dx.doi.org/10.1007/s10462-024-11023-7>]
26. Xiao, Y., H. Cui, A.G. Hussien, and F.A. Hashim, *MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications*. Advanced Engineering Informatics, 2024. **61**: p. 102464. [<http://dx.doi.org/10.1016/j.aei.2024.102464>]
27. Sörensen, K., *Metaheuristics—the metaphor exposed*. International Transactions in Operational Research, 2015. **22**(1): p. 3-18. [<http://dx.doi.org/10.1111/itor.12001>]
28. Kennedy, J. and R. Eberhart, *Particle Swarm Optimization*, in *Proceedings of ICNN'95 - International Conference on Neural Networks*. 1995. p. 1942-1948.
29. Faramarzi, A., M. Heidarinejad, S. Mirjalili, and A.H. Gandomi, *Marine Predators Algorithm: A Nature-Inspired Metaheuristic*. Expert Systems with Applications, 2020. **152**. [<http://dx.doi.org/10.1016/j.eswa.2020.113377>]
30. Mirjalili, S. and A. Lewis, *The Whale Optimization Algorithm*. Advances in Engineering Software, 2016. **95**: p. 51-67. [<http://dx.doi.org/10.1016/j.advengsoft.2016.01.008>]
31. Faramarzi, A., M. Heidarinejad, B. Stephens, and S. Mirjalili, *Equilibrium Optimizer: A Novel Optimization Algorithm*. Knowledge-Based Systems, 2019. [<http://dx.doi.org/10.1016/j.knosys.2019.105190>]

32. Askari, Q., I. Younas, and M. Saeed, *Political Optimizer: A novel socio-inspired meta-heuristic for global optimization*. Knowledge-Based Systems, 2020. **195**: p. 105709. [<http://dx.doi.org/10.1016/j.knosys.2020.105709>]
33. Ong, S. and N. Clark, [dataset] *Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States*. 2014. [<http://dx.doi.org/10.25984/1788456>]
34. Zheng, S., G. Huang, and A.C.K. Lai, *Techno-economic performance analysis of synergistic energy sharing strategies for grid-connected prosumers with distributed battery storages*. Renewable Energy, 2021. **178**: p. 1261-1278. [<http://dx.doi.org/10.1016/j.renene.2021.06.100>]
35. National Renewable Energy, L., [dataset] *Typical Meteorological Year 3 (TMY3)*. 2005. [https://web.archive.org/web/20190805120215/https://rredc.nrel.gov/solar/old_data/nsrdb/1991-2005/tmy3/by_state_and_city.html]
36. Wilcox, S. and W. Marion, *User's Manual for TMY3 Data Sets*. 2008, National Renewable Energy Laboratory: Golden, Colorado.
37. Nawawi, S., M. Yi, M. Craig, T. Deetjen, and P. Vaishnav, *Cross-sectoral trade-offs in a changing climate: Surrogate models to balance home energy bills, occupant comfort, and power system externalities*. iScience, 2025. **28**(6). [[https://www.cell.com/iscience/fulltext/S2589-0042\(25\)00989-7](https://www.cell.com/iscience/fulltext/S2589-0042(25)00989-7)]
38. Zitzler, E., L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca, *Performance assessment of multiobjective optimizers: an analysis and review*. IEEE Transactions on Evolutionary Computation, 2003. **7**(2): p. 117-132. [<http://dx.doi.org/10.1109/TEVC.2003.810758>]
39. Deb, K. and H. Jain, *An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints*. IEEE Transactions on Evolutionary Computation, 2014. **18**(4): p. 577-601. [<http://dx.doi.org/10.1109/TEVC.2013.2281535>]
40. Coello Coello, C.A. and M.S. Lechuga, *MOPSO: a proposal for multiple objective particle swarm optimization*, in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. 2002. p. 1051-1056.
41. Chen, L., X. Cai, K. Jin, and Z. Tang, *MOMPA: a high performance multi-objective optimizer based on marine predator algorithm*, in *Proceedings of the genetic and evolutionary computation conference companion*. 2021. p. 177-178.
42. Jangir, P. and N. Jangir, *Non-Dominated Sorting Whale Optimization Algorithm (NSWOA): A Multi-Objective Optimization algorithm for Solving Engineering Design Problems*. Global Journal of Research in Engineering, 2017. **17**(F4): p. 15-42.
43. Ghazaan, M.I., P. Ghaderi, and A. Rezaeizadeh, *A fast convergence EO-based multi-objective optimization algorithm using archive evolution path and its application to engineering design problems*. Journal of Supercomputing, 2023. **79**: p. 18849-18885. [<http://dx.doi.org/10.1007/s11227-023-05362-5>]

44. Zitzler, E., K. Deb, and L. Thiele, *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*. Evolutionary Computation, 2000. **8**(2): p. 173-195. [<http://dx.doi.org/10.1162/106365600568202>]
45. Zitzler, E. and L. Thiele, *Multiobjective Optimization Using Evolutionary Algorithms -- A Comparative Case Study*, in *Parallel Problem Solving from Nature -- PPSN V*, A.E. Eiben, T. Bäck, M. Schoenauer, and H.P. Schwefel, Editors. 1998, Springer. p. 292-301.
46. Li, M. and X. Yao, *Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey*. ACM Computing Surveys, 2019. **52**(2). [<http://dx.doi.org/10.1145/3300148>]
47. Chen, G., L. Liu, P. Song, and Y. Du, *Chaotic improved PSO-based multi-objective optimization for minimization of power losses and L index in power systems*. Energy Conversion and Management, 2014. **86**: p. 548-560. [<http://dx.doi.org/10.1016/j.enconman.2014.06.003>]
48. U.S. Energy Information Administration, *State electricity profiles*. 2022. [https://www.eia.gov/electricity/state/unitedstates/state_tables.php]