Equations for Multiple Methods in A Multi-Objective Optimization Implementation Towards Scalability of Distributed Green Hydrogen Systems in Urban Residential Communities

JUAN CAMILO RINCÓN AGUILAR, SERGIO RAUL RIVERA RODRIGUEZ
Department of Electrical and Electronic Engineering, Faculty of Engineering,
Universidad Nacional de Colombia,
COLOMBIA

Abstract: - This paper presents a methodology for finding scalable implementation paths for distributed green hydrogen systems in urban residential communities, based on a pre-2024-2025 study. Specifically, a multi-objective optimization approach is applied using the MOPSO (Multi- Objective Particle Swarm Optimization) algorithm, which efficiently addresses the many factors that influence the adoption of these systems. Furthermore, the research is extended to two other optimization methods: AGEII (Approximation-guided Evolutionary Multi-Objective Algorithm II) and FDV (Fuzzy Decision Variables Framework). These methods are compared in terms of their performance and effectiveness in generating optimal solutions. The results show that the AGE-II method significantly outperforms the other two methods, achieving more efficient and suitable solutions for the implementation of green hydrogen systems in urban environments. In this way, this study contributes to the discussion on the development of sustainable technologies and the transition to renewable energy.

Key-words: -Green hydrogen, residential communities, multi-objective optimization, algorithm, MOPSO, AGE-II, FDV, Pareto front.

Received: October 23, 2025. Revised: February 25, 2026. Accepted: March 28, 2026. Published: July 22, 2026.

1. Introduction

THE need to decarbonize urban residential communities has driven the search for innovative energy solutions that integrate renewable sources and flexible storage systems. In this context, green hydrogen systems are emerging as a promising alternative, combining rooftop photovoltaic panels with hybrid storage that integrates batteries and hydrogen storage. The study developed by Wu and Zhong [1] proposes a bottom-up energy model that comprehensively links climate uncertainties, human behaviors, and community characteristics to evaluate the deployment trajectories of these systems in various North American climate zones between 2030 and 2050. The results indicate that, under similar community conditions, the choice of deployment trajectory can lead to cost differences of up to 60%. In particular, the centralized household-level program model offers significant advantages by leveraging scale effects to reduce system costs, especially when energy storage demand increases. Moreover, the study highlights that uncertainties on both the supply side (mainly affected by climate conditions) and the demand side (influenced by human behaviors and specific building characteristics) have a differential impact on system performance, making it critical to adjust the storage integration strategy to maximize energy autonomy and reduce grid dependency.

This paper explicitly builds upon the conceptual advances and underlying energy system models presented in related works such as [1] and [2], which establish the foundational framework for distributed green hydrogen systems. **The distinct novelty of this submitted work lies not in the development of a new energy system model, but in its rigorous and unprecedented comparative methodological analysis of three advanced multi-objective optimization algorithms: MOPSO, AGE-II, and FDV.** Our research meticulously applies and tailors these algorithms to the complex sizing problem of green hydrogen systems in urban residential communities in Colombia. The primary contribution is the identification of the most effective algorithmic approach (AGE-II) for this challenging context, demonstrating its superior performance in generating robust and efficient solutions, thereby offering a crucial methodological tool for informed decision-making in future energy projects.

This approach is particularly relevant in the context of Colombian energy policy. Recently, the Ministry of Mines and Energy has established the regulatory framework for the creation of Energy Communities, a figure that empowers citizens to become 'prosumers,' capable of generating, managing, and even commercializing their own energy within communities [3]. While this regulation creates an opportunity, it leaves open the question of how to design and optimize these distributed systems to ensure their technical and economic viability. This work directly addresses that challenge, offering a methodological tool for these communities to make informed decisions about the sizing and operation of their infrastructure, thus aligning technological innovation with the goals of the national energy transition [4].

This document presents a methodology for defining implementation paths oriented towards the scalability of distributed green hydrogen systems in urban residential communities, building on previous conceptual advances such as those by Wu and Zhong [1] and [2]. In particular, a multi-objective optimization approach based on the MOPSO (Multi-Objective Particle Swarm Optimization) algorithm is proposed and applied, chosen for its ability to efficiently address the multiple factors—technical, economic, and social—that influence the adoption of these systems. The research is also extended by comparing the performance and effectiveness of MOPSO with two other multi-objective optimization methods: AGE-II (Approximation-guided Evolutionary Multi-objective Algorithm II) and FDV (Fuzzy Decision Variable Framework), to compare them in terms of their performance and effectiveness in generating suitable solutions [5].

2. Methodology

2.1 Optimization Algorithms

The methodology is based on the implementation of the optimization algorithms MOPSO, AGE-II, and FDV in MATLAB. The code for these algorithms comes from the PlatEMO (Evolutionary Multi-Objective Optimization Platform) [6]. The goal is to evaluate the performance of each algorithm by comparing them and thus find the most effective one. Figure 2 (the flow diagram illustrating the optimization process) is presented later in this section. Below are the main characteristics of each algorithm [4].

1) MOPSO (Multi-Objective Particle Swarm Optimization): The MOPSO algorithm is based on the particle swarm optimization (PSO) technique to solve problems with multiple objectives simultaneously. Instead of seeking a single best value, MOPSO explores how to improve multiple criteria at the same time, building a set of solutions that represent different trade-offs between those objectives (Pareto front) [7]. It works by iterating several cycles in which it compares each possible solution, evaluating the objectives and assigning dominance among them. This cycle is repeated until a number of iterations or a stopping criterion is met [8].

Key Equations for MOPSO

1. **Particle Velocity Update**: The velocity of each particle in the swarm is updated based on its current velocity, its personal best position, and the global best position. The velocity update equation is given by [9]:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t)), \quad (1)$$

where:

- $v_i(t)$ is the velocity of particle i at iteration t,
- w is the inertia weight,
- c_1 and c_2 are acceleration coefficients,
- r_1 and r_2 are random numbers in the range $[0, 1]$,
- $pbest_i$ is the personal best position of particle i.
- $gbest$ is the global best position in the swarm,
- $x_i(t)$ is the current position of particle i.

2. **Particle Position Update**: The position of each particle is updated using the velocity update equation. The position update is given by:

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (2)$$

3. **Objective Function Evaluation**: For each particle, the objective functions are evaluated to determine its fitness. The objective function for a multi-objective problem is given by:

$$F(x_i) = [f_1(x_i), f_2(x_i), \ldots, f_M(x_i)], \quad (3)$$

where M is the number of objectives, and $f_j(x_i)$ is the j-th objective function for particle i.

4. **Pareto Dominance**: A solution x_1 is said to dominate another solution x_2. This dominance criterion is formally defined as: x_1 dominates x_2 if $\forall j, f_j(x_1) \leq f_j(x_2)$ and $\exists j, f_j(x_1) < f_j(x_2)$.

This dominance criterion is used to update the Pareto front, which contains all non-dominated solutions.

5. **Global Best Selection**: The global best position $gbest$ is selected from the non-dominated solutions in the Pareto front. This selection is based on a crowding distance or other diversity-preserving mechanisms to ensure a well-distributed Pareto front.

6. **Stopping Criterion**: The algorithm terminates when a stopping criterion is met, such as reaching a maximum number of iterations T_{max}:

$$t \geq T_{max}. \quad (4)$$

Explanation of MOPSO Process

1) **Initialization**: Particles are initialized with random positions and velocities.
2) **Evaluation**: The objective functions are evaluated for each particle.
3) **Pareto Front Update**: Non-dominated solutions are added to the Pareto front.
4) **Velocity and Position Update**: Particles move in the search space based on their velocity and position updates.
5) **Global Best Selection**: The global best solution is selected from the Pareto front.
6) **Termination**: The algorithm stops when the stopping criterion is met.

2) AGE-II (Approximation-guided Evolutionary Multi-objective Algorithm II): AGE-II is an evolutionary algorithm designed to continuously improve the approximation to the Pareto front in multi-objective problems, explicitly controlling the quality of that approximation [10]. It introduces the idea of measuring the error or additive distance between the current approximation and an ideal version of the front, and uses that indicator to guide the evolution. In each cycle, it maintains a set of representative solutions of the front and calculates how much they reduce the global additive error. Then it selects the best candidates and adjusts them to maintain a balance between precision and cost.

Key Equations for AGE-II

1. **Additive Error (Distance)**: The additive error E measures the distance between the current approximation of the Pareto front P and the ideal Pareto front P^*. The additive error for a solution x is given by:

$$E(x) = \sum_{j=1}^{M} \max(0, f_j(x) - f_j^*(x)), \tag{5}$$

where:

- M is the number of objectives,
- $f_j(x)$ is the j-th objective value of solution x,
- $f_j^*(x)$ is the j-th objective value of the ideal solution x^*.

2. **Global Additive Error**: The global additive error E_{global} is the sum of the additive errors for all solutions in the current approximation P:

$$E_{global} = \sum_{x \in P} E(x). \tag{6}$$

3. **Approximation Quality**: The quality of the approximation is measured by how much the global additive error is reduced. The reduction in error ΔE is given by:

$$\Delta E = E_{global}(t) - E_{global}(t+1), \tag{7}$$

where $E_{global}(t)$ is the global additive error at iteration t, and $E_{global}(t+1)$ is the global additive error at iteration $t+1$.

4. **Selection of Candidates**: The algorithm selects the best candidates based on their contribution to reducing the global additive error. The selection criterion for a candidate x is:

$$\text{Selection}(x) =_{x \in P} \Delta E(x), \tag{8}$$

where $\Delta E(x)$ is the reduction in error contributed by solution x.

5. **Adjustment of Solutions**: The solutions are adjusted to maintain a balance between precision and cost. The adjustment is based on the following update rule:

$$x_{new} = x_{old} + \alpha \cdot \Delta x, \tag{9}$$

where:

- x_{new} is the adjusted solution,
- x_{old} is the current solution,
- α is a learning rate (step size), controlling the magnitude of the adjustment.
- Δx is the direction of adjustment, typically calculated based on the gradient of the objective functions or identified search direction for improvement.

6. **Stopping Criterion**: The algorithm terminates when the global additive error falls below a predefined threshold ϵ or when a maximum number of iterations T_{max} is reached:

$$E_{global} < \epsilon \quad \text{or} \quad t \geq T_{max}. \tag{10}$$

Explanation of AGE-II Process

1) **Initialization**: A population of solutions is initialized randomly.
2) **Evaluation**: The objective functions are evaluated for each solution.
3) **Additive Error Calculation**: The additive error $E(x)$ is calculated for each solution.
4) **Global Error Reduction**: The global additive error E_{global} is computed and reduced iteratively.
5) **Selection**: The best candidates are selected based on their contribution to reducing the global error.
6) **Adjustment**: The solutions are adjusted to improve the approximation quality.
7) **Termination**: The algorithm stops when the stopping criterion is met.

3) FDV (Fuzzy Decision Variable Framework): The FDV algorithm is a framework that consists of progressively reducing and reconstructing the search space in multi-objective problems with thousands of variables, using fuzzy logic concepts to guide when and how to group or translate decision variables [11]. It is characterized mainly by having two phases of work: a fuzzy phase and a precise phase. In the fuzzy phase, each decision variable is associated with promising fuzzy sets that are grouped and optimized to approach the Pareto region. While in the precise phase, the grouping is undone, and each variable is individually optimized so that, in each iteration, only those that contribute to improving the Pareto front survive.

Key Equations for FDV

1. **Fuzzy Membership Function**: In the fuzzy phase, each decision variable x_i is associated with a fuzzy set A_i defined by a membership function $\mu_{A_i}(x_i)$. The membership function is given by:

$$\mu_{A_i}(x_i) = \begin{cases} 1 & \text{if } x_i \in A_i, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

This function determines the degree to which a variable belongs to a fuzzy set.

2. **Fuzzy Aggregation**: The fuzzy sets are aggregated to form a combined fuzzy set A. The aggregation is performed using a fuzzy union operator, such as the maximum operator:

$$\mu_A(x) = \max_i \mu_{A_i}(x_i). \tag{12}$$

This combined fuzzy set represents the promising region of the search space.

3. **Fuzzy Optimization**: In the fuzzy phase, the optimization problem is reformulated to optimize over the fuzzy set A. The objective function is modified to include the membership function:

$$F(x) = [f_1(x) \cdot \mu_A(x), f_2(x) \cdot \mu_A(x), \ldots, f_M(x) \cdot \mu_A(x)], \tag{13}$$

where M is the number of objectives, and $f_j(x)$ is the j-th objective function.

4. **Precise Optimization**: In the precise phase, the grouping is undone, and each variable is optimized individually. The objective function is restored to its original form:

$$F(x) = [f_1(x), f_2(x), \ldots, f_M(x)]. \tag{14}$$

The precise optimization ensures that only the variables that contribute to improving the Pareto front survive.

5. **Variable Selection**: The selection of variables for optimization is based on their contribution to the Pareto front. The selection criterion for a variable x_i is:

$$\text{Selection}(x_i) =_{x_i} \Delta F(x_i), \tag{15}$$

where $\Delta F(x_i)$ represents the computed improvement in the Pareto front's quality (e.g., hypervolume contribution or crowding distance change) attributable to optimizing variable x_i, specific to the FDV algorithm's internal heuristic for guiding local search.

6. **Stopping Criterion**: The algorithm terminates when the improvement in the Pareto front falls below a predefined threshold ϵ or when a maximum number of iterations T_{max} is reached:

$$\Delta F < \epsilon \quad \text{or} \quad t \geq T_{max}. \tag{16}$$

2.2 Problem Configuration

The underlying energy system model, including its operational logic and scenarios (Case 1 for energy surpluses and Case 2 for energy deficits), is fundamentally aligned with the modeling framework presented in previous conceptual advances by Wu and Zhong [1] and [2], and related studies. This paper adapts this proven framework for the specific context of Colombian urban residential communities to allow for a direct comparative analysis of optimization algorithms.

The optimization was performed considering two main objectives ($obj.M = 2$): minimizing the total system cost and minimizing grid interaction, promoting self-sufficiency. The problem is configured under the following decision variables ($obj.D = 5$):

- Maximum electrolyzer capacity: $pelemax$
- Maximum fuel cell capacity: $pfcmax$
- Hydrogen tank capacity: $lohcap$
- Battery capacity: $soccap$
- Number of photovoltaic panels: $pvin$

The model will be implemented for a hypothetical number of households of 50 (participating family number $pfn = 50$), which is the number of families that make up this energy community. For each of the 5 decision variables, the algorithm sets minimum and maximum allowed values, $lower\_bound$ and $upper\_bound$, where it will search for solutions only within these ranges.

2.3 Input Data for the Energy Community

The energy demand of the community is based on an Excel file called *DATOS_COMUNIDAD*, which contains two important parameters: on one hand, information about the demand of two typical household profiles (one more intensive than the other) for the 24 hours of the day over a one-year period; and on the other hand, two profiles of electrical energy generation from the photovoltaic system, for the same period of time. As mentioned earlier, the number of households established within the simulation is 50, but instead of entering 50 consumption and photovoltaic generation profiles as input data, the code scales these two profiles and assumes them as the representative behavior of the entire community.

For this simulation exercise, the data obtained from the demand and photovoltaic generation profiles of the households were obtained using artificial intelligence tools, approximating the data to a city like Bogotá.

2.4 Objective Function

The objective function is calculated based on the energy management principles derived from previous conceptual advances in distributed green hydrogen systems. This involves defining the system's operational responses to different energy balance conditions.

$$\text{Minimize } \boldsymbol{F(x)} = [\boldsymbol{f_1(x)}, \boldsymbol{f_2(x)}] \tag{17}$$

Where $f_1(x)$ and $f_2(x)$ represent the two main objectives mentioned earlier ($obj.M = 2$), and are calculated as:

$$\begin{aligned} f_1(x) = CRF \cdot (&425 \cdot 1.0.1 \cdot \text{pv} + 205 \cdot 1.01 \cdot \text{SOCcapin} \\ &+ 6.5 \cdot 1.02 \cdot \text{LOHcapin} + 3200 \cdot 1.02 \cdot \text{pfcmaxin} \\ &+ 584 \cdot 1.02 \cdot \text{pfcmaxin} + 710 \cdot 1.02 \cdot \text{pelemaxin} \\ &+ 1087 \cdot 1.02 \cdot \text{pelemaxin}) + \text{gridint}, \end{aligned} \tag{18}$$

$$f_2(x) = \text{tgi} \tag{19}$$

The underlying logic for this objective function involves managing energy within the community, considering two primary operational conditions for energy flow:

- **Scenario 1: Energy Surpluses** When the system produces more energy than it consumes, the surplus is first directed to the battery until it reaches its full capacity or maximum charge rate. If excess energy still remains, it is utilized to produce hydrogen via the electrolyzer for storage in the tank. Should both battery and electrolyzer reach their maximum capacities, any further surplus energy is sold to the main grid.
- **Scenario 2: Energy Deficits** When energy demand exceeds local generation, the system first draws power from the battery to cover the deficit. If the battery cannot meet the demand, stored hydrogen is converted back to electricity using the fuel cell. If neither the battery nor the fuel cell can fully cover the demand, the remaining deficit is purchased from the main grid.

Equation	Description
$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t))$	Velocity update for particle i.
$x_i(t+1) = x_i(t) + v_i(t+1)$	Position update for particle i.
$F(x_i) = [f_1(x_i), f_2(x_i), \ldots, f_M(x_i)]$	Multi-objective function for particle i.
$\forall j, f_j(x_1) \leq f_j(x_2)$ and $\exists j, f_j(x_1) < f_j(x_2)$	Pareto dominance criterion (x_1 dominates x_2).
$t \geq T_{max}$	Stopping criterion based on maximum iterations.

TABLE I
KEY EQUATIONS IN MOPSO

Equation	Description
$E(x) = \sum_{j=1}^{M} \max(0, f_j(x) - f_j^*(x))$	Additive error for solution x.
$E_{global} = \sum_{x \in P} E(x)$	Global additive error for the current approximation P.
$\Delta E = E_{global}(t) - E_{global}(t+1)$	Reduction in global additive error.
Selection$(x) =_{x \in P} \Delta E(x)$	Selection criterion for candidate x.
$x_{new} = x_{old} + \alpha \cdot \Delta x$	Adjustment of solution x.
$E_{global} < \epsilon$ or $t \geq T_{max}$	Stopping criterion for the algorithm.

TABLE II
KEY EQUATIONS IN AGE-II

$$\text{PV1} = \begin{cases} \text{energydata.pv}(2, 2:8761) \cdot (2 \cdot \text{pvin} \\ -\sum \text{energydata.pv}(1, 2:\text{pvin}+1)) \\ +\text{energydata.pv}(3, 2:8761) \cdot \\ (\sum \text{energydata.pv}(1, 2:\text{pvin}+1) - \text{pvin}), & \text{if } \sum \text{energydata.pv}(1, 2:\text{pvin}+1) > \text{pvin} \\ \\ \text{energydata.pv}(2, 2:8761) \cdot \text{pvin}, & \text{otherwise} \end{cases} \tag{20}$$

The energy model is based on the following equations:

a) Solar Energy Production: The energy production of the photovoltaic system is given by equation 20.

b) Energy Balance:

$$\text{pnet}(i) = \begin{cases} \text{Supply}(i) - \text{Load1}(i), & \text{if Supply}(i) > \text{Load1}(i) \\ \text{Load1}(i) - \text{Supply}(i), & \text{otherwise.} \end{cases} \tag{21}$$

c) Battery State of Charge (SOC):

$$\text{SOC}(i+1) = \text{SOC}(i) + \eta \cdot \text{pbatch}(i) - \frac{\text{pbatdis}(i)}{\eta}, \tag{22}$$

where η is the battery efficiency.

d) Hydrogen Tank State (LOH):

$$\text{LOH}(i+1) = \text{LOH}(i) + \eta_1 \cdot \text{pele}(i) - \eta_2 \cdot \text{pfc}(i), \tag{23}$$

where η_1 and η_2 are the efficiencies of the electrolyzer and the fuel cell, respectively.

e) Grid Interaction:

$$\begin{aligned} \text{pgridsell}(i) &= \text{pnet}(i) - \text{pele}(i) - \text{pbatch}(i) \\ \text{pgridbuy}(i) &= \text{pnet}(i) - \text{pfc}(i) - \text{pbatdis}(i) \end{aligned} \tag{24}$$

3. Results

The results obtained by the model were analyzed in a specific region of convergence of interest: $[3e5\ 6e5]$. These results indicated that the AGE-II method achieved more efficient solutions than the MOPSO and FDV methods. The primary performance metric used was the **Inverted Generational Distance (IGD)**, which is a widely accepted performance index to evaluate both the quality and convergence of the solutions, reflecting an improvement in optimization. Figure 1 shows the resulting Pareto Front for the three implemented methods.

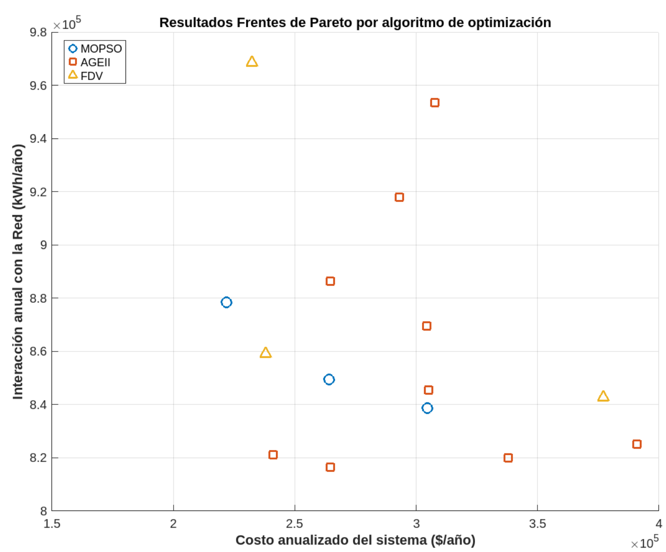

Fig. 1. Comparison of results from the different optimization methods.

The following IGD scores were obtained by the simulation program, confirming AGE-II's superior performance:

- **MOPSO Score (IGD): 238681,18**
- **AGE-II Score (IGD): 219341,17**
- **FDV Score (IGD): 254664,39**

4. Discussion

The most immediate and critical finding from the results is the significant difference in performance observed among the evaluated algorithms. Both the visual analysis of the Pareto fronts presented in Figure 1 and the numerical IGD scores unequivocally indicate that the AGE-II and MOPSO algorithms substantially outperform FDV for this particular problem. Among these, AGE-II distinctly emerges as the most effective method, achieving the lowest IGD score (219341,17). This indicates that AGE-II was able to locate a design solution significantly closer to the ideal reference point compared to its competitors. This result strongly suggests that for high-complexity problems, such as the sizing of green hydrogen energy systems that involve continuous variables and a complex multi-objective surface, modern evolutionary algorithms like AGE-II, which utilize an approximation-guided search mechanism, are more effective in exploring the solution space than classical metaheuristics or FDV. Furthermore, the Pareto fronts obtained confirm the fundamental conflict between the two objectives: minimizing the annualized system cost and achieving greater grid independence. For a community, this trade-off means there is no single optimal solution; instead, a range of optimal options exists. For example, by analyzing the AGE-II front (shown in red in Figure 1), a community might choose a solution with a low annual cost of approximately 2.6×10^5 /year, accepting a correspondingly higher grid interaction of almost $8.9 \times 10^5\,\mathrm{kWh/year}$. This highlights that our proposed methodology serves not only as an optimization tool but also as a crucial decision-support framework, enabling stakeholders to visualize and negotiate the trade-offs that best align with their specific economic priorities and strategic goals. These findings are particularly relevant in the context of the recent regulation of Energy Communities in Colombia. The results demonstrate that the choice of optimization tool is not a minor detail: it can have a direct and substantial impact on the viability of a project. The difference in IGD scores between AGE-II and MOPSO indicates that a community could end up with a significantly more costly or less autonomous system design simply by using a suboptimal optimization methodology. Therefore, this work not only validates a green hydrogen system model but also offers a crucial methodological recommendation for future energy communities seeking to implement technological solutions efficiently and cost-effectively.

5. Conclusions

The selection of the optimization algorithm is a critical factor for the efficient design of green hydrogen systems in residential communities. It was demonstrated that the AGE-II algorithm significantly outperformed MOPSO and FDV, achieving solutions that offer a better balance between the annualized system cost and energy autonomy. This finding underscores that the adoption of modern evolutionary algorithms can translate into substantial improvements in the technical and economic viability of these projects. The methodology presented not only allows for the optimization of energy infrastructure sizing but also functions as a decision-support tool for future Energy Communities in Colombia and other regions. Additionally, it opens a path for future research that integrates advanced optimization methods into the planning of sustainable energy systems.

Appendix

In this appendix, the MATLAB code used to perform the proposed simulations is presented.

The following is the general execution code for the entire algorithm:

```
rng(0);
[Dec,Obj,Con]=platemo('problem',@BT2,'
    algorithm',@MOPSO,'N',10,'maxFE',10,'
    save',1);
PopObj=Obj;
Result1=PopObj;
[Dec,Obj,Con]=platemo('problem',@BT2,'
    algorithm',@AGEII,'N',10,'maxFE',10,'
    save',1);
PopObj=Obj;
Result2=PopObj;
[Dec,Obj,Con]=platemo('problem',@BT2,'
    algorithm',@FDV,'N',10,'maxFE',10,'
    save',1);
PopObj=Obj;
Result3=PopObj;
methods = {'MOPSO', 'AGEII', 'FDV'};
dataCellArray = {Result1, Result2,
    Result3};
plotOptimizationResults(dataCellArray,
    methods)
PopObj=Obj;

%% SCORE FOR EACH METHOD
reference_point = [3e5 6e5];

score_MOPSO = mean(min(pdist2(
    reference_point, Result1), [], 2));
score_AGEII = mean(min(pdist2(
    reference_point, Result2), [], 2));
score_FDV = mean(min(pdist2(
    reference_point, Result3), [], 2));

fprintf('\n--- Performance Scores ---\n')
    ;
fprintf('Score for MOPSO: %.2f\n',
    score_MOPSO);
fprintf('Score for AGEII: %.2f\n',
    score_AGEII);
fprintf('Score for FDV:   %.2f\n',
    score_FDV);
```

The following code defines and executes the problem:

```
%% Default settings of the problem
classdef BT2 < PROBLEM
methods
function Setting(obj)
```

DOI: 10.37394/232021.2026.6.2


```
obj.M = 2;
obj.D = 5;
city=50;
pfn=city;
pelemax=600;
pelemin=20;
pfcmax=600;
pfcmin=20;
lohcapmax=800*pfn;
lohcapmin=10;
soccapmax=2000;
soccapmin=10;
ic1 = [0.5,2.5]; % Acceleration factor
iw = [0.5 0.001]; % Inertial factor
max_iter =10; % Maximum number of
    iterations
for n=1:5
if n==1
lower_bound(n)=pelemin;
upper_bound(n) =pelemax;
end
if n==2
lower_bound(n)=pfcmin;
upper_bound(n) =pfcmax;
end
if n==3
lower_bound(n)=lohcapmin;
upper_bound(n)=lohcapmax;
end
if  n==4
upper_bound(n) = soccapmax;
end
if n==5   % transfer pfn to problem
lower_bound(n)=pfn;
upper_bound(n) =pfn;
end
end
obj.lower    = lower_bound;
obj.upper    = upper_bound;
obj.encoding = ones(1,obj.D);
end

%% Calculate objective values
function PopObj = CalObj(obj,X)
datafile=['DATOS_COMUNIDAD.xlsx'] %
    energy data
load_table = readtable(datafile, 'Sheet',
     'load');
pv_table = readtable(datafile, 'Sheet', '
    pv');

energydata.load = table2array(load_table
    (:, 2:end));
energydata.pv = table2array(pv_table(:,
    2:end));

for hh=1:10
x=X(hh,:);
gridbuyprice=0.1775; % grid buy price
gridsellprice=0.7*gridbuyprice;%
    community sell price

pvin=x(5);
Load1=sum(energydata.load(2:3, 2:end), 1)
    ;

if sum(energydata.pv(1,2:pvin+1))>pvin
pv=(energydata.pv(4,1)*(2*pvin-sum(
    energydata.pv(1,2:pvin+1)))+energydata
    .pv(4,2)*(sum(energydata.pv(1,2:pvin
    +1))-pvin));
PV1=energydata.pv(2,2:end)*(2*pvin-sum(
    energydata.pv(1,2:pvin+1)))+energydata
    .pv(3,2:end)*(sum(energydata.pv(1,2:
    pvin+1))-pvin);
else
pv=130*pvin;
PV1=energydata.pv(2,2:end)*pvin;
end
Supply=PV1; % Energy production
pelemax=x(1);
pfcmax=x(2);
pelemaxin=x(1);
pfcmaxin=x(2);
LOHcapin=x(3);
SOCcapin=x(4);

pbatmax=x(4)*0.3;
n=0.9;     % bat efficiency
SOCmin=x(4)*0.1;
SOCmax=x(4)*0.9;
LOHmin=x(3)*0.1;
LOHmax=x(3)*0.9;
SOC=[];
LOH=[];
SOC(1,1:length(PV1))=x(4)*0.1;
LOH(1,1:length(PV1))=x(3)*0.1;
pbatch(1,1:length(PV1))=0;
pbatdis(1,1:length(PV1))=0;
pele(1,1:length(PV1))=0;
pgridsell(1,1:length(PV1))=0;
pgridbuy(1,1:length(PV1))=0;
pfc(1,1:length(PV1))=0;
pnet(1,1:length(PV1))=0;
n2=1.176;
n1=0.1976;%0.1976
for  i=1:length(PV1)
if Supply(i)>Load1(i)
pnet(i)=Supply(i)-Load1(i);
if SOC(i)>= SOCmax
if LOH (i)>= LOHmax
pgridsell(i)= pnet(i);
else
pele(i)=pnet(i);
if pele(i)>pelemax
pele(i)=pelemax;
```

```
LOH(i+1)=LOH(i)+n1*pele(i);
if  LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i);
else
pgridsell(i)= pnet(i)-pele(i);
end
else
pele(i)=pnet(i);
LOH(i+1)=LOH(i)+n1*pele(i);
if  LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i);
end
end
end
else
if pnet(i)>pbatmax
pbatch(i) = pbatmax;
pele(i)=pnet(i)-pbatch(i);
SOC(i+1)=SOC(i)+n*pbatch(i);
if  SOC(i+1)> SOCmax
pbatch(i)=(SOCmax-SOC(i))/n;
pele(i)=pnet(i)-pbatch(i);
if pele(i)>pelemax
pele(i)=pelemax;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
else
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
end
else
if pele(i)>pelemax
pele(i)=pelemax;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
else
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
end
end
else
pbatch(i)=pnet(i);
SOC(i+1)=SOC(i)+n*pbatch(i);
if  SOC(i+1)> SOCmax
pbatch(i)=(SOCmax-SOC(i))/n;
pele(i)=pnet(i)-pbatch(i);
if pele(i)>pelemax
pele(i)=pelemax;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
else
LOH(i+1)=LOH(i)+n1*pele(i);
if LOH(i+1)> LOHmax
pele(i)=(LOHmax-LOH(i))/n1;
pgridsell(i)= pnet(i)-pele(i)-pbatch(i);
end
end
end
end
SOC(i+1)=SOC(i)+n*pbatch(i);
LOH(i+1)=LOH(i)+n1*pele(i);
else
pnet(i)=Load1(i)-Supply(i);
if SOC  (i)<SOCmin
if LOH(i)<LOHmin
pgridbuy(i)= pnet(i);
else
pfc(i)=pnet(i);
if pfc(i)>pfcmax
pfc(i)=pfcmax;
pgridbuy(i)= pnet(i)-pfc(i);
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pfc(i);
end
else
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2);
pgridbuy(i)= pnet(i)-pfc(i);
end
end
end
else
pbatdis(i)=pnet(i);
if pbatdis(i)>pbatmax
pbatdis(i)=pbatmax;
SOC(i+1)=SOC(i)-pbatdis(i)/n;
if SOC(i+1)<SOCmin
pbatdis(i)=(SOC(i)-SOCmin)*n;
pfc(i)= pnet(i)-pbatdis(i);
if pfc(i) > pfcmax
pfc(i)=pfcmax;
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
```

```
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
else
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
else
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
end
else
pfc(i)= pnet(i)-pbatdis(i);
if pfc(i) > pfcmax
pfc(i)=pfcmax;
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
else
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
else
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
end
end
else
SOC(i+1)=SOC(i)-pbatdis(i)/n;
if SOC(i+1)<SOCmin
pbatdis(i)=(SOC(i)-SOCmin)*n;
pfc(i)= pnet(i)-pbatdis(i);
if pfc(i) > pfcmax
pfc(i)=pfcmax;
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
else
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
else
LOH(i+1)=LOH(i)-n2*pfc(i);
if  LOH(i+1) < LOHmin
pfc(i)= (LOH(i)-LOHmin)/n2;
pgridbuy(i)= pnet(i)-pbatdis(i)-pfc(i);
end
end
end
end
end
SOC(i+1)=SOC(i)-pbatdis(i)/n;
LOH(i+1)=LOH(i)-n2*pfc(i);
end
end
if LOH(1,length(PV1)+1)>LOHmin
gridint=gridbuyprice*sum(pgridbuy)-
   gridsellprice*sum(pgridsell)-(LOH(1,
   length(PV1)+1)-LOHmin)/n2*
   gridsellprice;
else
gridint=gridbuyprice*sum(pgridbuy)-
   gridsellprice*sum(pgridsell);
end
tgi=sum(pgridbuy)+sum(pgridsell);
PP=zeros(8,0);
CRF=0.058*((1+0.058)^20)/(((1+0.058)^20)
   -1);
cvalue=CRF*425*1.01*pv+205*1.01*SOCcapin
   +6.5*1.02*LOHcapin+3200*1.02*pfcmaxn
   +584*1.02*pfcmaxin+710*1.02*pelemaxin
   +1087*1.02*pelemaxin)+gridi;
y(1)=cvalue;
%% objective 2
y(2) = tgi;
%% reasonable capacity of hydrogen tank
if max(LOH) < 0.9 * LOHmax
    c = 1;
else
    c = 0;
end
PopObj(hh, 1) = y(1);
PopObj(hh, 2) = y(2);
end
end
end
end
```

The following MATLAB function is used to plot the optimization results:

```
function plotOptimizationResults(
   dataCellArray, methods)
figure;
hold on;
colors = lines(length(dataCellArray));
markers = {'o', 's', '^'};  % circle,
   square, triangle
for i = 1:length(dataCellArray)
    data = dataCellArray{i};
    if size(data, 2) >= 2
        scatter(data(:, 1), data(:, 2),
           60, ...
            markers{i}, ...
            'DisplayName', methods{i},
               ...
            'MarkerEdgeColor', colors(i,
               :), ...
            'LineWidth', 1.5);
    else
        warning('The method %s does not
           have at least two objectives
           to plot.', methods{i});
```

```
    end
end
xlabel('Annualized System Cost ($/year)',
    'FontWeight', 'bold');
ylabel('Annual Grid Interaction (kWh/year
   )', 'FontWeight', 'bold');
title('Pareto Front Results by
   Optimization Algorithm', 'FontWeight',
    'bold');
legend('show', 'Location', 'best');
grid on;
hold off;
end
```

The following pseudo-code summarizes the optimization process:

Algorithm 1 Multi-Objective Optimization Algorithm

1: Initialize population with random solutions
2: Evaluate initial population
3: **while** stopping criterion not met **do**
4: Apply optimization algorithm (MOPSO, AGE-II, or FDV)
5: Evaluate new solutions
6: Update Pareto front
7: **end while**
8: Return Pareto front and optimal solutions

The following key equations are used in the optimization process:

$$\text{pnet}(i) = \begin{cases} \text{Supply}(i) - \text{Load1}(i), & \text{if Supply}(i) > \text{Load1}(i) \\ \text{Load1}(i) - \text{Supply}(i), & \text{otherwise.} \end{cases} \tag{25}$$

$$\text{SOC}(i+1) = \text{SOC}(i) + \eta \cdot \text{pbatch}(i) - \frac{\text{pbatdis}(i)}{\eta}, \tag{26}$$

$$\text{LOH}(i+1) = \text{LOH}(i) + \eta_1 \cdot \text{pele}(i) - \eta_2 \cdot \text{pfc}(i), \tag{27}$$

$$\text{pgridsell}(i) = \text{pnet}(i) - \text{pele}(i) - \text{pbatch}(i), \tag{28}$$

$$\text{pgridbuy}(i) = \text{pnet}(i) - \text{pfc}(i) - \text{pbatdis}(i). \tag{29}$$

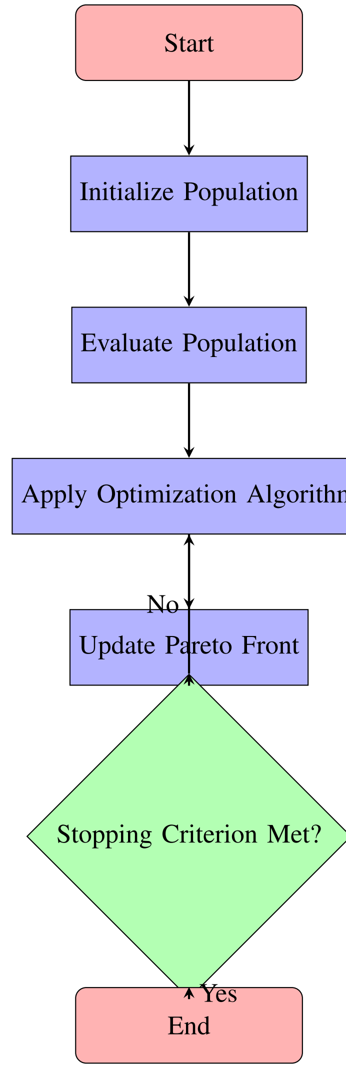

Fig. 2. Flow Diagram of the Optimization Process

References

[1] Y. Wu and L. Zhong, "Decarbonizing urban residential communities with green hydrogen systems," *Nature Cities*, vol. 2, pp. 81–92, 2025. [Online]. Available: https://doi.org/10.1038/s44284-024-00178-7
[2] L. Zhong and Y. Wu, "Probabilistic deployment pathways of scaling up distributed green hydrogen systems for urban residential communities in north america," 6 2024. [Online]. Available: https://www.researchsquare.com/article/rs-4327177/v1
[3] M. de Minas y Energía, "Resolución 40509 de 2024," 11 2024. [Online]. Available: https://normativame.minenergia.gov.co/normatividad/7079/norma/
[4] T. Valencia-Zuluaga, D. Agudelo-Martinez, D. Arango-Angarita, C. Acosta-Urrego, S. Rivera, D. Rodríguez-Medina, and J. Gers, "A fast decomposition method to solve a security-constrained optimal power flow (scopf) problem through constraint handling," *IEEE Access*, vol. 9, pp. 52 812–52 824, 2021.
[5] J. Garcia-Guarin, D. Alvarez, A. Bretas, and S. Rivera, "Schedule optimization in a smart microgrid considering demand response constraints," *Energies*, vol. 13, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/17/4567
[6] GitHub, "Platemo," 2025, accedido el 17 de Mayo de 2025. [Online]. Available: https://github.com/BIMK/PlatEMO
[7] C. Coello Coello and M. Lechuga, "Mopso: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2, 2002, pp. 1051–1056 vol.2.
[8] A. M. Sanchez, G. E. Coria, A. A. Romero, and S. R. Rivera, "An improved methodology for the hierarchical coordination of pev charging," *IEEE Access*, vol. 7, pp. 141 754–141 765, 2019.
[9] D. Baimbetov, M. Salem, S. Syrlybekkyzy, N. Ermek Talantovich, Y. Muralev, and M. Alhuyi Nazari, "Optimal scheduling of a multi-energy hub with renewables, hydrogen vehicles, and storage systems," *IEEE Access*, vol. 13, pp. 161 709–161 728, 2025.
[10] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 687–694. [Online]. Available: https://doi.org/10.1145/2463372.2463448
[11] X. Yang, J. Zou, S. Yang, J. Zheng, and Y. Liu, "A fuzzy decision variables framework for large-scale multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 445–459, 2023.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.