Energy & Buildings 297 (2023) 113433

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

A novel efficient multi-objective optimization algorithm for expensive building simulation models

Riccardo Albertin [a,*], Alessandro Prada [b], Andrea Gasparella [a]

[a] Faculty of Engineering, Free University of Bozen-Bolzano, Piazza Università 5, 39100 Bolzano, Italy
[b] Department of Civil, Environmental and Mechanical Engineering, University of Trento, Via Mesiano 77, 38123 Trento, Italy

ARTICLE INFO

Keywords:
Multi-objective optimization
Building performance optimization
Energy-efficient buildings
Bayesian optimization
Metamodeling

ABSTRACT

The energy design of a building is often an activity of finding trade-offs between several conflicting goals. However, a large number of expensive simulation runs is usually required to complete a Building Performance Optimization (BPO) process with a high confidence of the optimal solutions. Although evolutionary algorithms have been enhanced with surrogate models, complex BPO problems with many design variables still require a prohibitive number of expensive simulations, or lead to solutions with related low accuracy. Hence, performing multi-objective optimizations of actual building designs is still one of the most challenging problems in building energy design. A novel efficient multi-objective algorithm for expensive models based on a probabilistic approach is presented in this work. The new algorithm reduces the computational time needed for the optimization process, while increasing the quality of the solutions found. The algorithm was tested on the optimization problem of three groups of analytical test functions and on the BPO problem related to the refurbishment of three reference buildings. For the latter case, the efficiency, efficacy, and quality of the Pareto solutions found with the proposed algorithm were compared with the true Pareto front previously sought with a brute force approach. The results show that, for the most complex case among the three reference buildings, the algorithm can find about 50 % of the solutions on the true Pareto front with 100 % accuracy. In comparison, other algorithms tested on the same problem and with the same number of expensive simulations, are able to find at best 5 % of solutions on the true Pareto front with an accuracy around 5–10 %.

1. Introduction

The European Directive 2010/31/EU [1] mandates member states to reduce the energy demand of buildings in Europe through cost-effective design solutions known as the “cost-optimal approach.” However, achieving high-efficiency building refurbishment standards can sometimes lead to poor comfort conditions [2–3]. Therefore, designers need to consider additional performance aspects, resulting in multi-objective optimization problems with conflicting goals.

The number of research papers focusing on building performance optimization (BPO) has significantly increased [4]. Deterministic methods like linear or non-linear programming are not efficient for solving BPO problems due to the specific characteristics of building simulation models. Hence, three main categories of stochastic multi-objective algorithms are commonly used namely evolutionary algorithms (EA), swarm-based algorithms (PS) [5–6], and hybrid algorithms that combine both approaches. These techniques incorporate decision-making mechanisms based on probabilistic concepts. EA algorithms simulate natural evolution by using generations, parents, children, and mutations. The best solutions (parents) are selected to create the next set of solutions (children) in an iterative and probabilistic process. On the other hand, PS algorithms are inspired by biological systems and collective intelligence behavior. Stochastic algorithms are suitable for various optimization problems, including high-dimensional, integer or real parameters, and continuous or discrete variables. In the BPO process, a detailed building model, often referred to as an expensive model, is typically used for dynamic simulations and to collect new data necessary for the optimization process. This approach is known as simulation-based optimization. However, obtaining optimal solutions with high confidence requires a large number of expensive simulation runs, which can hinder the initial adoption of optimization in practice [7].

Several literature reviews focus on the optimization process for building design problems. Evins [8] provides a review of computational

[*] Corresponding author.
E-mail addresses: riccardo.albertin@natec.unibz.it (R. Albertin), alessandro.prada@unitn.it (A. Prada), andrea.gasparella@unibz.it (A. Gasparella).

https://doi.org/10.1016/j.enbuild.2023.113433
Received 18 April 2023; Received in revised form 17 July 2023; Accepted 6 August 2023
Available online 7 August 2023

optimization models for sustainable building design problems. Machairas et al. [4] note that the number of papers addressing building design optimization is still relatively small compared to building control optimization. Shi et al. [9] approach the same topic from an architect's perspective and find that 60 % of the works on building optimization utilize evolutionary algorithms. According to these reviews, one of the main challenges in BPO is the requirement for many expensive simulations before obtaining satisfactory results. Efficient optimization is crucial to find trade-off solutions in building design and refurbishment and to promote the practical application of BPO. However, an effective algorithm should also avoid premature convergence, which occurs when local optimal solutions (or dominated solutions) are selected instead of global optimal solutions. This limitation often arises from an incomplete analysis of the solution space.

Functional approximation is a common method to improve the optimization's efficiency while maintaining good accuracy. It consists of approximating the expensive model of a building with a mathematical function, hereafter named metamodel, which is then optimized by means of an optimization algorithm [10–17]. Among the different types of metamodels used as surrogate for building models, the most used are:

- Polynomial Regression, a popular metamodel that approximates the relationship between the input variables and the objective function or constraints. It is a simple and widely used approach, but it may struggle to capture complex nonlinear relationships [18–19].
- Kriging, known as Gaussian process regression, is a powerful metamodel that uses a stochastic process to model the objective function or constraint surface. It provides a flexible framework to capture complex and nonlinear relationships. Kriging models estimate the mean and covariance of the process using training data and generate predictions along with confidence intervals. Kriging is commonly used in engineering design optimization and simulation-based optimization problems [20–21].
- Radial Basis Functions (RBF) metamodels use a set of radial basis functions to approximate the objective function or constraints. These functions are centered at training points and their influence on the metamodel is defined by their shape and spread. RBF metamodels can capture both global and local behavior and have been successfully applied in various domains, including engineering design, finance, and environmental modeling [22–23].
- Support Vector Machines (SVM) is a well-known machine learning algorithm that can also be used as a metamodel for optimization problems. SVM constructs a hyperplane in a high-dimensional feature space that maximally separates the data points. In the context of metamodeling, SVM is trained on the input–output pairs generated by the original function and can approximate both linear and nonlinear relationships. SVM-based metamodels have been applied in various fields, including engineering design, finance, and computer science [24–25].
- Artificial neural networks (ANN) have gained increasing attention in recent years [26]. For instance, Kalogirou [27] combined an ANN with a GA to optimize a solar energy system, reducing the time required to find an optimal solution. Yizhe Xu et al. [28] also used an ANN coupled with an optimization algorithm to optimize the building envelope.

Metamodels, which are surrogate models, have been employed to search for optimal solutions by coupling them with either an EA or PS algorithm, and the results have been compared. In this case, the chosen algorithm is a Genetic Algorithm called NSGA-II [29]. Similarly, Yujun Jung et al. [30] successfully performed multi-objective optimization of a residential building by combining an ANN with NSGA-II, achieving accurate and efficient identification of optimal solutions. Other types of metamodels can be found in the literature. For example, Jianli Chen et al. [31] compared the performance of a Gaussian Process and Multiple Linear Regression in terms of computational time and reliability when applied to a calibration process.

Among the various strategies for integrating metamodels into the optimization process [32], a common approach involves fitting the metamodel to the available data (collected before the start of the optimization process) and then optimizing the metamodel using an EA to find a set of global solutions. These solutions are subsequently evaluated by the expensive model, and the results are added to the existing data set. The process is repeated iteratively until convergence is reached. The works of Xu et al. [17] and Brownlee and Wright [16] follow this strategy. Nonetheless, some challenges may arise regarding the number of design variables and/or the complexity of the metamodel, especially when it is a global approximation of the expensive model [33]. The higher the complexity of the costly model, the lower the accuracy with which the metamodel can approximate the model throughout the space of variables. For this, Knowles [34], applied the Chebyshev function to a multi-objective optimization problem guiding the selection of new offspring towards a portion of the objective space, thus avoiding potential issues associated with a globally approximate model. Nguyen et al. [35] analyzed works dealing with simulation-based optimization and concluded that future research should focus on enhancing the efficiency of search techniques.

Nevertheless, even an efficient algorithm may mistakenly identify local solutions as global or yield a set of solutions that do not significantly differ from each other during the optimization process. Not only does the algorithm evaluation consider the efficiency, but it takes into account also the efficacy and solution quality. Efficiency measures the computational cost of the algorithm, while efficacy quantifies the distance between the predicted Pareto front (the set of solutions found by the algorithm at the end of the optimization process) and the true Pareto solution (the global solutions to the optimization problem). Finally, solution quality assesses the uniformity of the Pareto front in the objective space.

Efficiency can be affected by the sampling technique chosen for the selection of the initial sample. Sobol Sequence Sampling (SSS) is among the most frequently used techniques. It is based on Sobol low-discrepancy sequences designed to cover the entire search space more evenly compared to random sampling methods. They have a deterministic and quasi-random nature and are well-suited for high-dimensional problems. The Latin Hypercube Sampling (LHS) is also a widely used method to determine the initial population. LHS is a stratified sampling method that ensures a more even coverage of the search space compared to simple random sampling. Finally, the Simple Random Sampling (SRS) is a basic and widely used sampling technique where each candidate solution is selected independently and with equal probability from the search space. Although it lacks the systematic coverage of LHS or Sobol sequences, simple random sampling is computationally efficient and easy to implement.

The comparison of optimization algorithms in BPO processes is often challenging due to the absence of known true solutions. As a result, different authors employ various methods to evaluate and compare algorithms. One common approach is to set a simulation budget and compare the optimal solutions obtained by each algorithm against the best solution found within that budget. Multiple simulation runs are often performed, and the best solution, along with the standard deviation or average solution, is reported. Alternatively, some authors compare algorithms based on the number of iterations required for a solution to reach a tolerance error compared to a reference value. Nevertheless, there is no guarantee that algorithms will find any optimal solution within a finite number of iterations when algorithms are evaluated using noisy estimates of solutions, as. This issue can impact the conclusions drawn when comparing two or more suboptimal solutions and may lead to incorrect assessments of the final performance of algorithms applied to BPO problems, as highlighted by Kämpf et al. [36].

The nature of the parameters influencing building optimization often lends itself to discrete optimization [37]. While geometric building features are typically continuous, most other parameters have discrete commercial options. This characteristic enables the evaluation of the true solution in BPO through a brute force approach, where all possible combinations of discrete options are exhaustively evaluated.

The literature review shows that there is room for improvement in the efficiency, effectiveness and quality of optimization algorithms, especially when employed in the optimization of expensive functions.

This paper presents a new efficient algorithm for the multi-objective optimization process of expensive models based on the local approximation method (i.e., the metamodel replaces the expensive model only for a portion of the objective space). The novelty in this work is achieved through a combination of different approaches. Firstly, a metamodel is employed to search for optimal solutions on behalf of the expensive models. This helps in simulating only the most promising solutions within the search space using the expensive models. The fitting process of the metamodel is based on a Bayesian probabilistic approach. This ensures a high-quality fitting process by fitting a multi-variate polynomial on each iteration without restricting the search to specific zones within the design variables space. Secondly, the metamodel fitting process changes its strategy as the optimization process progresses. Instead of using all available points, only the most promising ones are utilized. This narrows down the search specifically to the Pareto front zone within the design variable space, thereby improving approximation accuracy and training efficiency. Finally, the quality of the fitting process is greatly enhanced by moving from a multi-objective optimization to a single objective optimization. Consequently, the number of expensive simulations performed with design variables that are far from the Pareto front is significantly reduced. The main objective of this work is to reduce the computational time needed for the optimization process to be carried out, while increasing the quality of the solutions found. Considering that the algorithm’s runtime is negligible compared to the time needed for an expensive simulation, reducing the number of such simulations directly leads to a decrease in optimization time. Consequently, the focus of the problem shifts towards minimizing the number of expensive simulations required by the process to achieve satisfactory results. The new algorithm is therefore tested on a series of multi-objective problems and the results compared to other optimization algorithms in terms of efficiency, efficacy and solution quality by means of several metrics. For the comparison process, the solutions of an integer optimization problem found by a brute force approach, regarding the refurbishment of three simplified reference buildings, is utilized as a reference for the evaluation of the algorithm’s results. In a previous contribution, the effectiveness of SSS, LHS and SRS sampling methods in creating diverse initial populations was examined [32]. Since the Latin Hypercube Sampling (LHS) method consistently outperformed the other approaches, it has been chosen as the only sampling strategy in this work. However, different population sizes are used to evaluate the sensitivity of the algorithm performance to the characteristics of the initial sampling.

2. Method

The new algorithm is a multi-objective genetic algorithm implemented in MATLAB to solve computationally expensive optimization problems. The following paragraph provides a brief overview of the working principles of the algorithm (Fig. 1).

Solving an optimization problem involves finding the optimal combination or configuration of variables (referred to as design variables) that leads to the desired objective function value, representing a specific aspect of system performance or behavior. In multi-objective optimization problems, multiple objective functions or performance aspects are considered simultaneously, resulting in multiple sets of optimal configurations for the design variables. These sets of optimal configurations form the Pareto front. Each set of values for the design variables can be seen as a point in the variable space R^{p+n}, where p is the number of design variables and n is the number of objective functions.

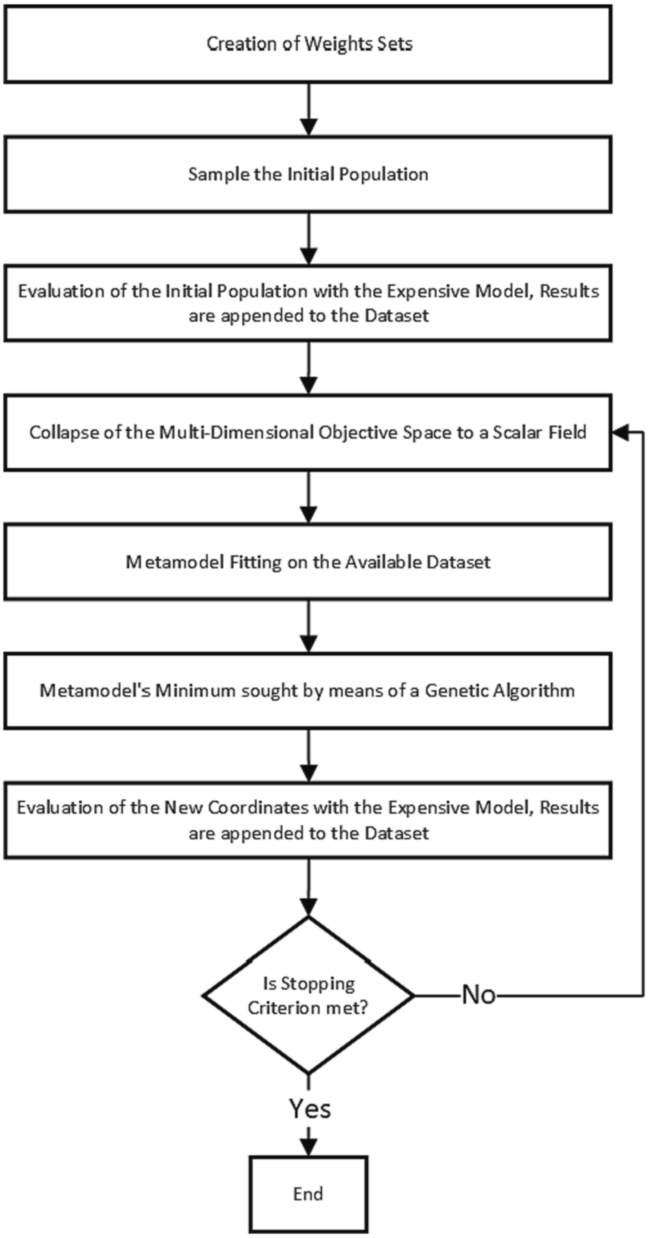
Creation of Weights Sets
Sample the Initial Population
Evaluation of the Initial Population with the Expensive Model, Results are appended to the Dataset
Collapse of the Multi-Dimensional Objective Space to a Scalar Field
Metamodel Fitting on the Available Dataset
Metamodel's Minimum sought by means of a Genetic Algorithm
Evaluation of the New Coordinates with the Expensive Model, Results are appended to the Dataset
Is Stopping Criterion met?
No
Yes
End

Fig. 1. Algorithm’s diagram representing the main phases of the optimization process.

The optimization process begins with the random sampling of an initial population of combinations of design variables, represented as a set of points in the variable space. Each point’s performance is evaluated using the expensive model, associating each point with a set of values corresponding to the objective functions. The evaluation process generates a dataset consisting of the points and their respective objective function values. At this stage, a matrix of weights is created and used to calculate the weighted mean value of the objective functions for the entire dataset during each iteration of the loop. By calculating the weighted mean value, the multi-objective optimization problem is transformed into a single-objective problem, simplifying the optimization process. Multiple sets of weights are generated randomly using Latin Hypercube Sampling (LHS) to explore the objective space of each objective function multiple times. The dataset obtained with the weighting process is then searched for points with a weighted mean value closer to the desired value, typically defined as the minimum or maximum. For sake of simplicity, optimization problems are considered as minimization problems from this point onward. Then, the points with the lowest weighted mean value are used to fit the metamodel. Since a different set of weights is used for the weighting process in each iteration of the loop, a new metamodel is created and fitted each time with a different set of points. This ensures a higher quality of the metamodel fitting process, as only the most suitable points are selected for its creation. The minimum of the metamodels is determined using a genetic algorithm (GA) at this stage. The values of the design variables associated with the minimum found by the GA are then used in the expensive simulation model to obtain the corresponding objective function values. Finally, the new point found by the GA and its related objective function values are added to the dataset. The algorithm checks if the iteration budget is exceeded and/or if the convergence criterion is met. If either condition is fulfilled, the main loop ends, and the Pareto front is evaluated.

2.1. Weights creation and sampling process

The sets of weights are created with an LHS process before the start of the main loop to investigate the entire objective space. This approach enables the dynamic prioritization of various objective functions during different stages, thereby addressing the potential issue of the algorithm persistently converging to identical solutions when similar weight values are used. Nevertheless, the weights for the objective functions are determined in a manner that ensures their sum always equals one.

In each iteration of the loop, each set is utilized to calculate the weighted mean value of the objective functions for the available dataset. To ensure compatibility with the maximum number of possible iterations, the total number of weights' sets must equal n_{imax}. Within each set, a weight is randomly generated for each objective function, n_{fobb}, using the Latin Hypercube Sampling (LHS) process. This weight is then applied to the entire dataset during the corresponding iteration.

Additionally, the weights are created in groups of n_{LHS} elements using the LHS process. Each group is designed to cover the entire objective space independently. This approach enables multiple searches for the Pareto front within the same optimization process. Consequently, a matrix of weights, denoted as W, is constructed. The matrix has a size of n_{imax} by n_{fobb}, obtained by concatenating submatrices of weights' sets in an iterative process. Each submatrix is derived through an adjusted LHS design process and has dimensions of n_{LHS} by n_{fobb}.

The ratio between n_{imax} and n_{LHS} defines how many times the Pareto front is searched for. The typical relationship between the two quantities is:

$$n_{LHS} = \frac{n_{imax}}{10} \tag{1}$$

this means that if no ending conditions is met during the process, the Pareto front is then searched for a total of 10 times (n_{imax}/n_{LHS}).

In a similar process, the Latin Hypercube Sampling (LHS) method is employed to sample the values of the design variables to create the initial population. These sampled values are stored in the matrix X. If any constraints exist, all points in the sampling set are checked. If a point violates any constraints, it is randomly replaced with another set of values. This replacement process continues until all points satisfies the constraints. Once the sampling set is constraint-compliant, it is evaluated using the expensive model. This evaluation results in the creation of a matrix Y which contains the values of the objective functions corresponding to each point in the sampling set. The sampled design variable points and their respective objective function values obtained during the sampling process are then combined to form the initial dataset, denoted as $D = [XY]$. The dataset matrix is not fixed in size, as in each iteration of the main loop, a new set of design variable values and their corresponding objective function values are appended to the dataset. This iterative process allows for the continuous expansion of the dataset throughout the optimization procedure.

2.2. Main loop

The main loop in the algorithm consists of several steps. Firstly, the weighted mean value of the objective functions is calculated by taking the scalar product of matrix Y and the corresponding row of weights from matrix W. This calculation is performed for each point in the dataset D. The resulting vector $\widehat{Y}$ contains the weighted mean values for each point in the search space R^p present in the dataset.

Next, a set of points in matrix X is chosen based on their related values in $\widehat{Y}$, sorted in descending order. The selected points are the top ones with the lowest weighted mean values, up to a predetermined number set before the optimization process begins. These selected points are then used for the metamodel fitting process. The metamodel fitted on the selected points during each iteration of the main loop is a multivariate polynomial. The polynomial fitting is performed using the Horseshoe method, a Bayesian approach [38].

Once the metamodel is fitted, a Genetic Algorithm (GA) is employed to search for the minimum of the metamodel. If there are constraints, the GA avoids the infeasible region of the search space by penalizing the associated objective functions. The coordinates of the minimum obtained from the GA are then checked. If these coordinates are not already present in the dataset, they are evaluated using the expensive simulation model, and the resulting objective function values are appended to the existing dataset. However, if the new solution is already in the dataset, the loop proceeds to the next iteration.

In some cases, when many design variables are integers, finding new coordinates that are not already in the dataset may require additional effort. As a solution, the algorithm can optionally search for a new minimum within the hammering distance: if a new solution found with the GA matches a solution in the dataset, a random design variable from the newly found solution is adjusted by adding or subtracting a relative step value. The sign (positive or negative) of the step value is randomly determined. The choice of using the hammering distance option is arbitrary and is decided during the algorithm initialization.

Finally, the ending conditions are checked. If none of the ending criteria are met, the loop proceeds to the next iteration, selecting a new set of weights, calculating the new objective mean values, and repeating the entire process. Once an ending condition is met, the algorithm stops, and the Pareto front is evaluated.

2.3. Convergence criteria

Ending conditions are utilized to interrupt the main loop and advance the algorithm to the next stage, that is the calculation and printing of the outputs. The code stops if it meets at least one of the three convergence criteria.

1. The first condition requires no new point is found on the Pareto front in n_{end} iterations (excluding the simulations of the sampling process). n_{end} is calculated with different equations dependently on whether the hammering distance option is enabled or not, to ensure that the entire Pareto front is searched at least once after the sampling process, with an arbitrary precision related to the choice of the parameter n_{LHS} (i.e., the number of weights' sets created within a LHS instance of the iterative process that leads to the weights' matrix W);

$$n_{end} = \frac{n_{LHS}}{10} \; if \; Hammering - distance \; is \; enabled \tag{2}$$

$$n_{end} = n_{LHS} \; if \; Hammering - distance \; is \; disabled \tag{3}$$

2. The second condition is met if all the objective functions' values are less than an arbitrary threshold, defined as the maximum acceptable value for each objective function
3. Finally, the last ending condition is related to the expensive simulation budget. The code stops if the number of expensive simulations

exceed the maximum number arbitrarily allocated before the start of the optimization process (i.e., n_{imax}). When this final condition is met, the algorithm displays a warning about the accuracy of the solutions found, which may not be satisfactory.

3. Testbench

The efficiency and efficacy of the proposed algorithm are assessed through evaluations on multiple multi-objective problems. Initially, the algorithm is tested on three analytical functions chosen from Deb's work [39]. These functions are specifically selected to include constraints, two objective functions, and a well-defined Pareto front that is densely populated. Furthermore, since other widely used and validated algorithms have been tested on the same optimization problems, it was possible to provide a comparison for the results found with the proposed algorithm.

While analytical benchmarking solutions do not always align with the characteristics of building simulation codes, which can result in misleading conclusions [36], the algorithm is further tested on a multi-objective optimization problem based on building simulation as described in Prada et al.'s study [32]. This problem involves discrete design variables and the non-linearity of the energy balance, which present additional challenges for the proposed algorithm [40].

This section introduces the analytical test functions as well as the case study from Prada et al.'s work [32]. Furthermore, the metrics used for comparing the algorithms are defined.

3.1. Analytical test cases

Three multi-objective test cases were selected for the testbench of the algorithm. The first test case is relative to the BNH test problem, from Binh and Korn [41]. This simple case does not have a practical application and consists of two objective functions (Equation (4) and (5), which must be minimized while also considering two constraints (Equation (6) and (7). The search space is defined by two continuous variables (x,y), which feasible ranges are set in Equation (8) and (9).

$$minimize\ f_1(x,y) = 4*\left(x^2+y^2\right) \tag{4}$$

$$minimize\ f_2(x,y) = (x-5)^2+(y-5)^2 \tag{5}$$

$$constraint\ C_1(x,y) = (x-5)^2+y^2 \leq 25 \tag{6}$$

$$constraintC_2(x,y) = (x-8)^2+(y+3)^2 \geq 7.7 \tag{7}$$

$$0 \leq x \leq 5 \tag{8}$$

$$0 \leq y \leq 3 \tag{9}$$

While the second constraint function is redundant as it is not in the feasible range of values for both variables x and y, the first constraint function does not eliminate any solution from the feasible search space, but rather reduce the density of feasible points present in it [39].

The second test case is relative to the Two-Bar truss design originally studied by Palli et al., [42]. In this case, three variables ($\bar{x}$, y) representing geometrical properties of a two-bar system (i.e., the cross-sectional area of the two bar and the vertical distance between the two ends of the bars respectively – Fig. 2) are used to minimize the volume of the truss (Equation (10) as well as the stresses along each of the two bars (Equation (11). Also in this case, the variables are continuous. The constraint function ensures that the truss does not go under elastic failure if a load specified in Equation (12) is applied to the system.

$$minimize\ f_1(\bar{x},y) = x_1\sqrt{16+y^2}+x_2\sqrt{1+y^2} \tag{10}$$

$$minimize\ f_2(\bar{x},y) = max(\sigma_1,\sigma_2) \tag{11}$$

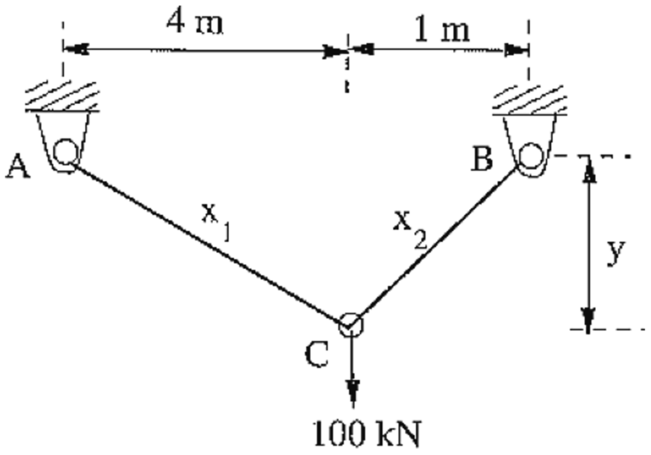

Fig. 2. Two-bar truss problem [39].

$$constraint\ C_1(\bar{x},y) = max(\sigma_1,\sigma_2) \leq 10^5 \tag{12}$$

$$\bar{x} \geq 0 \tag{13}$$

$$1 \leq y \leq 3 \tag{14}$$

$$with\sigma_1(x_1,y) = \frac{20\sqrt{16+y^2}}{yx_1};\sigma_2(x_2,y) = \frac{80\sqrt{1+y^2}}{yx_2} \tag{15}$$

The last test case utilized for the algorithm benchmark is relative to the Gear Train Design problem [43] (Fig. 3). The aim of the two objectives functions is to minimize the error given by the obtained gear ratio in relation to a required one of 1/6.931 and the dimensions of four gears, by changing their number of teeth. Given the nature of the problem, four variables are considered ($\bar{x} = x_a, x_b, x_d, x_f$), representing the number of teeth for each gear and so, admitting only integers values.

$$minimize\ f_1(\bar{x}) = \left(\frac{1}{6.931} - \frac{x_1x_2}{x_3x_4}\right)^2 \tag{16}$$

$$minimize\ f_2(\bar{x}) = max(x_1,x_2,x_3,x_4) \tag{17}$$

$$12 \leq \bar{x} \leq 60 \tag{18}$$

3.2. Building simulation optimizations

The optimization process for the refurbishment of three buildings was used as a secondary testbench to evaluate the proposed algorithm. In a previous study by Prada et al. [32] conducted in 2018, they

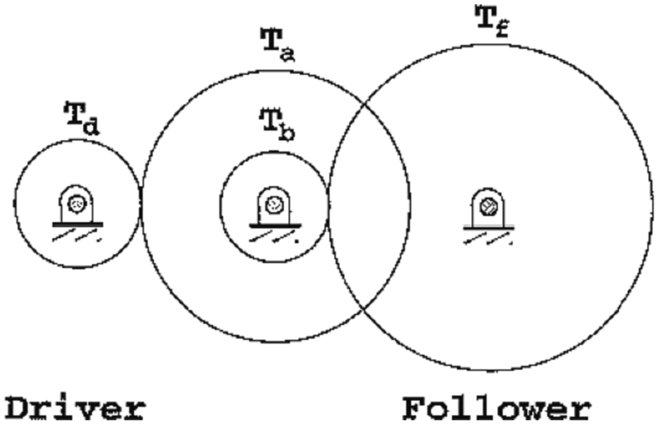

Fig. 3. Gear train design problem[39].

developed an algorithm that combined the multi-objective Genetic Algorithm NSGA-II with various metamodels. The BPO deals with the refurbishment of three specific buildings (Fig. 4): a penthouse (PH), an intermediate flat (IF), and a semi-detached house (SD).

The three reference buildings present different compactness ratios S/V, where S represent the dispersing surface and V the conditioned volume: 0.97 for the detached house, 0.63 for the penthouse, and 0.3 for the intermediate flat in multi-story building.

The typical envelopes of constructions built before the first Italian energy legislation in 1976, which have not undergone renovation, have been chosen for all three reference buildings. They are characterized by an opaque envelope resistance of 0.97 $m^2K\ W^{-1}$ and single pane glass with a standard timber frame. The thermal bridges in these reference cases have two-dimensional thermal coupling coefficients calculated according to EN ISO 10211. This calculation results in a linear transmittance of 0.098 $W\ m^{-1}K^{-1}$ for corners, 0.182 $W\ m^{-1}K^{-1}$ for intermediate floors and walls, and 0.06 $W\ m^{-1}K^{-1}$ for the perimeter of windows.

The infiltration rate for all the building reference configurations is determined based on the calculations specified in UNI EN 12207 and EN 15242. The heating system in the reference buildings consists of a standard boiler coupled with radiators and an on–off control system. To represent a climate typical of Northern Italy (Climatic zone E in the Italian classification), the weather conditions of Milan was chosen.

The design variables related to the optimization process of the three buildings are relative to the energy-saving measures (ESMs) that it is possible to apply to the non-renovated buildings:

1. Placement of an additional layer of expanded polystyrene with varying thicknesses (ranging from 0 to 20 cm) to the vertical walls, roof, and floor independently. Different initial costs were considered.
2. Replacement of windows: this measure focuses on replacing existing windows with more efficient glazing systems, such as double or triple pane windows, with either high or low solar heat gain coefficients.
3. Boiler replacement: the existing boiler is replaced with either a modulating or condensing boiler that includes an outside temperature reset control.
4. Installation of a mechanical ventilation system: this measure involves installing a mechanical ventilation system equipped with a cross-flow heat recovery system.

A cost-optimal framework utilizing a multi-objective optimization approach has been employed to determine the performance of different retrofit strategies. The first objective focuses on enhancing energy efficiency by minimizing the Primary Energy for Heating (EP_H). The second objective involves minimizing the total cost of the building, following the comparative framework methodology of cost-optimal levels. To measure the total cost of the building over a 30-year lifespan, the Net Present Value (NPV) indicator is used. The NPV is calculated by summing the cash flows associated with each intervention over time. The initial costs, derived from a regional price list, are considered for all the energy-saving measures (ESMs), along with annual energy costs, maintenance costs, replacement costs, and residual values for equipment with longer lifespans.

A brute force approach was employed to evaluate all possible combinations of the variables listed earlier. For the intermediate flat, there were 630 potential combinations, while for the penthouse, there were 13,230 combinations. In the case of the semi-detached house, the number of combinations reached 277,830. By evaluating all possible combinations, it was possible to assess the effectiveness of the models in accurately identifying the true Pareto front.

3.3. Metrics

In this study, some of the metrics proposed by Prada et al. [32] were used to evaluate the performance of the algorithm in solving the building refurbishment optimization problems. However, these metrics could not be applied to the three test functions optimization problems due to the continuous nature of some design variables and the inability to use brute force methods to search for the true Pareto front solutions required for metric assessment.

The metrics are categorized into efficacy, efficiency, and quality metrics. Efficiency metrics aim to assess the effort required by the algorithm to converge, while efficacy metrics measure the distance between the true Pareto front and the one found by the algorithm. Quality metrics evaluate the variety or dissimilarity of solutions generated by the algorithm [32]. It is important to note that all objective values were normalized with respect to the objective functions of the existing buildings before evaluating these metrics.

For efficiency, the ratio between the number of costly building energy simulations and the total number of combinations in the variable search space (NE) was chosen. NE allows for comparing different algorithms regardless of computational time, as the time required for executing the MATLAB code or the computational time of the code itself is negligible compared to the time needed for each expensive simulation.

Efficacy is quantified by two metrics: the fraction of true Pareto front solutions found by the algorithm (PS) and the number of wrong optimal solutions identified by the algorithm (C).

Finally, the selected quality metric is the pure diversity of the Pareto front, which is normalized based on the pure diversity of the true Pareto front (nPD).

4. Results

This section presents the results obtained applying the proposed algorithm to both the analytical test cases and the optimization process for refurbishing the three buildings.

Firstly, the optimization of the analytical test cases is presented, and the algorithm's performance is evaluated based on the minimum relative objectives. Whenever possible, the results obtained from applying

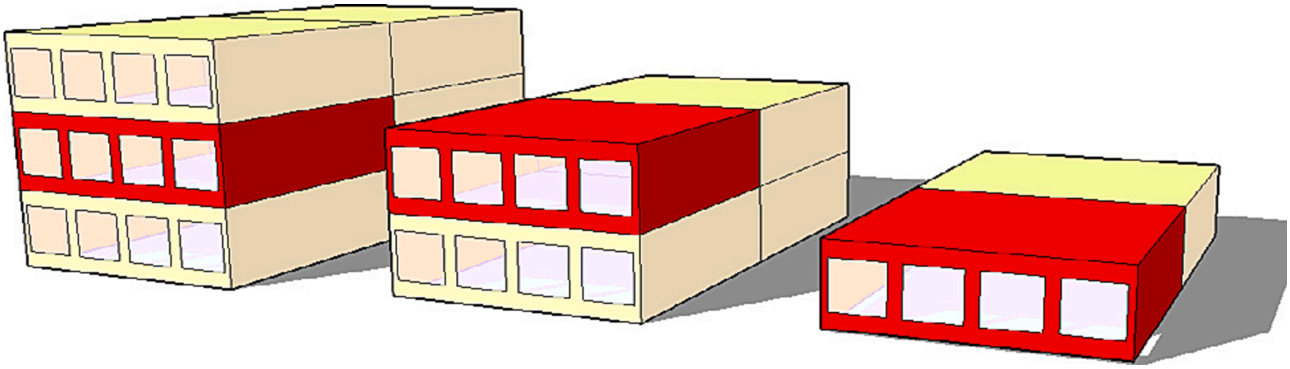

Fig. 4. Reference buildings used in the optimization problems: from the left - intermediate flat (IF), penthouse (PH) and semidetached house (SD) [32].

other algorithms to the same test cases are reported to allow for comparison. Additionally, graphical representations of the Pareto fronts obtained using the proposed algorithm are provided for a direct comparison against the results reported in Deb's work [39].

Subsequently, the optimization of the computationally expensive building models is discussed. In this section, the metrics described earlier are used to compare the performance of the new algorithm against existing literature.

By presenting these results and conducting the performance comparisons, this section aims to demonstrate the effectiveness and capabilities of the proposed algorithm in addressing the optimization challenges posed by both analytical test cases and real building refurbishment scenarios.

4.1. Analytical test cases

The algorithm parameters selected for each test case are shown in Table 1. These values were chosen arbitrarily in order to limit the number of expensive simulations with respect to the number of possible combinations of the design variables. A reduced number of weights' sets has been selected for the last test case, to focus the search of optimal solutions on the Pareto front extremities. This was required given the wide range of possible values related to the first objective of the gear train design problem (see Table 2 and Table 3).

The first test case/problem to which the algorithm was applied is the BNH problem. The solutions found by Binh and Korn [41] are $\min(f_1) = 0$ and $min(f_2) = 4$, respectively when $x_1, x_2 = 0$ and $x_1 = 5, x_2 = 3$.

The Pareto front found by the proposed algorithm is reported in Fig. 5. The black markers represent the objective functions of the points evaluated by the algorithm, while the red dots are the points of the Pareto front. The proposed algorithm was able to quickly find the two objective functions' minimums as well as many solutions on the Pareto front before reaching the maximum number of possible evaluations selected for the problem. The graph highlights the high percentage of points found by the algorithm belonging to the Pareto front, thus highlighting the algorithm's effectiveness in simulating only those solutions that are near the Pareto front. It is important to highlight that, among the black markers, are also present the points of the initial population, coming from the sampling process.

The second test case is relative to the two-bar truss design problem. In the original study [42], the ε-constraint method was used to minimize the volume and the tensional stresses in each of the two bars, with a minimum of $min(f_1) = 0.00445\ \mathrm{m}^3$ and $min(f_2) = 83268$ kPa respectively. The NSGA-II algorithm was also applied to the same problem, which was able to find the following minimums: $min(f_1) = 0.00407\ \mathrm{m}^3$ and $min(f_2) = 8439$ kPa.

The proposed algorithm was able to find lower values for both minimum with respect to NSGA-II results, within the maximum number of possible expensive evaluations: $min(f_1) = 0.00385\ \mathrm{m}^3$ and $min(f_2) = 8433$ kPa, with relative coordinates $x_1 = 5.0*10^{-4}\,\mathrm{m}^2, x_2 = 9.0*10^{-4}\mathrm{m}^2, y = 1.6\mathrm{m}$ and $x_1 = 5.6*10^{-3}\mathrm{m}^2, x_2 = 1.0*10^{-2}\mathrm{m}^2, y = 3\mathrm{m}$. It is worth noting that the minimum of f_1 is lower than what was identified by ε-constraint method and NSGA-II. Furthermore, it is possible to notice that also in this case, the solutions are mainly located near the Pareto front (Fig. 6). In this case, the algorithm did not reach convergence within the maximum number of evaluations. The reason is most likely the continuous nature of the variables related to the problem, which leads to a high number of possible combinations of the decision variables' values.

Table 1
Values of the algorithm' parameters for each test case.

Analytical test case	n_{LHS}	n_{imax}	n_{end}	Initial Population Size
BNH problem	100	1000	100	20
Two-bar truss design problem	100	1000	100	20
Gear train design problem	10	1000	100	20

Table 2
Efficiency (NE), efficacy (PS and C) and quality (nPD) metrics related to the optimal solutions found by applying the proposed algorithm to the IF optimization problem both with and without hammering distance option enabled.

	–				Hammering distance			
n_{samp}	NE (%)	PS (%)	C (%)	nPD (%)	NE (%)	PS (%)	C (%)	nPD (%)
32	15	47	7	130	39	73	5	93
64	19	45	4	87	44	84	2	94
128	26	33	4	66	52	78	2	91
256	44	24	20	82	74	53	13	83

Table 3
Efficiency (NE), efficacy (PS and C) and quality (nPD) metrics related to the optimal solutions found by applying the proposed algorithm to the PH optimization problem both with and without hammering distance option enabled.

	–				Hammering distance			
n_{samp}	NE (%)	PS (%)	C (%)	nPD (%)	NE (%)	PS (%)	C (%)	nPD (%)
32	1.1	35	0	98	2.3	62	0	149
64	1.5	42	0	95	2.5	58	0	127
128	1.7	46	0	100	2.8	58	0	146
256	2.4	17	0	89	3.8	56	0	98

Finally, the results relative to the gear train design problem are presented. In this case, the task was to minimize the error between the required and calculated gear ratios, as well as the size of the gears. NSGA-II was able to find the following solutions: $min(f_1) = 1.83*10^{-8}$ and $min(f_2) = 13$ cm, with coordinates equal to $x_a = 12, x_b = 12, x_d = 27, x_f = 37$ relative to $min(f_1)$, and $x_a = 12, x_b = 12, x_d = 13, x_f = 13$ to $min(f_2)$.

The minimums reached for the two objective functions are: $min(f_1) = 7.78*10^{-7}$ and $min(f_2) = 12$ cm, respectively at $x_a = 12, x_b = 12, x_d = 32, x_f = 31$ and $x_a = 12, x_b = 12, x_d = 12, x_f = 12$. In this case, the proposed algorithm was able to cover most of the Pareto front before meeting a stopping condition and so, without reaching the maximum value of evaluations, n_{imax}. The total number of evaluations performed was 499, including the initial sampling, while the number of possible combinations is 5 764 801. The Pareto front is represented in Fig. 7.

4.2. Building simulation optimizations

The optimization problem related to the refurbishment of three building models - i.e., the penthouse (PH), the intermediate flat (IF), and the semi-detached house (SD) – has been used to test the new algorithm starting from different samples sets, which are the same used for the benchmark tests in Prada et al., [32].

For all building typologies a number of weights' sets (i.e., n_{LHS}) of 200, a maximum number of allowed coordinates which lie outside the Pareto front (n_{end}) of 200 (no Hamming distance) or 20 (Hamming distance) are set. The maximum number of evaluations (n_{imax}) was set to 500 for the intermediate flat (to be less than the number of possible combinations, i.e., 600) and to 1000 for the two other cases. Given the discrete nature of the design variables and the high step on feasible range ratio for some of them, the optimization problem has been solved with and without the hammering distance option enabled. The increase of n_{LHS} from 100 to 200 allows to densify the solutions in the central part of the Pareto front.

The results' figures are presented by means of the metrics described in section 3.3 for each building and initial sampling size. Furthermore, the Pareto front found with the new algorithm and the true Pareto front found through a brute force method are shown for the smallest initial sampling size without the hammering distance. The Pareto front found

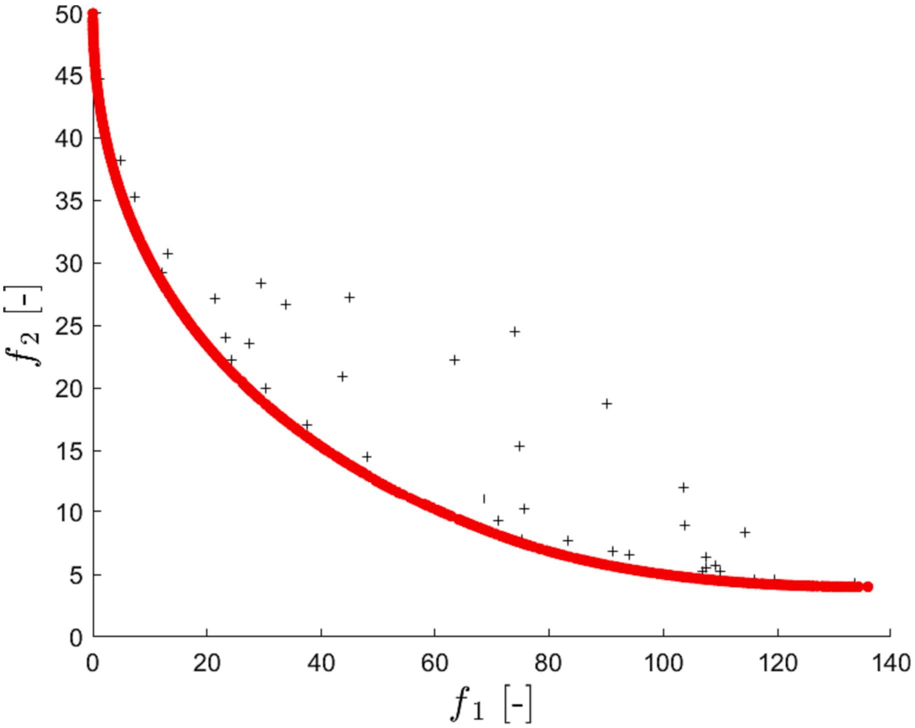

Fig. 5. Pareto front found by applying the proposed algorithm to the BNH problem.

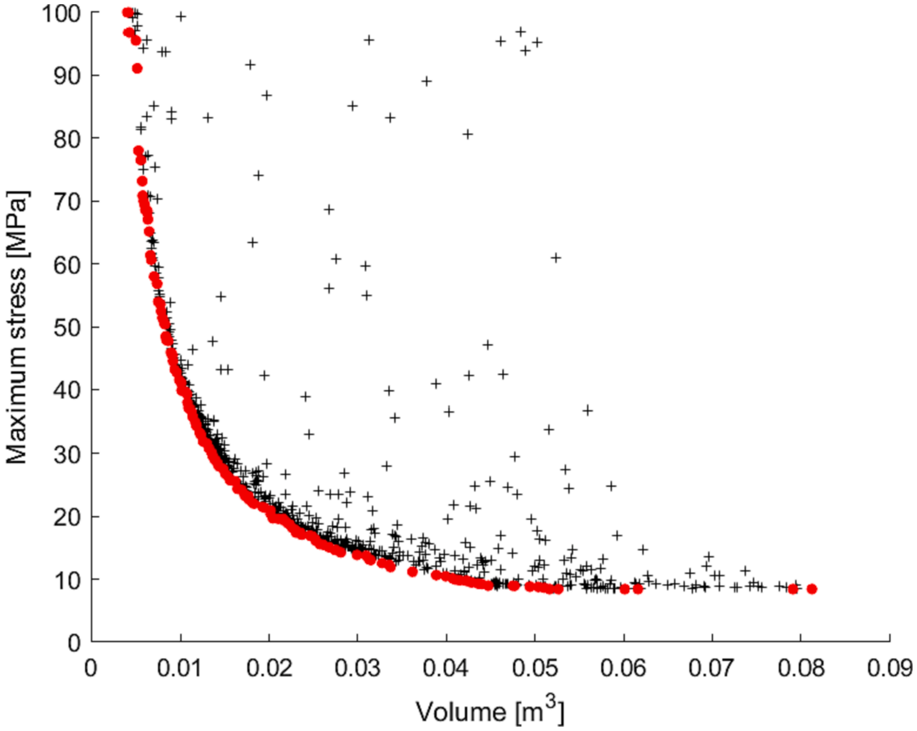

Fig. 6. Pareto front found by applying the proposed algorithm to the two-bar truss design problem.

by the algorithm is represented by red dots, while the true Pareto front by blue circles, and the black marks represent the coordinates evaluated, including the sample.

4.2.1. *Intermediate flat (IF)*

The intermediate flat case is characterized by a high percentage of costly building simulations (up to 44 % and 74 % respectively for the simulations with and without the hammering distance) with respect to the maximum number of possible combinations of design variables' values. The high percentage of costly building simulations however is balanced by a high number of true Pareto solutions found, especially when the hammering distance is used (Fig. 8). Furthermore, if the hammering distance option is not used, the number of costly simulations increases as well as the number of true Pareto solutions found decreases

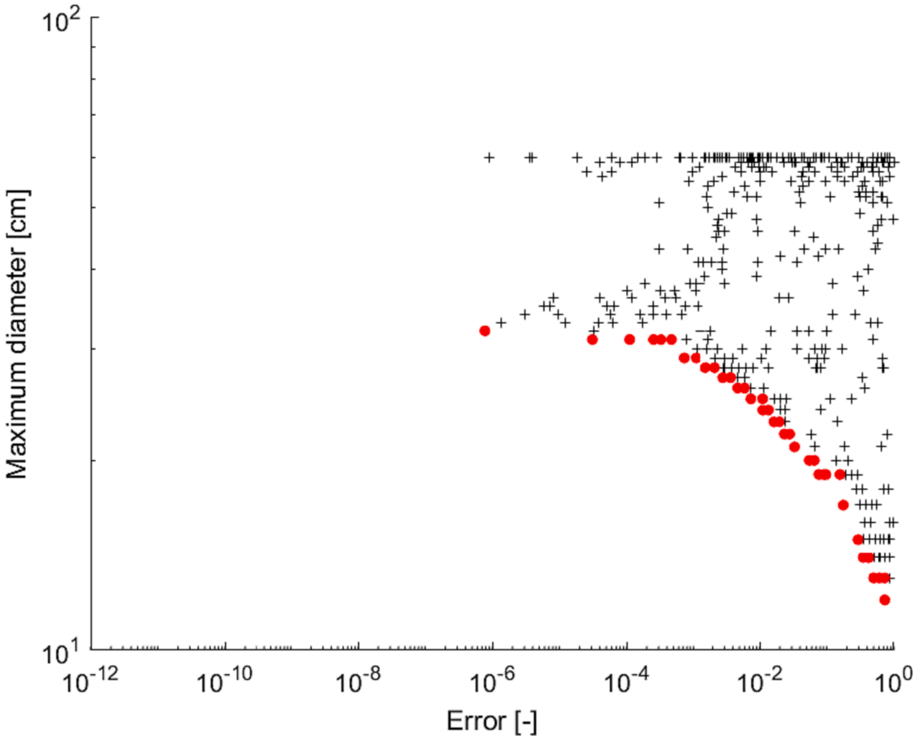

Fig. 7. Pareto front found by applying the proposed algorithm to the gear train design problem.

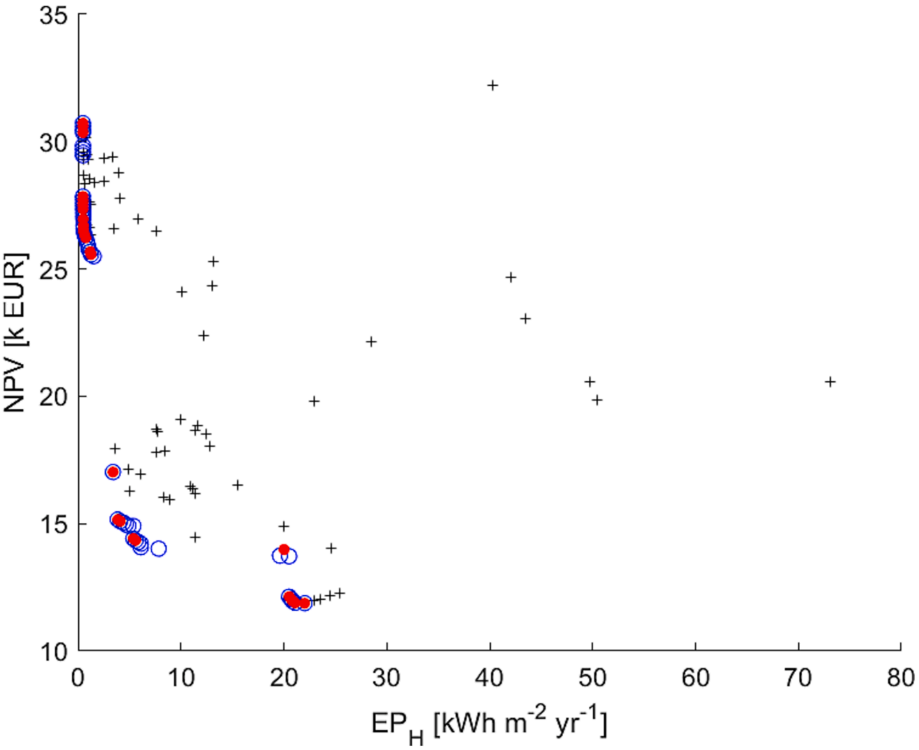

Fig. 8. Solutions found related to the IF optimization problem both with the proposed algorithm (red dots) and with the brute force approach (blue circles – true Pareto front). The black marks represent the non-optimal solutions found with the proposed algorithm.

(Table 2). This result highlights the key role of the initial sample size and the dependence on the number of possible combinations. Indeed, increasing the sampling size from 64 to 128 does not improve the algorithm accuracy, but rather worsens its efficiency due to more expensive simulation runs. Similarly, as the initial population increases, the value of the C metric increases, identifying suboptimal solutions that are in reality dominated by others. A clear trend on the quality of the front is not easily identifiable. In fact, while for an initial sample of 32 points the hammering distance worsens the diversity (*nPD*) of front solutions, there are weak improvements in the other cases.

4.2.2. Penthouse (PH)

In the PH case the number of available combinations increases thus stressing the efficiency metric of the algorithm which converges with higher accuracy with respect to the cases of the other buildings, simulating less than 3 percent of the possible combinations. Again, the increase in the initial sampling size does not always contribute to solution accuracy if the hammering option is not enabled (Table 3). The Coverage metric (C) always equal to 0 % indicates that all the solutions found by the algorithm are part of the true Pareto front. The case with an initial sampling size of 32 points outperforms all the other cases, characterized by a bigger sampling size, lower percentage of true Pareto solutions found and similar diversity metric (Fig. 9). In this test case, unlike the previous one, there is an improvement in solution diversity when the hammering distance is adopted.

Overall, with respect to the previous cases, the efficiency has increased substantially without compromising both efficacy and quality, compared to the previous building case.

4.2.3. Semidetached house (SD)

The third test case is certainly the most challenging given the greatest number of combinations available (Fig. 10). The data reported in Table 4 show the identification of about 25 % of the Pareto front solutions with less than 1 % of combinations simulated with the expensive model. The reduced percentage of costly simulation performed does not affect the number of true Pareto solutions found and if the hammering distance option is enabled, the PS metrics are comparable with those obtained for PH although NE is significantly lower.

Again, C metric highlights the algorithm's ability to identify only those solutions that actually belong to the Pareto front, thus avoiding the identification of false optimums. As in PH, there is a benefit of hammering distance, that leads to a greater diversity of front solutions.

4.3. Performance compared with other efficient optimization frameworks

In the final step, the results obtained in this study are compared with the results from the previous work conducted by Prada et al. [32]. Since in the previous work multiple metamodels were tested for the same cases, to facilitate the comparison, the process is divided into two steps. In the first step, the results for each building category are graphically represented for the metrics NE (number of expensive simulations), PS (efficacy), and C (accuracy). Fig. 11 illustrates these graphical representations, showcasing the comparison between the results obtained in the present work and those from the previous study. In the second step, the comparison focuses on each building typology and each sampling size. The results from the present work are compared with the results from the previous study, taking into consideration the proximity in terms of the number of expensive simulations (NE). By matching the NE number, a direct comparison can be made in terms of efficacy (PS), accuracy (C), and quality (nPD). This two-step comparison process allows for a comprehensive evaluation of the results, enabling a meaningful assessment of the effectiveness, accuracy, and quality of the optimization algorithm employed in the present study in relation to the findings of the previous work conducted by Prada et al. [32].

In Fig. 11, the results of the present work are represented by red dots, while the results of the previous work are represented by light blue dots. The metric PS is reported as the complement to the unit value (1-PS), meaning that the best solutions are in the left-bottom corner of all graphs. The results related to the IF show similarities in terms of NE, but significant differences in terms of PS. This indicates that while the proposed algorithm quickly identified the best solutions within the initial simulations, it struggled to find new solutions that were different from the ones already present in the dataset. This behavior is highlighted by the C-NE graphs, which demonstrate high accuracy that is independent of the number of simulations performed. Moving on to the PH, the new algorithm consistently outperforms the previous work in terms of the number of solutions found on the true Pareto front (PS) when NE values are similar. However, due to the relatively small number of possible combinations, if some solutions from the true Pareto front are already found through the sampling process or within a few simulations, the presented algorithm faces difficulties in finding solutions that are not

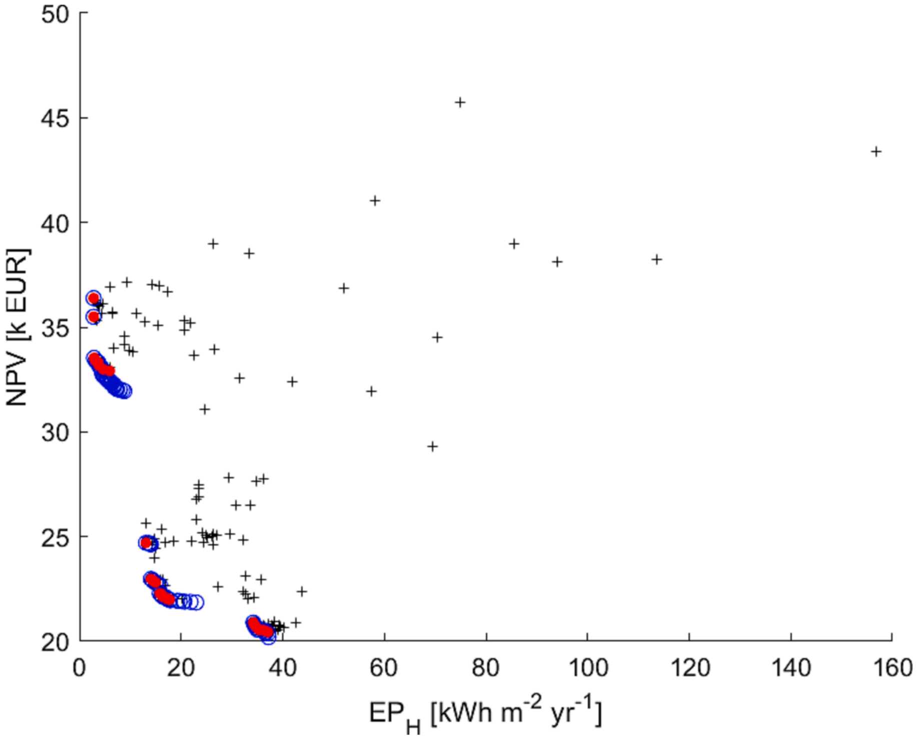

Fig. 9. Solutions found related to the PH optimization problem both with the proposed algorithm (red dots) and with the brute force approach (blue circles – true Pareto front). The black marks represent the non-optimal solutions found with the proposed algorithm.

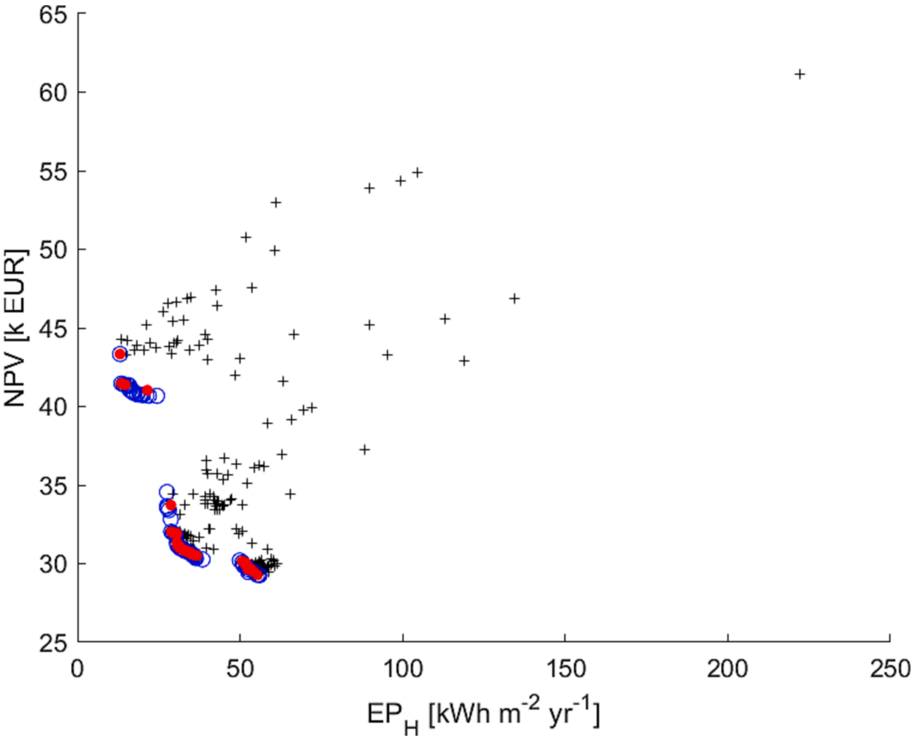

Fig. 10. Solutions found related to the SD optimization problem both with the proposed algorithm (red dots) and with the brute force approach (blue circles – true Pareto front). The black marks represent the non-optimal solutions found with the proposed algorithm.

Table 4
Efficiency (NE), efficacy (PS and C) and quality (nPD) metrics related to the optimal solutions found by applying the proposed algorithm to the SD optimization problem both with and without hammering distance option enabled.

	–				Hammering distance			
n_{samp}	NE %	PS %	C %	nPD %	NE %	PS %	C %	nPD %
32	0.11	25	0	78	0.17	51	0	91
64	0.12	27	0	77	0.19	49	0	74
128	0.13	22	0	76	0.19	52	0	75
256	0.15	16	0	81	0.24	54	0	67

yet present in the dataset. As a result, in some cases, certain algorithms from Prada et al. [32] outperform the new algorithm in terms of PS. This observation is validated by the last building category, where the larger number of possible combinations leads to better results for all the algorithms in terms of PS. The sampling process initially struggles to find solutions that are close or directly on the true Pareto front. However, the new algorithm's greater efficiency and effectiveness in identifying the front can be better appreciated given the limited impact of the initial sampling size. Moreover, the Pareto front found by the new algorithm is almost always included within the true Pareto front (C metric equal to zero). This outcome allows the decision-maker to avoid considering suboptimal solutions, even though they are generally close to the true front. Furthermore, the accuracy of the algorithm is not strongly dependent on the sampling size or the number of expensive simulations (NE).

Table 5, 6 and 7 show the metrics relative to the results obtained with the proposed algorithm with hammering distance compared with the results of the previous contribution in case of similar NE values. By matching the results in terms of NE it is possible to have a quantitative comparison on the accuracy and quality of the Pareto fronts.

In the case of the IF, when comparing the results in terms of NE, both the nPD and PS metrics are similar to those obtained in the previous work (Table 5). However, the C metric tends to be significantly lower, except for the case with a sampling size of 256 points where the algorithm performs worse with respect to all metrics. The lower accuracy observed in the C metric can be attributed to the true Pareto solutions found within the sampling process. Since the algorithm has difficulties in finding new solutions close to points that are already on the Pareto front, it struggles to improve upon the solutions obtained through the initial sampling. As a result, the algorithm may not explore the full extent of the Pareto front, leading to lower accuracy compared to the previous work. Overall, while the new algorithm shows similar results to the previous work in terms of nPD and PS when matched in terms of NE, the lower accuracy observed in the C metric highlights the algorithm's limitations in finding new solutions in proximity to the Pareto front.

In the penthouse case, the results with the hammering distance option enabled show significant improvements in terms of PS and C found by the proposed algorithm for almost all initial population sizes (Table 6). The only exception is the case with a sampling size of 256 points, where the results differ. In this scenario, characterized by a higher number of possible combinations if compared to the intermediate flat, there could be two possible explanations why the proposed algorithm cannot perform as well as for the other initial population sizes. Firstly, it is possible that some solutions were found during the sampling process that were already on or close to the true Pareto front. This would explain the greater values in PS and C metrics found in the previous contribution, which heavily depend on the quality of the initial population. Alternatively, the points selected during the sampling process might have influenced the metamodel to develop with a wrong "shape," affecting the accuracy of the results. This observation is further supported by the trend observed across all three building typologies. The solutions obtained with a sampling size of 256 points are consistently worse compared to the other cases, indicating that a smaller initial population size generally leads to better results. Overall, the hammering distance option proves beneficial in improving the efficacy (PS) and accuracy (C) of the algorithm for the penthouse case, while the quality (nPD) remains comparable. The exception with the sampling size of 256 points suggests that caution should be exercised when selecting the

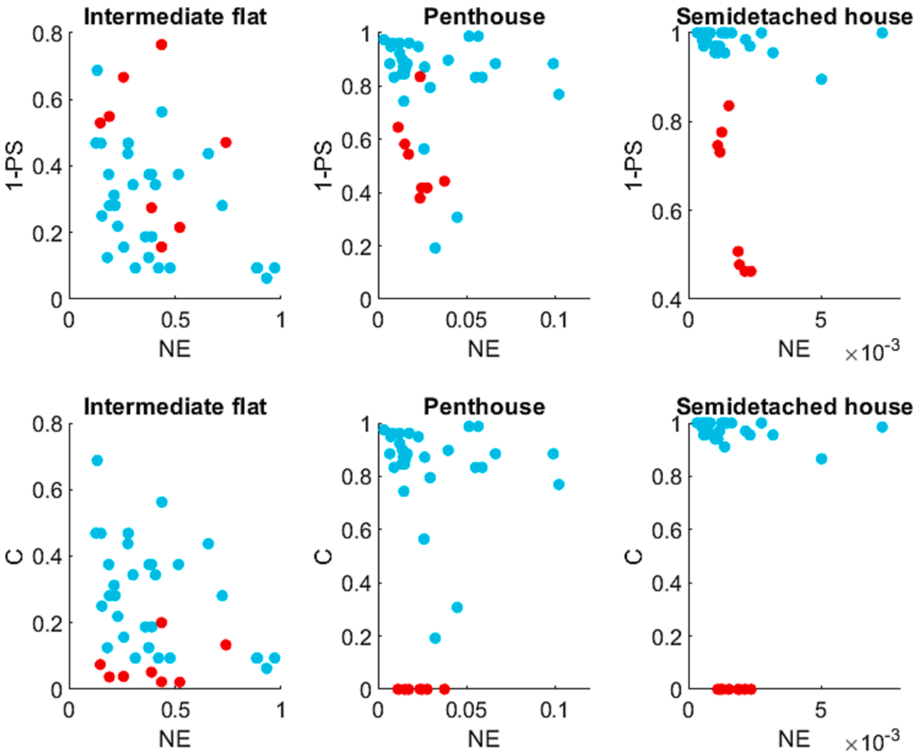

Fig. 11. Efficiency (ne) and efficacy (ps and c) metrics related to the solutions found with the proposed algorithm (red dots) and with the algorithm in prada et al. [32], for all three reference buildings (IF, PH and SD).

Table 5
Metrics related to the solutions of the proposed algorithm with hammering distance option enabled compared with those similar in terms of ne found with the algorithm in prada et al., [32] for the IF case.

Intermediate flat (IF)								
	Algorithm in Prada et al., [32]				New algorithm			
n_{samp}	NE %	PS %	C %	nPD %	NE %	PS %	C %	nPD %
32	44	44	56	86	39	73	5	93
64	26	84	16	101	44	84	2	94
128	66	56	44	99	52	78	2	91
256	93	94	6	99	74	53	13	83

Table 6
Metrics related to the solutions of the proposed algorithm with hammering distance option enabled compared with those similar in terms of ne found with the algorithm in prada et al.,[32] for the PH case.

Penthouse (PH)								
	Algorithm in Prada et al., [32]				New algorithm			
n_{samp}	NE %	PS %	C %	nPD %	NE %	PS %	C %	nPD %
32	2.6	44	56	173	2.3	62	0	149
64	1.6	12	89	146	2.5	58	0	127
128	1.5	15	85	145	2.8	58	0	146
256	4.5	69	31	154	3.8	56	0	98

initial population size, as a smaller size may generally yield better results.

From the analysis of the semidetached house (Table 7), the presented algorithm excels in finding a higher number of true Pareto front solutions with high accuracy compared to the previous work, especially in relation to high-dimensional problems, while the quality of the Pareto front found by the algorithm is comparable, considering the same number of expensive simulations.

Table 7
Metrics related to the solutions of the proposed algorithm with hammering distance option enabled compared with those similar in terms of ne found with the algorithm in prada et al.,[32] for the SD case.

Semidetached house (SD)								
	Algorithm in Prada et al., [32]				New algorithm			
n_{samp}	NE %	PS %	C %	nPD %	NE %	PS %	C %	nPD %
32	0.14	0	100	69	0.17	51	0	91
64	0.16	0	100	80	0.19	49	0	74
128	0.14	5	91	76	0.19	52	0	75
256	0.23	3	96	87	0.24	54	0	67

The semidetached house case presents a higher number of possible combinations compared to the other building typologies but despite this increased complexity, even with a sampling size of 256 points, the algorithm can achieve better performance compared to the previous work. This indicates the effectiveness and robustness of the presented algorithm in tackling more challenging optimization problems.

Overall, the algorithm demonstrates its ability to generate a larger number of solutions on the true Pareto front with high accuracy since no suboptimal solutions were selected as part of the Pareto front. These results underscore the algorithm’s improved performance and its suitability for handling complex optimization problems with a higher number of possible combinations.

5. Discussion

The results presented in this section reveal several features of the new algorithm, as well as its advantages and disadvantages, especially when compared to the outcomes of a previous contribution on the same optimization problems.

The application of the algorithm to the three analytical test functions demonstrates its capability to find the same or even better solutions compared to well-established algorithms such as NSGA-II and the ε-constraint method. The Pareto fronts depicted in the respective figures further confirm the algorithm's ability to focus on the region of the search space characterized by points near the global minimum. This enables the algorithm to avoid unnecessary expensive simulations, resulting in improved efficiency.

Considering the gear train design problem, even for cases with a wide range of objective function values, the algorithm successfully identifies the Pareto front with a limited number of simulations compared to the total number of possible combinations. This showcases the algorithm's effectiveness in handling complex optimization problems.

By comparing the results with the algorithms in Prada et al. [32] for the optimization problems regarding the refurbishment of three reference buildings, certain advantages and disadvantages can be observed. One notable difficulty of the algorithm is in finding new solutions near the ones already identified on the true Pareto front. When the number of possible combinations is low, optimal solutions can be found early on, even during the sampling process. In such cases, the algorithm faces challenges in discovering different solutions, and the results deteriorate as the number of sampled points increases. On the other hand, the potential of the new algorithm becomes more apparent when dealing with problems of increased complexity, characterized by a high number of possible combinations and/or design variables. For simpler cases, such as the intermediate flat, the algorithm typically finds initial solutions on the true Pareto front within the first 100 expensive simulations. The initial Pareto front identified by the algorithm may consist of sparse solutions but still effectively covers a significant portion of the objective space. As the iterations progress, the algorithm achieves higher PS but at the expense of an increase in NE. The trade-off between PS and NE is influenced by the setting of n_{end}, which needs to be carefully determined based on whether the goal is to find as many solutions as possible on the true Pareto front or to limit the number of expensive simulations.

It is noteworthy that even with a low number of simulations, the Pareto front identified by the algorithm is generally accurate, indicating that the solutions lie on the true Pareto front. This observation is reflected in the C metric, which is typically close to zero and shows weak correlation with the initial population density or NE. Thus, even with a limited number of expensive simulations, the solutions found by the new algorithm are typically reliable. Additionally, the relationship between PS and NE is weak, particularly for complex cases. More expensive simulations do not always lead to increased PS, especially if the increase is due to a larger number of sampling points. Regardless of the settings, a wider initial population often leads to suboptimal results. This consideration may vary depending on the case, particularly in relation to the number of design variables, but it highlights that a high number of sampling points is often unnecessary or even unfavourable.

In summary, the new algorithm demonstrates promising features and advantages in terms of finding optimal solutions, accurately identifying the Pareto front, and managing the trade-off between PS and NE. However, it also exhibits challenges when encountering solutions near the true Pareto front and in balancing the initial population density and the number of expensive simulations. Understanding these characteristics allows for better utilization and optimization of the algorithm in different problem scenarios.

6. Conclusion

A novel probabilistic optimization algorithm has been developed, leveraging metamodels to reduce the computational burden of expensive simulation runs in optimization processes. The algorithm incorporates multiple approaches to mitigate the number of expensive simulations required to achieve satisfactory results:

- the optimization algorithm employs a metamodel to explore the search space on behalf of the expensive models. By utilizing the metamodel, only the most promising solutions are selected for further evaluation using the expensive models. This selective approach significantly reduces the number of expensive simulations required.
- the metamodel fitting process is based on the Horseshoe method, which is a Bayesian probabilistic approach. This method ensures high-quality fitting by employing a multi-variate polynomial fitting process in each iteration, without constraining the search to specific regions within the design variable space.
- as the optimization process progresses, the metamodel fitting process adapts its strategy. Instead of considering all available points, it focuses on the most promising ones, specifically targeting the region around the Pareto front within the design variable space. This adaptive approach further narrows down the search, reducing the reliance on expensive simulations.
- by transitioning from a multi-objective optimization to a single objective optimization, the quality of the fitting process is significantly enhanced. As a result, the number of expensive simulations performed with design variables that are far from the Pareto front is greatly reduced.

By incorporating these strategies, the new algorithm showcases its ability to effectively reduce the number of expensive simulations required to achieve satisfactory results, while maintaining or even surpassing the performance of established methods like the ε-constraint method and NSGA-II. The algorithm's applicability and advantages are demonstrated across various optimization scenarios, validating its efficacy and highlighting its potential for practical use. To this end, the algorithm's performance is evaluated on three multi-objective test functions with constraints, as well as on the optimization problems relative to the refurbishment of three reference buildings. The results are compared with those obtained using the ε-constraint method and NSGA-II for selected test functions, and, by means of several metrics, with NSGA-II coupled with different metamodels for the three reference building optimization problems.

Through the comparison of test functions and building optimization problems results, it was possible to highlight several key capabilities of the new algorithm:

- the new algorithm successfully identifies minimum values for all test functions that are on par with those obtained using extensively validated algorithms like NSGA-II and the ε-constraint method. This demonstrates the algorithm's ability to achieve competitive optimization results.
- the algorithm exhibits a general ability to focus on the portion of the objective space near the Pareto front. This significantly reduces the need for a large number of expensive simulations, making the optimization process more efficient.
- the application of the algorithm is not limited to problems with objective functions of different magnitudes or specific types of design variables. It can effectively handle continuous, discrete, and categorical variables, making it versatile for a wide range of optimization problems.
- solutions on the true Pareto front are found with a limited number of expensive evaluations. Across all building optimization problems, including the sampling process, fewer than 100 expensive simulations are required to find the minimum values of the objective functions.
- increasing the number of expensive simulations tends to result in densification of the solutions on the true Pareto front. The number of design variables has a weak relationship with this characteristic, as the complexity of the problems increases without significantly impacting the number of solutions found on the true Pareto front.
- the new algorithm achieves a significantly higher level of accuracy while maintaining a comparable diversity of solutions found. This indicates that the algorithm not only generates solutions with a wide

range of characteristics but also ensures their accuracy and high quality.

- the Pareto front identified by the new algorithm in the initial simulations generally covers a significant portion of the objective space. This indicates that the algorithm quickly explores and captures important regions of interest.

For what concerns the optimization problems of the three reference buildings, the number of solutions found by the proposed algorithm on the true Pareto front is generally equal to or greater than those found in a previous contribution, except if specific conditions are met. Specifically, for the intermediate flat and penthouse cases, if the initial population size is 256, there is a potential issue where the algorithm may consistently find the same solutions when encountering the true optimal solutions during the sampling process or early iterations. As a result, the number of unique solutions on the true Pareto front may be smaller compared to the previous contribution in these specific cases. However, in other scenarios characterized by a smaller initial population or a greater number of design variable possible combinations, the proposed algorithm demonstrates its ability to find a comparable or even greater number of solutions on the true Pareto front with respect to results available in literature. It is important to note that the accuracy of the algorithm is not significantly affected by the number of expensive simulations, since the solutions found by the algorithm are usually part of the true Pareto front.

From the results, it is possible to conclude that the proposed algorithm excels at finding satisfying results with great accuracy within a limited number of expensive simulations, especially for high-dimensional optimization problems. In cases where a higher density of solutions on the Pareto front is desired, the algorithm can provide additional solutions at the expense of efficacy, by increasing the number of expensive simulations.

Overall, the algorithm can successfully reduce the number of expensive simulations, and so, the computational time required to complete an optimization process, while maintaining a high level of accuracy, which was the main objective of this work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Directive 2010/31/EU, of the European parliament and of the council of 19 may 2010 on the energy performance of buildings OJ L 153/2010, 2010.

[2] P. Penna, A. Prada, F. Cappelletti, A. Gasparella, Multi-objectives optimization of energy efficiency measures in existing buildings, Energ. Buildings 95 (2015) 57–69.

[3] E. Carlon, M. Schwarz, A. Prada, L. Golicza, V.K. Verma, M. Baratieri, A. Gasparella, W. Haslinger, C. Schmidl, On-site monitoring and dynamic simulation of a low energy house heated by a pellet boiler, Energ. Buildings 116 (2016) 296–306.

[4] V. Machairas, A. Tsangrassoulis, K. Axarli, Algorithms for optimization of building design: A review, Renew. Sustain. Energy Rev. 31 (2014) 101–112.

[5] S. Sharma, V. Kumar, Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future, Arch. Comput. Methods Eng. 29 (7) (2022) 5605–5633.

[6] J. Wang, Z.J. Zhai, Y. Jing, C. Zhang, Particle swarm optimization for redundant building cooling heating and power system, Appl. Energy 87 (12) (2010) 3668–3679, https://doi.org/10.1016/j.apenergy.2010.06.021.

[7] S. Attia, M. Hamdy, W. O'Brien, S. Carlucci, Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design, Energ. Buildings 60 (2013) 110–124.

[8] R. Evins, A review of computational optimisation methods applied to sustainable building design, Renew. Sustain. Energy Rev. 22 (2013) 230–245.

[9] X. Shi, Z. Tian, W. Chen, B. Si, X. Jin, A review on building energy efficient design optimization from the perspective of architects, Renew. Sustain. Energy Rev. 65 (2016) 872–884.

[10] K. Klemm, W. Marks, A.J. Klemm, Multicriteria optimisation of the building arrangement with application of numerical simulation, Build. Environ. 35 (2000) 537–544.

[11] J.H. Lee, Optimization of indoor climate conditioning with passive and active methods using GA and CFD, Build. Environ. 42 (9) (2007) 3333–3340.

[12] L. Magnier, F. Haghighat, Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network, Build. Environ. 45 (3) (2010) 739–746.

[13] B. Eisenhower, Z. O'Neill, S. Narayanan, V.A. Fonoberov, I. Mezić, A methodology for meta-model based optimization in building energy models, Energ. Buildings 47 (2012) 292–301.

[14] E. Tresidder, Y. Zhang, A. I. J. Forrester, *Acceleration of building design optimisation through the use of kriging surrogate models,* In: BSO12 Proceedings of the 1st IBPSA England conference building simulation and optimization, Loughborough, UK; 2012. p. 118–125.

[15] C. J. Hopfe, M. Emmerich, R. Marijt, J. L. M. Hensen, *Robust multi-criteria design optimisation in building design,* In: BSO12 Proceedings of the 1st IBPSA-England conference building simulation and optimization, Loughborough, UK; 2012. p. 118–125.

[16] A.E.I. Brownlee, J.A. Wright, Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation, Appl. Soft Comput. 33 (2015) 114–126.

[17] W. Xu, A. Chong, O.T. Karaguzel, K.P. Lam, Improving evolutionary algorithm performance for integer type multi-objective building system design optimization, Energy and Building 127 (2016) 714–729.

[18] D.C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, 2017.

[19] R.H. Myers, D.C. Montgomery, C.M. Anderson-Cook, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons, 2016.

[20] Santner, T. J., Williams, B. J., & Notz, W. I. *The Design and Analysis of Computer Experiments*, Springer, (2003).

[21] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P.K. Tucker, Surrogate-Based Analysis and Optimization, Prog. Aerosp. Sci. 41 (1) (2005) 1–28.

[22] M.J.D. Powell, Radial Basis Functions for Multivariable Interpolation: A Review, Algorithms for Approximation 2 (1992) 143–167.

[23] I. Babuška, J.M. Melenk, The Partition of Unity Method, Int. J. Numer. Meth. Eng. 40 (4) (1996) 727–758.

[24] I. Steinwart, A. Christmann, Support Vector Machines, Springer, 2008.

[25] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 2013.

[26] N.D. Roman, F. Bre, V.D. Fachinotti, R. Lamberts, Application and characterization of metamodels based on artificial neural networks for building performance simulation: A systematic review, Energ. Buildings 217 (2020) 109972.

[27] S. Kalogirou, Optimization of solar systems using neural-networks and genetic algorithms, Appl. Energy 77 (2004).

[28] Y. Xu, G. Zhang, C. Yan, G. Wang, Y. Jiang, K. Zhao, A two-stage multi-objective optimization method for envelope and energy generation systems of primary and secondary school teaching buildings in China, Building and Environment 204 (2021), 108142, https://doi.org/10.1016/j.buildenv.2021.108142. ISSN 0360-1323.

[29] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction, Springer, London, 2011.

[30] Y. Jung, Y. Heo, H. Lee, Multi-objective optimization of the multi-story residential building with passive design strategy in South Korea, Build. Environ. 203 (2021), 108061, https://doi.org/10.1016/j.buildenv.2021.108061.

[31] X. Gao, Y. Hu, Z. Zeng, Y. Liu, A meta-model-based optimization approach for fast and reliable calibration of building energy models, Energy 188 (2019), 116046, https://doi.org/10.1016/j.energy.2019.116046.

[32] A. Prada, A. Gasparella, P. Baggio, On the performance of meta-models in building design optimization, Appl. Energy 225 (2018), https://doi.org/10.1016/j.apenergy.2018.04.129.

[33] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Trans. Evol. Comput. 6 (2002) 481–494.

[34] J. Knowles, ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Trans. Evol. Comput. 10 (1) (2006) 50–66.

[35] A.T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis, Appl. Energy 113 (2014) 1043–1058.

[36] J.H. Kämpf, M. Wetter, D. Robinson, A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus, J. Build. Perform. Simul. 3 (2) (2010) 103–120.

[37] D. Tuhus-Dubrow, M. Krarti, Comparative analysis of optimization approaches to design building envelope for residential buildings, ASHRAE Trans. 115 (2000) 554–562.

[38] E. Makalic, D. Schmidt, A Simple Sampler for the Horseshoe Estimator, IEEE Signal Process Lett. (2015), https://doi.org/10.1109/LSP.2015.2503725.

[39] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, 2001.

[40] M. Wetter, J. Wright, A comparison of deterministic and probabilistic optimization algorithms for non-smooth simulation-based optimization, Build. Environ. 39 (8) (2004) 989–999.

[41] T.T. Binh, U. Korn, MOBES: A multiobjective evolution strategy for constrained optimization problems, in: The Third International Conference on Genetic

Algorithms, Proceeding of the 3rd International Conference on Genetic Algorithm Mendel, Brno, Algorithm Mendel, Brno, 1997, pp. 176–182.

[42] N. Palli, S. Azram, P. McCluskey, R. Sundararajan, An interactive multistage ε-inequality constraint method for multiple objectives decision making, ASME J. Mech. Des. 120–4 (1999) 678–686.

[43] B.K. Kannan, S.N. Kramer, *An augmented lagrange multiplier based method for mixed integer discrete* continuous optimization an*d its applications to mechanical design*, ASME J. Mech. Des. 116–2 (1994) 405–411.