

## **Integrating robustness indicators into multi-objective optimization to find robust optimal low-energy building designs**

### **Abstract**

Uncertainties can have a large influence on building performance and cause deviations between predicted performance and performance during operation. It is therefore important to quantify this influence and identify robust designs that have potential to deliver the desired performance under uncertainties. Generally, robust building designs are identified by assessing the performance of multiple design configurations under various uncertainties. When exploring a large design space, this approach becomes computationally expensive and infeasible in practice. Therefore, we propose a simulation framework based on multi-objective optimization and sampling strategies to find robust optimal designs at low computational costs. The genetic algorithm parameters of optimization are fine tuned to further enhance the computational efficiency. Furthermore, a modified fitness function is implemented to use minimax regret robustness method in the optimization loop. The implemented simulation framework can save up to 94-99% of computational time compared to full factorial approach, while identifying the same robust designs.

**Keywords:** Multi-objective optimization; scenario sampling; uncertainties; robust design; low-energy buildings; performance robustness assessment

## I. Introduction

In a typical low-energy building design process, multiple design configurations regarding building envelope and energy systems such as insulation levels, window to wall ratios, air tightness, heating and cooling systems and photovoltaic systems are considered to find an optimal design. It is well understood that occupant behavior and weather conditions are among the major factors that influence building performance (Hoes et al. 2009; de Wilde and Tian 2009; Guerra-Santin and Itard 2010; de Wilde and Coley 2012; Yan et al. 2015; Tian et al. 2018), especially in low-energy buildings (McLeod, Hopfe, and Kwan 2013; Rysanek and Choudhary 2013; Van Gelder, Janssen, and Roels 2014). When predicting building performance in the design phase, uncertainties in occupant behavior and weather conditions can therefore result in deviations between the predicted performance and the operational performance (de Wilde 2014; Gram-hanssen and Georg 2017). It is thus important to quantify the impact of these uncertainties during the design process (Woloszyn and Beausoleil-Morrison 2017) to reduce the performance gap between measured and predicted performance (de Wilde 2014) to ensure the desired performance not only in the near future but over the building's life-span (Fawcett et al. 2012). This quantification is also important to aid decision makers to make informed design decisions considering uncertainties and thus enhance confidence in design decisions (Østergård, Jensen, and Maagaard 2017). Furthermore, there is growing need for optimization of building performance under uncertainty considering a large number of uncertain factors in occupant behavior, climate change etc. in the design phase to reach robust optimal low-energy building designs (Tian et al. 2018).

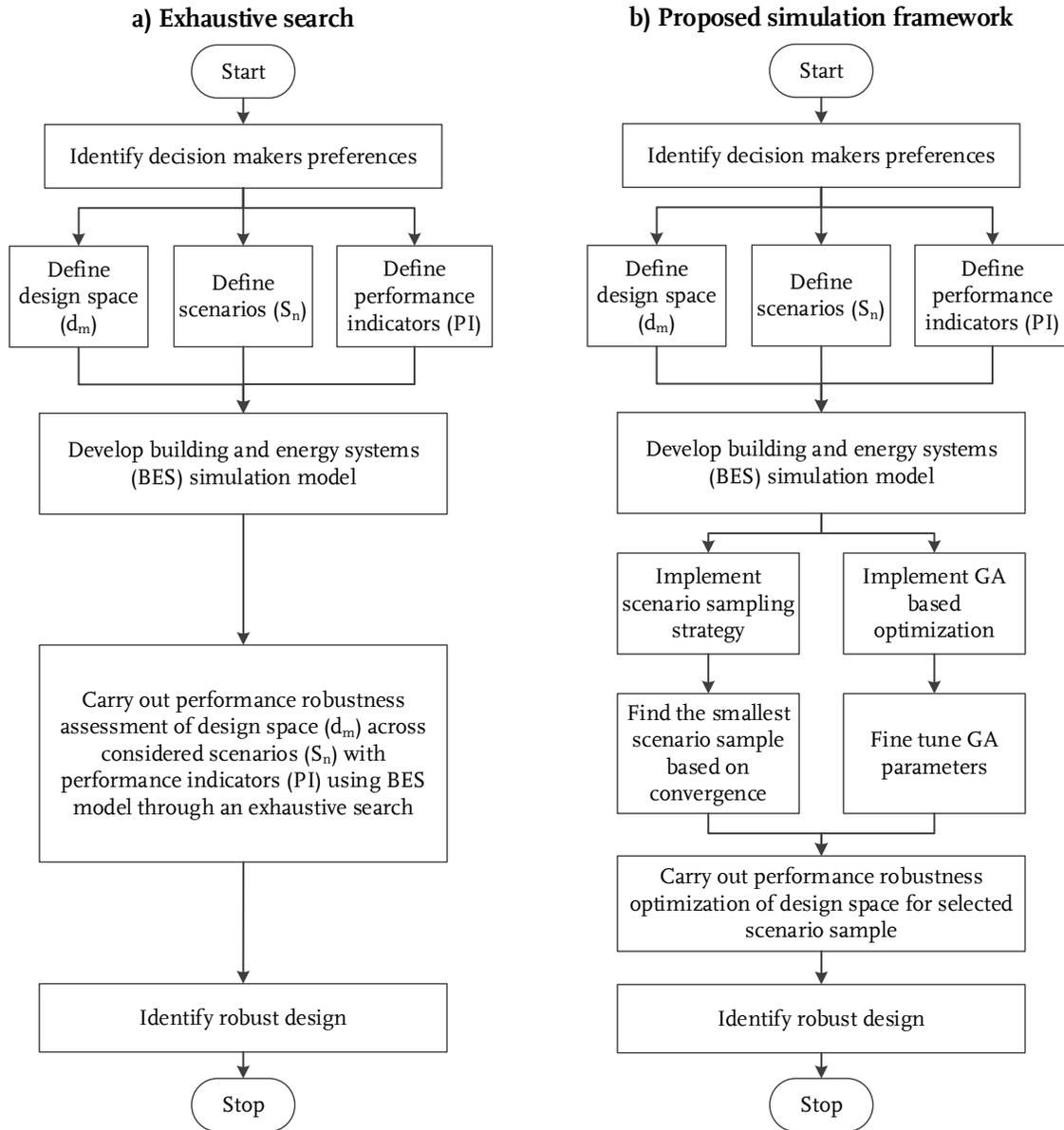
The probability of occurrence of many influential factors (like occupant behavior) is largely unknown over a building's life-span, and as such it is difficult for designers to quantify their impact. For those influential factors it is possible to use scenario analysis in order to understand the impact of their uncertainty (Kotireddy, Hoes, and Hensen 2018). In scenario analysis, alternative futures for the influential factors are formulated, which can be used to identify designs that perform well (robust performing) in these formulated futures (Moss et al. 2010). As such, integration of scenario analysis

into performance robustness assessment allows for a thorough investigation of uncertainties (Struck and Hensen 2013; Kim 2013). Accordingly, a non-probabilistic robustness assessment based on scenario analysis is used to identify robust designs (Hoes et al. 2011; Rysanek and Choudhary 2013; Kotireddy, Hoes, and Hensen 2018). The max-min method and the minimax regret method are widely used for robustness assessment using scenario analysis (Averbakh 2000; Aissi, Bazgan, and Vanderpooten 2009). Therefore, in this article, we present a simulation framework comprising of robustness assessment using scenario analysis to identify robust optimal low-energy building designs.

Generally, robust designs are identified by assessing the performance of multiple design configurations under a large number of future scenarios. When exploring a large design space this approach becomes computationally very expensive and infeasible in practice. In literature, sampling strategies are reported to reduce computational costs associated with running large sets of simulations (Macdonald 2009; Burhenne, Jacob, and Henze 2011; Hu and Augenbroe 2012; O'Neill and Eisenhower 2013). In our proposed simulation framework (Figure 1b), we also implemented a sampling strategy to find the smallest scenario sample that can predict similar performance as that of the full scenario sample (all possible scenario combinations).

The computational costs of the simulation framework can be further reduced by using optimization methods (Nguyen, Reiter, and Rigo 2014; Hamdy and Sirén 2016). Genetic algorithms (GA) are widely used in building design optimization (Evins 2013; Machairas, Tsangrassoulis, and Axarli 2014; Evins 2016). In our proposed framework (Figure 1b), we use a genetic algorithm (Deb et al. 2002) in order to find robust optimal solutions in a large design space. However, to include performance robustness as an objective in the multi-objective optimization, the robustness indicators such as maximum performance regret can only be calculated after the performance assessment is conducted for the entire design space for all scenarios (Kotireddy, Hoes, and Hensen 2017). Therefore, in this article, we show how the original fitness function of the genetic algorithm is modified to use the various robustness assessment methods. Furthermore, we show how the parameters of the genetic algorithm can be tuned

to the optimization problem in order to optimize the optimization process. Figure 1a shows the simulation process that can be followed in an exhaustive search to identify robust design solutions. Figure 1b shows the proposed simulation process which makes use of scenario sampling and a genetic algorithm to reduce the computational cost of the search.



**Figure 1** Simulation process flow implemented in this work to reduce computational costs compared to an exhaustive search in identifying robust designs.

This paper is organized as follows. Section 2 describes the adoption of different robustness assessment methods that are commonly reported in general literature. Section 3 describes a case study used for the demonstration of the simulation framework. The simulation framework is presented in Section 4. The integration of robustness indicators in multi-objective optimization is also presented in this section. The improvement of the computational efficiency of the simulation framework achieved through the use of scenario sampling strategies and multi-objective optimization is presented in Section 5. The developed simulation framework is validated in Section 6 and the conclusions of this study are summarized in Section 7.

## **2. Robustness assessment methods**

Generally, in a building design project, decision makers with different attitudes towards risk acceptance are involved. Therefore, to cater for the needs of different attitudes towards risk acceptance by decision makers, it is important to identify appropriate robustness assessment methods (Kotireddy, Hoes, and Hensen 2017). Different robustness assessment methods were reviewed from other fields (Averbakh 2000; Aissi, Bazgan, and Vanderpooten 2009; Polasky et al. 2011; Xidonas et al. 2017) and it was found that the max-min method (Wald 1945) and the minimax regret method (Savage 1951) are commonly used for robustness assessment using scenario analysis (Averbakh 2000; Aissi, Bazgan, and Vanderpooten 2009). The max-min method is a conservative approach, as it yields a robust design that has the least variations across extreme scenarios. Conversely, the minimax regret method is a less conservative approach (Aissi, Bazgan, and Vanderpooten 2009) as it yields a robust design that performs as closely as possible to the optimal performance for every scenario (Kotireddy, Hoes, and Hensen 2018). Therefore, these two robustness assessment methods are selected in this work to address conservative and non-conservative approaches in the design decision making process. Adoption of these methods in the building performance context is described below.

## 2.1 The max-min method

In this method, the spread of a performance indicator is used as a robustness indicator of a design, and is defined as the difference between maximum performance and minimum performance across all scenarios. Using this method, the following steps are implemented in the present study to identify the most robust design of a design space across the considered scenarios.

1. Assess the performance of a design for all scenarios ( $S_n$ ) using a performance indicator (PI).
2. Find the maximum and minimum performance of a design across all scenarios, as shown in Table 1.
3. Calculate the spread of a design across all scenarios. The spread is the performance difference between the maximum and minimum performance, as shown in Table 1.
4. Repeat steps 1-3 for all designs.

The spread is used as a measure of robustness, and the design that has the smallest spread is the most robust solution in a design space. Ideally, the design with zero spread is the most robust solution of a design space.

**Table 1** Calculation of performance robustness (spread) using the max-min method.

Designs	Scenarios				Maximum performance (PImax)	Minimum performance (PImin)	Performance spread (PImax-PImin)
	$S_1$	$S_2$	...	$S_n$			
$d_1$	$PI_{11}$	$PI_{12}$	...	$PI_{1n}$	$\max(PI_{11}, PI_{12}, \dots, PI_{1n})$	$\min(PI_{11}, PI_{12}, \dots, PI_{1n})$	$PImax_1 - PImin_1$
$d_2$	$PI_{21}$	$PI_{22}$	...	$PI_{2n}$	$\max(PI_{21}, PI_{22}, \dots, PI_{2n})$	$\min(PI_{21}, PI_{22}, \dots, PI_{2n})$	$PImax_2 - PImin_2$
...	...	...	...	...	...	...	...
$d_m$	$PI_{m1}$	$PI_{m2}$	...	$PI_{mn}$	$\max(PI_{m1}, PI_{m2}, \dots, PI_{mn})$	$\min(PI_{m1}, PI_{m2}, \dots, PI_{mn})$	$PImax_m - PImin_m$
<b>The most robust design</b>							$\min(PImax - PImin)$

## 2.2 The minimax regret method

In this method (Savage 1951), for a given scenario, performance regret is the performance difference between a design and the optimal design in that scenario (Aissi, Bazgan, and Vanderpooten 2009). The maximum performance regret of a design across all scenarios is the measure of its robustness. Using this method, the following steps are implemented in the present study to select the most robust design of a design space across the considered scenarios.

1. Assess the performance of designs ( $d_m$ ) for all scenarios ( $S_n$ ) using a performance indicator (PI).
2. Find the optimal design for each scenario by comparing the performance of all designs. In this work, we assume that the optimal design is the one with the minimum value of a PI for a scenario.

**Table 2** Calculation of performance robustness (maximum regret) using the minimax regret method.

Designs	Scenarios				
	$S_1$	$S_2$	...	$S_n$	
$d_1$	$PI_{11}$	$PI_{12}$	...	$PI_{1n}$	
$d_2$	$PI_{21}$	$PI_{22}$	...	$PI_{2n}$	
...	...	...	...	...	
$d_m$	$PI_{m1}$	$PI_{m2}$	...	$PI_{mn}$	
Minimum performance for each scenario (A)	$A_1 = \min (PI_{11}, PI_{21}, \dots, PI_{i1}, PI_{m1})$	$A_2 = \min (PI_{12}, PI_{22}, \dots, PI_{i2}, PI_{m2})$	...	$A_n = \min (PI_{1n}, PI_{2n}, \dots, PI_{in}, PI_{mn})$	
	Performance regrets (R)				Maximum performance regret (Rmax)
	$S_1$	$S_2$	...	$S_n$	
$d_1$	$R_{11} = PI_{11} - A_1$	$R_{12} = PI_{12} - A_2$	...	$R_{1n} = PI_{1n} - A_n$	$R_{max1} = \max (R_{11}, R_{12}, \dots, R_{1n})$
$d_2$	$R_{21} = PI_{21} - A_1$	$R_{22} = PI_{22} - A_2$	...	$R_{2n} = PI_{2n} - A_n$	$R_{max2} = \max (R_{21}, R_{22}, \dots, R_{2n})$
...	...	...	...	...	...
$d_m$	$R_{m1} = PI_{m1} - A_1$	$R_{m2} = PI_{m2} - A_2$	...	$R_{mn} = PI_{mn} - A_n$	$R_{maxm} = \max (R_{m1}, R_{m2}, \dots, R_{mn})$
<b>The most robust design</b>					$\min(R_{max})$

3. Calculate the performance regret ( $R$ ) of a design for each scenario, as shown in Table 2. The regret is the performance difference between the design and the optimal design for a scenario.
4. Find the maximum performance regret for each design across all considered scenarios (see Table 2).

The maximum performance regret is the measure of robustness; the lower the maximum performance regret, the higher the robustness. Therefore, the most robust design is the design with the lowest maximum performance regret, as shown in Table 2.

In summary, it can be noted that only the scenarios that cause extreme performance are considered for robustness assessment in the max-min method. In the minimax regret method, the performance of all designs across a scenario is compared to find the optimal design, and the performance regret of other designs is the difference between a design and optimal design in that scenario. Therefore, the evaluation of performance regret includes inter-comparison of performance of other designs and the maximum performance regret is calculated only after conducting the performance assessment of the entire design space. It is noteworthy that for both robustness assessment methods, the preferred robust design is based on optimal performance and the lowest value of robustness indicator calculated using the corresponding method.

### **3. Case study**

The implementation of the scenario sampling strategy and multi-objective optimization are carried out through a Dutch residential building case study, which is a semi-detached terraced house (see Figure 2) a typical Dutch residence (Agentschap NL 2013). The demonstration is carried out with a policymaker as a decision maker. Two performance indicators, CO<sub>2</sub> emissions and additional investment cost (IC<sub>a</sub>), based on the preferences of a policymaker, are considered for demonstration. The case study details and the description of the design space, scenarios and performance indicators can be found in (Kotireddy, Hoes, and Hensen 2017). The case study design space and scenarios considered for the

sampling strategy and multi-objective optimization are shown in Table 3 and Table 4 respectively. The design space is defined by varying building envelope and energy system properties to arrive at multiple low-energy building configurations. In addition, these building configurations meet various Dutch building codes and regulations (RVO 2015; RVO 2016).



**Figure 2** Example of a Dutch terraced house (Agentschap NL 2013).

Scenarios are defined considering uncertain and influential parameters that can impact the preferred performance indicators over the building's lifespan. In this case study, scenarios are used as formulated alternatives because in practice probabilities of uncertainties are unknown beforehand in most of the building design projects. The integration of such approaches in building performance predictions can provide a better understanding of the impact of uncertainties and also facilitate decision making during the design selection process with the goal of choosing a design that is robust to a variety of possible future situations (Moss et al. 2010). These scenarios constitute occupant, usage, and climate scenarios and they are defined explicitly (discrete scenarios) considering all possible future situations (Aissi, Bazgan, and Vanderpooten 2009). Four occupant scenarios representing the potential occupants of the building over its life-span are formulated based on Dutch household statistics (CBS 2016a), which show 37% single person household, 33% of two-person household, 12% and 13% for three and four person

households, respectively. For each of the occupant scenarios, usage scenarios are formulated based on occupant behavior with respect to energy use in the building. These usage scenarios span very careful energy users to energy-wasting users; well-informed to poorly informed users and also cover different types of equipment with low to very high efficiencies. For usage scenarios, occupancy patterns, heating setpoint temperatures, lighting and appliance use, ventilation rates, domestic hot water consumption and shading control are varied from low to high values around an average usage scenario (Ministerie van VROM 2009; CBS 2016b; NEN7120+C2 2012; Hoes 2014). It is worth noting that some of these scenarios are varied together as they are inter-dependent (see Table 4). For instance, internal heat gains due to appliances and lighting depends on the usage of lighting and appliances. In addition to occupants and their behavior scenarios, five climate scenarios are considered. One is a typical climate reference year, NEN 5060, which is based on average months of 20 years of historical weather data (NEN 2008). Others are climate change scenarios, which represent an increase of global temperatures in 2050 relative to 1990 (van den Hurk et al. 2006).

Ideally, the performance and robustness of the design space should be assessed for all scenario combinations as the likelihood of any scenario combination is not known. There are 6 design options with 3240 combinations and 9 scenarios with 29160 combinations. Performance assessment of this design space (Table 3) across all scenario combinations (Table 4) would require 94 million simulations, but this assessment is computationally very expensive. Therefore, to reduce this number of simulations, the scenario sampling strategy and optimization methods are implemented, which are discussed in the following sections. Full factorial simulations (design options  $\times$  all scenario combinations) is used as a reference to calculate savings in computational costs.

Building and energy systems models for the case study are developed in TRNSYS. MATLAB is used as a process integrator that couples building and energy system models, and is used as a platform to carry out multi-objective optimization of the design space for the considered scenarios using a multi-objective optimization genetic algorithm from the MATLAB optimization tool box (MathWorks 2016).

**Table 3** Design options of the case study considered to implement scenario sampling and multi-objective optimization in order to enhance the computational efficiency of the simulation framework.

Design variant	Options
Building envelope properties (Rc-wall/roof/floor, m <sup>2</sup> k/W; Windows U value W/m <sup>2</sup> K)	[4.5/6/3.5, 6/7/5, 7/8/6, 9/9/7, 10/10/10; 1.43, 1.01, 0.81, 0.68, 0.4]
WWR (%)	[20, 40, 60]
Thermal mass	[Light-weight, Medium-weight, Heavy-weight]
Infiltration, ach	[0.12, 0.24, 0.36, 0.48]
PV system, m <sup>2</sup>	[5, 10, 15, 20, 25, 30]
Solar DHW system, m <sup>2</sup>	[0, 2.5, 5]

**Table 4** Scenarios of the case study considered to implement the scenario sampling and multi-objective optimization to enhance the computational efficiency of the simulation framework.

Scenario parameter	Options
<i>Occupant scenarios</i>	
Household size	[1, 2, 3, 4]
<i>Occupant behavior (usage) scenarios</i>	
Heating setpoint (occupied), °C	[18, 20, 22]
Heating setpoint (un-occupied) *, °C	[14, 16, 18]
Occupancy profile	[Evening, All-day]
Average electricity use for lighting, W/m <sup>2</sup>	[1,2,3]
Average electricity use for appliances, W/m <sup>2</sup>	[1,2,3]
Internal heat gains due to lighting and appliances*, W/m <sup>2</sup>	[2, 3, 4, 5, 6]
Domestic hot water consumption, l/person per day	[40, 60, 100]
Ventilation, ach	[0.9, 1.2, 1.5]
Shading control ON if radiation is above, W/m <sup>2</sup> and if T <sub>indoor</sub> >24°C	[250, 300, 350]
Shading control OFF if radiation is below*, W/m <sup>2</sup> and if T <sub>indoor</sub> <24°C	[200, 250, 300]
<i>Climate scenarios</i>	
Reference climate and climate change scenario	[NEN5060-2008, G, W, G+, W+]

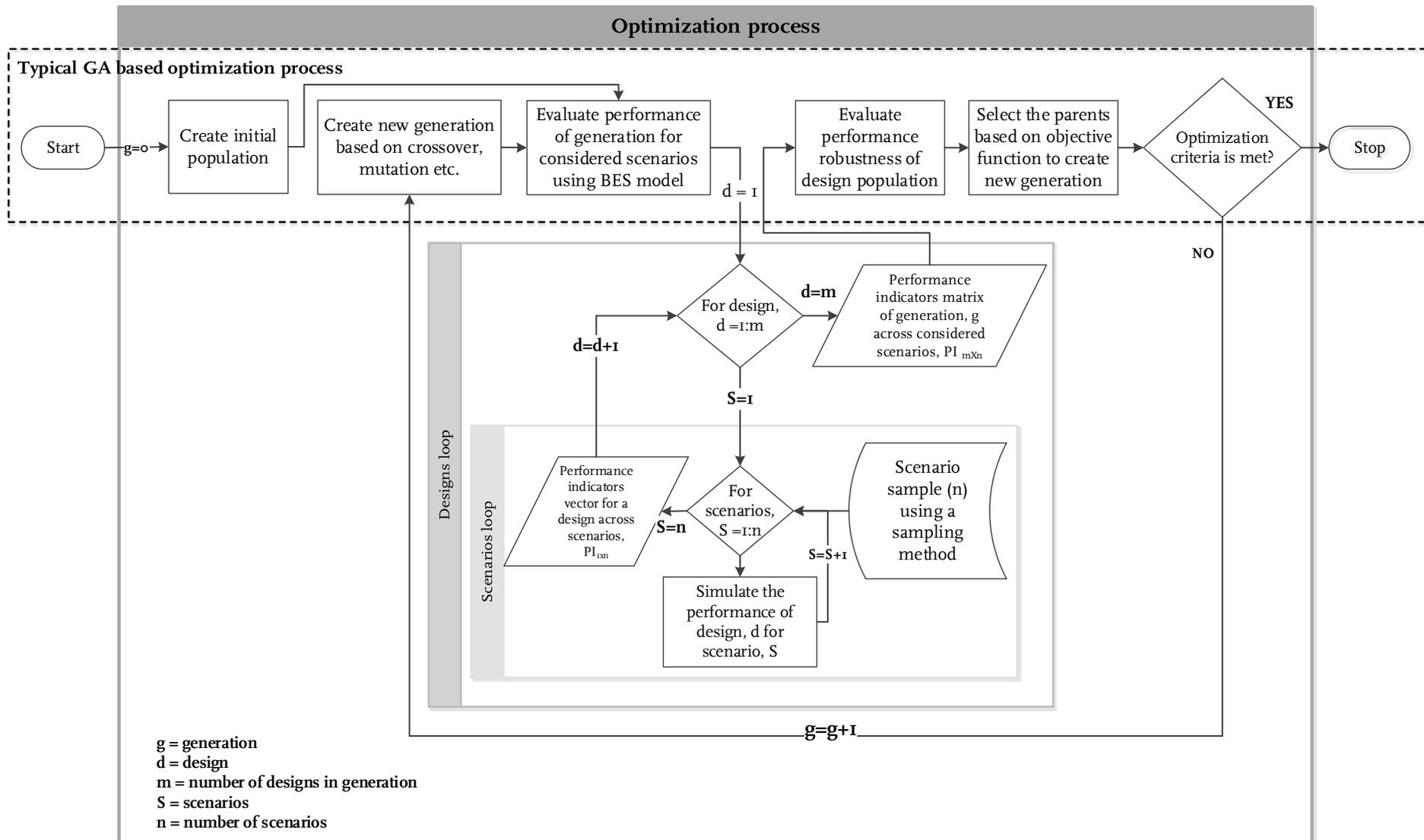
\* This scenario is varied together with the previous scenario.

## 4. Integration of robustness indicators into the optimization framework

### 4.1 Simulation framework

In a typical GA based optimization process, the GA creates a new generation through cross-over and mutation of the previous generation, then the objective function is evaluated for each individual (i.e., each design solution) in the new generation. This optimization process can also include scenario analysis as is shown in Figure 3. For considered scenarios, the performance and robustness of each design in a generation is assessed using a building and energy systems simulation model (BES). The process continues until the optimization criterion is met. However, this optimization process should be different when the maximum performance regret is included as an objective in the multi-objective optimization, since the maximum performance regret can only be calculated after all designs in the current generations are evaluated for the considered scenarios (Kotireddy, Hoes, and Hensen 2017). Therefore, the fitness function (objective function) has to be defined in such a way that the optimization process halts after every generation until the calculation of the performance robustness is finished. Hence, the optimization process in the current study is nested across three loops (see Figure 3) as discussed below, to ease calculation of robustness indicators that require pausing of the GA algorithm:

1. **Main loop** – In this loop, population of design alternatives is updated for different generations based on objectives. Robustness indicators are also calculated in this loop.
2. **Designs loop** – This is a sub loop of the main loop, where the performance of a design population across considered scenarios is calculated and the performance indicators matrix from this loop are returned to the main loop.
3. **Scenarios loop** – This loop is a sub loop of the designs loop, where the performance of each design is assessed for each scenario and the performance indicator vector of a design across the considered scenarios is returned to the designs loop.



**Figure 3** The extended GA based optimization implemented in this study compared to a typical GA based optimization (indicated in dotted line) for performance robustness optimization using scenario analysis.

In this optimization process, for a particular generation, the performance of a design is calculated for the considered scenarios in the scenario loop and the performance indicator vector of a design is returned to the designs loop. In order to evaluate the performance of a design population across the considered scenarios, the scenario loop needs to be nested within the designs loop. As a result of this nesting, the performance indicator's matrix of design population across considered scenarios is returned to the main loop, where the robustness assessment method is applied. Based on the predicted performance and performance robustness, the design space for the new generation is updated by the genetic algorithm. This process continues until the optimization criterion is met. In this work, the optimization process stops if the average relative change in the best fitness function value over 20 generations is less than 0.001. This stopping criterion is tested with the true Pareto front resulting from full factorial analysis.

#### *4.2 Modifying fitness function of GA by storing design archive*

In order to find a robust design within a design space, the robustness of all designs are compared and the most robust design is the design with the lowest or ideally zero spread/maximum regret using the max-min/minimax regret methods. As noted earlier, using the max-min method, the robustness of each design is calculated separately without any inter-comparison of performance of other designs. Therefore, performance and robustness can be calculated simultaneously and the most robust design in a generation could be the most robust design in the entire design space using the max-min method. In contrast, using the minimax regret method, the robustness of each design is calculated with inter-comparison of the performance of all designs in a design space and the most robust design is the design with the lowest maximum regret in the entire design space. To enable inter-comparison of all designs, designs in the current and previous generations should be stored. This requires some modifications to a standard GA fitness function to store all designs in a design archive. Without this design archive, the optimization would yield completely different Pareto fronts. This is illustrated with an example in Figure 4, which compares the Pareto front of an optimization run with a standard fitness function and the modified fitness function.

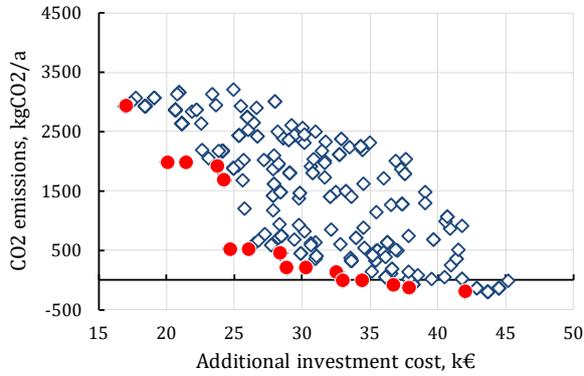
In the case of the standard fitness function, maximum performance regret is calculated for each generation without storing any design archive of previous generations, which thus results in zero maximum performance regret for at least one design in each generation. The same can be observed from Figure 4(a) that there are many designs with zero maximum regret of CO<sub>2</sub> emissions, especially in the Pareto front where all designs, except designs with IC<sub>a</sub> of 23.7 k€ and 30.2k€, have zero maximum regret of CO<sub>2</sub> emissions. This is because the performance regret is calculated based on the optimal design for a scenario and for the first design in each generation the regret is always zero as there is no other design to compare with. Therefore, using the standard fitness function, there are as many as 17 most robust designs (with zero maximum regret of CO<sub>2</sub> emissions) in the IC<sub>a</sub> range of 17-45 k€. Generally, these designs may not be the most robust when compared to other designs in each generation and also with the entire design archive, as seen in the Pareto front with the modified fitness function in Figure 4(b). For instance, it can be inferred from Figure 4 that designs with IC<sub>a</sub> of 17 k€ and 20 k€ have zero maximum regrets of CO<sub>2</sub> emissions with a standard fitness function (Figure 4(a)), whereas these designs' maximum performance regrets are above 3000 kgCO<sub>2</sub>/a with the modified fitness function (Figure 4(b)). Similar observations can be made for designs with IC<sub>a</sub> of 21.5 k€, 24.2-28.8 k€ and 32.5-42 k€. Furthermore, each design has different maximum regrets of CO<sub>2</sub> emissions (e.g. design with an IC<sub>a</sub> of 17 k€ has maximum regrets of 0-3671 kgCO<sub>2</sub>/a) across different generations with a standard fitness function as seen in Figure 4(a).

Using the modified fitness function, the maximum performance regret of a design is calculated and updated after every generation by comparing performance of all designs of current and previous generations (design archive). This update of maximum regret by inter-comparison results in actual maximum performance regret of a design within a design space and thus results in only one most robust design (IC<sub>a</sub> of 44.48 k€) of the entire design space. Therefore, the design archive of previous generations must be stored by the GA and the maximum performance regret of each design should be updated before proceeding to the next generation (Figure 5). This update of maximum regret at the end of each generation cannot be done in a straightforward approach by typical optimization software tools. In the modified fitness function, the GA pauses after every

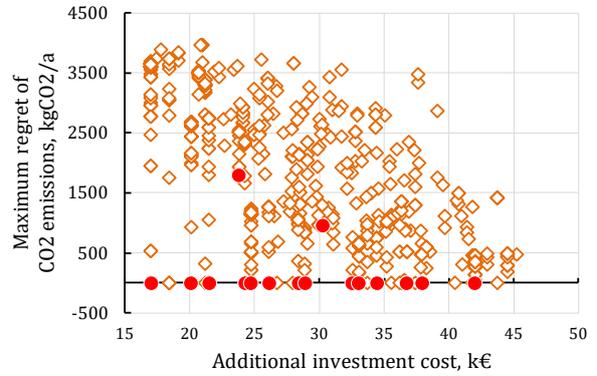
generation to update maximum regret as well as store the design archive of previous generations.

This optimization process is also shown in Figure 5.

**a) Standard fitness function**

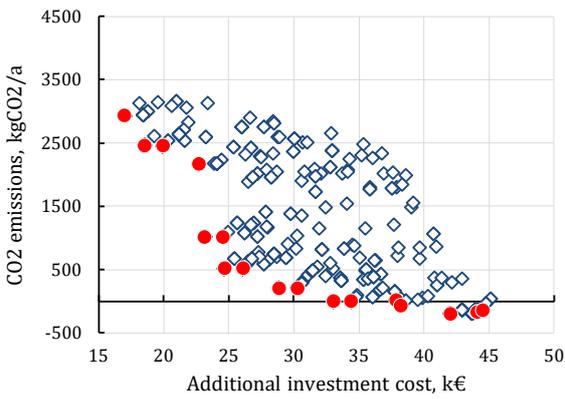


◇ All solutions of an optimization run ● Pareto solutions

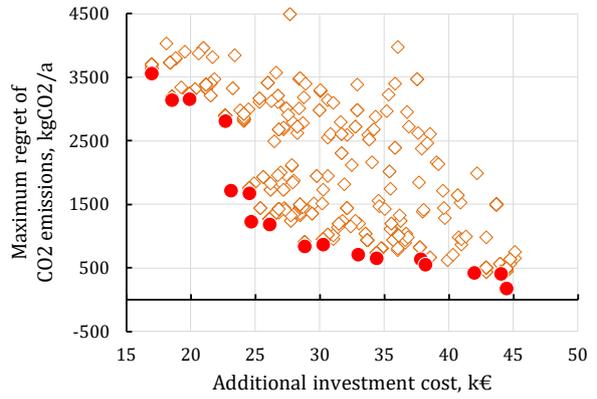


◇ All solutions of an optimization run ● Pareto solutions

**b) Modified fitness function**

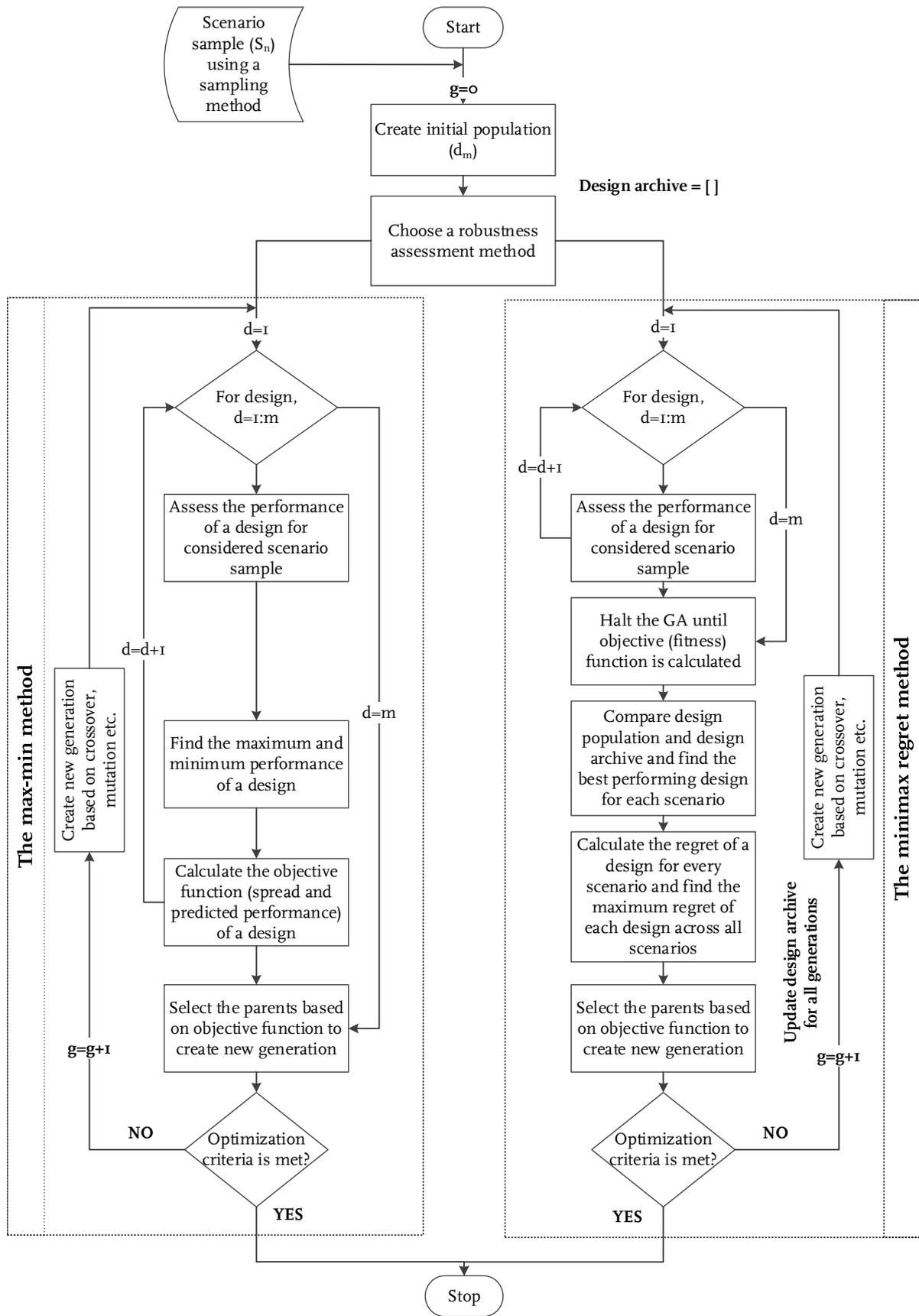


◇ All solutions of an optimization run ● Pareto solutions



◇ All solutions of an optimization run ● Pareto solutions

**Figure 4** The Pareto front and design archive of an optimization run with maximum performance regret as an objective in the fitness (objective) function. a) A standard fitness function b) The modified fitness function.



**Figure 5** A multi-objective optimization approach considering multiple performance indicators and robustness indicators calculated using both robustness assessment methods.

## 5. Improving simulation framework efficiency

The improvement in computational efficiency using a scenario sampling strategy and multi-objective optimization is presented in this section and the computational cost savings from these methods are tabulated at the end of this section.

### 5.1 Scenario sampling

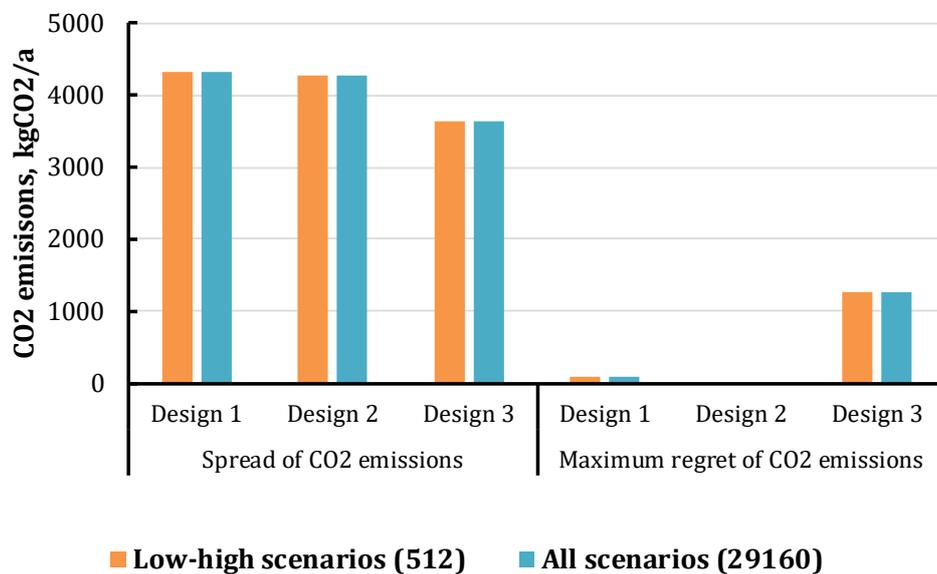
#### *i. All scenario combinations vs low-high scenario combinations*

In a conventional approach, the sampling strategy is selected based on convergence i.e., mean performance and variance (Janssen 2013). However, for performance robustness assessment, the performance range or distribution is also crucial in selecting a sampling strategy and determining its smallest sample size. Low-high scenario combinations are generally sufficient for performance robustness assessment (Kotireddy, Hoes, and Hensen 2018) because the low-high scenario combinations typically result in a performance range, and performance with the remaining scenario combinations might be within this range. Therefore, for low-high scenario combinations the sampling strategy is used instead of investigating all scenario combinations. To justify this selection, the performance robustness of three designs (Table 5), selected from the design space presented in Table 3, is assessed with all scenario combinations and low-high scenario combinations to evaluate if low-high scenario combinations are sufficient for the performance robustness assessment for multiple performance indicators.

**Table 5** Selected designs from the design option space to implement scenario sampling strategies.

Design parameter	Design 1	Design 2	Design 3
Rc (Wall/Roof/ Floor), m <sup>2</sup> K/W	4.5/6.5/3.5	6/7/5	10/10/10
Windows, W/m <sup>2</sup> K	1.43	1.01	0.4
WWR (%)	40	40	40
Thermal mass	Heavy-weight	Heavy-weight	Heavy-weight
Infiltration, ach	0.48	0.24	0.12
PV system, m <sup>2</sup>	30	25	15
SDHW system, m <sup>2</sup>	2.5	2.5	5

The robustness of CO<sub>2</sub> emissions of three designs is calculated using the two robustness assessment methods and results are presented in Figure 6. It can be observed that *low-high scenario* and *all scenario* combinations result in similar calculated robustness for both methods. Therefore, it can be concluded that *low-high scenario* combinations are sufficient for performance robustness assessment. This approach would itself save about 98% of computational costs, but would still require a total of 1.6 million simulations (design options × low-high scenario combinations). Therefore, it is still necessary to reduce the number of scenario combinations. It is expected that not all scenario combinations will influence the building performance in the same way. Scenario combinations that are not influencing the performance could be discarded. Typically, a sensitivity analysis would be a suitable method to identify these scenario combinations. However, it seems impractical to perform a sensitivity analysis for each design in the considered design space. Therefore, a sampling strategy based on Monte Carlo sampling is proposed in the next section. It aims to find the smallest sample size of scenario combinations that predicts similar performance to that of all *low-high scenario* combinations.

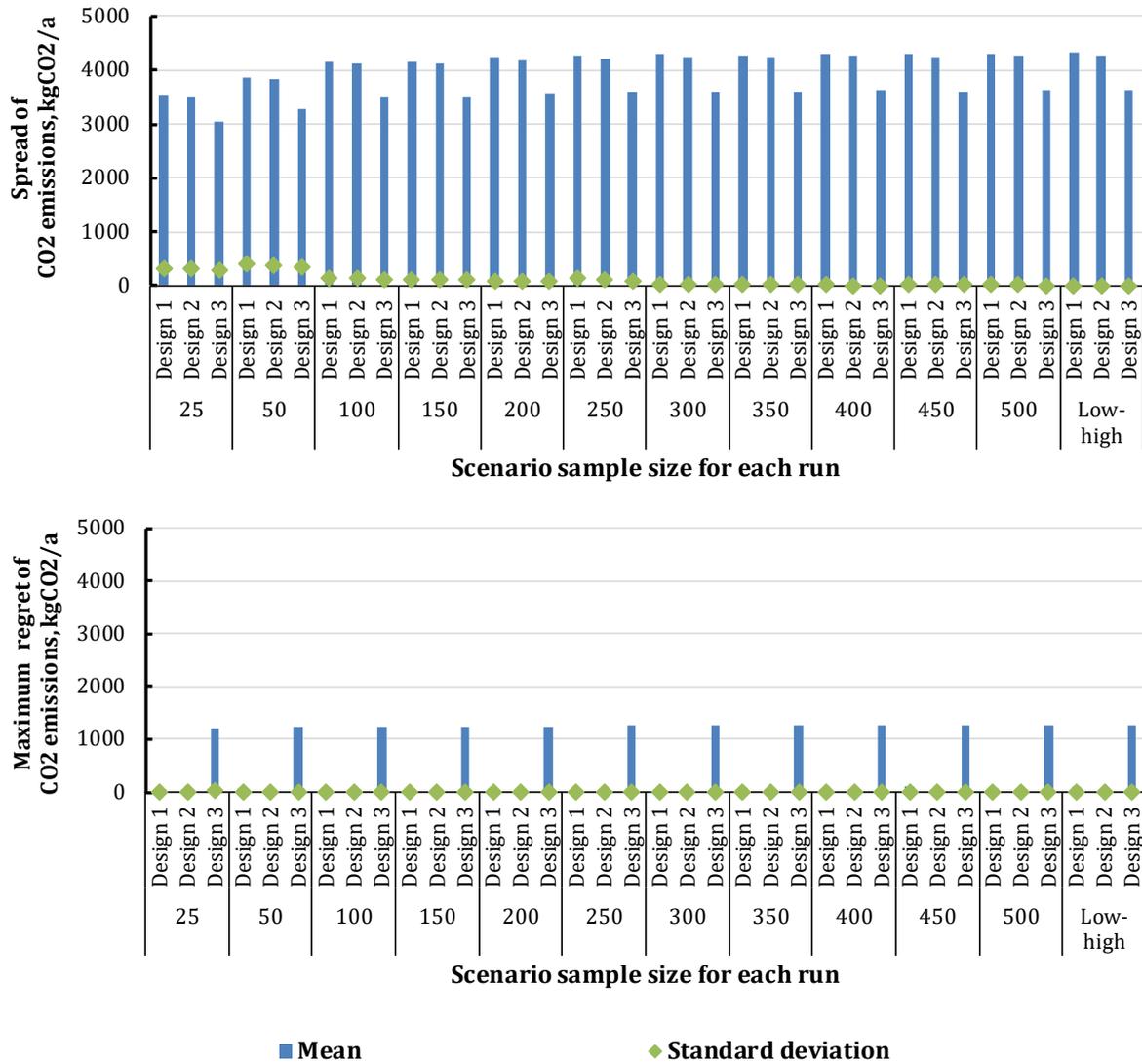


**Figure 6** Comparison of performance robustness of CO<sub>2</sub> emissions of three designs calculated using both robustness assessment methods for all scenario combinations (29160) and low-high scenario combinations (512).

ii. *Latin hypercube sampling*

Commonly used Monte Carlo sampling strategies in building performance simulations are random sampling, Sobol sampling and Latin hypercube sampling (LHS) (Macdonald 2009; Burhenne, Jacob, and Henze 2011). In random sampling, the sample is generated according to a random distribution, which results in clusters and gaps. These clusters and gaps are avoided in the SOBOL method as samples are generated as uniformly as possible (Burhenne, Jacob, and Henze 2011). LHS has fast convergence (Janssen 2013; Helton et al. 2006) compared to the other two methods. The sample is generated by stratification and input is divided into sub-intervals. Each interval has the same number of samples, which are randomly generated. Due to the efficient stratification of the LHS sampling method, small sample sizes are sufficient to achieve desired outcomes (Helton et al. 2006). This sampling efficiency can be further enhanced by using the uniform Latin hypercube (ULH) sampling method, because the ULH method provides desired outcomes even at smaller sample sizes (Janssen 2013). Thus, the ULH sampling is the preferred sampling method in this study.

The performance of the three previously described designs is assessed with different ULH samples sizes ranging from 25-500 scenario combinations. Due to the stochastic nature of the ULH sampling method, the performance assessment of these samples is carried out multiple (10) times to reduce stochasticity in sample generation. The smallest sample size that has a similar mean robustness indicator (across the multiple runs) as that of all *low-high scenario* combinations is preferred. Additionally, the standard deviation should be close to zero. Figure 7 shows the mean and the standard deviation of the robustness of CO<sub>2</sub> emissions for different ULH sample sizes across the multiple runs. It can be observed that for all ULH sample sizes, except 25 and 50, the standard deviation is close to zero and the mean is similar to that of all low-high scenario combinations. Hence, the ULH sample size of 100 scenario combinations is chosen for the performance robustness assessment in this study.



**Figure 7** Variation of mean and standard deviation of performance robustness of CO<sub>2</sub> emissions of three designs for different ULH scenario samples across multiple runs (10) compared to low-high scenario combinations (512).

## 5.2 Multi-objective optimization

By definition, the full factorial approach results in a true Pareto front for a design space, while the GA based optimization typically results in an approximation of the true Pareto front, which is inherent to the stochastic nature of the GA algorithm. The objective of this experiment is to find the best approximation of the true Pareto front using the GA based optimization method in the least possible number of iterations. GA parameter settings have a strong influence on the performance of the GA, and hence, it is important to determine the optimal settings to enhance computational efficiency in the process of converging to the Pareto front.

The main parameters of the genetic algorithm GA (Deb et al. 2002) are:

- **Population size** (PS) determines the number of individuals (designs) in a population at each generation.
- **Generations** (g) determine the number of evaluations in an optimization run.
- **Crossover fraction** (CF) determines the fraction of population at the next generation.
- **Pareto fraction** (PF) controls the elite members of the population for every generation to maintain the diversity of the population for convergence to an optimal Pareto front.
- **Selection function**, which is tournament size (TS) in the case of multi-objective optimization, determines how the GA selects the parents of the crossover members and selects the mutation members for the next generation.

The default values of these GA parameters in MATLAB are  $CF=0.8$ ,  $g=100 \times \text{number of design variables}$ ,  $PF=0.35$  and  $TS=4$ . The optimal values depend on the design space and fitness function and could be different for the two robustness assessment methods. Therefore, optimal values of the GA parameters are investigated below using the aforementioned case study.

The optimal settings of GA parameters are determined based on the following aspects:

- i. **Fast convergence:** Minimum number of iterations (defined as the product of generations and population size) required to meet the optimization criterion.
- ii. **Reaching the true Pareto front:** A high matching index (defined as the percentage of Pareto solutions for a certain set of a GA parameter values that matches the true Pareto solutions).

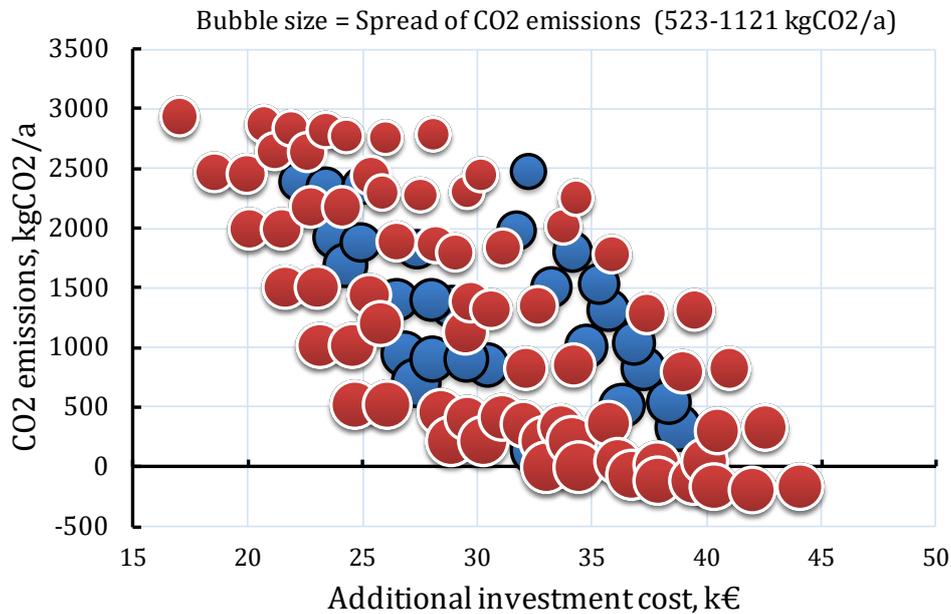
It is worth mentioning that uniform creation and mutation functions are considered in this study to avoid non-integer values of design variants. In addition, a uniformly distributed initial population which covers the design space uniformly is provided for all optimization runs.

*i. Default settings*

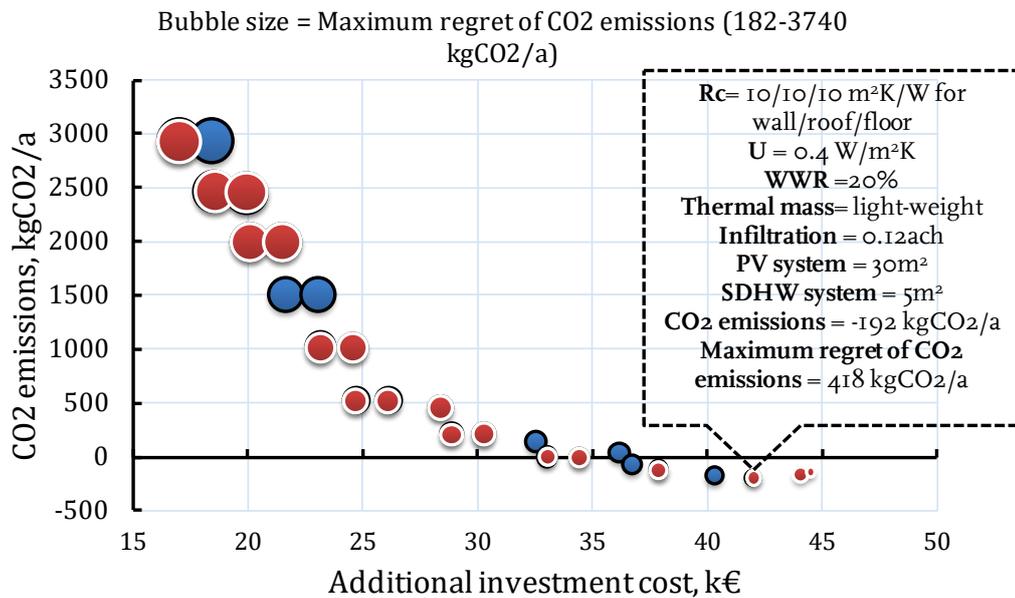
The Pareto fronts of the two robustness assessment methods with default MATLAB values of the GA parameters over multiple runs are compared with their corresponding true Pareto fronts, calculated using the full factorial approach. This comparison is presented in Figure 8.

Each bubble represents a design and bubble size depicts the robustness. The smaller the bubble size the more robust is the design. The maximum size of bubble is fixed and the bubble size is varied in proportion to the range of robustness indicator values. The details of a design (bubble) are shown in Figure 8. The blue bubbles represent the true Pareto front and the red bubbles represent the calculated Pareto front with default GA parameters values. A matching index of 66.6% and 69.5% is achieved with an average of 876 and 792 iterations over 5 runs using the max-min method and the minimax regret method, respectively. The results show that using the default settings there is a risk of losing about 30% of robust designs compared to the full factorial approach even though these settings save a considerable amount of computational cost. In the next section, it is investigated if the matching index can be improved using other parameter values.

## i. The max-min method



## ii. The minimax regret method



- True Pareto front calculated using full factorial approach
- Pareto front calculated using default GA parameters settings

**Figure 8** Comparison of the true Pareto front and a Pareto front with default GA parameter values (provided in MATLAB) for both robustness assessment methods. Blue bubbles represent the true Pareto front and red bubbles represent a Pareto front for the default GA parameter's settings. The details of a design are shown in the dotted box.

ii. *Optimal GA parameter values*

To find the optimal parameter values, the following steps are executed.

1. Define the range of the various GA parameters. In this case study, we use the following values per parameter: CF=[0.5, 0.6, 0.7, 0.8]; PF=[0.2, 0.3, 0.4, 0.5]; TS=[2, 3, 4, 5]; PS=[20, 30, 40]. This leads to a total of 192 parameter value combinations.
2. Run the optimization for every combination of GA parameter values.
3. Repeat the optimization process multiple times for every combination of GA parameter values to reduce the stochasticity effect of the GA. In this case study each combination is ran 5 times.
4. Calculate average number of iterations required to meet the optimization criterion and matching index for each combination of GA parameter values.
5. Repeat steps 2-4 for each robustness assessment method.
6. Find the optimal GA parameter values for each robustness assessment method; the combination of GA parameter values that has the highest matching index and the least number of iterations is considered the most optimal.

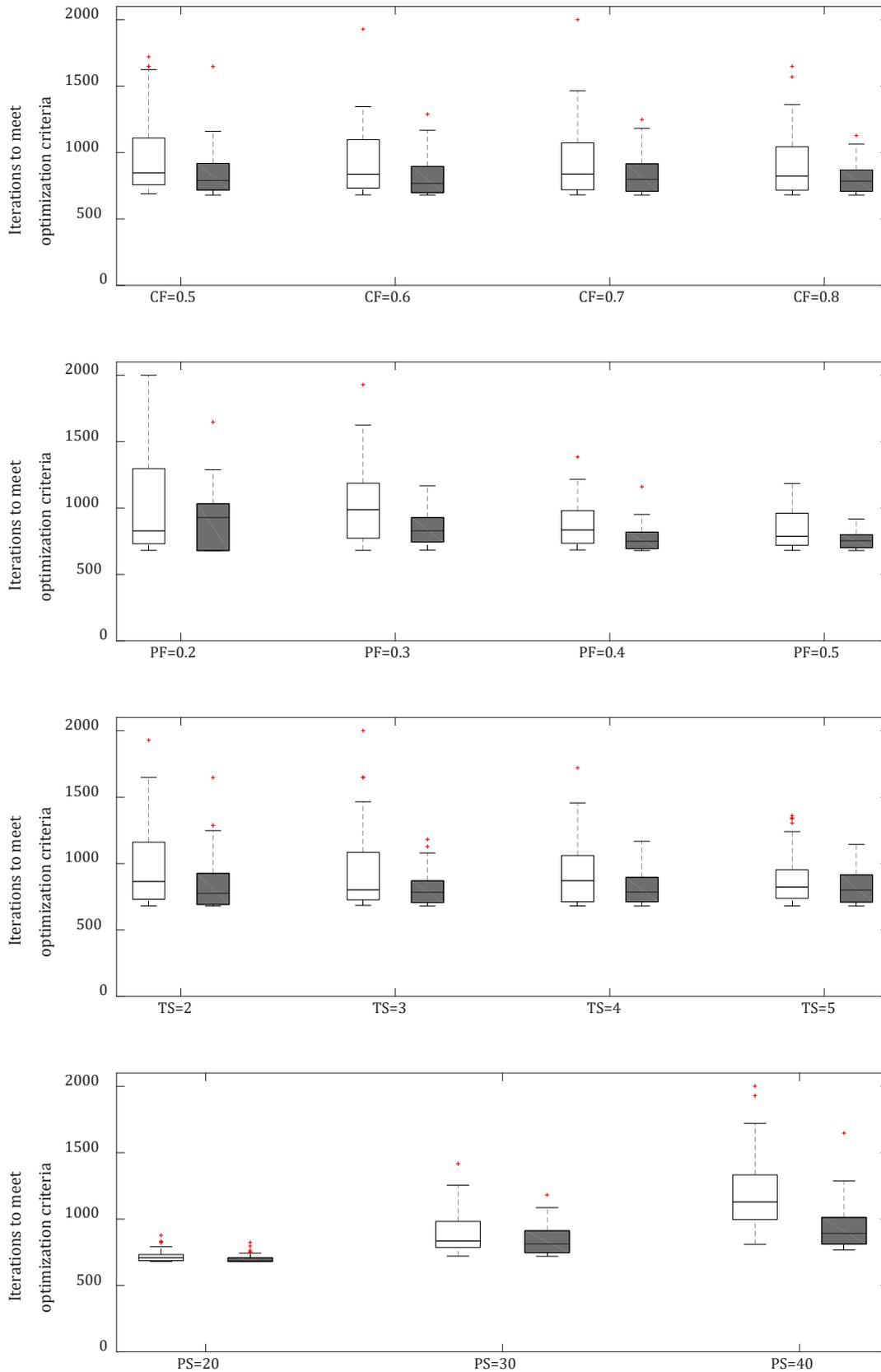
Note that the same stopping criterion is used in the optimization process for the two robustness assessment methods. The number of iterations required to meet the optimization criterion with different GA parameters for two robustness assessment methods is shown in Figure 9. Each box shows the results of one fixed parameter value (indicated below the box), while the other parameter values are varied. The matching index for each fixed parameter value is shown in Figure 10.

Figure 9 shows that the cross-over fraction (CF) values do not influence the required number of iterations across different values of other GA parameters (small range of box) for the minimax regret method. However, lower cross-over fractions reduce the range in the matching index, as seen in Figure 10. A small cross-over fraction results in the highest matching index for the minimax regret method. A CF of 0.7 results in the highest matching index for the max-min method. Therefore, CF of 0.5 and 0.7 are optimal for the respective methods. A high Pareto fraction leads

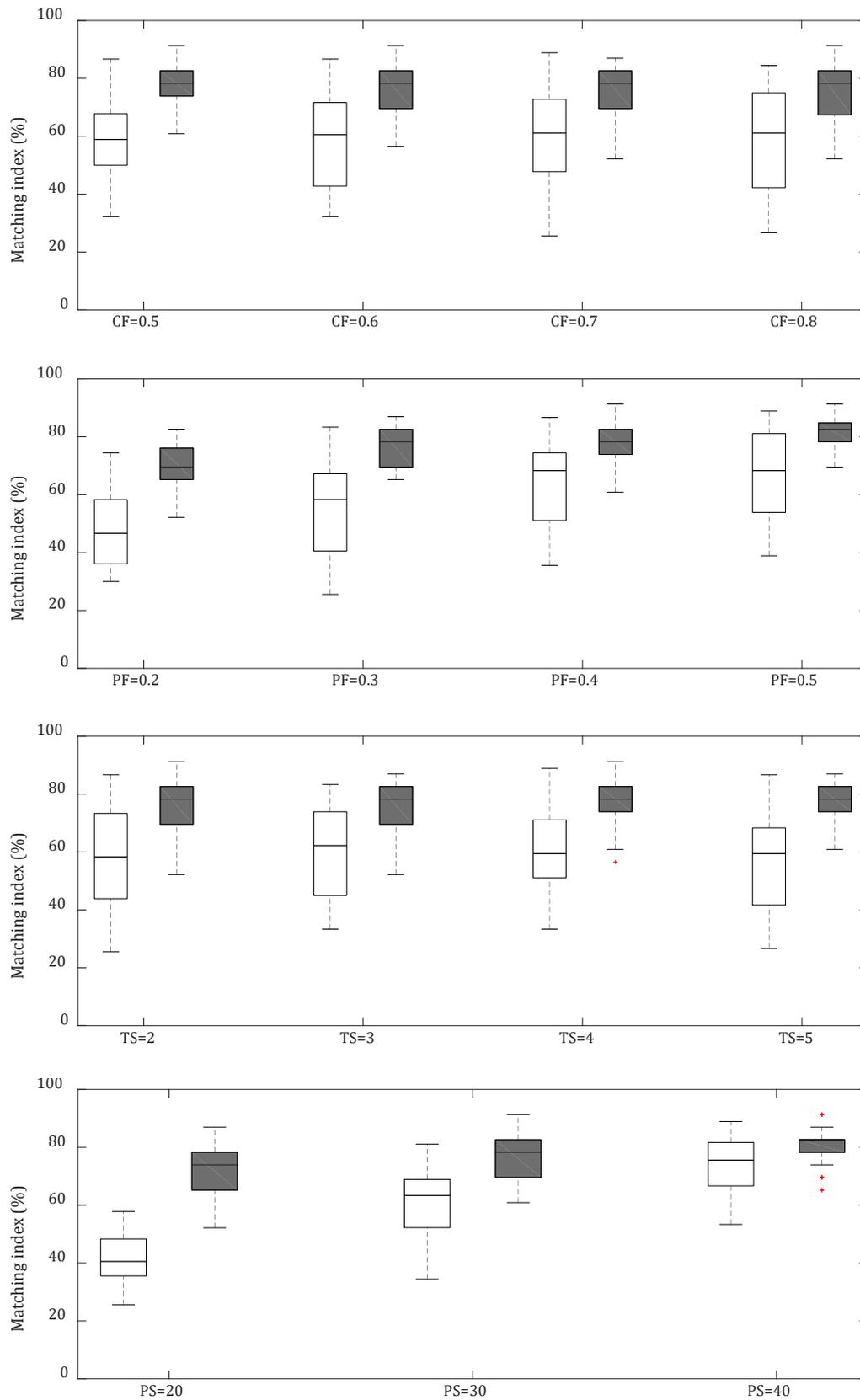
to faster convergence because optimization reaches a local optimum with high Pareto fractions. Contrariwise, a low PF requires more iterations as it tries to reach the global optimum. High Pareto fractions are optimal if the design archive is considered when evaluating objectives, which can be justified by a higher matching index (see Figure 10) for maximum regret with a PF of 0.5. Furthermore, in the case of high PF, other parameters have a limited effect (small range of boxplot as seen in Figure 9 and Figure 10) on the number of iterations required and on the matching index. It can be concluded that higher Pareto fractions and lower crossover fractions are optimal values for GA parameters for optimization using the minimax regret method.

It can be inferred from Figure 9 that larger values for the Tournament size (TS) reduce the range in the required iterations for the max-min method, while the matching index is not influenced that much. Figure 9 shows that the largest population size (PS=40) requires more iterations to meet the optimization criterion compared to the other population sizes for both methods. As expected, a population size of 20 requires the lowest number of iterations. However, the matching index with a population size of 20 is significantly lower compared to that of the population size of 40, as shown in Figure 10. A population size of 30 is an optimal trade-off between iterations and the matching index for the minimax regret method. A population size of 40 is optimal for the max-min method as the matching index with PS of 40 is significantly higher compared to other population sizes.

The matching index is improved up to 90% on average for both methods when the optimal values for the GA are used. In Section 5.3, the GA parameter values that we consider optimal for this case study are discussed. Also, the improvement in the matching index and the required computational cost compared to the MATLAB default values are discussed.



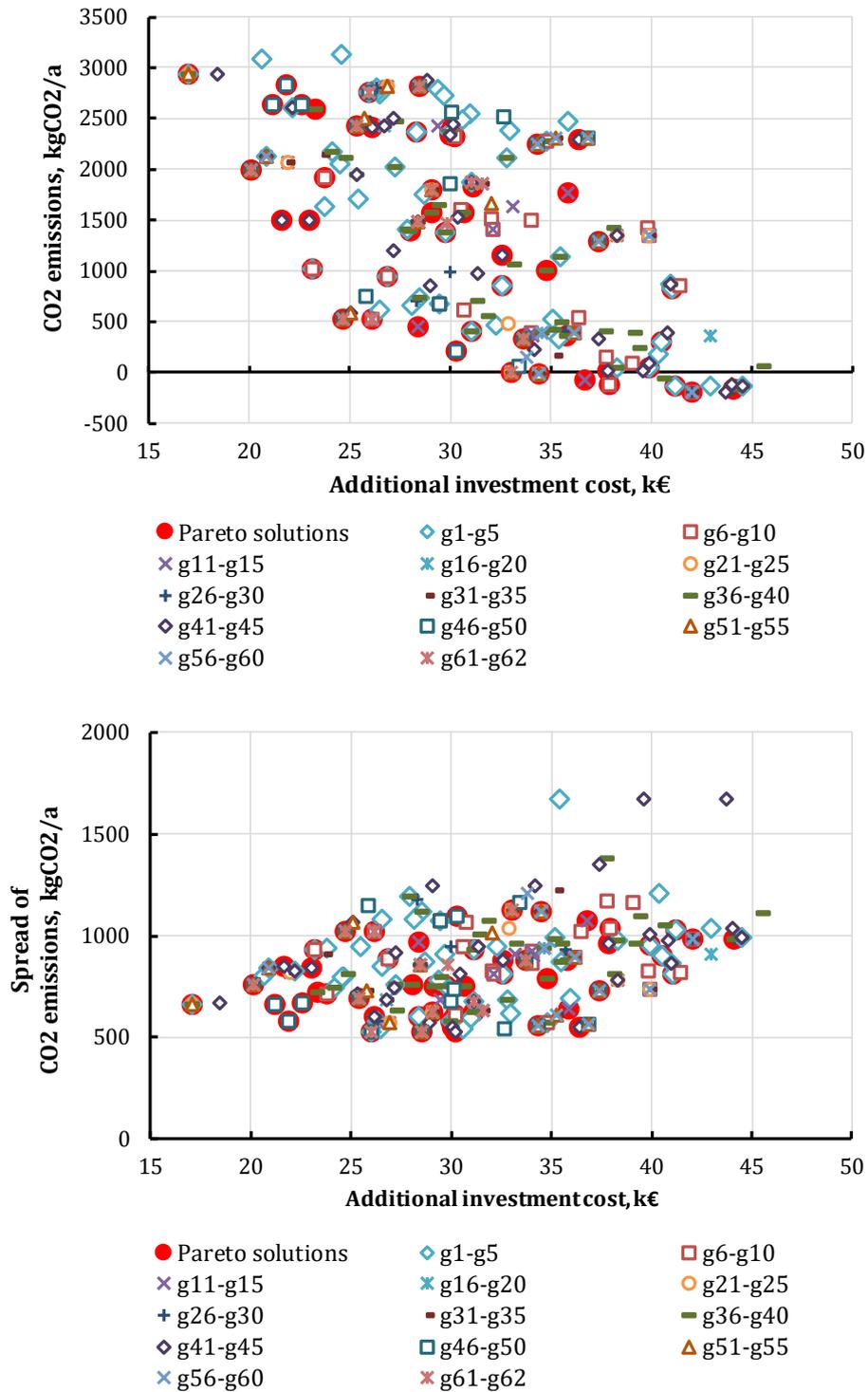
**Figure 9** Number of iterations required to meet the optimization criterion for different GA parameter values for both robustness assessment methods. White boxes represents the spread and filled boxes represent the maximum performance regret. Each box consists of all values of other parameters and multiple runs.



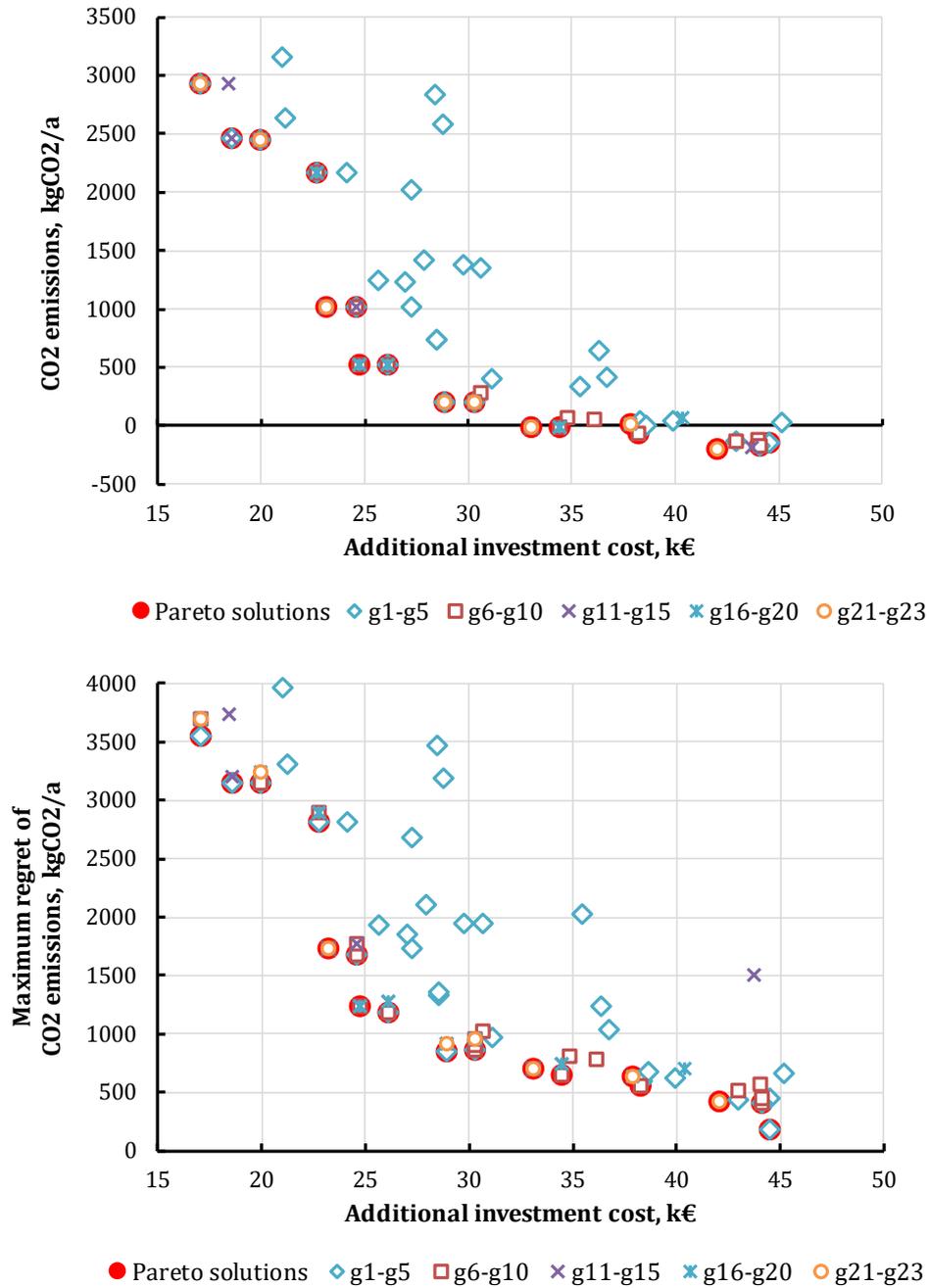
**Figure 10** Matching index of different GA parameters for both robustness assessment methods. White boxes represents the spread and filled boxes represent the maximum performance regret. Each box consists of all value of other parameters and multiple runs.

Optimization using the max-min method requires more iterations compared to the minimax regret method to meet the optimization criterion, as shown by the larger boxes in Figure 9. This is also shown in Figure 11 and Figure 12, which show optimization runs with default settings for both robustness methods. It is worth noting that optimization is carried out with default settings for both methods to allow a fair comparison. In these figures, the Pareto front grouped for five generations is shown separately for actual performance and performance robustness, and different scales are used for better visualization purposes. It can be observed that optimization using the max-min method took about 62 (1860 iterations) generations to converge, whereas the minimax regret method took only 23 (690 iterations) to converge. The difference in convergence rates for these two methods is because in the max-min method, robustness (spread) is optimized with respect to the best performing scenario of a design and there is no inter-comparison of designs. Therefore, actual performance and robustness do not necessarily follow the same trend, but often conflict, as observed in Figure 11. This means that, for example, the design with very high CO<sub>2</sub> emissions has the least spread across the scenarios. In addition, the spread of a design population of a particular generation is quite scattered. Therefore, the max-min method requires a higher number of generations to converge.

In contrast, in the minimax regret method, robustness (maximum performance regret) of a design is optimized with respect to the optimal design. The design with the optimal performance will have the least maximum performance regret, and thus both the actual performance and performance robustness follow the same trend, as demonstrated in Figure 12. Furthermore, for each generation, objectives are calculated considering the current population and the design archive of previous generations. Since the GA takes the design history into account, the population at the next generation depends on the entire design archive instead of on the previous generation. Thus, the relative change in the fitness function reduces as the Pareto front converges more quickly with an increase in generations, resulting in less iterations required to meet the optimization criterion. Therefore, the minimax regret method requires a lower number of iterations to converge compared to the max-min method.



**Figure 11** Variation of the Pareto front across different generations of an optimization run with default settings using the max-min method. The top graph shows actual performance and the bottom graph shows robustness (performance spread).



**Figure 12** Variation of the Pareto front across different generations of an optimization run with default settings using the minimax regret method. The top graph shows actual performance and the bottom graph shows robustness (maximum performance regret).

### 5.3 Computational cost savings using the ULH scenario sampling strategy and optimal settings of GA parameters for multi-objective optimization

The reduction in computational costs is calculated by comparing the computational costs arising from using the optimal settings of GA and the ULH sampling strategy with the computational costs of using the full factorial approach. It can be observed from Table 6 that for both methods on average when using optimal settings, a matching index of about 90% is achieved at less than 0.1

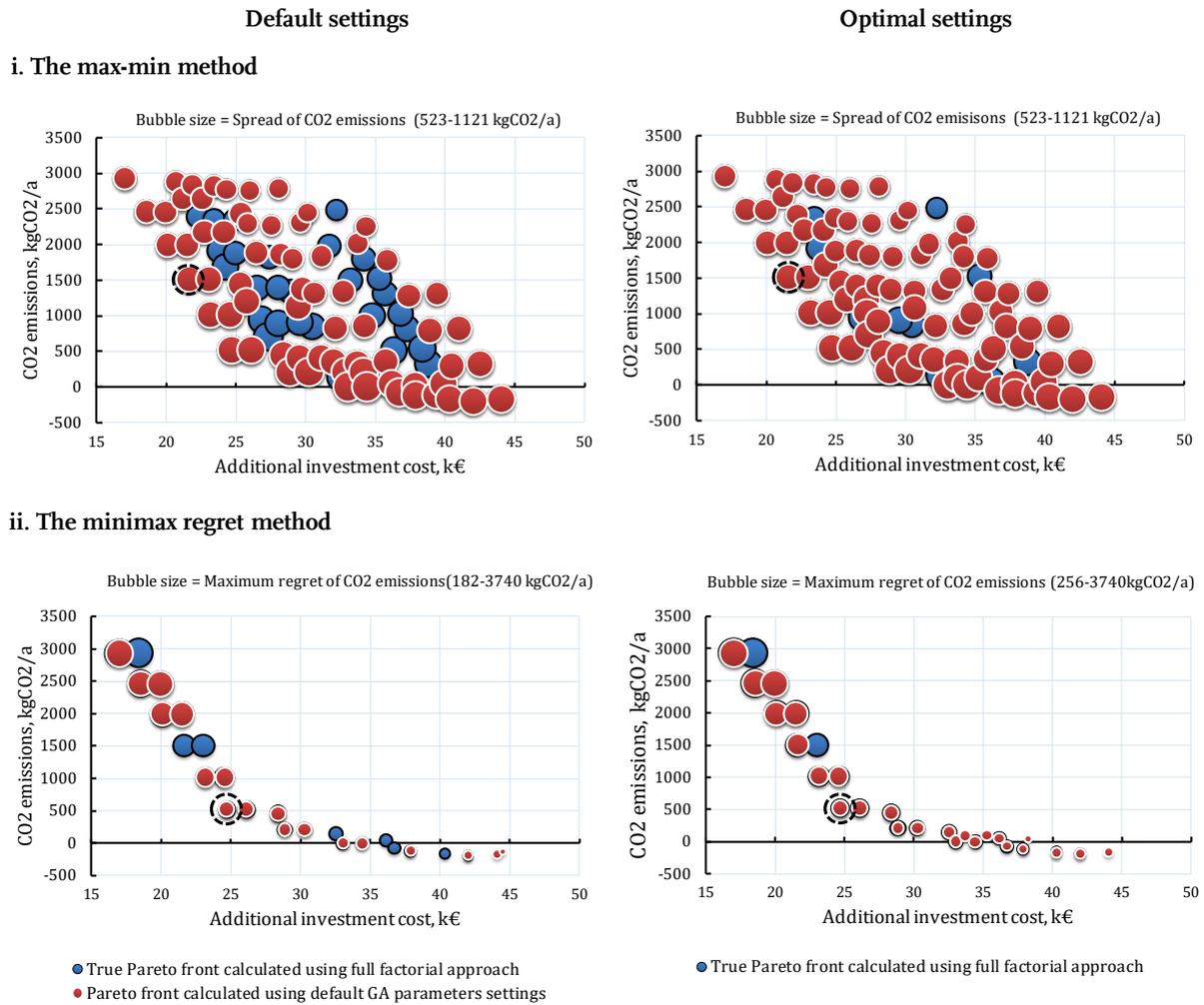
% of the computational cost of a full factorial assessment. Similarly, the matching index is improved by up to 22-28% at more or less the same number of iterations by using optimal settings compared to using default MATLAB values.

**Table 6** *Optimal settings of GA parameter values selected based on the matching index and required number of iterations for the case study with policymaker as decision maker.*

Parameter	Default settings	Optimal settings	
		Max-min method	Minimax regret method
Crossover fraction (CF)	0.8	0.7	0.5
Pareto fraction (PF)	0.35	0.5	0.5
Selection function (TS)	4	4	4
Population size	30	40	30
ULH scenario sample	100	100	100
Matching index (%)	<b>66 – 71.5</b>	<b>88.8</b>	<b>91.3</b>
Iterations required to meet optimization criterion (averaged over multiple runs)	744 – 876	945	798
Computational costs reduced compared to FF with low-high scenario combinations (%)	<b>94.72-95.52</b>	<b>94.3</b>	<b>95.19</b>
Computational costs reduced compared to FF with all scenario combinations (%)	<b>99.91</b>	<b>99.9</b>	<b>99.2</b>

## 6. Validation

The Pareto fronts with the optimal settings are compared in Figure 13 with those of the default values and the true Pareto fronts (from the full factorial approach). The blue bubbles represent true Pareto fronts and red bubbles represent Pareto fronts for the corresponding optimal and default settings. It is noteworthy that the range of maximum performance regret of CO<sub>2</sub> emissions is slightly different for Pareto fronts optimized with optimal settings of GA parameters, as seen in Figure 13. This difference is attributed to the optimal settings yielding more Pareto solutions compared to the Pareto solutions calculated using the full factorial approach. Accordingly, the different values of maximum performance regret for the same design are due to the inter-comparison of designs in the corresponding Pareto front for the maximum performance regret calculations (see Table 7).



**Figure 13** Comparison of the Pareto fronts using default MATLAB values and corresponding optimal values with the true Pareto front obtained using the full factorial approach for two robustness assessment methods. Blue bubbles represent the true Pareto front and red bubbles represent Pareto front with corresponding settings. The most robust design selected using the Hurwicz criterion is indicated in dotted lines.

It can be seen from Figure 13 that when using default values, there is a risk of losing some robust designs. Conversely, with the optimal settings this risk is reduced to a significant extent, especially with the minimax regret method. However, the selection of optimal GA parameter settings depends on whether these parameters lead to the same robust design as the full factorial approach. For instance, this can be validated by comparing the most robust design obtained using the optimal settings with the equivalent robust design obtained using the full factorial approach. The most robust design is identified based on the highest design score calculated using the Hurwicz criterion (Hurwicz 1952; Rysanek and Choudhary 2013). The design score of a design for a robustness assessment method is calculated considering additional investment cost, CO<sub>2</sub> emissions and

corresponding robustness. The most robust designs are indicated by dotted line in Figure 13 and the design details are tabulated in Table 7. It can be observed that both optimization methods (FF and GA based optimization) result in the same most robust design for the corresponding robustness assessment method, indicating that GA based optimization with optimal settings is valid and can be used to reduce computational time without compromising the outcome.

**Table 7** The most robust designs selected based on the highest design score calculated using the Hurwicz criterion for both robustness assessment methods using the full factorial approach (FF) and GA based optimization with corresponding optimal settings.

Design variants	Max-min method		Minimax regret method	
	FF	Optimal GA	FF	Optimal GA
Rc-Wall, m <sup>2</sup> K/W	4.5	4.5	4.5	4.5
Rc-Roof, m <sup>2</sup> K/W	6	6	6	6
Rc-floor, m <sup>2</sup> K/w	3.5	3.5	3.5	3.5
Windows U value, W/ m <sup>2</sup> k	1.43	1.43	1.43	1.43
WWR (%)	20	20	20	20
Thermal mass	Light-weight	Light-weight	Light-weight	Light-weight
Infiltration, ach	0.12	0.12	0.12	0.12
PV size, m <sup>2</sup>	20	20	30	30
SDHW, m <sup>2</sup>	0	0	0	0
Additional investment cost, k€	21.6	21.6	24.7	24.7
CO <sub>2</sub> emissions, kgCO <sub>2</sub> /a (median value across considered scenarios)	1506	1506	522	522
Performance robustness of CO <sub>2</sub> emissions, kgCO <sub>2</sub> /a	843	843	1274	976

## 7. Summary and conclusion

A simulation framework was developed to integrate robustness indicators into the optimization process. GA multi-objective optimization in combination with the ULH scenario sampling method were implemented in this framework to enhance its usability in practice. The GA parameters are fine-tuned to further improve computational efficiency. The fitness function of the GA was modified to implement robustness indicators as objectives of the optimization problem.

The following conclusions can be drawn from this study:

- In the max-min method, robustness (performance spread) is optimized with respect to the best performing scenario of a design. In the minimax regret method, robustness (maximum performance regret) is optimized by minimizing the maximum performance difference across all scenarios between the performance of a design and the optimal design of the corresponding scenario, similar to study by (Aissi, Bazgan, and Vanderpooten 2009). Therefore, the max-min method can be used when a design has to deliver the desired performance for all scenarios including extreme scenarios, whereas the minimax regret method can be used when a design should yield optimal or close to optimal performance for each scenario. In other words, the max-min method can be used when the cost/risk associated with the failure of design is very high. The minimax regret method can be used when a decision maker can accept a certain range of performance variation/risk as a trade-off. If computational costs are the main criterion, then minimax regret method is more preferred based on this case study results.
- The method of calculation of fitness function differs for both robustness indicators. Performance spread is calculated for each design of the population without any inter-comparison of the performance of other designs of the population, and thus one design at a time is considered when calculating the robustness. Contrariwise, maximum performance regret is calculated with inter-comparison of the performance of other designs, and thus robustness is calculated after the calculation of the performance of the entire design population. Furthermore, for each generation, maximum regret is calculated considering the current design population and design archive of previous generations because of the need for inter-comparison of the performance of all designs.
- Integration of maximum performance regret as an objective in multi-objective optimization cannot be done in a straightforward way due to inter-comparison of performance of designs in the calculation of maximum performance regret. In such cases, the typical GA fitness function is not recommended; instead, a modified fitness function as presented in this study may be used.

- Using the ULH sampling strategy in the current case study, a sample of 100 scenario combinations was the smallest sample size that yielded similar performance robustness as that of low-high scenario combinations for the considered performance indicators.
- The matching index of a Pareto front can be improved by up to 90% on average with optimal settings compared to default values (average matching index of 68%) for both robustness assessment methods.
- The simulation framework implemented in this study using scenario sampling and multi-objective optimization methods could save up to 94-99% of computational costs compared to full factorial approach with low-high and all scenario combinations, which require millions of simulations.

The developed simulation framework could be useful for designers and consultants to identify robust designs at low computational costs. However, the implemented GA based optimization can only provide an approximation of the true Pareto front and therefore there is still a risk of missing some robust designs. This risk might be reduced further by implementing different optimization techniques (Nguyen, Reiter, and Rigo 2014). Determining the optimal settings of GA parameters largely depends on the considered design space and objective functions, consequently making them case study dependent.

## References

- Agentschap NL. 2013. "Ministerie van Economische Zaken: Referentiewoningen Nieuwbouw," 37.
- Aissi, Hassene, Cristina Bazgan, and Daniel Vanderpooten. 2009. "Min-Max and Min-Max Regret Versions of Combinatorial Optimization Problems: A Survey." *European Journal of Operational Research* 197 (2): 427–438. doi:10.1016/j.ejor.2008.09.012.
- Averbakh, Igor. 2000. "Minmax Regret Solutions for Minimax Optimization Problems with Uncertainty." *Operations Research Letters* 27 (2): 57–65. doi:10.1016/S0167-6377(00)00025-0.
- Burhenne, Sebastian, Dirk Jacob, and Gregor P Henze. 2011. "Sampling Based on SOBOL Sequences for Monte Carlo Techniques Applied to Building Simulations." In *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November.*, 14–16.
- CBS. 2016a. "Central Bureau of Statistics Netherlands - Households." <http://statline.cbs.nl/StatWeb/publication/?VW=T&DM=SLen&PA=82905ENG&LA=EN>.
- CBS. 2016b. "CBS StatLine - Aardgas En Elektriciteit, Gemiddelde Prijzen van Eindverbruikers."

- de Wilde, P. 2014. "The Gap between Predicted and Measured Energy Performance of Buildings: A Framework for Investigation." *Automation in Construction* 41: 40–49. doi:10.1016/j.autcon.2014.02.009.
- de Wilde, P, and David Coley. 2012. "The Implications of a Changing Climate for Buildings." *Building and Environment* 55: 1–7. doi:10.1016/j.buildenv.2012.03.014.
- de Wilde, P, and Wei Tian. 2009. "Identification of Key Factors for Uncertainty in the Prediction of the Thermal Performance of an Office Building under Climate Change." *Building Simulation* 2 (3): 157–174. doi:10.1007/s12273-009-9116-1.
- Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197. doi:10.1109/4235.996017.
- Evins, Ralph. 2013. "A Review of Computational Optimisation Methods Applied to Sustainable Building Design." *Renewable and Sustainable Energy Reviews* 22: 230–245. doi:10.1016/j.rser.2013.02.004.
- Evins, Ralph. 2016. "A Bi-Level Design and Operation Optimization Process Applied to an Energy Centre." *Journal of Building Performance Simulation* 9 (3): 255–271. doi:10.1080/19401493.2015.1045034.
- Fawcett, William, Martin Hughes, Hannes Krieg, Stefan Albrecht, and Anders Vennström. 2012. "Flexible Strategies for Long-Term Sustainability under Uncertainty." *Building Research & Information* 40 (5): 545–557. doi:10.1080/09613218.2012.702565.
- Gram-hanssen, Kirsten, and Susse Georg. 2017. "Energy Performance Gaps: Promises , People , Practices." *Building Research & Information*, 1–9. doi:10.1080/09613218.2017.1356127.
- Guerra-Santin, O, and Laure Itard. 2010. "Occupants' Behaviour: Determinants and Effects on Residential Heating Consumption." *Building Research and Information* 38 (3): 318–338. doi:10.1080/09613211003661074.
- Hamdy, M, and Kai Sirén. 2016. "A Multi-Aid Optimization Scheme for Large-Scale Investigation of Cost-Optimality and Energy Performance of Buildings." *Journal of Building Performance Simulation* 9 (4): 411–430. doi:10.1080/19401493.2015.1069398.
- Helton, J. C., J. D. Johnson, C. J. Sallaberry, and C. B. Storlie. 2006. "Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis." *Reliability Engineering and System Safety* 91 (10–11): 1175–1209. doi:10.1016/j.ress.2005.11.017.
- Hoes, P. 2014. "Computational Performance Prediction of the Potential of Hybrid Adaptable Thermal Storage Concepts for Lightweight Low-Energy Houses." Eindhoven University of Technology. doi:10.13140/2.1.2329.8562.
- Hoes, P., J. L M Hensen, M. G L C Loomans, B. de Vries, and D. Bourgeois. 2009. "User Behavior in Whole Building Simulation." *Energy and Buildings* 41 (3): 295–302. doi:10.1016/j.enbuild.2008.09.008.
- Hoes, P, M Trcka, J L M Hensen, and B Hoekstra Bonnema. 2011. "Optimizing Building Designs Using a Robustness Indicator with Respect to User Behavior." *Proceedings of the 12th Conference of the International Building Performance Simulation Association*, 14–16.
- Hu, Huaifen, and Godfried Augenbroe. 2012. "A Stochastic Model Based Energy Management System for off-Grid Solar Houses." *Building and Environment* 50. Elsevier Ltd: 90–103. doi:10.1016/j.buildenv.2011.10.011.
- Hurwicz, L. 1952. "A Criterion for Decision Making under Uncertainty." *Technical Report 355, Cowles Commission*.
- Janssen, Hans. 2013. "Monte-Carlo Based Uncertainty Analysis: Sampling Efficiency and Sampling Convergence." *Reliability Engineering and System Safety* 109: 123–132.

doi:10.1016/j.res.2012.08.003.

- Kim, Sean Hay. 2013. "An Evaluation of Robust Controls for Passive Building Thermal Mass and Mechanical Thermal Energy Storage under Uncertainty." *Applied Energy* 111: 602–623. doi:10.1016/j.apenergy.2013.05.030.
- Kotireddy, Rajesh, P Hoes, and Jan L. M. Hensen. 2018. "A Methodology for Performance Robustness Assessment of Low-Energy Buildings Using Scenario Analysis." *Applied Energy* 212: 428–442. doi:https://doi.org/10.1016/j.apenergy.2017.12.066.
- Kotireddy, Rajesh, P Hoes, and Jan L.M. Hensen. 2017. "Simulation-Based Comparison of Robustness Assessment Methods to Identify Robust Low Energy Building Designs." In *Proceedings of 15th IBPSA Conference, San Francisco, CA, USA*, 892–901. doi:10.26868/25222708.2017.240.
- Macdonald, Iain A. 2009. "Comparison of Sampling Techniques on the Performance of Monte-Carlo Based Sensitivity Analysis." *BS2009: 11th Conference of International Building Performance Simulation Association, Glasgow, Scotland, July 27-30*, no. 3: 992–999.
- Machairas, Vasileios, Aris Tsangrassoulis, and Kleo Axarli. 2014. "Algorithms for Optimization of Building Design: A Review." *Renewable and Sustainable Energy Reviews* 31 (1364): 101–112. doi:10.1016/j.rser.2013.11.036.
- MathWorks. 2016. "MATLAB (Matrix Laboratory)." www.mathworks.nl.
- McLeod, Robert S., C.J. Hopfe, and Alan Kwan. 2013. "An Investigation into Future Performance and Overheating Risks in Passivhaus Dwellings." *Building and Environment* 70: 189–209. doi:10.1016/j.buildenv.2013.08.024.
- Ministerie van VROM. 2009. *Energiegedrag in De Woning*.
- Moss, Richard H, Jae a Edmonds, Kathy a Hibbard, Martin R Manning, Steven K Rose, Detlef P van Vuuren, Timothy R Carter, et al. 2010. "The next Generation of Scenarios for Climate Change Research and Assessment." *Nature* 463 (7282). Nature Publishing Group: 747–756. doi:10.1038/nature08823.
- NEN. 2008. *Netherlands Norm, NEN 5060- Hygrothermal Performance of Buildings - Climatic Reference Data*.
- NEN7120+C2. 2012. *Energy Performance of Buildings-Determination Method*. Vol. 6.
- Nguyen, Anh-Tuan, Sigrid Reiter, and Philippe Rigo. 2014. "A Review on Simulation-Based Optimization Methods Applied to Building Performance Analysis." *Applied Energy* 113: 1043–1058. doi:10.1016/j.apenergy.2013.08.061.
- O'Neill, Zheng, and B Eisenhower. 2013. "Leveraging the Analysis of Parametric Uncertainty for Building Energy Model Calibration." *Building Simulation*, 1–13. doi:10.1007/s12273-013-0125-8.
- Østergård, Torben, Rasmus L. Jensen, and Steffen E. Maagaard. 2017. "Early Building Design: Informed Decision-Making by Exploring Multidimensional Design Space Using Sensitivity Analysis." *Energy and Buildings* 142. Elsevier B.V.: 8–22. doi:10.1016/j.enbuild.2017.02.059.
- Polasky, Stephen, Stephen R. Carpenter, Carl Folke, and Bonnie Keeler. 2011. "Decision-Making under Great Uncertainty: Environmental Management in an Era of Global Change." *Trends in Ecology and Evolution* 26 (8): 398–404. doi:10.1016/j.tree.2011.04.007.
- RVO. 2015. "Hernieuwbare Energie in Bijna Energieneutrale Gebouwen (BENG)."
- RVO. 2016. "Current Dutch Building Standards." https://www.rvo.nl/onderwerpen/duurzaam-ondernemen/gebouwen/wetten-en-regels-gebouwen/nieuwbouw/energieprestatie-epc/referentiewoningen-epc/tussenwoning.
- Rysanek, A. M., and R. Choudhary. 2013. "Optimum Building Energy Retrofits under Technical

- and Economic Uncertainty.” *Energy and Buildings* 57: 324–337. doi:10.1016/j.enbuild.2012.10.027.
- Savage, L. 1951. “The Theory of Statistical Decision.” *Journal of the American Statistical Association* 46: 55–67.
- Struck, C, and J L M Hensen. 2013. “Scenario Analysis for the Robustness Assessment of Building Design Alternatives – a Dutch Case Study.” In *Proceedings of CISBAT 2013*, 939–944.
- Tian, Wei, Yeonsook Heo, Pieter de Wilde, Zhanyong Li, Da Yan, Cheol Soo Park, Xiaohang Feng, and Godfried Augenbroe. 2018. “A Review of Uncertainty Analysis in Building Energy Assessment.” *Renewable and Sustainable Energy Reviews* 93 (January 2017). Elsevier Ltd: 285–301. doi:10.1016/j.rser.2018.05.029.
- van den Hurk, Bart, Albert Klein Tank, Geert Lenderink, Aad van Ulden, Geert Jan van Oldenborgh, Caroline Katsman, Henk van den Brink, et al. 2006. *KNMI Climate Change Scenarios 2006 for the Netherlands*.
- Van Gelder, L, Hans Janssen, and Staf Roels. 2014. “Probabilistic Design and Analysis of Building Performances: Methodology and Application Example.” *Energy and Buildings* 79: 202–211. doi:10.1016/j.enbuild.2014.04.042.
- Wald, A. 1945. “Statistical Decision Functions Which Minimize the Maximum Risk.” *The Annals of Mathematics* 46 (2): 265–280.
- Woloszyn, Monika, and Ian Beausoleil-Morrison. 2017. “Treating Uncertainty in Building Performance Simulation.” *Journal of Building Performance Simulation* 10 (1): 1–2. doi:10.1080/19401493.2017.1261641.
- Xidonas, Panos, George Mavrotas, Christis Hassapis, and Constantin Zopounidis. 2017. “Robust Multiobjective Portfolio Optimization: A Minimax Regret Approach.” *European Journal of Operational Research* 262 (1): 299–305. doi:10.1016/j.ejor.2017.03.041.
- Yan, Da, William O’Brien, Tianzhen Hong, Xiaohang Feng, H. Burak Gunay, Farhang Tahmasebi, and Ardeshir Mahdavi. 2015. “Occupant Behavior Modeling for Building Performance Simulation: Current State and Future Challenges.” *Energy and Buildings* 107: 264–278. doi:10.1016/j.enbuild.2015.08.032.