# Multi-objective mixed-integer evolutionary algorithms for building spatial design

Blom, K. van der

**Author**: Blom, K. van der
**Title**: Multi-objective mixed-integer evolutionary algorithms for building spatial design
**Issue Date**: 2019-12-11

# Chapter 5

# Problem Specific Constraint Handling and Multi-Objective Optimisation

In the previous chapter first steps have been made towards constraint handling techniques for building spatial design, as is required to answer RQ2. The main conclusion was that although the proposed penalty based methods were reasonably effective for the small scale designs considered in the experiments there, they are unlikely to be sufficient for the larger scale designs that are frequently needed in the real world. To resolve this issue, in this chapter problem specific operators are developed which ensure that only feasible designs are found. Furthermore, the two considered objectives (structural, and energy performance) have so far only been considered separately. Since there is an interaction between these disciplines, and they both affect the spatial design, in this chapter they will be considered in a multi-objective context.

The use of multi-objective optimisation in building design is not new, and has been shown to be effective in [54] when two objectives from the same discipline are considered. However, traditionally, energy efficiency and structural design objectives are dealt with in different engineering disciplines. Multidisciplinary optimisation aims to combine different disciplines in order to find building designs that perform well with respect to criteria from each of them. While in the building design domain the use of multidisciplinary optimisation is still limited, it has already been used with great success in areas such as automotive and aerospace engineering [70].

This chapter contributes towards the use of multidisciplinary optimisation in building spatial design. Here the previously introduced (Section 4.3) objectives from structural design (compliance) and energy efficiency (total surface area) are considered again. By proposing a multi-objective optimisation approach, the problem of conflicting objectives is also taken into account. In this case a Pareto front of building spatial designs is computed that can be used in preparation of decision making, to understand design principles that lead to high performance in one discipline or the other discipline, and to find valid compromise solutions.

To achieve these goals (evolutionary) multi-objective optimisation algorithms, such as SMS-EMOA [39] and NSGA-II [32], are employed. Within the algorithm structure, the problem specific constraint handling operators are developed. In this manner these traditional algorithms are adapted to handle the heavily constrained, and mixed-integer search space of the supercube representation. First it is shown that the use of problem specific operators (here initialisation and mutation are considered) is indeed a promising direction by developing initial versions of them. Next, based on the positive results, these methods are developed further to navigate the search space without bias.

The unbiased initialisation operator that will be introduced should be well suited for use in landscape analysis. By using landscape analysis based on the initialisation operator, insight is gained into the objective landscape corresponding to the objective functions. Various methods for landscape analysis have been proposed in the literature, see e.g. [80, 76]. A basic approach for landscape analysis is to look at the distribution of function values over random samples, for example by density of states analysis [83]. Additionally, landscape analysis can be used to identify how variation operators behave in the objective landscape [57]. In this chapter it is applied to evaluate the improved mutation operator, and to investigate its behaviour for different step sizes on multiple problem instances.

Naturally, obtaining the optimal performance from the developed algorithms is desirable. To achieve this, parameter tuning can be used, which aims at finding the optimal settings for an algorithm to solve a problem. For instance, in [68] the tuning and configuration of an image processing pipeline in coronary vessel image analysis is considered as a mixed-integer optimisation problem. Specific algorithmic and statistical solutions for tuning also exist, e.g. SMAC [55], irace [71], and SPOT [9]. Another way to improve algorithm performance is by integrating problem knowledge into operators, which can significantly improve performance as shown for chemical process design in [84]. This is also done in this chapter, and the settings of these operators are tuned to further improve performance.

Many real world optimisation problems, such as building spatial design, are characterised by expensive evaluation functions. As a result, tuning optimisation algorithms for this type of problem involves an additional challenge. The irace package [71], based on statistical significance of configurations, and the Mixed-Integer Evolution Strategy (MIES) [69], a self-adaptive evolutionary algorithm with specialised operators for mixed-integer problems, are compared here for the tuning of the parameters of the standard SMS-EMOA [39] and a tailored version of SMS-EMOA that uses the newly introduced problem specific operators. This comparison serves three purposes. Firstly, it allows fair insight into how the tailored and standard versions of SMS-EMOA differ. Secondly, the qualities of two tuning methods, with different philosophies behind them, can be compared. Finally, while state-of-the-art tuning methods have been extensively evaluated on academic case studies, the comparison here considers a real world problem.

Given these preliminaries, this chapter contributes as follows. Initialisation and mutation operators suitable to the problem specific constraints, and the mixed-integer nature of the problem are developed. Next, first results on multi-objective optimisation of building spatial designs are provided and discussed. Following that, the problem specific mutation and initialisation operators are improved to avoid biases. Using the unbiased operators, landscape analysis of the building spatial design problem is performed to identify problem features and to investigate the behaviour of the mutation operator. Finally, parameter tuning is conducted for the considered algorithms and discussed for optimisation problems with time consuming evaluation functions. Moreover, two parameter tuning algorithms are compared and their different merits are discussed.

This chapter continues with Section 5.1, where algorithm details and the problem specific operators are introduced. Thereafter, in Section 5.2 experiments are described, followed by a discussion of the numerical results. In Section 5.3 the improved unbiased initialisation and mutation operators are described. Section 5.4 then investigates the objective landscapes for structural design and energy efficiency, as well as how the mutation operator behaves in this landscape. To compare the standard SMS-EMOA algorithm and a tailored version with the unbiased operators described in this chapter, Section 5.5 discusses parameter tuning and algorithm performance for both of these approaches. Finally, Section 5.6 summarises the chapter and discusses directions for future work.

# 5.1 Algorithms

This section first describes how standard multi-objective evolutionary algorithms are adapted for building spatial design by using the same techniques as applied for single objective optimisation in Chapter 4. After that, problem specific initialisation and mutation operators are introduced to replace the standard versions in an algorithm tailored to the building spatial design problem.

## 5.1.1 Standard Algorithms Applied to Building Spatial Design

NSGA-II [32] and SMS-EMOA [39] are both standard algorithms in evolutionary multi-objective optimisation (EMO), and will be used as a baseline in the experiments here. Both algorithms largely work according to the same principles and are described together in the following, and their differences are indicated as needed.

Initialisation works by setting binary variables to one with probability $1/N_{cells}$, or to zero otherwise. Here $N_{cells}$ is again defined as $N_w \times N_d \times N_h$. The continuous variables are initialised to a uniformly random value in $[lb, ub]$, where $lb = 3$ and $ub = 19.8$ (both in metres). As a final step of the initialisation procedure, the volume is repaired to be within 1 % of the desired volume $V_0 = 4^3 \times N_{cells}$ (in cubic metres). Here volume repair works as previously described in Section 4.4.3.

To generate offspring an individual is selected uniformly at random from the parent population. Then recombination is applied with probability $RP = 0.5$, in which case a second parent is selected uniformly at random. Otherwise the first parent is copied to the offspring individual. Binary variables are swapped between the parents with a probability of 0.25, while simulated binary crossover [30] is applied to the continuous variables with probability 0.5. Variables that exceed their bound are set to $lb$ or $ub$, depending on which one is exceeded. Finally, either of the children is selected with probability 0.5.

Mutation is applied with a probability of $MT = 1/N_{dims}$, where $N_{dims} = N_{cells} \times N_{spaces} + N_w + N_d + N_h$ is once more the total number of variables. Binary variables are mutated by bit flips, while polynomial mutation [32] is applied to continuous variables above the lower bound. Continuous variables that are exactly at the lower boundary are reinitialised (according to the previously described initialisation procedure). Following mutation, variables exceeding their bounds are set to their appropriate boundary values again, and the volume of the produced offspring is repaired as described in Section 4.4.3.

Both algorithms use a $(\mu + \lambda)$ selection strategy, where for NSGA-II $(20 + 20)$

is considered, and for SMS-EMOA (50 + 1). NSGA-II does so by applying non-dominated/crowding distance sortingto the population, and then selecting the first $\mu$ individuals for the next parent population. On the other hand, SMS-EMOA selects the population of size $\mu$ that retains the largest hypervolume, using $(1.1e9, 1.1e9)$ as reference point during optimisation.

Equivalent to the process in the previous chapter, in case of constraint violations a penalty value $pen = 999\,999\,998 + CV$ is returned according to the graded penalty method (Section 4.4.2). The objective functions are only evaluated when no constraints are violated. As such, infeasible solutions do not influence the remaining evaluation budget.

### 5.1.2 Tailored Algorithm for Building Spatial Design

The general mutation and recombination operators used in NSGA-II and SMS-EMOA have difficulties navigating heavily constrained objective landscapes, such as considered for the building spatial design problem. To resolve this, a problem specific mutation operator is proposed which only produces mutants that do not violate the constraints. Since in preliminary experiments the algorithms have fairly similar performance, only SMS-EMOA is considered with the problem specific mutation operator.


**(a)** Initialisation. **(b)** Mutation.

**Figure 5.1:** Problem specific operators that prevent constraint violations.

Additionally, initialisation is adapted to ensure all initial solutions are feasible as well. Initialisation of the continuous variables causes no problems and is kept as described in Section 5.1.1. Binary variables are initialised by selecting a non-fully occupied pillar from the supercube for every space uniformly at random. The first cell from the bottom that does not belong to any previously initialised space is then set to active ($b^{\ell}_{i,j,k} = 1$) for this space. This process is visualised in Figure 5.1a, where spaces $A, B$, and $C$ can be imagined as having been dropped – one after the other – into the supercube from above. Since the initial population resulting from this process consists solely of single-cell spaces, it is not very diverse. To resolve this, twenty mutations (as described next) are applied to each individual in the initial population.

Having defined the initialisation procedure, a variation operator is needed to guide the search. To this end, a problem specific mutation operator is defined next. With a probability of 0.25 a binary mutation is applied, and otherwise a continuous mutation is applied. In case of a continuous mutation, polynomial mutation [32] is always applied to a single continuous variable, selected uniformly at random from those variables that are relevant to at least one active cell. In case of a binary mutation, a random space is selected and contracted or expanded by a surface of cells in a random direction as shown in Figure 5.1b (where the arrows indicate possible moves for space $B$). This is achieved by selecting one of the following faces of the space to make either an outward or an inward move: left, right, top, bottom, front, or back. All moves are applied to all cells along the selected face of a space, such that the space remains cuboid when adding and removing cells. These moves are of size one, meaning that the width, depth or height (depending on the selected face) of a space grows or shrinks by a single cell. For any move Constraint C3 (all spaces must have at least one cell assigned to them) and the supercube borders must be respected. In other words, any move violating these constraints cannot be chosen. From all possible mutation moves that do not result in a constraint violation one is chosen uniformly at random. To avoid issues with the no overlap constraint (C1), whenever an outward move adds a cell to a space $B$ that is already part of another space $A$ the cell is set to inactive ($b^{\ell}_{i,j,k} = 0$) for space $A$.

A new offspring individual is then created as follows. A parent is selected uniformly at random, and mutation is applied. No recombination is used, but this may be considered in future work. Although recombination may be able to aid the search, designing a problem specific recombination operator that does not violate any constraint is complicated. Moreover, at the same time as not violating constraints, it should also result in sensible new designs. Penalty values are also no longer used since all offspring is now guaranteed to be feasible. The remaining procedures (such as selection based on hypervolume contribution) are the same as before in Section 5.1.1.

## 5.2 Tailored Algorithm Evaluation

### 5.2.1 Experiments

Having adapted NSGA-II and SMS-EMOA to operate on the building spatial design problem, it is now possible to compare them empirically. They will also form a baseline to compare with the tailored SMS-EMOA variant. For these experiments a similar setup is used as in the previous chapter (Section 4.5). The 2221, 2223, 2225, 3331, 3333, and 3335 configurations are considered again. Here the first three digits again indicate the supercube size in width, depth, and height, while the last digit indicates the number of spaces. The most important change is that now the two objectives (minimal compliance, and minimal surface area) are optimised at the same time in a multi-objective setting. Besides this, here an evaluation budget of 2500 is used for five runs of each configuration and algorithm pair. The resulting Pareto front approximations of the five runs are then averaged and analysed visually with median attainment curves [49].

Further settings, as previously mentioned in the algorithm descriptions, are summarised in Table 5.1.

| $(\mu+\lambda)$ NSGA-II | $(\mu+\lambda)$ SMS-EMOA | $V_0$ | $lb$ | $ub$ | $RP$ | $MT$ |
|---|---|---|---|---|---|---|
| (20+20) | (50+1) | $4^3 \times N_{cells}$ | 3.0 | 19.8 | 0.5 | $1/N_{dims}$ |

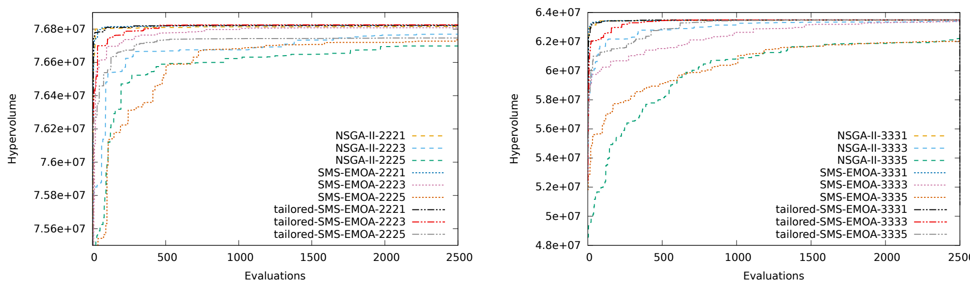
**Table 5.1:** Parameter settings for experiments with standard and tailored algorithms.

### 5.2.2 Results

Figure 5.2 shows the average convergence rate of the various problem configurations over five runs. Here, the hypervolume is computed over $\log(1 + \text{compliance})$, with a reference point $(35000, 2500)$ (compliance, surface area). Moreover, extreme outliers in either objective (points beyond the reference point), are left out of the analysis.

Most configurations and algorithms show a quick convergence to a relatively stable hypervolume value. In Figure 5.2 it can be observed that the tailored SMS-EMOA variant produces similar results to the other two approaches for problems with a single space. However, when three spaces are considered the tailored method improves over the other two by a decent margin, and for five spaces it is clearly better, both in terms of convergence speed and with regard to the final solution. In addition, for the most complex configuration (3335) in particular it is not clear whether NSGA-II and SMS-EMOA have converged,whereas the tailored SMS-EMOA does converge.

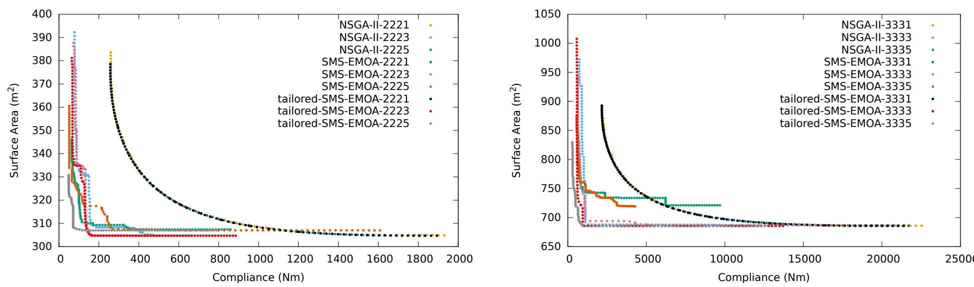
Next, the median attainment curves [49] produced by the algorithms will be analysed, and are shown in Figure 5.3. The attainment curve serves as a measure of the probability to attain (dominate) certain nondominated points by the approximation set. Both NSGA-II and SMS-EMOA produce similar curves as depicted in Figure 5.3.




**(a)** 222x configurations. **(b)** 333x configurations.

**Figure 5.2:** Average hypervolume growth over five runs, for one, three, and five spaces.

This indicates the considered optimisation process works, and is able to discover a Pareto front approximation. The same behaviour as seen for the hypervolume convergence can be observed from the attainment curves. Namely, the differences in performance become more pronounced with larger problem sizes. Clearly, problem specific operators are able to produce a better Pareto front approximation.




**(a)** 222x configurations. **(b)** 333x configurations.

**Figure 5.3:** Median attainment curves from five runs for one, three, and five spaces.

The standard deviations of the hypervolume at the final generation are relatively small for most problem configurations and do not change the numerical result. Only for the 3335 configuration (Table 5.2) large deviations occur for the generic methods, but even their highest hypervolume solutions do not outperform the smallest hypervolume found by the tailored method.

Since the 3335 configuration is the largest, it is also the best indicator for performance on even larger building spatial designs that may be considered in real world applications. As such, a Wilcoxon signed-rank test[1] [104] is conducted for this configu-

[1] A non-parametric test to compare the distributions of different samples

| algorithm | min | max | mean | median | std dev |
|---|---|---|---|---|---|
| NSGA-II | 60665475 | 63379795 | 6.22089e+07 | 62247270 | 1.05263e+06 |
| SMS-EMOA | 59642594 | 63387974 | 6.20227e+07 | 62313768 | 1.25616e+06 |
| tailored SMS-EMOA | 63492022 | 63495486 | 6.34941e+07 | 63494483 | 1145.48 |

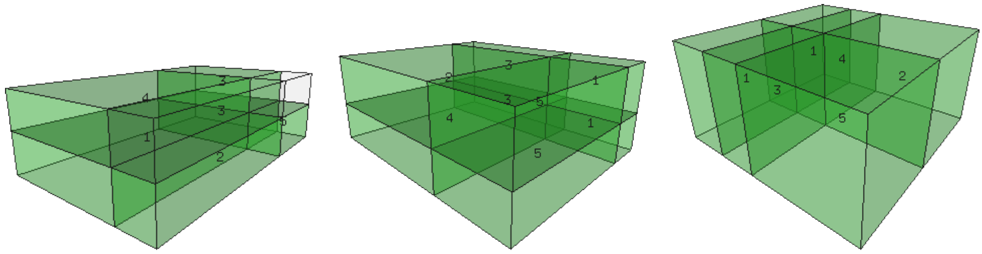
**Table 5.2:** Hypervolume statistics per algorithm for the 3335 problem instance.

ration to identify whether performance differences are statistically significant. Results of this test are summarised in Table 5.3. First NSGA-II and SMS-EMOA are compared to see whether one is preferable over the other for this application. This comparison results in $W = -1$, indicating there is no significant difference. Next, the same test is applied between NSGA-II, and the tailored SMS-EMOA. This results in $W = 15$, indicating the tailored method is better than NSGA-II with a statistical significance of 0.05. The exact same outcome is found when the same comparison is made between the standard SMS-EMOA, and the tailored SMS-EMOA. Evidently, the problem specific constraint handling operators make a valuable contribution to the performance, and are worth developing further.

| run | NSGA-II | SMS-EMOA | tailored | sgn × rank NS | sgn × rank NT | sgn × rank ST |
|---|---|---|---|---|---|---|
| 1 | 60665475 | 59642594 | 63492022 | -5 | 5 | 5 |
| 2 | 61445894 | 62300678 | 63493947 | 4 | 4 | 4 |
| 3 | 62247270 | 62313768 | 63494483 | 2 | 3 | 3 |
| 4 | 63305880 | 62468595 | 63494488 | -3 | 2 | 2 |
| 5 | 63379795 | 63387974 | 63495486 | 1 | 1 | 1 |
| W | | | | -1 | 15 | 15 |

**Table 5.3:** Wilcoxon signed-rank test results for the hypervolume attained for the 3335 problem instance.

Figure 5.4 shows the best found spatial designs in terms of each objective, as well as a compromise solution at the knee point of the median attainment curve. This is accompanied by some additional information on these spatial designs in Table 5.4. As can be expected, the optimal spatial design in terms of minimal surface area has a cuboid shape. The knee point solution is largely similar, but has a slightly lower structure, and as a result it is stretched in both width and depth to maintain the volume. Finally, the minimal compliance solution has an elongated structure. Moreover, it has a lower structure, which can be explained since it results in less strain on the structural elements, which in turn reduces the compliance.



**(a)** Minimal compliance. **(b)** Knee point. **(c)** Minimal surface area.

**Figure 5.4:** Best spatial designs found with the tailored SMS-EMOA for the 3335 configuration.

| | Compliance (N m) | Surface area (m) | Soil surface (m) | Height (m) | Longest edge (m) | Shortest edge (m) |
|---|---|---|---|---|---|---|
| Minimal compliance | 214.658 | 741.181 | 303.228 | 6.361 | 22.182 | 13.670 |
| Knee point | 451.351 | 690.077 | 252.545 | 6.845 | 17.658 | 14.302 |
| Minimal surface area | 9483.320 | 685.562 | 227.177 | 7.603 | 15.128 | 15.017 |

**Table 5.4:** Details about the best spatial designs found with the tailored SMS-EMOA for the 3335 configuration.

Table 5.5 shows the CPU time used by SMS-EMOA with problem specific operators. The other methods performed similarly because the compliance computations used by far the most CPU time. Each experiment used a single core of an i7-3770 CPU @ 3.40GHz processor, and had 16GiB DIMM DDR3 Synchronous 1600 MHz memory available.

| problem configuration | 2221 | 2223 | 2225 | 3331 | 3333 | 3335 |
|---|---|---|---|---|---|---|
| CPU time (minutes) | 42 | 342 | 888 | 42 | 620 | 1008 |

**Table 5.5:** Average runtime of SMS-EMOA with problem specific operators over five runs, rounded to the closest whole minute.

## 5.3 Unbiased Operators

Based on the success of the problem specific operators introduced in the previous section, they are developed further. First, bias free versions of the initialisation and mutation operators are introduced. Then, a new version of the tailored SMS-EMOA and its tunable parameters is summarised.

### 5.3.1 Initialisation

The initialisation approach presented in Section 5.1.1 contains some clear biases. Starting from single cell spaces results in a much lower probability of any space being stacked on top of another. While the application of a number of mutations results in a more varied initial population, the starting state of each initialisation is similar. The outcome of the mutations depends on this initial state, and as such does not lead to a truly random initialisation.

Here, an improved initialisation operator is presented that aims to produce an unbiased initial population. This provides a better distribution of the initial population over the search space. In addition, an unbiased random initialisation strategy is useful for a number of landscape analysis approaches.

This newly proposed initialisation method works from the principle that spaces of any shape should have equal probability to be included, as long as there is a large enough area to place them. As a starting point, for a supercube of a given size with a given number of spaces, all possible shapes and all possible positions of those shapes are considered. For a supercube of a size given by its width, depth and height $(N_w, N_d, N_h)$ there are exactly $N_w \times N_d \times N_h$ possible shapes that do not violate any of the constraints, this is depicted in Figure 5.5. The number of possible shapes is limited by the number of spaces $N_{spaces}$ that need to fit in the supercube. Therefore, the largest shape should leave at least $N_{spaces} - 1$ cells empty, such that the remaining spaces can use at least one cell each. It follows from these observations that the maximum size of a shape for a supercube described by the four parameters $N_w, N_d, N_h, N_{spaces}$ is limited to $N_w \times N_d \times N_h - (N_{spaces} - 1)$.

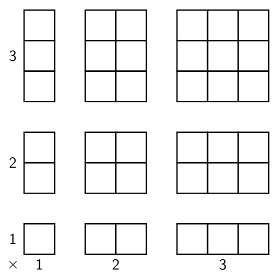



**Figure 5.5:** Possible feasible shapes for a 3 × 3 2D grid, this pattern extends to 3D and therefore to the supercube representation.

A space is then placed by uniformly at random selecting a pair of a possible shape $s$ and a feasible position for that shape. The shape's possible positions are limited by the number of cells belonging to the shape. A heightmap of size $N_w \times N_d$ is considered, and a shape has a width $s_w$, depth $s_d$, and height $s_h$. The heightmap $M_{i,j}$ is initialised as a matrix of zeros, where $i \in \{0, \ldots, N_w - 1\}$ and $j \in \{0, \ldots, N_d - 1\}$. The possible positions of a shape are then every pair $i, j$ where $i + s_w \leq N_w$ and $j + s_d \leq N_d$ hold. A position is feasible if and only if $M_{i,j} + s_h \leq N_h$ and $\forall_{m,n} : M_{m,n} = M_{i,j}$, where $m \in \{i, \ldots, i + s_w\}$ and $n \in \{j, \ldots, j + s_d\}$. Once a shape and position combination is selected the corresponding bits $b^{\ell}_{m+1,n+1,k+1}$ are activated, where $m$ and $n$ are the same as before, $k \in \{M_{m,n}, \ldots, M_{m,n} + s_h\}$ and $\ell$ is the space under consideration. For every space that is placed, the heightmap is updated by $M_{m,n} + s_h$, again for the same values of $m$ and $n$. Finally, once every space is placed, the continuous parameters are initialised uniformly at random within their bounds as in Section 5.1.2.

### 5.3.2 Mutation

Mutation in Section 5.1.2 used a probability to either apply a binary mutation or a continuous mutation. In case of a binary mutation a random space was selected and contracted or expanded in a random direction (Figure 5.1b). For this move the existence constraint (C3: all spaces must consist of at least one cell) and the supercube borders are respected. In other words, any move violating these constraints cannot be chosen. For a continuous mutation, polynomial mutation [32] was always applied to a single continuous variable, randomly selected from those variables that are relevant to at least one active cell. In the following this operator is improved and extended.

Firstly, when continuous mutation is selected, polynomial mutation is applied with some probability to each continuous variable. This has the following implications. There is now a chance that nothing is mutated, but continuous mutations are also no longer limited to a single variable. This means that both small and large changes in the continuous space are possible.

The binary mutation introduced in Section 5.1.2 contains a bias, making some moves more likely to occur than others. However, in the design of evolutionary algorithms, biases in the mutation operator should be avoided [103]. In this case, the bias is an artefact of the used procedure, where first a space is selected, and then a feasible (resulting in no constraint violations) move for that specific space is selected. As a result, when space $A$ has one possible feasible move and space $B$ has three possible feasible moves, the probability to select one of the moves for space $B$ is lower than the probability of selecting the move for space $A$. This is resolved here by selecting a combination of a space and a move instead, resulting in equal probabilities for every move.

Finally, the binary mutation operator is extended to allow mutations consisting of multiple steps. With multiple steps a mutation may result in larger changes to candidate solutions, which may also help to escape local optima. Additionally, multi-step mutations may be able to traverse infeasible regions in case of multiple disconnected feasible regions. In Section 5.1.2 inward moves from the bottom of a space were disallowed because they always result in an infeasible (constraint violating) space by introducing a vertical gap (Constraint C2) in the topological design. Here, such moves are allowed, along with most other constraint violating moves (only Constraint C3 and the supercube boundaries may never be violated). As a result, intermediate moves may lead to infeasible states.

Allowing infeasible intermediate states has a few implications. Figure 5.6 shows how it is possible to move from an initial feasible state to both feasible and infeasible intermediate states. Moreover, the same figure shows how an infeasible intermediate state could lead to both a feasible and an infeasible final state. Moves to infeasible final states are disallowed because only feasible final states are considered for evaluation.

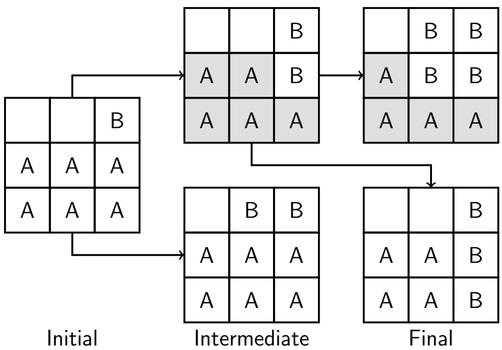



**Figure 5.6:** An initial state can result in both feasible and infeasible intermediate states, in turn an infeasible intermediate state may or may not result in a feasible final state. Infeasible spaces are shaded in grey, in both cases space $A$ is not cuboid and thus violates Constraint C4.

Next, Figure 5.7 shows how spaces with an infeasible shape are mutated. Like with a feasible shape, only the outermost line of spaces in one direction is contracted or extended. Note that moving up from the initial state in the figure would be infeasible since the space would expand beyond the supercube boundary, it would not result in a feasible space of six cells as might be the intuition.

Mutations are then applied as follows: (1) apply a move to a uniformly at random selected combination of a space and movement direction as long as all spaces keep at least one cell (Constraint C3) and supercube boundaries are respected, (2) if this is the final move, try space and direction combinations, in a uniformly at random selected order, until a move is found that satisfies all constraints. If no feasible state is found after exhausting all possible moves the offspring is returned without any mutation (all previous moves are discarded).

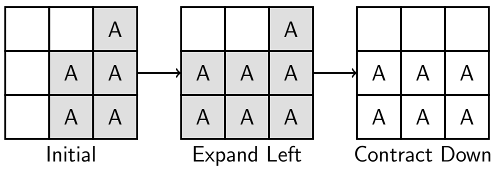



**Figure 5.7:** Expansions and contractions are applied exclusively to the outermost line of cells, even for a space in an infeasible state. The initial state here is infeasible (Constraint C4) and only shown for illustrative purposes. Infeasible spaces are shaded in grey.

### 5.3.3 Tailored SMS-EMOA

The improved operators are integrated into a tailored version of SMS-EMOA, outlined in Algorithm 3. A number of options are available that will be used to tune the algorithm later in this chapter (Section 5.5). Firstly, the population size is as usual controlled by $\mu$. One of two initialisation techniques is selected with $IT$: (1) initialisation as in Section 5.1.2; or (2) unbiased initialisation as proposed in Section 5.3.1. Next, $IM$ controls how many mutations are applied with initialisation technique (1) to increase initial diversity. Parameter $MT$ controls the type of mutation that is applied. It represents the probability to mutate the binary variables with the unbiased mutation operator from Section 5.3.2, and the inverse probability to apply polynomial mutation to the continuous variables. With $ST$ a technique to control the step size in the binary mutation operator is selected: (1) a fixed number of moves $FS$; or (2) pooling, where uniformly at random either a local move of one step or an explorative move of three steps is chosen. In the future, more options, such as step size adaptation, may be explored. When a fixed step size is used $FS$ controls the number of steps, otherwise it has no effect on the algorithm. $MC$ denotes the probability to apply polynomial mutation (if chosen instead of binary mutation) for each continuous variable. The tailored SMS-EMOA (using the improved problem specific operators) is compared to the standard SMS-EMOA (described in Section 5.1.1) after tuning the parameters of both versions in Section 5.5.

**Algorithm 3** Outline of the tailored SMS-EMOA and tunable parameters

**Require:** $IT$, $\mu$, $IM$, $MT$, $ST$, $FS$, $MC$
**if** $IT = 1$ **then**
    Generate population $X$ of $\mu$ parents as in Section 5.1.2
    **for** $i \in \{1, \ldots, IM\}$ **do**
        Mutate all individuals in $X$ as in Section 5.1.2
    **end for**
**else** ▷ $IT = 2$
    Generate population $X$ of $\mu$ parents as in Section 5.3.1
**end if**
**while** Stop condition not met **do**
    $X' \leftarrow$ A uniform random individual from $X$
    **if** $U(0, 1) \leq MT$ **then**
        **if** $ST = 1$ **then** ▷ (Fixed step size)
            n_steps $\leftarrow FS$
        **else** ▷ $ST = 2$ (Pooling)
            **if** $U(0, 1) \leq 0.5$ **then**
                n_steps $\leftarrow 1$ ▷ Local move
            **else**
                n_steps $\leftarrow 3$ ▷ Explorative move
            **end if**
        **end if**
        Mutate binary variables in $X'$ with n_steps as in Section 5.3.2
    **else**
        Apply polynomial mutation to each continuous variable in $X'$ with probability $MC$
    **end if**
    $X \leftarrow$ Select $\mu$ individuals from $X \cup X'$
**end while**

## 5.4 Landscape Analysis

As mentioned in the introduction to this chapter, it is possible to employ landscape analysis both to learn about the objective landscape, and to evaluate variation operators (e.g. mutation, recombination). With both an unbiased initialisation operator, and an unbiased mutation operator having been defined in the previous section, everything is now in place to do landscape analysis. First, Section 5.4.1 describes the considered setup, and then Section 5.4.2 discusses the results.

### 5.4.1 Setup

The landscape of the multi-objective problem is analysed by randomly sampling solutions generated by the newly introduced unbiased initialisation operator (Section 5.3.1). This provides insight in the distribution of solutions over the objective landscape. In addition to random sampling, mutations are applied to investigate the landscape around the randomly sampled points. Through these mutations, the appearance of the local landscape can be investigated. Moreover, it gives insight into the behaviour of the unbiased mutation operator introduced in Section 5.3.2.

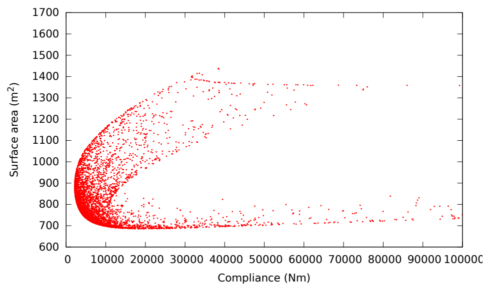
As before, problem instances are defined by four numbers corresponding to the supercube parameters $N_w, N_d, N_h$ and $N_{spaces}$. Experiments are conducted for three different instances: 3331, 3333 and 3335. For each instance fixed step sizes from 1 to 5 are considered. In the experiment a single parent individual is generated with the unbiased initialisation operator, and for this parent a single offspring individual is generated using the unbiased mutation operator. This process is repeated 1000 times for each instance and step size combination, totalling 5000 executions per instance.

In these experiments only binary mutations are considered. The rationale behind this is that each possible distribution of spaces in the supercube corresponds to a different – possibly overlapping – subspace in the objective landscape. When only the continuous variables are changed the parent and offspring would be in the same subspace. Since the number of possible space distributions in the supercube is large, binary mutations will lead to better coverage of the objective landscape. Future work may explore the subspaces more extensively, and for instance investigate the degree of their overlap.

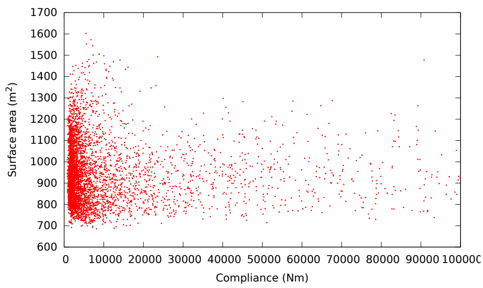
Continuous variables are initialised in $]3, 19.8]$ and a volume $V_0 = 4^3 \times N_{cells}$ is maintained where $N_{cells} = N_w \times N_d \times N_h$. Recall that every individual is scaled to this volume whenever it deviates (e.g. after mutation).

### 5.4.2 Results

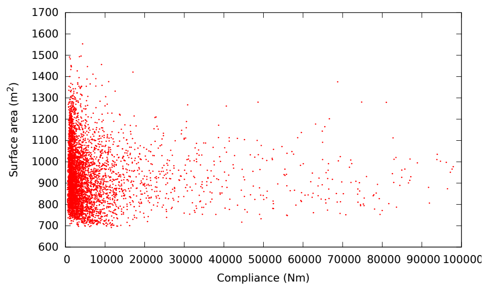
In Figure 5.8 all parent individuals are plotted for the different problem instances (5000 each). For the 3331 instance (Figure 5.8a) a clear Pareto front approximation is visible, which is very similar to the one found in Section 5.2.2. Additionally, many other parts of the landscape also show smooth curves. This suggests that these areas are well covered by random sampling. The largest concentration of points is seen reasonably close to the Pareto front. This makes it probable that random sampling results in a decent solution for this instance.




**(a)** 3331.




**(b)** 3333.




**(c)** 3335.

**Figure 5.8:** Landscapes per problem instance, each with 5000 randomly initialised samples, limited to points with compliance below 100 000 N m.

Both the 3333 and 3335 problem instances (Figures 5.8b and 5.8c) show a different distribution over the objective landscape, indicating that these are, as expected, more complicated. On the other hand, both of these also show a large concentration of points in an area of decent solutions. It appears that finding fairly good solutions is feasible with random sampling, but finding exact optima is still challenging.

Next, the absolute change in the compliance objective values is shown for each instance and step size combination in Table 5.6. First of all, it may be observed that the variance in the change to objective values is large for all considered cases. For instances 3331 and 3333 the change in compliance from small to large step sizes shows a very slight trend towards greater change with larger step sizes. For the 3335 instance the absolute change for different step sizes shows a more parabolic behaviour, where both small and large step sizes have less impact on the compliance than medium step sizes. This is likely caused by larger step sizes reaching infeasible intermediate states in the mutation more frequently, and then being unable to find a feasible final move and reverting back to the original.

| steps | min | max | mean | median | std dev | zeros |
|---|---|---|---|---|---|---|
| | | | instance 3331 | | | |
| 1 | 3.10 | 333667 | 9359.36 | 2833.29 | 24848.95 | 0 |
| 2 | 0.00 | 401687 | 9228.16 | 2048.79 | 30041.31 | 209 |
| 3 | 0.00 | 3621254 | 16002.67 | 3758.13 | 118798.72 | 10 |
| 4 | 0.00 | 36910000 | 50354.11 | 3638.98 | 1166743.33 | 79 |
| 5 | 0.00 | 403216 | 11749.73 | 3516.93 | 32740.83 | 18 |
| | | | instance 3333 | | | |
| 1 | 0.01 | 3.20e+16 | 3.22e+13 | 2672.43 | 1.01e+15 | 0 |
| 2 | 0.00 | 1.29e+15 | 1.31e+12 | 2283.77 | 4.07e+13 | 44 |
| 3 | 0.00 | 9.63e+13 | 1.61e+11 | 4527.37 | 3.58e+12 | 8 |
| 4 | 0.00 | 7.89e+18 | 8.13e+15 | 4319.85 | 2.49e+17 | 36 |
| 5 | 0.00 | 4.28e+18 | 5.00e+15 | 4823.08 | 1.37e+17 | 35 |
| | | | instance 3335 | | | |
| 1 | 1.02 | 3.07e+17 | 4.75e+14 | 1235.32 | 1.11e+16 | 0 |
| 2 | 0.00 | 5.45e+33 | 5.45e+30 | 1331.23 | 1.72e+32 | 19 |
| 3 | 0.00 | 8.68e+22 | 8.68e+19 | 1999.31 | 2.74e+21 | 12 |
| 4 | 0.00 | 7.78e+12 | 1.74e+10 | 1709.64 | 3.26e+11 | 16 |
| 5 | 0.00 | 3.96e+17 | 3.96e+14 | 1153.29 | 1.25e+16 | 28 |

**Table 5.6:** Statistics for absolute change in compliance per instance.

Absolute change to the surface area in Table 5.7 shows a somewhat clearer trend across the board. For instance, the trend of larger changes for larger step sizes seems to be more pronounced when, for instance, looking at the mean and median.

Figure 5.9 visualises how objective values are distributed in both objectives for the 3333 instance. The other instances produced largely similar visuals. Although the differences in absolute change do not appear to be very large between the various step sizes, there does appear to be a trend towards larger changes for larger step sizes.

For both objectives zero changes also occur (see the rightmost column in Tables 5.6 and 5.7). There are multiple explanations for this. Firstly, when multiple steps are taken a second move may be the inverse of the first and as such return to the original solution. A second explanation is found in the design of the mutation operator, where the algorithm reverts to the original individual if no feasible final moves are possible. Finally, there are moves where only the interior of the building design changes, in other words, where only walls are moved. In those cases the compliance changes, but the surface area does not.

| steps | min | max | mean | median | std dev | zeros |
|---|---|---|---|---|---|---|
| | | | instance 3331 | | | |
| 1 | 0.22 | 632.13 | 77.51 | 51.28 | 82.98 | 0 |
| 2 | 0.00 | 534.44 | 74.70 | 36.49 | 96.36 | 209 |
| 3 | 0.00 | 657.30 | 95.10 | 61.32 | 99.20 | 10 |
| 4 | 0.00 | 667.93 | 100.61 | 64.83 | 115.66 | 79 |
| 5 | 0.00 | 662.83 | 104.26 | 67.63 | 107.07 | 18 |
| | | | instance 3333 | | | |
| 1 | 0.00 | 450.32 | 64.02 | 40.59 | 70.70 | 20 |
| 2 | 0.00 | 641.03 | 70.54 | 42.61 | 84.05 | 50 |
| 3 | 0.00 | 478.77 | 84.81 | 57.22 | 86.48 | 16 |
| 4 | 0.00 | 540.57 | 93.02 | 62.35 | 96.33 | 38 |
| 5 | 0.00 | 642.32 | 96.46 | 63.63 | 104.26 | 36 |
| | | | instance 3335 | | | |
| 1 | 0.00 | 482.75 | 50.78 | 28.10 | 67.51 | 15 |
| 2 | 0.00 | 573.65 | 57.96 | 28.76 | 77.71 | 27 |
| 3 | 0.00 | 532.80 | 68.83 | 40.19 | 84.21 | 17 |
| 4 | 0.00 | 520.17 | 71.95 | 40.28 | 92.19 | 15 |
| 5 | 0.00 | 470.09 | 63.18 | 27.68 | 83.76 | 28 |

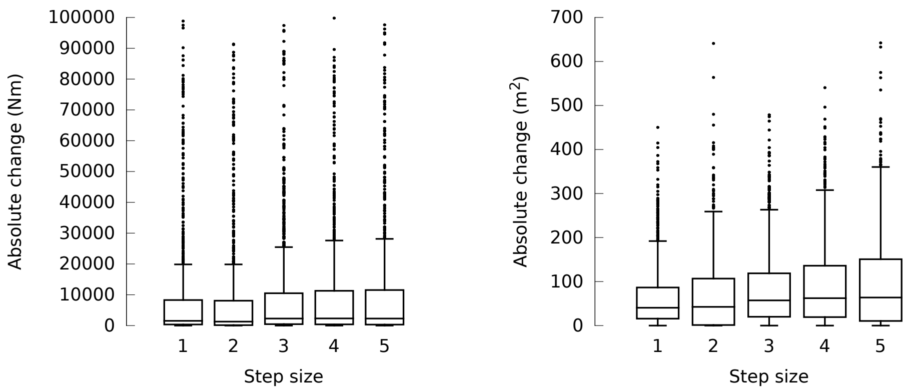
**Table 5.7:** Statistics for absolute change in surface area per instance.

## 5.5 Optimisation and Parameter Tuning

The landscape analysis in the previous section showed that the unbiased initialisation operator is able to generate a diverse set of solutions, and the unbiased mutation operator produces larger changes with larger step sizes. Having confirmed these desirable properties of the operators, they will now be evaluated in an optimisation setting. This section first briefly reviews the experimental setup from Section 5.2.1, which will be used here as well. Next, the considered setup for algorithm tuning is introduced. Finally, the results of these experiments are discussed.

### 5.5.1 Optimisation Setup

SMS-EMOA as used here is the standard SMS-EMOA from [39], with the only changes being the volume rescaling and the evaluation as described in the following paragraphs (see also Section 5.1.1). A tailored version of SMS-EMOA with problem specific operators for building spatial design is used, this will be referred to as tailored SMS-EMOA. Changes are as follows: (1) problem specific initialisation is used, where tuning will select between the biased version from Section 5.1.2, and the unbiased one from Sec-




**(a)** Compliance.

**(b)** Surface area.

**Figure 5.9:** Absolute change to the objective values with 1000 samples for a single mutation of 1 through 5 steps for the 3333 problem instance.

tion 5.3.1, and (2) offspring is generated with the unbiased problem specific mutation operator as described in Section 5.3.2. All other settings, except for the evaluation budget and tunable parameters detailed later, are exactly as in Section 5.2.1. Note that no recombination operator is used with the tailored SMS-EMOA. The reason behind this is that the standard recombination operators result in many constraint violations because of the many infeasible solutions in the supercube as reported in Section 3.2.4. Due to the complexity of developing such an operator, and the promising results when only using mutation as shown in Section 5.2.2, development of a problem specific recombination operator is deferred to future work.

For both the standard and tailored versions of SMS-EMOA the considered optimisation problem is as follows. Objective functions are (1) the compliance in N m (newton metre), and (2) the surface area in $m^2$ (square metre), both to be minimised. Binary and continuous variables are considered as described in Section 3.2. The binary variables are subject to four topological constraints as described mathematically in Section 3.2.2. To comply to the volume constraint (Section 3.2.3), the continuous variables are rescaled for any new individuals that are produced. Both algorithms consider the reference point $(1.1e9, 1.1e9)$.

Function evaluations for both of these algorithms proceed as follows. First constraints are checked. For constraint violations a penalty of $999\,999\,998 + CV$ is returned as the objective value for both objectives, for a number of constraint violations $CV$. As in previous experiments, the fourth constraint is considered in two parts, resulting in $CV \in \{1, \ldots, 5\}$. If the objective value is 999 999 999 or higher (i.e. at least one constraint is violated) no evaluation is counted, but the individual may still be selected, provided it passes the selection criteria. In other words, any individual violating a constraint or showing extremely poor performance is considered infeasible. Note that although constraint violations do not occur with the tailored SMS-EMOA, there may be individuals with extremely poor performance. In all other cases the remaining evaluation budget is decreased by one.

### 5.5.2 Parameter Tuning Setup

Acknowledgement

The author gratefully acknowledges the helpful advice of Leslie Pérez Cáceres and Manuel López-Ibáñez on setting up and using the irace package.

To tune the parameters of the standard and tailored SMS-EMOA for building spatial design, two approaches are compared: The irace package [71] and the Mixed-Integer Evolution Strategy (MIES) [69]. The objective is the maximisation of the hypervolume indicator (HVI, sometimes also referred to as just hypervolume), with reference point $(100000, 1500)$. Note that here the hypervolume is taken over all evaluated individuals, not just over the individuals in the final population.

Objective function values are normalised to a $[0, 1]$ range for both objectives. For the compliance the used range is $[0, \ldots, 100000]$ and for the surface area a range of $[0, \ldots, 1500]$ is considered. Data points exceeding either of these ranges are excluded from the analysis. These ranges are based on investigation of the result data from Section 5.2.2. This ensures that only extreme outliers are discarded and the data can still be analysed visually.

For the standard SMS-EMOA three tunable parameters are considered as shown in Table 5.8. The range for the mutation probability is restricted for the standard SMS-EMOA because large mutations are likely to result in infeasible individuals. Too small population sizes are also avoided since an initial population without feasible individuals has great difficulty navigating to a set of feasible individuals even with the graded penalties for constraint violations. As a result, the algorithm would use extreme amounts of time without being able to evaluate feasible individuals.

For the tailored SMS-EMOA the parameters from Algorithm 3 (Section 5.3.3) are tuned within ranges as shown in Table 5.9 (parameters without type are categorical).

From the results of the landscape analysis in Section 5.4.2 it can be concluded

| parameter | symbol | range | type |
|---|---|---|---|
| Population size | $\mu$ | $\{10,\ldots,100\}$ | $\mathbb{Z}$ |
| Recombination probability | $RP$ | $[0.0, 1.0]$ | $\mathbb{R}$ |
| Mutation probability | $MP$ | $[0.005, 0.100]$ | $\mathbb{R}$ |

**Table 5.8:** Configurable parameters for the standard SMS-EMOA.

| parameter | symbol | range | type |
|---|---|---|---|
| Population size | $\mu$ | $\{10,\ldots,100\}$ | $\mathbb{Z}$ |
| Mutation type probability | $MT$ | $[0.0, 1.0]$ | $\mathbb{R}$ |
| Step size technique | $ST$ | $\{1, 2\}$ | |
| Fixed number of steps | $FS$ | $\{1,\ldots,5\}$ | $\mathbb{Z}$ |
| Continuous mutation probability | $MC$ | $[0.0, 1.0]$ | $\mathbb{R}$ |
| Initialisation technique | $IT$ | $\{1, 2\}$ | |
| Initialisation mutations | $IM$ | $\{1,\ldots,100\}$ | $\mathbb{Z}$ |

**Table 5.9:** Configurable parameters for the tailored SMS-EMOA.

that the 3331 (supercube of size $3 \times 3 \times 3$ with a single space) problem instance is rather simple. A more challenging tuning problem can be found in the 3333 instance. Moreover, the evaluation time for this problem is less prohibitive than for the 3335 instance, as reported in Section 5.2.2. As such, the 3333 instance is used in the experiments. The tuning budget consists of two parts, namely (1) the number of algorithm executions, and (2) the number of function evaluations per execution. Firstly, the number of algorithm executions is set to the minimum requirement enforced by the irace package, which was 180 algorithm executions for the given settings and tunable parameters. Secondly, the results in Section 5.2.2 show convergence to a near stable state in about 300 function evaluations. With this in mind an evaluation budget of 300 is considered for tuning.

Both irace [71] (version 2.1.1662) and MIES [69] are used primarily with default settings as described in their respective papers, any deviations from those settings will be explicitly stated. As a result performance of both approaches is likely not optimal, but the configuration of parameter tuners could be the subject of a whole other study, and is considered out of scope here.

MIES is used with multiple step size mode for both real valued and integer parameters, and single step size mode for categorical parameters. In [69] it is suggested to keep step size bounds for categorical variables in $[\frac{1}{n_d}, \frac{1}{2}]$ when using single step size mode for a number of categorical variables $n_d$. However, for $n_d = 2$ this means the step size is constant at $\frac{1}{2}$. Since two categorical variables are considered for this experiment, and a fixed step size is unable to adapt anything, categorical step sizes are instead bounded in $[\frac{1}{3}, \frac{1}{2}]$.

In [42] a $(\mu, \kappa, \lambda)$ strategy is shown to converge faster than a $(\mu, \lambda)$ strategy. Here $\kappa$ is the maximum number of generations that an individual can stay in the population. Due to the small number of available evaluations, faster convergence is desirable here. Instead of the $(3, 5, 10)$ strategy from [42], here a $(3, 3, 10)$ strategy is used. The intuition is that due to the small number of generations, keeping individuals in the population for a too large fraction of the generations would otherwise not differ from a plus-strategy (see Section 2.3 for an explanation of a plus-strategy).

### 5.5.3 Tuned Configurations

Table 5.10 shows the configurations for standard SMS-EMOA. Here an untuned configuration (US) is included for comparison purposes. The parameters of this configuration are almost the same as those used in Section 5.2.1. Only $MP$ is now 0.0111, instead of exactly $\frac{1}{N_{dims}}$.

| ID | $\mu$ | $RP$ | $MP$ | mean HVI |
|---|---|---|---|---|
| | | untuned configuration (US) | | |
| US | 50 | 0.5000 | 0.0111 | 0.5348 |
| | | irace configurations (IxS) | | |
| I1S | 31 | 0.8984 | 0.0385 | 0.5380 |
| I2S | 41 | 0.9650 | 0.0520 | 0.5385 |
| I3S | 73 | 0.8677 | 0.0427 | 0.5381 |
| Mean | 48 | 0.8942 | 0.0514 | N/A |
| Std | 19 | 0.1133 | 0.0105 | N/A |
| | | MIES configurations (MxS) | | |
| M1S | 15 | 0.9679 | 0.0323 | 0.5386 |
| M2S | 40 | 0.5567 | 0.0891 | 0.5365 |
| M3S | 5 | 0.9709 | 0.0351 | 0.5364 |
| Mean | 35 | 0.8088 | 0.0545 | N/A |
| Std | 35 | 0.1834 | 0.0262 | N/A |

**Table 5.10:** Parameter configurations for the standard SMS-EMOA: untuned and from three repetitions of irace and MIES.

Both irace and MIES used around two weeks of computation time on a single CPU core per repetition to tune the parameters. The number of output configurations for irace can vary, but in this case there were nine in total after three repetitions. Since irace clearly indicates which configuration it considers to be the best, this configuration is shown in Table 5.10 for each repetition. For MIES there are always three output individuals, because the population size $\mu = 3$. As a result, MIES also has a total of nine output configurations after three repetitions. For each repetition of MIES, the configuration with the largest hypervolume coverage is considered to be the best and shown in Table 5.10. The table also shows the mean and standard deviation for each parameter over all nine output configurations for both irace and MIES.

Compared to the untuned configuration (US), both tuning approaches found higher recombination ($RP$) and mutation ($MP$) probabilities for the standard SMS-EMOA. A possible explanation is that by producing larger and more frequent variations, the algorithm is able to explore more effectively, and in turn finds better solutions. While this is also likely to lead to more constraint violations, such solutions are not counted for the evaluation budget here, and thus in fact result in more solutions being considered. In terms of population size not much can be concluded, except that many different population sizes seem to allow for similar performance. Part of the reason may be that all considered solutions are counted towards the final Pareto front approximation, and therefore the population size has less influence. However, the population size also plays a role in the diversity of possible offspring individuals that can be reached, which is not explained by this argument. Combined with the larger mutation and recombination rates however, it may simply not play a very large role in reaching diverse solutions.

Parameter configurations for the tailored SMS-EMOA are reported in Table 5.11. Here too, the settings for the untuned configuration (UT) are mostly as in Section 5.2.1. Only $MC$ is adjusted because continuous mutation changed in the unbiased mutation operator (Section 5.3.2). For this application, irace found eleven solutions in total, of which again only the best reported per repetition is shown in the table. For MIES once more nine configurations were found, and for each repetition the one with the largest hypervolume coverage is shown. As before, the mean and standard deviation for each parameter are taken over the full set of output configurations, rather than only the best per repetition.

With the tailored version using a smaller population size ($\mu$), more frequent use of the binary mutation operator ($MT$) and an increased probability to apply continuous mutations ($MC$) seems to improve performance over the untuned configuration. While for the standard SMS-EMOA increased variation probabilities could be explained by being advantageous in the sense that constraint violations are not counted as evaluation anyway, this cannot be the reason here. For the tailored SMS-EMOA it appears that the increase in exploration caused by these settings is more advantageous than

| ID | $\mu$ | $MT$ | $ST$ | $FS$ | $MC$ | $IT$ | $IM$ | mean HVI |
|---|---|---|---|---|---|---|---|---|
| | | | | Untuned configuration (UT) | | | | |
| UT | 50 | 0.2500 | 1 | 1 | 0.3333 | 1 | 20 | 0.5384 |
| | | | | irace configurations (IxT) | | | | |
| I1T | 32 | 0.6890 | 1 | 4 | 0.6686 | 1 | 3 | 0.5390 |
| I2T | 21 | 0.2794 | 2 | N/A | 0.6894 | 2 | N/A | 0.5388 |
| I3T | 26 | 0.3960 | 2 | N/A | 0.3231 | 1 | 64 | 0.5393 |
| Mean | 28 | 0.5618 | 1.5 | 3.8 | 0.4752 | 1.4 | 34 | N/A |
| Std | 16 | 0.1463 | 0.5 | 0.4 | 0.2338 | 0.5 | 30 | N/A |
| | | | | MIES configurations (MxT) | | | | |
| M1T | 12 | 0.6212 | 1 | 2 | 0.7970 | 1 | 69 | 0.5374 |
| M2T | 6 | 0.4993 | 2 | N/A | 0.4381 | 1 | 60 | 0.5374 |
| M3T | 5 | 0.1176 | 2 | N/A | 0.5118 | 1 | 43 | 0.5365 |
| Mean | 14 | 0.4413 | 1.3 | 2.5 | 0.6780 | 1.4 | 53 | N/A |
| Std | 9 | 0.1791 | 0.5 | 0.5 | 0.1921 | 0.5 | 11 | N/A |

**Table 5.11:** Parameter configurations for the tailored SMS-EMOA: untuned and from three repetitions of irace and MIES.

staying close to known solutions. It is also notable that there is a slight preference for the biased initialisation technique ($IT = 1$). A possible explanation is that the use of a large number of initialisation mutations ($IM$) mitigates the bias. What may also play a role is that although the unbiased initialisation technique ($IT = 2$) is more random, such solutions are not necessarily better.

In order to compare the optimised configurations and their corresponding algorithms each of them is evaluated on the considered optimisation problem (Section 5.5.1). Each configuration is compared at two stages in the optimisation process. First with a budget of 300 evaluations using fifteen repetitions, and then with a budget of 1000 evaluations using thirteen repetitions. In both cases all other settings are the same as before. The mean hypervolume (HVI) values after 1000 evaluations are also included in Tables 5.10 and 5.11.

Median attainment curves [49] of the resulting data are used to compare the different configurations, as well as the two tuning methods. In addition to those configurations, the Pareto front approximation found by taking 5000 random samples (RS) for the landscape analysis from Section 5.4.2 is also included. This makes it possible to compare random sampling with both SMS-EMOA variants. Figure 5.10 shows the situation after 300 evaluations.

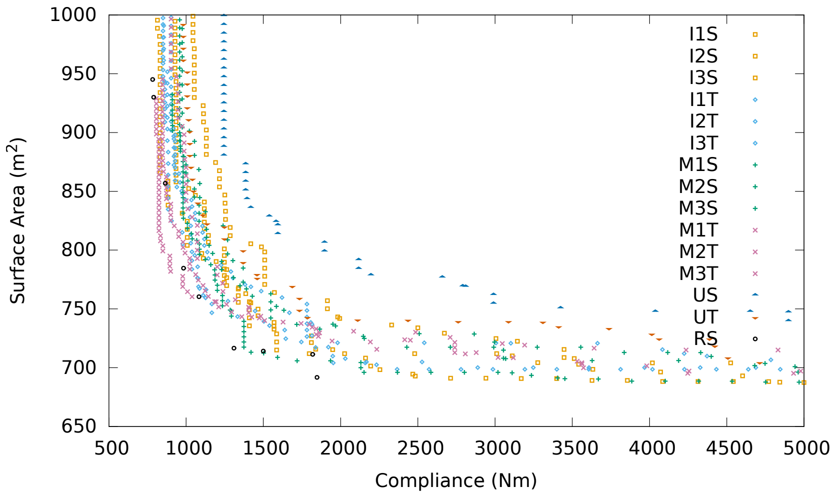



**Figure 5.10:** Median attainment curves after 300 evaluations, over fifteen repetitions for configurations tuned by irace and MIES for standard and tailored versions of SMS-EMOA, untuned configurations, and a Pareto front approximation from 5000 random samples.

A first observation is that even with only 300 evaluations, a number of configurations are already competitive with the results of 5000 random samples (RS). Evidently, optimisation is effective here. Furthermore, the untuned configuration for the standard SMS-EMOA (US) is clearly not competitive with anything else, while the untuned tailored SMS-EMOA (UT) performs better, it is among the worst of the other configurations. This shows that tuning has been worthwhile in improving performance. Notably, all configurations tuned for the standard SMS-EMOA (xxS) outperform the untuned configuration in every part of the Pareto front approximation (PFA). For the tailored SMS-EMOA (xxT), this is true for almost all parts of the PFA. In terms of performance differences between tuned configurations there does not seem to be a clear winner as far as optimal performance goes. However, there does seem to be less variance between configurations found for the tailored SMS-EMOA, indicating that it may be a more robust algorithm. With regard to the tuning approaches, it appears that MIES is prone to finding high quality solutions in one objective, but then slightly less so in the other objective. On the other hand irace seems to usually find a more balanced PFA, but as a result does not find the highest quality solutions for every part of the Pareto front.

Next, Figure 5.11 shows the situation for the same configurations after 1000 evaluations. It is obvious that everything progressed to find better solutions, and the use of more evaluations was a worthwhile investment. For the surface area part of the PFA, almost all configurations reach a very similar approximation. More variation is seen in the knee point region and the compliance objective, where sometimes MIES tuned tailored SMS-EMOA configurations are better, and other times the irace tuned

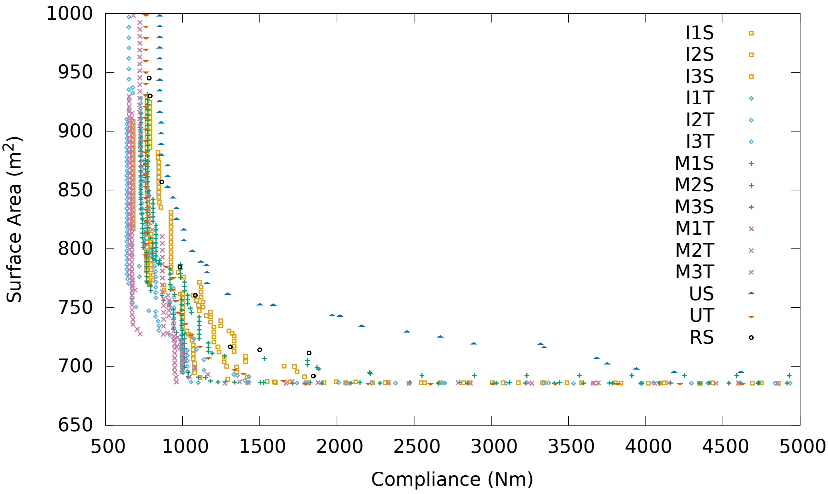



**Figure 5.11:** Median attainment curves after 1000 evaluations, over thirteen repetitions for configurations tuned by irace and MIES for standard and tailored versions of SMS-EMOA, untuned configurations, and a Pareto front approximation from 5000 random samples.

versions. Except for the occasional outlier, the standard SMS-EMOA configurations are not competitive in these areas of the PFA. Besides the untuned standard SMS-EMOA, all configurations now outperform, or are competitive with the 5000 random samples.

In summary it is evident that even with a relatively limited tuning budget both tuning approaches (irace and MIES) are able to improve over untuned configurations, but there is no clear preference for one or the other. Although the standard SMS-EMOA is not much worse in its final results, it appears to be less robust (even with tuned configurations) in its performance than the tailored version. Moreover, based on the results in Figure 5.11 it also appears to be the case that the standard SMS-EMOA is slower to converge to the true Pareto front, and might even get stuck before reaching it. Even so, these are promising results since based on Section 5.2.2 it can be expected that differences between the standard and tailored methods will only be larger for more difficult and realistic problem instances. Clearly, the development of the tailored SMS-EMOA has been beneficial.

# 5.6 Conclusion

## 5.6.1 Summary

Due to the limited success of penalty methods in navigating the constrained landscape of the building spatial design problem, this chapter continued the investigation of how to navigate these landscapes effectively (RQ2). To this end, problem specific operators have been developed. Specifically, initialisation and mutation operators. In addition, with a basic understanding of the objectives (thermal and structural performance) having been established, this chapter addressed them in a multi-objective fashion.

Development of the problem specific operators led to basic initialisation and mutation techniques that can only reach feasible solutions (non constraint violating). In order to assess the performance of an algorithm using these operators a comparison has been made to an algorithm using the penalty techniques previously developed in Chapter 4. The results showed that while both approaches work for the most basic problem instances, the penalty based algorithms quickly fall behind when larger problems are considered. Clearly, the problem specific operators are beneficial in navigating the search space.

Although these basic problem specific operators already showed promise, they contained clear biases. Moreover, while it is not clear whether any disconnected feasible regions exist, the basic mutation operator was unable to reach such regions due to fixed size moves. To resolve this, as well as the bias, an unbiased problem specific mutation operator has been developed. Likewise, an unbiased initialisation operator has also been introduced.

As a first test for these new unbiased operators, they have been subjected to landscape analysis. The initialisation operator showed differences in search space complexity between a very basic problem instance, and two more complex variants. In addition, the mutation operator showed a correlation between its step size and the degree of change to the produced solutions. This is a desirable property since it means that this parameter can control the balance between exploration and exploitation of this operator.

Following the landscape analysis, parameter tuning has been applied in order to assess whether the unbiased operators are indeed preferable over the basic ones. In addition, this served to compare the parameter tuning package irace [71] to the mixed-integer evolution strategy (MIES) [69]. The results showed that the unbiased mutation operator indeed reaches better performance than the basic one. For the initialisation operators there was no great preference for either, but since the unbiased operator initialises a diverse population with less computational effort it is recommended over the basic version. In terms of tuning no clear advantage has been shown for either the irace package or MIES. Evidently, although less commonly used, MIES is an adequate tool to tune the parameters of an algorithm.

### 5.6.2 Future Work

The results with the problem specific operators showed that a mutation only evolutionary algorithm is able to navigate the constrained landscape. However, in order to scale up to larger problem sizes (buildings with more spaces) further research in this direction is needed. One aspect that could help the search process is the inclusion of a recombination operator. What makes this a challenging research direction is that it is not obvious how to recombine the discrete part of the supercube representation in such a way that designs remain both feasible, and result in a reasonable combination of two parent designs.

An aspect of the problem specific mutation operator that could be improved is the inclusion of step size adaptation. This is a standard component of Evolution Strategies [81, 88], that allows the mutation strength to be adapted during the execution of the algorithm. Through this process, the algorithm can start with a more explorative approach, and become more exploitative as it converges. The main challenge here is figuring out what an effective adaption mechanism is for the newly introduced operator.

Landscape analysis showed how the considered problem quickly becomes more difficult when problem instances are considered beyond the most basic versions. What is not yet clear is how the interplay between the continuous and discrete components of the supercube representation influences the complexity. To elucidate this, it should be investigated to which degree different discrete subspaces overlap in objective space, as a result from continuous variations within these distinct discrete subspaces.

Since the considered problem of building spatial design requires costly evaluations, algorithms that require fewer evaluations would be valuable. One way to reduce the number of evaluations that are needed is the use of so-called surrogate models (also called metamodels). Such models aim to provide a cheaper alternative for the actual objective function, at the cost of accuracy. How to integrate this into the building spatial design problem, and the used mixed-integer representation is an open question.

Like in the optimisation process, surrogate models can also be applied in parameter tuning. As seen in this chapter, tuning algorithms for relatively small building spatial designs is already expensive. Future work can investigate how parameter tuning techniques with surrogate models (e.g. SPOT [9] and SMAC [55]) can be used to cope with tuning algorithms that optimise larger building spatial designs.