Configuring Advanced Evolutionary Algorithms for Multicriteria Building Spatial Design Optimisation

Koen van der Blom*, Sjonnie Boonstra†, Hèrm Hofmeyer†, Thomas Bäck* and Michael T. M. Emmerich*
*Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands
Email: {k.van.der.blom, t.h.w.baeck, m.t.m.emmerich}@liacs.leidenuniv.nl
†Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands
Email: {s.boonstra, h.hofmeyer}@tue.nl

***Abstract*—In this paper solution approaches for solving the building spatial design optimisation problem for structural and energy performance are advanced on multiple fronts. A new initialisation operator is introduced to generate an unbiased initial population for a tailored version of SMS-EMOA with problem specific operators. Improvements to the mutation operator are proposed to eliminate bias and allow mutations consisting of multiple steps. Moreover, landscape analysis is applied in order to explore the landscape of both objectives and investigate the behaviour of the mutation operator. Parameter tuning is applied with the irace package and the Mixed Integer Evolution Strategy to find improved parameter settings and explore tuning with a relatively small number of expensive evaluations. Finally, the performances of the standard and tailored SMS-EMOA algorithms with tuned parameters are compared.**

I. Introduction

Building design is a highly multi-disciplinary problem. Single objective optimisation in building design like in [1] or [2] is already challenging. Complex requirements for each discipline and intricate relations between the disciplines makes designing with multiple disciplines in mind prohibitively difficult for humans. Multicriteria optimisation is effective in building design as shown in [3] and may aid domain experts to integrate different disciplines. Here, the bi-objective problem of building spatial design optimisation for structural (compliance) and energy (surface area) efficiency is considered.

This building spatial design optimisation problem as previously described in [4] considers a number of constraints. Both in [4] and in [5] it was noted that standard single and multi-objective evolutionary algorithms have difficulty navigating the constraint landscape for anything but the simplest problem instances. In [5] domain specific operators were introduced to manage these constraints. Here, the mutation operator from [5] is extended to allow larger mutations and the previous naive initialisation operator is replaced by an unbiased variant. These techniques may in the future be used in combination with a superstructure free design space and heuristic methods as suggested in [6].

Landscape analysis provides insight into the objective landscape corresponding to an objective function. Various methods for landscape analysis have been proposed in the literature, see e.g. [7], [8]. A simple way of doing landscape analysis is to look at the distribution of function values over random samples, for example by density of states analysis [9]. Additionally, landscape analysis can be used to identify how variation operators (mutation, recombination) behave in the objective landscape [10]. In this work an improved mutation operator is evaluated to investigate its behaviour for different step sizes on multiple problem instances.

In the field of algorithm tuning the aim is to find the optimal settings for an algorithm to solve a problem. For instance, in [11] 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 [12], irace [13] and SPOT [14]. Another way to tune algorithms is by integrating problem knowledge into operators, which can significantly improve performance as shown for chemical process design in [15]. Many real-world optimisation problems, such as building 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 [13] based on statistical significance of configurations and the Mixed Integer Evolution Strategy (MIES) [16], a self-adaptive evolutionary algorithm with specialised operators for mixed integer problems, are compared here for the tuning and configuration of the standard SMS-EMOA [17] and a tailored version with the newly introduced domain specific operators.

In summary the novelties of this work are: (1) improved problem specific mutation and initialisation operators, (2) landscape analysis of the buildings spatial design optimisation problem to identify problem features and to investigate the behaviour of the mutation operator, and (3) parameter tuning for the considered algorithms and discussion of parameter tuning for optimisation problems with time consuming evaluation functions. Moreover, two parameter tuning algorithms are compared and their different merits are discussed.

The remainder of this paper is structured as follows. First, Section II provides a problem description and a mixed integer representation for solution candidates. In Section III the new initialisation and mutation operators are described. Section IV details the analysis of the objective landscapes for structural design and energy efficiency. To compare the standard SMS-EMOA algorithm and a tailored version with the operators described in this paper, Section V discusses parameter tuning for both of these approaches. Finally, Section VI contains the discussion and conclusions.

978-1-5090-4601-0/17/$31.00

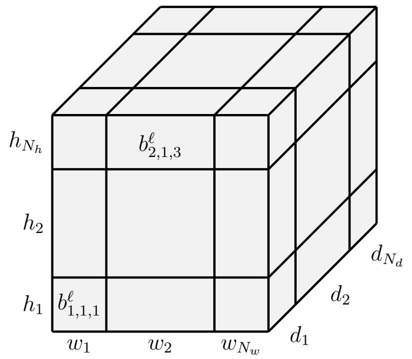


Fig. 1. Visualisation of the supercube representation.

II. Problem

A. Background and Definition

This work focuses on two aspects of building spatial design: (1) structural efficiency, and (2) energy efficiency.

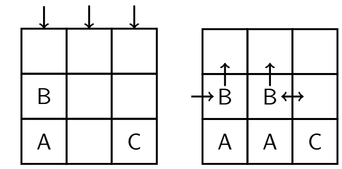
Structural efficiency is important in that it encompasses the maximal contribution of every individual structural element (e.g. a concrete wall) to the full structure, and thus aims at the optimal use of materials. For example, a structural element that has no strain on it, does not contribute to the structural efficiency. On the other hand, if all strain is on a single structural element, the total strain must be reduced to prevent material failure. This is achieved by distributing the strain over multiple structural elements. For this study the minimal total compliance, which relates to the stiffness of the structure, is used to measure the structural efficiency.

The importance of energy efficiency should be evident, as it allows reductions to both the impact on the environment and to monetary costs. Thermal control is a significant part of energy use in buildings, where heating and cooling are used to maintain the desired temperature. Here, thermal efficiency is measured by the minimal outer surface area (minus the ground floor surface) of the building spatial design, as a substitute for a more sophisticated simulation to be used in future work.

In summary the optimisation problem may then be stated as the minimisation of (1) the total compliance and (2) the outer surface area of a building spatial design.

B. Supercube Representation

To aid with the understanding of the rest of this study, first the supercube representation (Figure 1) and the accompanying constraints are briefly reintroduced. For more details the reader is referred to [4], [6]. The supercube representation defines a cuboid (3D rectangle) grid with $N_w \times N_d \times N_h$ subdivisions (cells), which is used as a superstructure for the topological design of a building. A *space* represents a part of a building (similar to a room), consisting of one or more cells. For each space $\ell \in \{1, \ldots, N_{spaces}\}$ each cell may be turned on or off by a binary variable $b^\ell_{i,j,k}$ (where $i \in \{1, \ldots, N_w\}$, $j \in \{1, \ldots, N_d\}$, and $k \in \{1, \ldots, N_h\}$). Moreover, different slices of the supercube may be resized by their continuous dimensioning variables w_i, d_j, and h_k, as shown in Figure 1.



(a) Initialisation (b) Mutation

Fig. 2. Operators as used in [5].

Which cells are active and which cells a space occupies is restricted by a number of constraints on the binary variables. These constraints are the following: (C1) An active cell ($b^\ell_{i,j,k} = 1$) may not have empty space below it, i.e. it must be at the lowest h index or have another active cell below it. (C2) Every space must consist of at least one active cell. (C3) Every cell (i, j, k) in the supercube belongs to at most one space. (C4+C5) All active cells belonging to a space must together form a cuboid shape, which is mathematically described as a two part constraint. The mathematical descriptions of these constraints can be found in [4], [6]. The continuous variables are subject to a constraint on the volume V_0 of the spatial design. Whenever an individual deviates from this volume (e.g. after initialisation, recombination, mutation), the continuous variables are rescaled in equal proportions to comply with the constraint (for details see [5]).

III. Improved Operators

A. Mutation

Mutation in [5] was defined as follows. With some probability either a binary mutation or a continuous mutation is applied. In case of a binary mutation a random space is selected and contracted or expanded in a random direction as shown in Figure 2b. For this move the existence constraint (C2: all spaces must keep 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 [18] is always applied to a single continuous parameter, randomly selected from those parameters 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 way, both small and large changes in the continuous space are possible.

The binary mutation introduced in [5] contains a bias where some moves are more likely to occur than others. However, in the design of EAs, bias in the mutation operator should be avoided [19]. This bias is an artifact 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

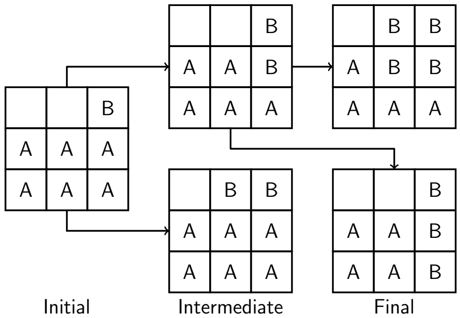


Fig. 3. Feasible and infeasible intermediate moves. An initial state can result in both feasible and infeasible intermediate states, in turn an infeasible intermediate state may or may not result in a feasible final state.

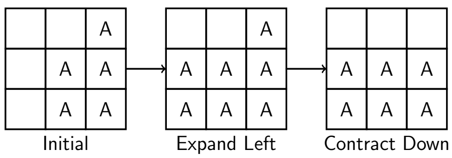


Fig. 4. Expansions and contractions are applied exclusively to the outermost line of cells, even for a space in an infeasible state. The initial state here is normally infeasible (C4+C5) and only shown for illustrative purposes.

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 might help to escape local optima. Additionally, multi-step mutations may be able to traverse infeasible regions in case of multiple disconnected feasible regions. In [5] inward moves from the bottom of a space were disallowed because they always result in an infeasible (constraint violating) space by introducing a vertical void (C1) in the topological design. Here, such moves are allowed, along with most other constraint violating moves. As a result intermediate moves may lead to infeasible states.

Allowing infeasible intermediate states has a few implications. Figure 3 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, which is disallowed because only feasible final states are considered for evaluation. Next, Figure 4 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 no spaces are deleted (C2) and supercube boundaries are not violated, (2) if this is the final move, try space and direction combinations in an order selected uniformly at random 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).

B. Initialisation

Initialisation as presented in [5] is defined by the following mechanism. As depicted in Figure 2a spaces were placed by assigning them to the lowest empty cell in a randomly selected pillar of the supercube. After assigning a single cell to each space, a number of smart mutations as described in [5] (see Subsection III-A) are applied to the entire individual.

The initialisation approach presented in [5] 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 smart mutations results in a more varied initial population, the starting state of the initialisation means that the result of these mutations has a higher probability to be close to the 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. The number of possible shapes is limited by the number of spaces N_{spaces} that need to fit in the supercube. The largest shape should therefore 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 $N_w \times N_d \times N_h - (N_{spaces} - 1)$.

A space is then placed by uniformly at random selecting a pair of a possible shape S and a feasible position for that shape S_p. A 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 $H_{i,j}$ 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 iff $H_{i,j} + S_h \leq N_h$ and $\forall_{m,n} : H_{m,n} = H_{i,j}$, 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, $k \in$

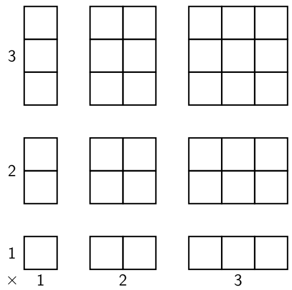


Fig. 5. Possible feasible shapes for a 3×3 2D grid, this pattern extends to 3D and therefore to the supercube representation.

$H_{n,m}, \ldots, H_{n,m} + S_h$ and ℓ is the space under consideration. For every space that is placed, the heightmap is updated by $H_{m,n}+S_h$, 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 [5].

C. Tailored SMS-EMOA

The improved operators are integrated into a tailored version of SMS-EMOA, outlined in Algorithm 1. A number of options are available that will be used to configure the algorithm later in this work. Firstly, the population size is as usual controlled by μ. One of two initialisation techniques is selected with IT: (1) initialisation as in [5]; or (2) initialisation as proposed in Subsection III-B. 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 mutation operator from Subsection III-A, 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 SF; and (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. With MC the probability to apply polynomial mutation (if chosen instead of binary mutation) for each continuous variable may be set. The tailored SMS-EMOA (using the improved operators) is compared to the standard SMS-EMOA (with standard operators) after tuning the parameters of both versions in Section V.

Algorithm 1 Outline of the tailored SMS-EMOA algorithm

Require: IT, μ, IM, MT, ST, FS, MC

if $IT = 1$ **then**
  Generate population P of μ parents as in [5]
  for $i \in \{1, \ldots, IM\}$ **do**
    Mutate all individuals in P as in [5]
  end for
else ▷ $IT = 2$
  Generate population P of μ parents as in Section III-B
end if
while Stop condition not met **do**
  $Q \leftarrow$ A uniform random individual from P
  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 Q with n_steps as in Section III-A
  else
    Apply polynomial mutation to each continuous variable in Q with probability MC
  end if
  $P \leftarrow$ Select μ individuals from $P \cup Q$
end while

IV. Landscape Analysis

A. Setup

The landscapes of both objectives are analysed by randomly sampling solutions generated by the newly introduced initialisation operator (Section III-B). 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 extended mutation operator introduced in Section III-A.

A problem instance is 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 is generated with the new initialisation operator and for this parent a single offspring is generated using the improved mutation operator. This is executed 1000 times for each instance and step size combination, totalling 5000 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 offspring and parent 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.

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).

B. Results

In Figure 6 all parent individuals are plotted for the different problem instances. For the 3331 instance a clear Pareto front approximation is visible, which is very similar to the one found in [5]. Additionally, many other parts also show smooth curves, suggesting that these areas are well covered by random sampling. The largest concentration of points is seen reasonably close to the Pareto front. This suggests that random sampling can attain a decent solution for this instance.

Both the 3333 and 3335 problem instances 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 good solutions is very feasible, but finding exact optima is still challenging.

Next the absolute change in both objective values is shown for each instance and step size combination in Tables I (compliance) and II (surface area). 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 (Table I) 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.

Absolute change to the surface area (Table II) shows a somewhat clearer trend across the board, although the 3335 instance seems to stagnate later. Figure 7 visualises how objective values are distributed for both objectives for the 3333 instance. The other instances produced largely similar visuals.

For both objectives zero changes also occur. 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 design structure. 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 a change is seen to the compliance, but not to the surface area.

TABLE I
STATISTICS FOR ABSOLUTE CHANGE IN COMPLIANCE PER INSTANCE

S	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 II
STATISTICS FOR ABSOLUTE CHANGE IN SURFACE AREA PER INSTANCE

S	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

V. PARAMETER TUNING

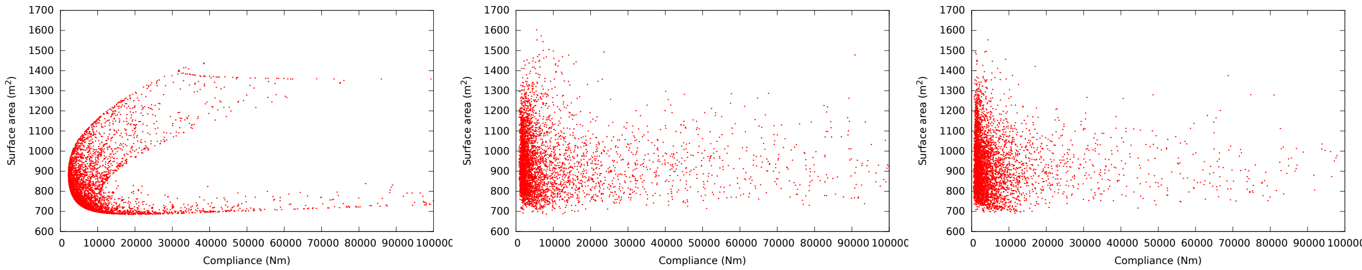
This section first briefly reintroduces the algorithmic setup from [5] which will be used here as well. Next, the employed experimental setup for algorithm tuning is introduced. Finally, the results of those experiments are discussed.

A. Optimisation Setup

SMS-EMOA as used here is the standard SMS-EMOA from [17], with the only changes being the volume rescaling and the evaluation as described in the following paragraphs, details are available in [5], [17]. The tailored SMS-EMOA that is used is an altered version of SMS-EMOA (as just described), tailored for this specific problem. Changes are as follows: (1) the initialisation operator is replaced by the one from Section III-B, and (2) offspring is generated with the mutation operator as described in Section III-A. All other settings except for the evaluation budget and tunable parameters detailed later are exactly as in [5].

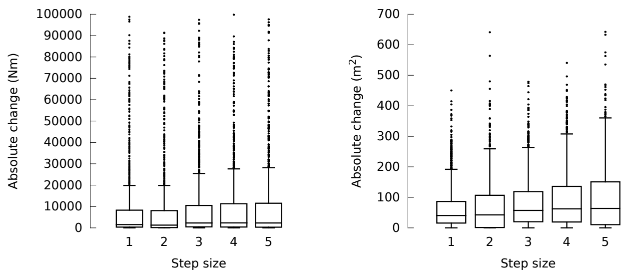
For both the standard and tailored versions of SMS-EMOA the considered optimisation problem is as follows. Objective functions are (1) the compliance in Nm, and (2) the surface area in m^2, both to be minimised. Binary and continuous variables are considered as described in Section II-B. The binary variables are subject to five topological constraints as mentioned in Section II-B and described mathematically in [4]. To comply to the volume constraint 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) Instance 3331 (b) Instance 3333 (c) Instance 3335

Fig. 6. Landscape for different problem instances, each with 5000 randomly initialised samples, limited to points with compliance below 100,000 Nm.



(a) Compliance (b) Surface area

Fig. 7. Absolute change to the compliance objective with 1000 samples for a single mutation of 1 through 5 steps for the 3333 problem instance.

a penalty of $999999998 + CV$ is returned as the objective value for both objectives, for a number of constraint violations CV. If the objective value is 999999999 or higher 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.

B. Tuning Setup

To tune the parameters of the standard and tailored SMS-EMOA for building spatial design, two approaches are compared: The irace package [13] and the Mixed Integer Evolution Strategy (MIES) [16]. The objective is the maximisation of the hypervolume (HV), with reference point $(100000, 1500)$.

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 discarded. These ranges are based on investigation of the result data from [5]. This ensures only extreme outliers are discarded and the data can be visually analysed.

For the standard SMS-EMOA three tunable parameters are considered as follows:

- Population size $\mu \in \{10, \ldots, 100\} \subset \mathbb{Z}$
- Crossover probability $CP \in [0.0, 1.0] \subset \mathbb{R}$
- Mutation probability $MP \in [0.005, 0.100] \subset \mathbb{R}$

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 metric 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 1 are tuned (parameters without type are categorical):

- Population size $\mu \in \{1, \ldots, 100\} \subset \mathbb{Z}$
- Mutation type probability $MT \in [0.0, 1.0] \subset \mathbb{R}$
- Step size technique $ST \in \{1, 2\}$
- Fixed number of steps $FS \in \{1, \ldots, 5\} \subset \mathbb{Z}$
- Cont. mutation probability $MC \in [0.0, 1.0] \subset \mathbb{R}$
- Initialisation technique $IT \in \{1, 2\}$
- Initialisation mutations $IM \in \{1, \ldots, 100\} \subset \mathbb{Z}$

From the landscape analysis in Section IV it can be concluded 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 [5]. The irace package enforced a minimum of 180 algorithm executions for the given settings and tunable parameters. The method proposed in [5] converges to a near stable state in about 300 function evaluations, so this evaluation budget is used here for tuning.

Both irace [13] (version 2.1.1662) and MIES [16] are used primarily with default settings as described in their respective papers, deviations from those settings are explicitly stated. As a result performance of both approaches is likely not optimal.

MIES is used with multiple step size mode for both real valued and integer parameters, and single step size mode for categorical parameters. In [16] 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. For $n_d = 2$ this means the step size is constant at $\frac{1}{2}$. Since two categorical variables are considered for this experiment categorical step sizes are instead bounded in $[\frac{1}{3}, \frac{1}{2}]$.

In [20] a (μ, κ, λ) strategy is shown to converge faster than a (μ, λ) strategy. Due to the small number of available evaluations, faster convergence is desirable here. Instead of the

TABLE III
PARAMETER CONFIGURATIONS FOR THE STANDARD SMS-EMOA: UNTUNED AND FROM THREE REPETITIONS OF IRACE AND MIES

ID	μ	CP	MP	MHV
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 IV
PARAMETER CONFIGURATIONS FOR THE TAILORED SMS-EMOA: UNTUNED AND FROM THREE REPETITIONS OF IRACE AND MIES

ID	μ	MT	ST	FS	MC	IT	IM	MHV
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

$(3, 5, 10)$ strategy from [20], 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.

C. Results

Both tuning approaches used around two weeks on a single CPU core, per repetition. The parameter configurations found by irace and MIES, as well as untuned configurations are found in Table III (standard SMS-EMOA) and Table IV (tailored SMS-EMOA). The untuned configurations are set as in [5] whenever possible, only MC is adjusted because continuous mutation changed. For irace the best found parameter configurations are shown, while for MIES the configurations with the largest hypervolume coverage in the final population were selected. For each parameter the mean and standard deviations taken over the set of all final populations (from every repetition) are also reported, for both irace and MIES.

For the standard SMS-EMOA higher crossover (CP) and mutation (MP) probabilities were found for tuned configurations compared to untuned. With the tailored version using a smaller population size (μ), 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. 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.

In order to compare the considered configurations and their corresponding algorithms each of them is evaluated on the considered optimisation problem. Each configuration is run for thirteen repetitions with a budget of 1000 evaluations, all other settings are the same as before. The mean hypervolume (MHV) values resulting from this are shown in the tables.

Median attainment curves [21] of the resulting data are used to compare the different configurations and tuning methods. From Figure 8 it is clear that of the untuned configurations the tailored version (UT) dominates a larger area than the standard version (US). Most configurations show very similar

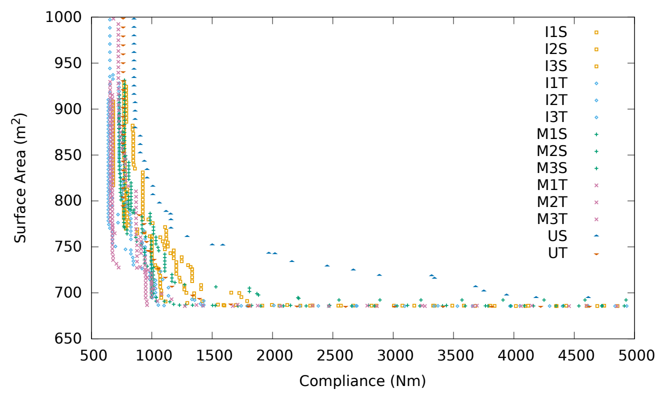


Fig. 8. Comparison of median attainment curves over thirteen repetitions for untuned configurations and tuned configurations by irace and MIES, all for both the standard and tailored versions of SMS-EMOA with 1000 evaluations.

performances in terms of surface area minimisation. The tailored SMS-EMOA configurations tuned by irace (IxT) and MIES (IxM) outperform their standard variants (IxS and IxM respectively), both in compliance minimisation and in the knee point area. Tailored SMS-EMOA achieves compliances as much as 500 Nm lower compared to the standard SMS-EMOA. The two tuning methods show similar performances in tuning for the standard and tailored versions of SMS-EMOA.

In summary it is evident that even with a relatively limited tuning budget both tuning approaches are able to improve over untuned configurations. These are promising results, since based on the results in [5] it can be expected that differences between the standard and tailored methods will only be larger for more difficult and realistic problem instances.

VI. CONCLUSION

The bi-objective problem of building spatial design for structural and energy performance is considered. New operators were introduced to eliminate bias in both the initialisation and mutation processes. With the use of landscape analysis the distribution of solutions over the objective landscape and the complexity of different problem instances were investigated. Additionally, the search behaviour of the mutation operator was evaluated. Furthermore, this work explores parameter tuning of standard and tailored (using the new operators) versions of SMS-EMOA with the irace [13] package and the Mixed Integer Evolution Strategy [16].

Despite tuning with a relatively small budget algorithm performance is improved compared to untuned configurations. Although tuning was successful, it should be noted that a relatively simple problem instance was considered. For larger real world problem instances parameter tuning approaches that are effective with fewer function evaluations have to be introduced. After tuning, the tailored SMS-EMOA is shown to improve over the standard SMS-EMOA.

Future work should focus on evaluating how the considered algorithms scale towards larger problem instances with five, ten, or even more rooms. This is interesting in regard to the optimisation algorithms, where the landscape is likely to increase in complexity. Moreover, it is interesting with regard to tuning, which may have to be done with an even smaller evaluation budget. In addition to this, the problem complexity may increase when surface area is substituted by heating and cooling simulations to measure energy performance.

Another direction for future work is the exploration of parameter tuning approaches that include meta-modelling such as SPOT [14] and SMAC [12]

ACKNOWLEDGMENT

The authors gratefully acknowledge the financing of this project by the Dutch STW (now part of NWO TTW) via project 13596 (Excellent Buildings via Forefront MDO, Lowest Energy Use, Optimal Spatial and Structural Performance).

Moreover, the authors would like to express their gratitude to Leslie Pérez Cáceres and Manuel López-Ibáñez for their helpful advice on setting up and using the irace package.

REFERENCES

[1] M. Wetter and E. Polak, "Building design optimization using a convergent pattern search algorithm with adaptive precision simulations," *Energy and Buildings*, vol. 37, no. 6, pp. 603 – 612, 2005.

[2] H. Hofmeyer and J. M. Davila Delgado, "Coevolutionary and genetic algorithm based building spatial and structural design," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 29, no. 4, p. 351370, Nov 2015.

[3] C. J. Hopfe, M. T. M. Emmerich, R. Marijt, and J. Hensen, "Robust multi-criteria design optimisation in building design," *Proceedings of the First Building Simulation and Optimization Conference, Loughborough, UK*, pp. 118–125, 2012.

[4] K. van der Blom, S. Boonstra, H. Hofmeyer, and M. T. M. Emmerich, "A super-structure based optimisation approach for building spatial designs," in *VII European Congress on Computational Methods in Applied Sciences and Engineering – ECCOMAS VII*, M. Papadrakakis, V. Papadopoulos, G. Stefanou, and V. Plevris, Eds., vol. 2. Athens, Greece: National Technical University of Athens, 2016, pp. 3409–3422.

[5] ——, "Multicriteria building spatial design with mixed integer evolutionary algorithms," in *Parallel Problem Solving from Nature – PPSN XIV*, ser. Lecture Notes in Computer Science, J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter, Eds., vol. 9921. Cham: Springer International Publishing, 2016, pp. 453–462.

[6] S. Boonstra, K. van der Blom, H. Hofmeyer, R. Amor, and M. T. M. Emmerich, "Super-structure and super-structure free design search space representations for a building spatial design in multi-disciplinary building optimisation," in *Electronic proceedings of the 23rd International Workshop of the European Group for Intelligent Computing in Engineering*. Jagiellonian University ZPGK, 2016, pp. 1–10.

[7] E. Pitzer and M. Affenzeller, "A comprehensive survey on fitness landscape analysis," in *Recent Advances in Intelligent Engineering Systems*, J. Fodor, R. Klempous, and C. P. Suárez Araujo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 161–191.

[8] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 829–836.

[9] H. Rosé, W. Ebeling, and T. Asselmeyer, "The density of states — a measure of the difficulty of optimisation problems," in *Parallel Problem Solving from Nature — PPSN IV*, ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., vol. 1141. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 208–217.

[10] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms." in *ICGA*, vol. 95, 1995, pp. 184–192.

[11] R. Li, M. T. M. Emmerich, J. Eggermont, E. G. P. Bovenkamp, T. Bäck, J. Dijkstra, and J. H. C. Reiber, "Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 2764–2771.

[12] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization*, C. A. Coello Coello, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 507–523.

[13] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[14] T. Bartz-Beielstein, C. W. G. Lasarczyk, and M. Preuß, "Sequential parameter optimization," in *2005 IEEE congress on evolutionary computation*, vol. 1. IEEE, 2005, pp. 773–780.

[15] G. Sand, J. Till, T. Tometzki, M. Urselmann, S. Engell, and M. Emmerich, "Engineered versus standard evolutionary algorithms: A case study in batch scheduling with recourse," *Computers & Chemical Engineering*, vol. 32, no. 11, pp. 2706 – 2722, 2008, enterprise-Wide Optimization.

[16] R. Li, M. T. M. Emmerich, J. Eggermont, T. Bäck, M. Schütz, J. Dijkstra, and J. H. C. Reiber, "Mixed integer evolution strategies for parameter optimization," *Evolutionary computation*, vol. 21, no. 1, pp. 29–64, 2013.

[17] M. Emmerich, N. Beume, and B. Naujoks, "An emo algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 62–76.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[19] D. Wiesmann, "From syntactical to semantical mutation operators for structure optimization," in *Parallel Problem Solving from Nature — PPSN VII*, ser. Lecture Notes in Computer Science, J. J. M. Guervós, P. Adamidis, H.-G. Beyer, H.-P. Schwefel, and J.-L. Fernández-Villacañas, Eds., vol. 2439. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 234–243.

[20] M. Emmerich, M. Grötzner, B. Groß, and M. Schütz, "Mixed-integer evolution strategy for chemical plant optimization with simulators," in *Evolutionary Design and Manufacture: Selected Papers from ACDM '00*, I. C. Parmee, Ed. London: Springer London, 2000, pp. 55–67.

[21] C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete, "Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function," in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 250–264.