Evaluating Memetic Building Spatial Design Optimisation Using Hypervolume Indicator Gradient Ascent

Koen van der Blom[1](✉), Sjonnie Boonstra[2], Hao Wang[1], Hèrm Hofmeyer[2], and Michael T. M. Emmerich[1]

[1] LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
{k.van.der.blom,h.wang,m.t.m.emmerich}@liacs.leidenuniv.nl
[2] Eindhoven University of Technology, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
{s.boonstra,h.hofmeyer}@tue.nl

Abstract. In traditional, single objective, optimisation local optima may be found by gradient search. With the recently introduced hypervolume indicator (HVI) gradient search, this is now also possible for multi-objective optimisation, by steering the whole Pareto front approximation (PFA) in the direction of maximal improvement. However, so far it has only been evaluated on simple test problems. In this work the HVI gradient is used for the real world problem of building spatial design, where the shape and layout of a building are optimised. This real world problem comes with a number of constraints that may hamper the effectiveness of the HVI gradient. Specifically, box constraints, and an equality constraint which is satisfied by rescaling. Moreover, like with regular gradient search, the HVI gradient may overstep an optimum. Therefore, step size control is also investigated. Since the building spatial designs are encoded in mixed-integer form, the use of gradient search alone is not sufficient. To navigate both discrete and continuous space, an evolutionary multi-objective algorithm (EMOA) and the HVI gradient are used in hybrid, forming a so-called memetic algorithm. Finally, the effectiveness of the memetic algorithm using the HVI gradient is evaluated empirically, by comparing it to an EMOA without a local search method. It is found that the HVI gradient method is effective in improving the PFA for this real world problem. However, due to the many discrete subspaces, the EMOA is able to find better solutions than the memetic approach, albeit only marginally.

Keywords: Multi-objective optimisation · Memetic algorithm
Hypervolume indicator gradient · Building spatial design

1 Introduction

Many engineering experts from different disciplines cooperate in the process of building design. Whenever these experts change the design their limited

L. Trujillo et al. (Eds.): NEO 2017, SCI 785, pp. 62–86, 2019.
https://doi.org/10.1007/978-3-319-96104-0_3

knowledge about the other disciplines may lead to a negative impact on the quality of the design with respect to these other disciplines. Automated multi-objective building spatial design optimisation can play a part in alleviating this issue. In [4,6] the authors introduced representations for building spatial design for the optimisation of structural and thermal performance. Following this in [2,3] the authors developed problem specific operators for evolutionary algorithms to ensure that only feasible designs are considered. Furthermore, parameter tuning is employed to maximise the performance of the algorithm [2]. However, finding exact optima remains a challenge.

The ultimate goal in optimisation is finding the global optimum. Convergence to the global optimum is quick for convex and continuous optimisation problems when exact methods like gradient search are used. In complex functions however, such exact methods may get stuck in local optima. Exploring multiple local optima requires different methods, such as evolutionary algorithms for example. However, heuristic methods like evolutionary algorithms may be slower to lock in on an exact (local) optimum. As such, a combination of these methods could provide advantages over either of the individual methods. Hybrids of such heuristic and exact methods are called memetic algorithms [21]. Here, a combination of evolutionary search and gradient search is proposed.

Gradient search, like other exact optimisation methods, has traditionally been used for the single-objective case. In fact, until recently gradients were only defined for single points. In the generalisation of gradient methods for multi-objective optimisation the challenge arises that typically a set of points is considered together, the so-called Pareto front. One way to measure the quality of a Pareto front approximation is the hypervolume indicator (HVI). In [11] the authors describe the HVI gradient, which allows a set of points to be moved towards the Pareto front, and to be distributed well across the Pareto front. Although the HVI gradient has been tested on benchmark functions [12], and improvements to its computation [11], as well as to the navigation of dominated points [26,27] have been proposed, it was never tested on real world problems.

Other gradient approaches for multi-objective optimisation exist as well. The key difference is the use of set gradients (in case of the HVI gradient), as opposed to single point gradients [13,22] and gradients that are used for the computation of bounds on subspaces [8]. Other ideas to use gradients in memetic search have been proposed in the literature, such as continuation methods that locally extend Pareto fronts by steps along tangent planes [20,23]. Moreover, directed search has been proposed, which steers points in a desired direction, either across or towards the PF. Such methods also use gradients in order to construct these directions in the decision space [24]. The HVI gradient is favoured here since it updates the Pareto front approximation as a whole and local optimality has been verified [12].

In this work the HVI gradient is applied to the real world problem of building spatial design for the first time. The combination of an evolutionary multi-objective algorithm (EMOA) and the HVI gradient results in a memetic multi-objective (MEMO) algorithm. Specifically, here the $\mathscr{S}$-metric (also known as HVI) selection EMOA (SMS-EMOA [10]), with specialised operators [2] for the building spatial design optimisation problem is used. For the HVI gradient component the hypervolume indicator gradient ascent multi-objective optimisation (HIGA-MO) approach from [26] is employed.

To summarise, key points of this work are as follows. The HVI gradient is used in a memetic setting for the first time. Since local optimality of the HVI gradient has been verified [12] it is an excellent candidate to explore the potential of local search for the considered problem. In addition, it may provide guarantees with regard to local convergence in the continuous subspace. Furthermore, the HVI gradient method is also subjected to constraints and a mixed-integer search space for the first time. These new challenges should provide insight into the effectiveness of the HVI gradient in more complex search spaces.

The remainder of this work is structured as follows. Section 2 introduces the problem of building spatial design and its representation. Next, Sect. 3 reviews the basics of multi-objective optimisation, and briefly describes the principles of the hypervolume indicator. In Sect. 4 the hypervolume indicator (HVI) gradient and its use in algorithms are discussed. Section 5 starts by describing a local search algorithm (based on the HVI gradient) and a global search algorithm (based on SMS-EMOA) for the building spatial design problem, and then considers how they can be combined into a memetic algorithm. Following this, Sect. 6 describes the experimental setup for the evaluation of the different algorithms. This is naturally followed by Sect. 7, where the results are analysed. Finally, Sect. 8 discusses the work as a whole, and summarises the resulting conclusions.

2 Problem

This section briefly introduces the problem of building spatial design optimisation in its first subsection. Following that the supercube representation, which encodes the problem in mixed-integer form, is introduced.

2.1 Building Spatial Design

Early stage building design consists of design decisions that have a profound influence on the many disciplines involved, and as such on their quality. By its very nature the shape and the layout of the building, i.e. the building spatial design, are normally decided early in the design process. The building spatial design includes not only the exterior building shape, but also the partitioning of spaces within the building.

Since the building spatial design influences numerous qualities of a building, it is also something worth optimising. Here, the goal is to optimise the building spatial design for two aspects. First, optimal structural performance is desirable, as it will minimise material use. For this the stiffness of a building's structural design has to be maximised. Here compliance is used as a fitting measure for a structure's stiffness. Compliance has an inverse relationship with the stiffness, and therefore has to be minimised. Second, the energy efficiency, which is here measured by the required heating and cooling energy of a building during operational hours. Efficient energy use may reduce both upkeep costs and the environmental impact. This too, has to be minimised. Both of these objectives are evaluated through simulation. Details of the simulation process are discussed later in Subsect. 6.1.

2.2 Supercube Representation

Algorithmic optimisation of building spatial designs requires a numerical representation. Here the representation from [4,6] is used, and will be briefly reintroduced on a conceptual level in the following.

In the basis the representation consists of a supercube (Fig. 1, left), which encompasses the different spaces that define the building. This supercube is a superstructure, i.e. the relevant subset of the design space. It should be noted that *space* here refers to a section of the building, somewhat similar to a room. While, *design space* refers to the collection of candidate solutions in the used representation.

The supercube is partitioned in cells by width, depth, and height indices ($i \in \{1, \ldots, N_w\}, j \in \{1, \ldots, N_d\}, k \in \{1, \ldots, N_h\}$ respectively). Each cell may then be set to the active or inactive state for a specific space $\ell \in \{1, \ldots, N_{spaces}\}$ by its corresponding binary variable $b^{\ell}_{i,j,k}$. A space may consist of multiple cells, allowing spaces to be elongated in one dimension such that they neighbour spaces that are shortened in that same dimension. Depending on which of the cells are active, and which are not, buildings with different shapes may be carved out of the supercube. Moreover, the different rows, columns, and beams in the supercube may be stretched and shrunk by their corresponding continuous variables $w_{i\in\{1,\ldots,N_w\}}, d_{j\in\{1,\ldots,N_d\}}, h_{k\in\{1,\ldots,N_h\}}$, which results in a more diverse set of possible shapes. An example of a spatial design that could be composed from the supercube is included on the right of Fig. 1.

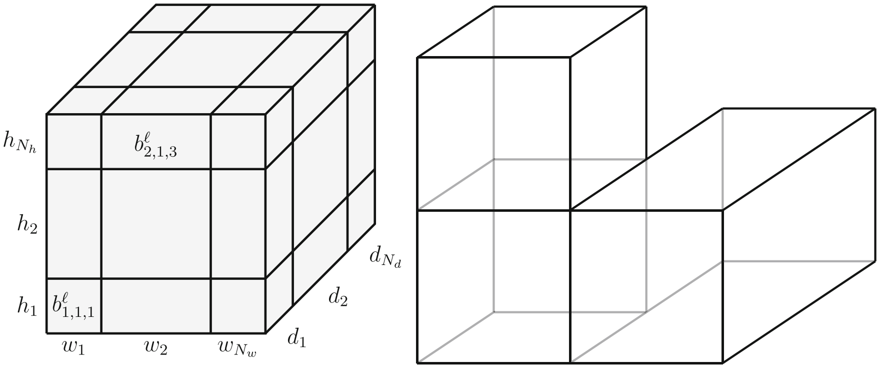

Fig. 1. Schematic of the supercube representation (left); and an example spatial design (right)

Naturally, this representation may lead to infeasible designs when it is not restricted by constraints, for example floating spaces could occur. On the binary variables the following constraints are defined: (C1) Each cell must have support below it, in the form of either another cell, or the ground. (C2) Each space must have at least one active cell assigned to it. (C3) Each cell is active for no more than one space. (C4 + C5) Each space forms a cuboid (3D rectangle) out of the active cells assigned to it (mathematically this is described in two parts). For the interested reader these constraints are described in mixed-integer nonlinear programming form in [4,6].

Further, a number of constraints are considered on the continuous subspace as well. All continuous variables are restricted between a lower bound, and an upper bound with box-constraints. In addition, a constant volume V_0 is maintained during optimisation to allow a fair comparison between different spatial designs. This is achieved by iteratively scaling the design according to the scaling functions given in [3], until the volume is within a cubic millimetre of the desired value.

3 Multi-objective Optimisation and the Hypervolume Indicator

Loosely speaking, in multi-objective optimisation problems (MOPs) the goal is to search for the candidate decision vector $\mathbf{x} = [x_1, \ldots, x_d]$ that optimises a tuple of objective functions $\mathbf{y} = \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]$, simultaneously. Without loss of generality, it is assumed that each objective function $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ has to be *minimised*.

Given that the different functions will rarely have common optimal values for $\mathbf{x}$, the outcome of a MOP is usually a Pareto front of solutions, with differing values for $\mathbf{y}$. Particularly in *continuous* MOPs, the Pareto front's efficient set [9] is approximated by a finite set (of size μ): $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mu\} \subset \mathbb{R}^d$. The corresponding Pareto front approximation $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_\mu\} \subset \mathbb{R}^m$ is the image of X under $\mathbf{f}$, namely $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i),\ i = 1, 2, \ldots, \mu$.

The quality of a Pareto front, or an approximation thereof, can be measured with the hypervolume indicator (also known as the $\mathscr{S}$-metric) [28,29]. This metric measures the volume (or area in bi-objective cases) that is dominated by a set of points in the objective space, with respect to a reference point $\mathbf{r} \in \mathbb{R}^m$. For notational brevity we denote the hypervolume indicator as H for mathematical use, while otherwise using the more expressive abbreviation HVI. As such the hypervolume indicator for Y can be found by $H(Y)$. Given this quality measure, it is also possible to compare different Pareto front approximations (PFAs) to each other. However, it should be noted that this measure is entirely dependent on how the reference point is chosen. In other words, the ranking of PFAs is determined in part by the value used for the reference point.

4 Hypervolume Indicator Gradient Ascent Multi-objective Optimisation

In this section, first the hypervolume indicator (HVI) gradient is introduced in its general form. This is followed by subsections for normalisation of the HVI gradient, step size adaptation for the HVI gradient, and finally update rules for the considered points.

4.1 Hypervolume Indicator Gradient

To give a proper derivation of the hypervolume indicator (HVI) gradient over approximation sets, it is proposed to use the so-called *set-oriented* approach: By concatenating the vectors in X the μd-vector $\mathbf{X} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_\mu^\top]^\top \in \mathbb{R}^{\mu d}$ is defined. Note that the restriction to $\mathbb{R}$ here is intentional, the gradients will be taken exclusively for the real subspace of the considered problem, while the discrete subspace remains constant. Likewise a μm-vector $\mathbf{Y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_\mu^\top]^\top \in \mathbb{R}^{\mu m}$ can be defined for the objective values. Furthermore, the following mapping can be introduced: $\mathbf{F} : \mathbb{R}^{\mu d} \to \mathbb{R}^{\mu m}, \mathbf{X} \mapsto \mathbf{Y}$. Using the mapping $\mathbf{F}$, the HVI can be related to the decision space: $\mathscr{H}_\mathbf{F}(\mathbf{X}) := H(\mathbf{F}(\mathbf{X})) = H(\mathbf{Y})$. Note that this is simply the definition of a more concise symbol for the same concept.

The full HVI gradient can then be expressed as in Eq. 1, and represents the direction of steepest improvement of the HVI for the entire Pareto Front Approximation (PFA).

$$\nabla \mathscr{H}_\mathbf{F}(\mathbf{X}) = \left[\frac{\partial \mathscr{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{x}^{(1)}}^\top, \ldots, \frac{\partial \mathscr{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{x}^{(\mu)}}^\top \right]^\top . \tag{1}$$

Subsequently, Eq. 2 defines subgradients for each point of the PFA. Note that although subgradients are computed for individual points, their combination is not merely the direction of maximal improvement for each point, but for the whole set.

$$\frac{\partial \mathscr{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} = \left[\frac{\partial \mathscr{H}_\mathbf{F}}{\partial x_1^{(i)}}, \ldots, \frac{\partial \mathscr{H}_\mathbf{F}}{\partial x_d^{(i)}} \right]^\top . \tag{2}$$

Each subgradient can then be computed as:

$$\frac{\partial \mathscr{H}_\mathbf{F}}{\partial x_j^{(i)}}(\mathbf{X}) = \sum_{k=1}^{m} \frac{\partial H}{\partial y_k^{(i)}}(\mathbf{Y}) \times \frac{\partial f_k(\mathbf{x}^{(i)})}{\partial x_j^{(i)}} . \tag{3}$$

For the $m = 2$ case, when the indices are given by the ascending order of the first objective f_1, the first term of the summation can be expressed as follows:

$$\frac{\partial H}{\partial y_1^{(i)}} = y_2^{(i)} - y_2^{(i-1)} , \qquad \frac{\partial H}{\partial y_2^{(i)}} = y_1^{(i)} - y_1^{(i+1)} .$$

Or, for a greater number of objective functions:

$$\forall_{k=1,\dots,m}: \; \frac{\partial H}{\partial y_k^{(i)}} = \prod_{\ell \neq k} y_\ell^{(i)} - y_\ell^{(i+)} \; , \quad k = 1, 2, \dots, m \; ,$$

where $i+$ indicates the next greater value in objective ℓ, or the reference point if no such value exists. For maximisation this would be $i-$, the next smaller value. Note that strictly greater (smaller) values are taken, since any points that are equivalent in some objective, should also move the same in that objective.

Applying the HVI gradient is only possible if the gradient can be computed. Since for many problems, like the one considered in this work, the analytical expression of derivatives is not available, numerical computation of the gradient is considered as alternative. As such, the second term of the summation in Eq. 3 may be computed numerically according to the finite difference method. For a small number h, the approximation reads,

$$\frac{\partial f_k(\mathbf{x}^{(i)})}{\partial x_j^{(i)}} = \frac{f_k(\mathbf{x}^{(i)} + \mathbf{e}_j h) - f_k(\mathbf{x}^{(i)})}{h} \; .$$

Note that $\mathbf{e}_j$ is the j-th standard basis in $\mathbb{R}^d$. Here $h = 0.01$ is chosen, which equates to a change of 10 mm (millimetre) in the building spatial design. This value was chosen such that it both represents a meaningful change to the design, and it is small enough such that it provides a sufficient accuracy to approximate the gradient. Moreover, it was ensured that the employed simulator for the evaluation of a solution's quality was sufficiently sensitive. In other words, that it gave different objective values for changes of this size.

4.2 Normalisation

A limitation of the HVI gradient method is the so-called *creepiness* behaviour, as analysed in [25]. Creepiness refers to how the points move towards the Pareto front in a suboptimal way. When the differences between subgradients are large, the steps taken by the points are largely unbalanced, leading to a non-uniform convergence to the Pareto front. To avoid creepiness, the subgradients are normalised according to Eq. 4 before using them to update the original points.

$$G_{norm} = \left(\frac{\frac{\partial \mathscr{H}_\mathbf{F}}{\partial \mathbf{x}^{(1)}}}{\left\| \frac{\partial \mathscr{H}_\mathbf{F}}{\partial \mathbf{x}^{(1)}} \right\|}, \dots, \frac{\frac{\partial \mathscr{H}_\mathbf{F}}{\partial \mathbf{x}^{(\mu)}}}{\left\| \frac{\partial \mathscr{H}_\mathbf{F}}{\partial \mathbf{x}^{(\mu)}} \right\|} \right), \quad \text{where} \quad \left\| \frac{\partial \mathscr{H}_\mathbf{F}}{\partial \mathbf{x}^{(i)}} \right\| = \sqrt{\sum_{j=1}^{d} \left(\frac{\partial \mathscr{H}_\mathbf{F}}{\partial x_j^{(i)}} \right)^2} \; . \quad (4)$$

4.3 Step Size Adaptation

In [27], the authors mentioned that the normalised subgradients may lead to oscillatory (even divergent) behaviour. To mitigate this effect, the step-size adaptation that has been proposed in [26] is adopted as follows. In Eq. 5 $\langle \cdot, \cdot \rangle$ stands for the dot product in $\mathbb{R}^d$. For each search point, I is calculated by the inner product of the normalised HVI subgradients in two consecutive iterations. This is used to find whether the step size should be increased, for positive values, or decreased otherwise. The subscript t on the subgradient is used to indicate the iteration.

$$I_t^{(i)} = \left\langle \left(\frac{\partial \mathscr{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}}\right)_{t-1}, \left(\frac{\partial \mathscr{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}}\right)_{t} \right\rangle , \quad i = 1, \ldots, \mu , \quad t = 1, 2, \ldots . \tag{5}$$

Since the inner product may vary largely between generations, the stabilisation is achieved by taking the cumulative p of this value, over t generations with exponential decay (Eq. 6). The accumulation coefficient $0 < c < 1$ controls how much new information will be incorporated.

$$p_t^{(i)} \leftarrow (1 - c) \times p_{t-1}^{(i)} + c \times I_t^{(i)} , \qquad i = 1, \ldots, \mu , \qquad t = 1, 2, \ldots . \tag{6}$$

Given the cumulative inner product the value of the step size $\sigma_{t+1}^{(i)}$ for the next time step can be found according to Eq. 7. The parameter α controls the rate of change for updates to the step size.

$$\sigma_{t+1}^{(i)} = \begin{cases} \sigma_t^{(i)} \times \alpha & \text{if } p_t^{(i)} < 0, \\ \sigma_t^{(i)} & \text{if } p_t^{(i)} = 0, \\ \sigma_t^{(i)}/\alpha & \text{if } p_t^{(i)} > 0. \end{cases} \qquad 0 < \alpha < 1 . \tag{7}$$

The parameters $c = 0.7$ and $\alpha = 0.8$ are used here as they were used in [26], but should ideally be tuned for the specific problem. Both $\left(\frac{\partial \mathscr{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}}\right)_{t-1}$ and $p_{t-1}^{(i)}$ are initialised to zeros, such that the starting position is neutral.

4.4 Update

Which points are updated, and when, has a large influence on how the Pareto front approximation (PFA) changes. The obvious choice is to move the points on the current PFA. However, what to do with dominated points is not immediately obvious. In [18] the authors suggested to move dominated points according to the there defined direction of Lara.

In the bi-objective case, this direction is defined as the sum of normalised gradients from two objective functions. It guarantees that dominated decision points move into the dominance cone [27]. However, such a method only considers the movement of single points, instead of a set of search points, and does not generalise naturally to higher dimensions.

Alternatively, in [26], the authors suggested to move all points, including the dominated points, according to the HVI gradient. In order to do this, the whole population is partitioned by the so-called nondominated sorting procedure [7], resulting in multiple subsets (fronts) of nondominated solutions. Subsequently, the HVI gradient is well-defined on each front by ignoring other fronts that dominate it. Since both approaches require the same number of evaluations the exact method from [26] is used here, as shown in Eq. 8. Given that the numerical computation of the gradients requires a large number of evaluations (equal to the number of continuous decision variables) investigating alternatives that use fewer, or no, evaluations could be a promising future direction.

In this work, the step size parameter σ is initialised to $0.0025 \times (ub_r - lb_r)$ according to practical usage of the algorithm. Here ub_r and lb_r refer to the upper and lower bounds of continuous decision variable r respectively. The gradient-based update is as follows,

$$\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \sigma^{(i)} \frac{\partial \mathscr{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}_j^{(i)}} \,, \qquad i = 1, \ldots, \mu \,, \qquad j = 1, \ldots, d \,. \tag{8}$$

5 Algorithms

Each subsection here describes one of the considered algorithms. First the HIGA-MO-SC approach, as adapted from the standard HIGA-MO [26] algorithm. Second the SMS-EMOA-SC algorithm, previously introduced in [2]. Finally, a combination of the two in the form of a memetic algorithm, MEMO-SC, is considered.

5.1 HIGA-MO-SC

In full, the HVI gradient method adapted to the context of building spatial design is described in Algorithm 1. There HIGA-MO [26] is adjusted to work with the supercube representation and forms the HIGA-MO-SC algorithm. An initial population is generated with the specialised initialisation procedure introduced in [2]. Following this, the population is sorted according to nondominated sorting [7]. Each front is then updated separately as follows. First, the HVI gradient is computed for the continuous subspace as previously described in Sect. 4.1. Second, the HVI gradients are normalised using Eq. 4. Third, step sizes are updated by employing Eqs. 5, 6 and 7. Finally, the old points are replaced with new points generated according to the normalised HVI gradients and the updated step sizes.

Note that HIGA-MO-SC as used here differs from HIGA-MO from [26] on two points. First, the step sizes are updated before moving points, rather than after. As a result the gradient information of the current iteration is immediately taken into account. Second, and most significantly, here gradients are numerically approximated. As a result they require a number of function evaluations equal to the number of continuous decision variables.

Discussion on whether to call this algorithm memetic or not is possible, since the HVI gradient operates on a population level. Here, it is important to note that there are two equivalent views on the HVI gradient. One view is, that the HVI gradient consists of the gradients of the hypervolume contributions. In that sense, a point can locally improve by increasing its hypervolume contribution. If this is performed simultaneously for all points, the second view, the effect is equivalent to following the set gradient of the HVI. A detailed discussion is provided in [11].

Algorithm 1. HIGA-MO-SC

```
input: $\mu, \lambda, \sigma, c, \alpha, h$
output: PFA based on all evaluated solutions
Initialise population $X$ of $\mu$ parents as in [2]
while Stop condition not met do
    while $X \neq \varnothing$ do
        $X_{nds} \leftarrow \mathrm{NDS}_1(X)$ ▷ Where $\mathrm{NDS}_1$ returns the first front after nondominated sorting
        $X \leftarrow X \setminus X_{nds}$
        Compute the HVI gradient for $X_{nds}$ according to Sect. 4.1
        Normalise HVI gradient of $X_{nds}$ according to Eq. 4
        Update step size of $X_{nds}$ according to Eqs. 5, 6 and 7
        Move $X_{nds}$ according to Eq. 8
        $X' \leftarrow X' \cup X_{nds}$
    end while
    $X \leftarrow X'$
end while
```

5.2 SMS-EMOA-SC

The SMS-EMOA SuperCube (SMS-EMOA-SC) algorithm was previously developed by the authors in [2,3] to navigate the heavily constrained landscape represented by the supercube. In [3] it was shown by the authors that specialised operators become essential to the building spatial design problem as soon as building designs beyond the trivially small cases are considered. Standard algorithms such as NSGA-II [7] and SMS-EMOA [10] simply do not handle the constraints well, and waste a large amount of time on infeasible solutions.

Through the use of specialised initialisation and mutation operators SMS-EMOA-SC as described in Algorithm 2 considers only feasible solutions. In discrete space the initialisation operator generates random building spatial designs composed of cuboid spaces consisting of a random number of cells, within the restrictions of the supercube representation. The continuous variables are initialised uniformly at random within their bounds. Either discrete mutations are applied with probability $MT = 0.4993$, or continuous mutations with probability $1 - MT$. Mutation in discrete space works by extending or contracting existing spaces to change their shape, while ensuring that these changes finally lead to another feasible design. Since this mutation procedure can consist of multiple steps, it is possible to move into infeasible regions, and then back to feasible space. As a result, disconnected feasible areas can be reached as well. For mutation of the continuous variables, polynomial mutation is applied with probability $MC = 0.4381$. Both MT and MC are used with values as found by parameter tuning in [2], although somewhat different objective functions were considered there.

Note that mutation is applied either in discrete space, or in continuous space. When mutations are applied on the discrete variables a design may get a significantly altered shape. As a result the optimal settings for the continuous variables change, and mutating them at the same time may have little meaning. Further, all designs are rescaled in the continuous domain (as described in [3]) to the same volume to be able to make a sensible comparison between them. Therefore, any

Algorithm 2. SMS-EMOA-SC

```
input: $\mu, MT, MC$
output: PFA based on all evaluated solutions
Initialise population $X$ of $\mu$ parents as in [2]
while Stop condition not met do
    $\mathbf{x}' \leftarrow$ A uniform random individual from $X$
    if $U(0,1) \leq MT$ then        ▷ Where $U(0,1)$ returns a uniform random number
        if $U(0,1) \leq 0.5$ then
            n_steps ← 1                                        ▷ Local move
        else
            n_steps ← 3                                        ▷ Explorative move
        end if
        Mutate binary variables in $\mathbf{x}'$ with n_steps as in [2]
    else
        Apply polynomial mutation to each continuous variable in $\mathbf{x}'$ with probability $MC$
    end if
    Rescale the continuous variables of $\mathbf{x}'$ until the design reaches the desired spatial volume
    $X \leftarrow$ Select $\mu$ individuals from $X \cup \mathbf{x}'$
end while
```

changes in the discrete domain automatically also result in changes in the continuous domain. Finally, mutations in discrete space may – chosen uniformly at random – consist either of a single step, to make a local move, or of three steps, to make an explorative move.

5.3 MEMO-SC

Algorithm 3 shows how the SMS-EMOA-SC [2,3] and the HIGA-MO [26] algorithms are combined into a new memetic algorithm. The evaluation budget is split between the two approaches according to a given fraction $frac = 0.5$ to be used for global search. Aside from this, the behaviour is the same as for the separate algorithms.

Although different hybridisation strategies are possible, here a relay-hybrid is chosen. This is favoured here over an alternate-hybrid for various reasons. Applying the relatively expensive HVI gradient at earlier stages of the optimisation process may result in costly updates to points in suboptimal discrete subspaces. Furthermore, optimising the points in low quality discrete subspaces may even impede finding better solutions in overlapping discrete subspaces. For instance, only 10% of the solutions in some subspace A may be able to improve over the Pareto front (PF) of subspace B. As such, the further the search is away from the PF of B, the more likely it is that a newly discovered solution of the higher quality subspace A is accepted into the population by the evolutionary algorithm. Despite these possible issues, evaluating alternatives to the considered

Algorithm 3. MEMO-SC

```
input: $\mu, MT, MC, \lambda, \sigma, c, \alpha, h$
output: PFA based on all evaluated solutions
Initialise population $X$ of $\mu$ parents as in [2]
while Stop condition not met do
    if $eval \geq eval_{max} \times frac$ then
        Generate a new population as in Algorithm 1
    else
        Generate a new population as in Algorithm 2
    end if
end while
```

relay-hybrid, with appropriate consideration for the noted pitfalls, may still be worth investigating in future work.

6 Experiments

For the comparison of the three algorithms (SMS-EMOA-SC, HIGA-MO-SC, and MEMO-SC) two objectives are considered as discussed in Subsect. 6.1. Following that, Subsect. 6.2 describes the experimental setup.

6.1 Objective Functions

In this work two objectives are considered for the building spatial design problem, related to two disciplines: structural design, and building physics. For both objectives measurements are taken through simulations [5,6]. Settings for each of the simulation models are described briefly in the following.

6.1.1 Structural Design

The Structural Design (SD) objective for a given building spatial design is obtained by taking the total strain energy, here defined as compliance, in N mm (newton millimetre) from a Finite Element (FE) analysis that has been performed on an SD model of that spatial design. An SD model is obtained by means of a design grammar, i.e. a set of design rules that add discipline specific details to a building spatial design. Specifically, the SD grammar adds structural aspects – like structural components, loads, and constraints – to the spatial design [6].

The following SD grammar has been defined for the studies in this work: To every surface in the spatial design a concrete slab is added with thickness $t = 150\,\text{mm}$ (millimetre), elasticity modulus $E = 30000\,\text{Nmm}^{-2}$ (newton per square millimetre), and Poisson's ratio $v = 0.3$. Furthermore, each edge of a surface will be constrained if both endpoints of that edge have an equal z-coordinate that is at or below zero (i.e. ground level). Next, a live load case

$p_{live} = 5.0\,\mathrm{kNm}^{-2}$ (kilo newton per square metre) in $-z$-direction is applied on each concrete slab with a surface normal oriented vertically. Finally, wind load cases are applied, with for each wind load case three load types: $p_{w,p} = 1.0\,\mathrm{kNm}^{-2}$ for pressure; $p_{w,s} = 0.8\,\mathrm{kNm}^{-2}$ for suction; and $p_{w,sh} = 0.4\,\mathrm{kNm}^{-2}$ for shear. Four wind load cases are defined; in positive and negative x- and y-direction respectively. The load types are assigned to all external surfaces in the building spatial design (except to the ground floor surface). This is carried out according to the orientation of the external surface normal vector with respect to the wind direction vector: pressure if they are opposing; suction if they have the same orientation; and shear if they are perpendicular to each other.

FE analysis starts with meshing all the components into finite elements and nodes. Here a structural component is divided into ten elements along every dimension, which results in 10^n elements for n-dimensional components. For each load case, loads and boundary conditions are then applied to nodes, and stiffness relations between the nodes are obtained via finite element formulations. The discretised structural design is formulated as a sparse linear system, which is then solved by the simplicial-LLT solver from the C++ library Eigen [15]. For each load case, the strain energy for each element can be computed once the system has been solved. Finally, the objective is then easily computed as the sum of strain energies over all elements, over all load cases. Note that here, for each element, the strain energy is calculated by $\mathbf{u}^{\top}\mathbf{K}\mathbf{u}$, where $\mathbf{u}$ is the displacement vector of an element and $\mathbf{K}$ is its stiffness matrix.

6.1.2 Building Physics

The building physics (BP) objective is computed as the sum of heating and cooling energy in kW h (kilo watt hour) that is required to keep the air of all spaces of the building spatial design within a certain temperature range during a given simulation time period. The BP design grammar adds thermal related aspects – like volumes of air, thermal separations (e.g. walls and floors), temperature set points, and temperature profiles – to the building spatial design.

The BP grammar starts by defining temperature profiles for the weather and the ground. The ground temperature is set to be constant at $T_g = 10\,^{\circ}\mathrm{C}$. The temperature data of the weather is obtained from real world measured data by KNMI at De Bilt, The Netherlands [16]. Two periods are simulated, three full hot summer days starting 1976, July 2, and three full cold winter days starting 1978, December 30. The grammar initialises all spaces of the building spatial design with their volume, and assigns a heat capacity $C_s = 3600\,\mathrm{J\,K}^{-1}\,\mathrm{m}^{-3}$ (joule per kelvin per cubic metre), a heating set point $T_h = 18\,^{\circ}\mathrm{C}$, a cooling set point $T_c 20\,^{\circ}\mathrm{C}$, a heating power $Q_h = 100\,\mathrm{W\ m}^{-3}$ (watt per cubic metre), a cooling power $Q_c = 100\,\mathrm{W\ m}^{-3}$, and a ventilation rate of one air change per hour. Subsequently, the thermal separations are added, with their heat conduction properties and their connections to the volumes and temperature profiles: All surfaces in the building spatial design are assigned a concrete slab with thickness $t = 150\,\mathrm{mm}$, density $\varrho = 2400\,\mathrm{kgm}^{-3}$ (kilogram per cubic metre), specific heat capacity $C = 850\,\mathrm{JK}^{-1}\,\mathrm{kg}^{-1}$ (joule per kelvin per kilogram), and thermal conductivity $\lambda = 1.8\,\mathrm{W\ K^{-1}\,m^{-1}}$ (watt per kelvin per metre). Additionally, each external surface is assigned insulation on the outside with thickness $t = 150\,\mathrm{mm}$, density $\varrho = 60\,\mathrm{kgm^{-3}}$, specific heat capacity $C = 850\,\mathrm{JK^{-1}\,kg^{-1}}$, and thermal conductivity $\lambda = 0.04\mathrm{W\ K^{-1}\,m^{-1}}$. A warm-up period is defined for each simulation period, starting to run backwards from four days after the beginning of the actual simulation period and ending when the start of the period is reached.

For the simulation, the BP model is first abstracted as a Resistor-Capacitor (RC) network [17], where each volume or separation is modelled by a temperature point called a state. Between each temperature point a resistance is modelled, and a grounded capacitor is attached to each temperature point. The heat flux through the capacitors and resistors in the RC-network can be described by a set of first order ordinary differential equations (ODEs) [6]. This system is solved using time steps of 15 minutes using the error controlled explicit Runge-Kutta-Dopri5 solver by odeint [1]. The simulated heating or cooling of spaces is controlled at each time step by first predicting the energy demand for that time step with the system of ODEs. Then the predicted heating or cooling demand is accepted if it is lower than the available power in a space, if not it is set to the available power. All heating and cooling energies are summed over all spaces and time steps to finally yield the BP objective.

Note that in comparison with earlier work [2–4], the objectives were obtained differently. For the BP objective now realistic heating and cooling performance is used like in [5], instead of only a measure of the outer surface area. And for the SD objective, the number of wind load cases has been reduced to four and the magnitudes of the loads have changed to the values as mentioned above. Moreover, error control has been introduced in the solver of the BP simulations to prevent possible erroneous results compared to [5].

6.2 Setup

A number of aspects of the proposed approach are evaluated empirically. Specifically, a comparison is made between the three described methods: SMS-EMOA-SC, HIGA-MO-SC, and MEMO-SC. Moreover, two versions of both the HIGA-MO-SC and the MEMO-SC algorithms are considered, one with gradient step size adaptation and one without.

Note that due to the mixed-integer nature of the problem the pure HIGA-MO-SC approach cannot be expected to be competitive with the other methods. It is considered here solely to study the behaviour of the HVI gradient on the constrained landscape of this real world problem, and the value of step size adaptation. It may also be used to show that the exploration of the discrete subspace, which HIGA-MO-SC lacks, is essential to find high quality solutions, but this is not new information.

A problem with a supercube size 3333 is considered here. Meaning the supercube has three cells in with, depth, and height dimensions, and also consists of three spaces. Although in earlier work [4] it was found that for a mid-sized supercube like this constraint navigation is still reasonably simple, this problem size already consists of nine continuous variables. For the hypervolume indicator

(HVI) gradient, including numerically computing the gradient (nine evaluations, one per continuous variable), this means each new point requires ten evaluations. Since the focus of this study is on analysing the behaviour of the HVI gradient, and not on constraint navigation, the problem size is considered to be sufficient here.

Given a $3 \times 3 \times 3$ supercube, 27 discrete variables, and nine continuous variables exist: three each for width, depth, and height. The continuous variables for width and depth are bounded in $]0.5, 20]$, while those for height are bounded in $]3, 20]$. This ensures all spaces in the building are sufficiently large for human occupation. During optimisation these variables are rescaled such that the volume of the building spatial designs is maintained at $V_0 = 300\,\text{cm}^3$ (cubic metre), like in [5].

For these experiments each algorithm is executed 35 times with an evaluation budget of 10000. The MEMO-SC approaches are set to switch halfway (i.e. $frac = 0.5$), and thus use 5000 evaluations each on evolutionary search and gradient search. This halfway switch is chosen in order to allow the evolutionary search to progress sufficiently in discrete space, while also giving the gradient search enough time to advance and adjust step sizes as needed. Note that although the evaluation budgets are equal, the number of sampled points is not. During evolutionary search each evaluation equates to a sampled point, while during HVI gradient search ten evaluations are used per sampled point.

Each algorithm considers a population size $\mu = 25$. This value is chosen to ensure a high likelihood of having a well covered PFA. Moreover, it is not so large that it would prohibit applying gradient approximation to the whole population. Note that with 25 individuals the initialisation costs 25 evaluations, leaving 9975 for the rest of the process. This means HIGA-MO-SC is not split exactly in two halves of 5000. Since HIGA-MO-SC stops when it has insufficient budget left to generate a new point (in this case 10 evaluations), it ultimately uses five evaluations less than the two other algorithms.

Settings for the SMS-EMOA-SC algorithm are given in Table 1. Parameters MT and MC control the probability to perform a discrete or continuous mutation, and the probability of mutation per continuous decision variable respectively (see Sect. 5.2 for details). A reference point of $(1.1e9, 1.1e9)$ is used as in previous work [2]. The settings for HIGA-MO-SC are available in Table 2,

Table 1. Settings for SMS-EMOA-SC

μ	MT	MC
25	0.4993	0.4381

Table 2. Settings for HIGA-MO-SC

μ	λ	σ	c	α	h
25	25	0.0025	0.7	0.8	0.01

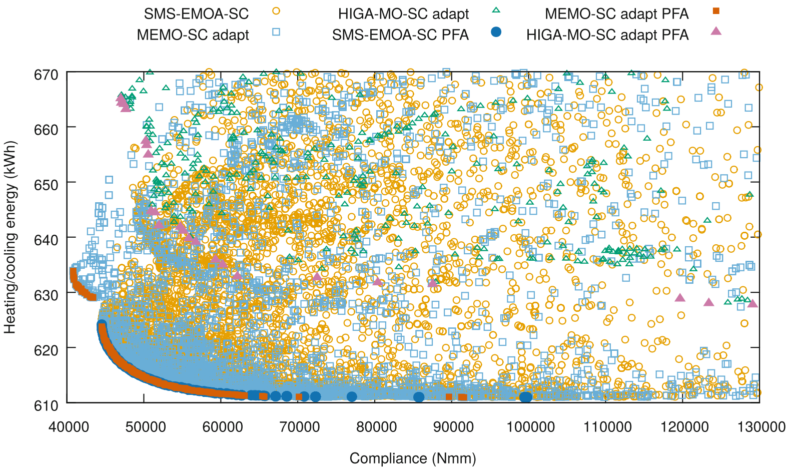

Fig. 2. Scatter plot of the PFA region for a single execution of the SMS-EMOA-SC, adaptive HIGA-MO-SC, and adaptive MEMO-SC approaches

details on their values are available in Sect. 4. Finally, MEMO-SC uses settings from either of the other two algorithms depending on whether it is in the global or local search phase.

7 Results

In Fig. 2 results are shown for a single execution of the SMS-EMOA-SC, adaptive HIGA-MO-SC, and adaptive MEMO-SC algorithms. Both the Pareto front approximations (PFAs), and the points considered during the search (limited to those sufficiently close to the PFAs) are shown. Evidently, SMS-EMOA-SC and MEMO-SC seem to perform similarly well. While, unsurprisingly, HIGA-MO-SC lags behind, unable to navigate the discrete landscape. Even so, HIGA-MO-SC is clearly able to navigate the continuous landscape within the discrete subspaces it is confined to upon initialisation.

Figures 3 and 4 display the behavioural difference in the search strategies of the SMS-EMOA-SC and MEMO-SC approaches during the second half of the optimisation process. MEMO-SC strongly focuses on local improvements to the PFA, while SMS-EMOA-SC continues to explore as well as exploit. Another interesting observation is that for this specific execution MEMO-SC seems to find two partially overlapping discrete subspaces that both contribute to the PFA. This results in a PFA consisting of two parts, one similar to what is found by SMS-EMOA-SC in Fig. 3, and an extra part in the upper-left corner of Fig. 4. Note that the differences in discrete subspaces that are discovered are an artifact of comparing single executions. Given a second execution, the discovered discrete subspaces might be reversed.

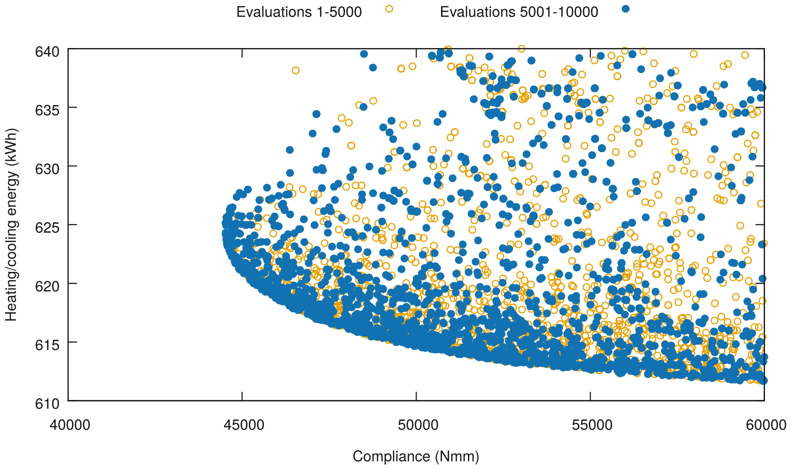

Fig. 3. Scatter plot of the PFA region for a single execution of SMS-EMOA-SC

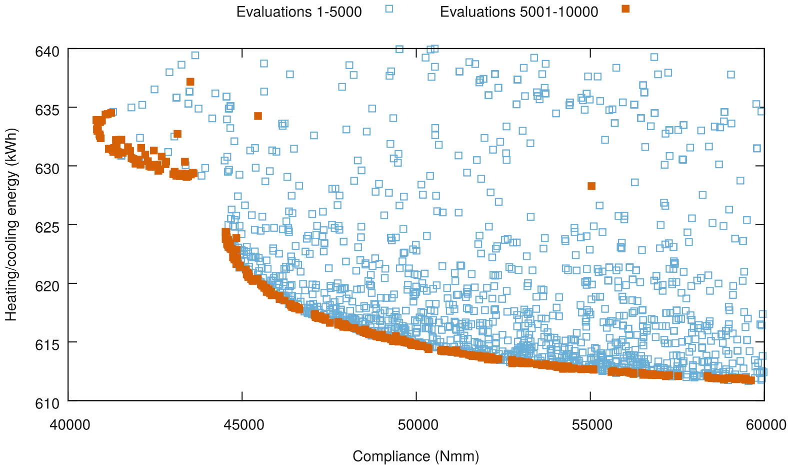

Fig. 4. Scatter plot of the PFA region for a single execution of adaptive MEMO-SC

A visual comparison of the results over multiple repetitions is done using median attainment curves [14]. Figure 5 shows the high level overview of the results, including all of the approaches. Moreover, the results are split in a first and a second half, to indicate how much the algorithms improved during the second half. From this figure it is clear that, as expected, the pure HVI gradient methods are not competitive. Even so, it is also evident that these methods work, and effectively improve their Pareto front approximations (PFA). It also becomes clear form this figure that the use of step size adaptation has a significant effect on the optimisation progress.

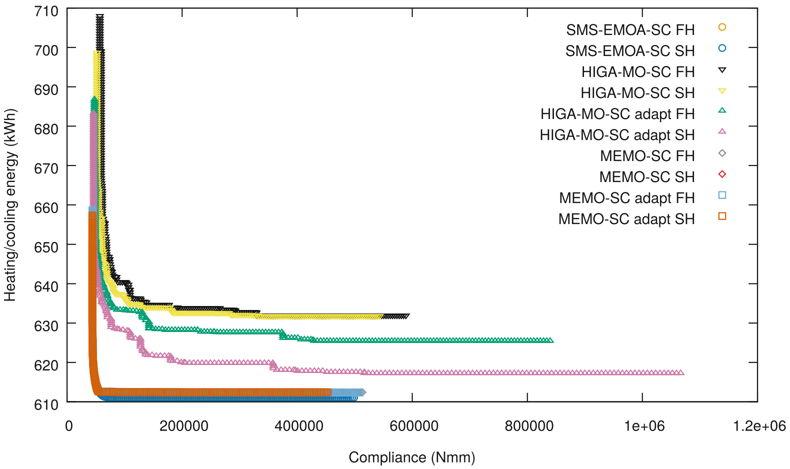

Fig. 5. Median attainment curves per algorithm (35 repetitions each), first halves (FH) and second halves (SH)

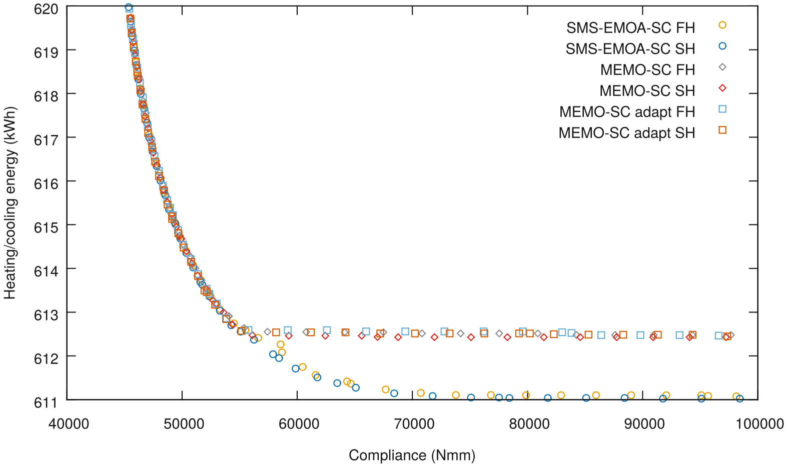

Fig. 6. Median attainment curves per algorithm (35 repetitions each), first halves (FH) and second halves (SH); zoomed in on the knee-point area

When zoomed in on the knee-point area of the median attainment curves in Fig. 6, it can be seen that there is not much difference between the adaptive MEMO-SC, and the regular MEMO-SC algorithms. While SMS-EMOA-SC appears to be able to find better solutions in the heating and cooling energy objective, even during the first half of the search. This is a striking result, given that these three algorithms behave exactly the same during the first half of the search process. Note that despite their equivalent behaviour, given their separately generated random seeds, they can still find different results due to chance.

To understand what is happening Fig. 7 shows the nondominated solutions of every repetition for all considered algorithms. In this figure multiple different PFAs, that are frequently found by all of the competitive approaches, can clearly be identified. These evidently represent the PFAs for different discrete subspaces. Looking back at Fig. 6, it appears that despite using 35 repetitions, the number of times each algorithm ends up in each discrete subspace differs sufficiently to end up with differing median attainment curves. After all, the median attainment curve may be different even if one of the algorithms ends up in (for instance) the optimal discrete subspace only a single time more than the other algorithms.

Based on the points found in Fig. 7 it is also possible to visualise the trade-off between the two objectives. In Fig. 8 three example solutions are shown. One for each objective, and one from the knee point area. For compliance it seems that long, evenly distributed walls with short floor spans are optimal for the distribution of strain across the structural elements. On the other hand, optimal

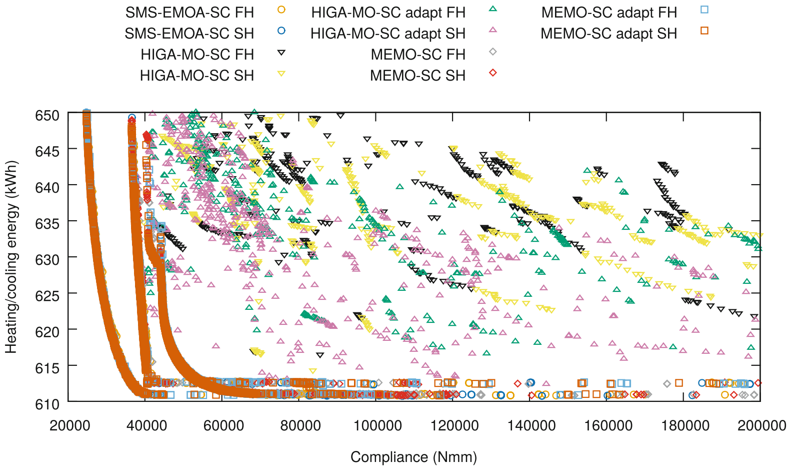

Fig. 7. Nondominated solutions from each of the 35 repetitions per algorithm, first halves (FH) and second halves (SH); zoomed in

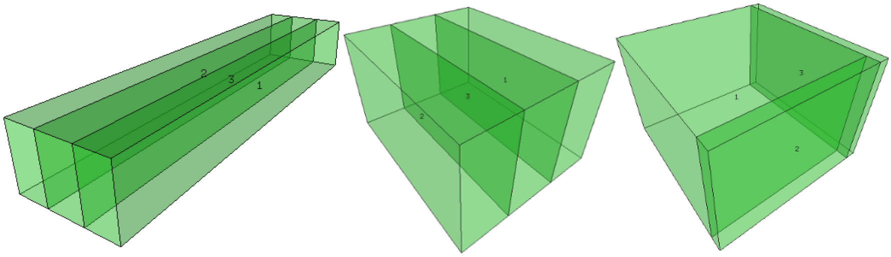

Fig. 8. Example solutions: optimal compliance (left), a knee-point solution (centre), and optimal energy efficiency (right)

energy efficiency is found by using a cubic shape and some spaces as padding to the outside, in order to provide insulation.

Based on the nondominated solutions found over all repetitions of all the approaches (Fig. 9) the objective values are normalised. From these nondominated solutions it is found that for compliance a range of $[0, 500000]$ can be considered, while in energy use a range of $[610, 660]$ is sufficient. All objective values are normalised from those ranges to a $[0, 1]$ range.

Given the normalised objective values, statistics over the hypervolume indicator (HVI) can be computed, with reference point (1,1). Table 3 shows these results per algorithm for the first half. Considering that SMS-EMOA-SC and both MEMO-SC approaches are equivalent in the first half, it is not surprising to see their very similar performance here. Although SMS-EMOA-SC performs slightly better overall, this is purely based on chance.

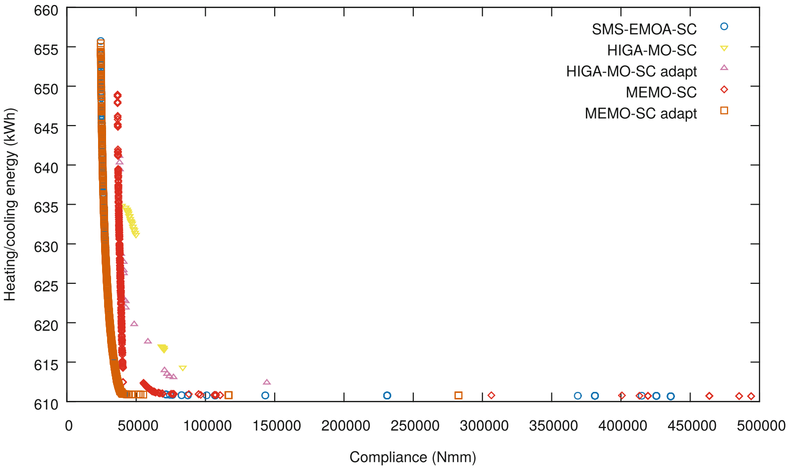

Fig. 9. Nondominated solutions over all 35 repetitions per algorithm, second halves only

Table 3. Statistics of the normalised HVI per algorithm after the first half, over 35 repetitions, best values in bold

Algorithm	Min	Max	Mean	Median	Std. dev.
SMS-EMOA-SC	0.86052	**0.92815**	**0.88674**	**0.88899**	0.01764
HIGA-MO-SC	0.25969	0.75421	0.45662	0.45798	0.11305
HIGA-MO-SC adapt	0.40257	0.76252	0.57560	0.57260	0.09315
MEMO-SC	**0.86112**	0.90129	0.88068	0.87780	**0.01220**
MEMO-SC adapt	0.85973	0.92788	0.88049	0.87633	0.01892

Further, it should be noted that these results largely depend on which discrete subspace an algorithm ends up in. For instance, consider two partially overlapping Pareto front approximations PFA_1 and PFA_2, and two equivalent algorithms A_1, A_2, executed for 10 repetitions each. Now, by chance A_1 could end up in PFA_1 8 out of 10 times, while A_2 does the reverse. Seemingly A_1 would then be better in one objective, and A_2 in the other, although they are actually the same algorithm. In other words, even for a simple case a reasonably large number of repetitions is required. Alternatively each discrete subspace could be analysed separately, but this is contrary to the goal of finding high quality discrete subspaces in the first place. Moreover, recall that the problem at hand does not merely consider two discrete subspaces, but many – often hard to reach – subspaces.

Table 4. Statistics of the normalised HVI per algorithm for the second half, over 35 repetitions, best values in bold

Algorithm	Min	Max	Mean	Median	Std. dev.
SMS-EMOA-SC	**0.86337**	**0.92910**	**0.89087**	**0.89048**	0.01602
HIGA-MO-SC	0.31499	0.77975	0.49822	0.48678	0.10857
HIGA-MO-SC adapt	0.49186	0.85363	0.69118	0.71397	0.08543
MEMO-SC	0.86146	0.90300	0.88163	0.87950	**0.01197**
MEMO-SC adapt	0.86193	0.92797	0.88168	0.87793	0.01843

Table 4 contains the results after completion of the second half of the search process. Once more, SMS-EMOA-SC seems to outperform all other approaches. Moreover, when comparing the results in Table 4 to those in Table 3 it can be observed that SMS-EMOA-SC shows the greatest improvement during the second phase of the search. Although this could be taken as surprising, when taken together with Figs. 6 and 7, it can be postulated that SMS-EMOA-SC simply found the best PFA more often. Moreover, SMS-EMOA-SC is still able to make discrete moves in the second phase, and may therefore find a new, better, front while the MEMO-SC approaches cannot. Regardless of the reason SMS-EMOA-SC appears to be better during the local search phase, the performance of the HIGA-MO-SC approach is also striking. It significantly improves in all metrics during the second phase, and is clearly a viable method for this problem, as long as it is given a good discrete subspace to work in.

All in all, it appears that both the SMS-EMOA-SC and MEMO-SC approaches are able to converge to good Pareto front approximations. Since this is already true after the first half of the search process, there is simply little to improve for either the HVI gradient in MEMO-SC, or the evolutionary approach in SMS-EMOA-SC during the second half. Further, it is observed that the quality of the found PFA depends more on the discrete, than on the continuous decision variables.

8 Conclusion and Discussion

In this work a building spatial design problem for two objectives has been considered. For this problem the shape of a building had to be optimised for structural performance and energy performance. Provided an existing mixed-integer representation, three algorithms have been applied to this optimisation problem. An algorithm based on the hypervolume indicator (HVI) gradient (HIGA-MO-SC), an adapted version of SMS-EMOA (SMS-EMOA-SC), and a memetic algorithm combining the two methods (MEMO-SC).

Results showed that the HVI gradient method by itself could not compete with the evolutionary and memetic approaches. Which, considering the mixed-integer nature of the problem, was not surprising. It has also been shown that the evolutionary approach performed slightly better during the local search phase than the memetic algorithm. However, this may be the result of larger global moves, rather than of its implied local search abilities.

Although the algorithm does not improve significantly in most cases, the HVI gradient remains useful in providing a guarantee of local convergence in the continuous subspace. The non-deterministic evolutionary algorithm cannot provide such guarantees. However, based on the results, in many cases it has a good practical performance in adjusting the continuous variables. Further, the effort spent on solving the integer problem appears to have a more significant impact on the overall result.

Improvement of the MEMO-SC algorithm may be possible by focusing on moving nondominated points, rather than all points. This could reduce the number of used evaluations on points that might never reach the Pareto front, because they are either simply too far away, or worse, stuck in a discrete subspace that is completely dominated by another. Challenges herein are found in how it is ensured that there are sufficiently many points on the Pareto front, and subsequently, how to ensure they remain on the Pareto front during the search.

Other hybridisation strategies could also be explored for the MEMO-SC algorithm. In this work a relay-hybrid, where first evolutionary search is applied and then HVI gradient ascent, has been considered. An alternate-hybrid strategy, which continuously alternates between the two approaches, may produce different results. Additionally, a comparison between HIGA-MO-SC and SMS-EMOA-SC on a single discrete subspace (such that only continuous variables are considered by both of them) could give interesting insights.

One limitation of the HVI gradient is that it cannot navigate mixed-integer space. To overcome this, metamodels can be considered. For instance, metamodels are used for the mixed-integer case in [19]. Use of the HVI gradient is possible in this case by taking the HVI gradient of the metamodel, rather than of the actual objective functions. Otherwise, it may be that methods integrating the HVI gradient are not well suited to problems with far more integer than continuous variables, as is considered here. To investigate this, a comparison to a problem with a small number of integer variables would be interesting as future work.

If the MEMO-SC algorithm as it was presented in this work is improved by the suggestions above or by other means, such that it provides advantages beyond the evolutionary approach, new directions open up. Then, in the future, combining an improved MEMO-SC algorithm and the cooperation between superstructure and free representations presented in [5] can lead to an optimisation strategy covering all levels of the building spatial design problem. Specifically, it would allow exploration with the free representation using co-evolutionary design simulation, followed by global search with the evolutionary algorithm and local search with the HVI gradient method when using the superstructure representation.

Despite the existence of a plethora of different measures to compare the quality of Pareto front approximations (PFAs), determining which PFA is better, or how two PFAs differ remains challenging. In particular this holds if, as in this work, there is a desire for statistical significance and comparisons are done over multiple repetitions per algorithmic approach. Moreover, mixed-integer landscapes further complicate the process, where different repetitions may end up in distinct, but possibly overlapping, discrete subspaces. Evidently, much work remains in the area of multi-objective quality measures.

Acknowledgements. This work is part of the TTW-Open Technology Programme with project number 13596, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO). The authors express their gratitude to the reviewers for their valuable comments.

References

1. Ahnert, K., Mulansky, M.: Odeint - solving ordinary differential equations in C++. AIP Conference Proceedings vol. 1389, no. 1, pp. 1586–1589 (2011). https://doi.org/10.1063/1.3637934
2. van der Blom, K., Boonstra, S., Hofmeyer, H., Bäck, T., Emmerich, M.T.M.: Configuring advanced evolutionary algorithms for multicriteria building spatial design optimisation. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1803–1810. IEEE (2017). https://doi.org/10.1109/CEC.2017.7969520
3. van der Blom, K., Boonstra, S., Hofmeyer, H., Emmerich, M.T.M.: Multicriteria building spatial design with mixed integer evolutionary algorithms. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.): Parallel Problem Solving from Nature – PPSN XIV, Lecture Notes in Computer Science, vol. 9921, pp. 453–462. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45823-6_42
4. van der Blom, K., Boonstra, S., Hofmeyer, H., Emmerich, M.T.M.: A super-structure based optimisation approach for building spatial designs. In: Papadrakakis, M., Papadopoulos, V., Stefanou, G., Plevris, V. (eds.): VII European Congress on Computational Methods in Applied Sciences and Engineering – ECCOMAS VII, vol. 2, pp. 3409–3422. National Technical University of Athens (2016). https://doi.org/10.7712/100016.2044.10063

5. Boonstra, S., van der Blom, K., Hofmeyer, H., Emmerich, M.T.: Combined superstructured and super-structure free optimisation of building spatial designs. In: Koch, C., Tizani, W., Ninić, J. (eds.): 24nd International Workshop of the European Group for Intelligent Computing in Engineering, pp. 23–34. University of Nottingham (2017)
6. Boonstra, S., van der Blom, K., Hofmeyer, H., Emmerich, M.T., van Schijndel, J., de Wilde, P.: Toolbox for super-structured and super-structure free multi-disciplinary building spatial design optimisation. Adv. Eng. Inf. **36**, 86–100 (2018). https://doi.org/10.1016/j.aei.2018.01.003
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
8. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering pareto sets by multilevel subdivision techniques. J. Optim. Theory Appl. **124**(1), 113–136 (2005). https://doi.org/10.1007/s10957-004-6468-7
9. Ehrgott, M.: Multicriteria Optimization, vol. 491. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-27659-9
10. Emmerich, M., Beume, N., Naujoks, B.: An emo algorithm using the hypervolume measure as selection criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.): Evolutionary Multi-Criterion Optimization, pp. 62–76. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_5
11. Emmerich, M., Deutz, A.: Time complexity and zeros of the hypervolume indicator gradient field. In: Schuetze, O., Coello Coello, C.A., Tantar, A.A., Tantar, E., Bouvry, P., Moral, P.D., Legrand, P. (eds.): EVOLVE - A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation III, pp. 169–193. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-01460-9_8
12. Emmerich, M., Deutz, A., Beume, N.: Gradient-based/evolutionary relay hybrid for computing pareto front approximations maximizing the s-metric. In: Bartz-Beielstein, T., Blesa Aguilera, M.J., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M. (eds.): Hybrid Metaheuristics, pp. 140–156. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75514-2_11
13. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**(3), 479–494 (2000). https://doi.org/10.1007/s001860000043
14. Fonseca, C.M., Grunert da Fonseca, V., Paquete, L.: Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.): Evolutionary Multi-Criterion Optimization, pp. 250–264. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_18
15. Guennebaud, G., Jacob, B., et al.: Eigen v3: A C++ linear algebra library (2010). http://eigen.tuxfamily.org. Accessed 7 May 2018
16. Koningklijk Nederlands Metereologisch Instituut: Measured weather data in the Netherlands (2018). http://www.knmi.nl/nederland-nu/klimatologie/daggegevens. Accessed 7 May 2018
17. Kramer, R., van Schijndel, J., Schellen, H.: Simplified thermal and hygric building models: a literature review. Front. Archit. Res. **1**(4), 318–325 (2012). https://doi.org/10.1016/j.foar.2012.09.001
18. Lara, A., Sanchez, G., Coello, C.A.C., Schütze, O.: Hcs: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **14**(1), 112–132 (2010). https://doi.org/10.1109/TEVC.2009.2024143

19. Li, R., Emmerich, M.T.M., Eggermont, J., Bovenkamp, E.G.P., Bäck, T., Dijkstra, J., Reiber, J.H.C.: Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In: IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2764–2771 (2008). https://doi.org/10.1109/CEC.2008.4631169
20. Martín, A., Schütze, O.: Pareto tracer: a predictorcorrector method for multi-objective optimization problems. Eng. Optim. **50**(3), 516–536 (2018). https://doi.org/10.1080/0305215X.2017.1327579
21. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program 158-79, Technical Report, pp. 1–68 (1989)
22. Schäffler, S., Schultz, R., Weinzierl, K.: Stochastic method for the solution of unconstrained vector optimization problems. J. Optim. Theory Appl. **114**(1), 209–222 (2002). https://doi.org/10.1023/A:1015472306888
23. Schütze, O., Coello, C.A.C., Mostaghim, S., Talbi, E.G., Dellnitz, M.: Hybridizing evolutionary strategies with continuation methods for solving multi-objective problems. Eng. Optim. **40**(5), 383–402 (2008). https://doi.org/10.1080/03052150701821328
24. Schütze, O., Hernández, V.A.S., Trautmann, H., Rudolph, G.: The hypervolume based directed search method for multi-objective optimization problems. J. Heuristics **22**(3), 273–300 (2016). https://doi.org/10.1007/s10732-016-9310-0
25. Sosa Hernández, V.A., Schütze, O., Emmerich, M.: Hypervolume maximization via set based newton's method. In: Tantar, A.A., Tantar, E., Sun, J.Q., Zhang, W., Ding, Q., Schütze, O., Emmerich, M., Legrand, P., Del Moral, P., Coello Coello, C.A. (eds.): EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V, pp. 15–28. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07494-8_2
26. Wang, H., Deutz, A., Bäck, T., Emmerich, M.: Hypervolume indicator gradient ascent multi-objective optimization. In: Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., Grimme, C. (eds.) Evolutionary Multi-Criterion Optimization, pp. 654–669. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54157-0_44
27. Wang, H., Ren, Y., Deutz, A., Emmerich, M.: On steering dominated points in hypervolume indicator gradient ascent for bi-objective optimization. In: Schütze, O., Trujillo, L., Legrand, P., Maldonado, Y. (eds.): NEO 2015: Results of the Numerical and Evolutionary Optimization Workshop NEO 2015 held at September 23–25 2015 in Tijuana, Mexico, pp. 175–203. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44003-3_8
28. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature – PPSN V, pp. 292–301. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056872
29. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003). https://doi.org/10.1109/TEVC.2003.810758