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

Blom, K. van der

Chapter 6

Local Search with Set Gradients

In the previous chapter problem specific constraint handling operators were introduced to efficiently navigate the search space, and as such RQ2 was answered. Furthermore, parameter tuning was employed to maximise the performance of the algorithm. However, finding exact optima remains a challenge. To this end this chapter follows up on RQ3, and investigates whether local search can improve the solutions found during the global search performed by evolutionary optimisation.

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 [77]. 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 (PFA) is the hypervolume indicator (HVI). In [40] the HVI gradient is described, which allows a set of points to be moved towards the Pareto front, and to be distributed well across the Pareto front. The HVI gradient has been tested on benchmark functions [41], and improvements to its computation [40], as well as to the navigation of dominated points [97, 98] have been proposed, but 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 [46, 85], and gradients that are used for the computation of bounds on subspaces [34]. Other ideas to use gradients in memetic search have been proposed in the literature, such as continuation methods that start from a pre-computed Pareto optimal point and locally extend Pareto fronts by steps along tangent planes [74, 86]. Moreover, directed search has been proposed, which steers points in a desired direction, either across or towards the Pareto front. Such methods also use gradients in order to construct these directions in the decision space [87]. Further, recent developments on derivative free exact local methods are summarised in [29]. The HVI gradient is favoured here since it updates the Pareto front approximation as a whole, and local optimality has been verified [41].

In this work the HVI gradient is applied to the real world problem of building spatial design for the first time. Furthermore, this chapter employs simulations to measure energy efficiency, rather than the previous substitute of outside surface area. The combination of an evolutionary multi-objective optimisation algorithm (EMOA) and the HVI gradient results in a memetic multi-objective optimisation (MEMO) algorithm. Specifically, here the $\mathcal{S}$-metric (also known as HVI) selection EMOA (SMS-EMOA [39]), with specialised operators (Section 5.3) 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 [97] is employed.

To summarise, key points of this chapter 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 [41] 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. Finally, more accurate measures of energy and structural

performance are considered in this chapter.

The remainder of this chapter is structured as follows. Section 6.1 reviews the basics of multi-objective optimisation, and briefly describes the principles of the hypervolume indicator. In Section 6.2 the hypervolume indicator (HVI) gradient, and its use in algorithms are discussed. Section 6.3 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, Section 6.4 describes the experimental setup for the evaluation of the different algorithms. This is naturally succeeded by Section 6.5, where the results are analysed. Finally, Section 6.6 summarises the chapter as a whole, and discusses future work resulting from the study.

6.1 Multi-Objective Optimisation and the Hypervolume Indicator

Given their importance to the introduction of the HVI gradient, key concepts of multi-objective optimisation and the HVI are briefly summarised here. A more gentle introduction is available in Section 2.2.

In continuous 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 $\mathbf{y} = \mathbf{f(x)} \coloneqq [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]$, simultaneously. Without loss of generality, it is assumed that each objective function $f_i : \mathbb{R}^d \to \mathbb{R}$ 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 $\mathbf{y}$. In continuous MOPs, the efficient set [38] is typically 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 (HVI), in the early literature also known as the $\mathcal{S}$-metric [106, 107]. The HVI 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 $\boldsymbol{\rho} \in \mathbb{R}^m$. For notational brevity the hypervolume indicator is denoted as H for mathematical use, while otherwise the more expressive abbreviation HVI is used. As such, the hypervolume indicator for Y is denoted with $H(Y)$. 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.

6.2 HVI Gradient Ascent Multi-Objective Optimisation

This section first introduces the hypervolume indicator (HVI) gradient in its general form. Following this, subsections for normalisation of the HVI gradient, step size adaptation for the HVI gradient, and finally update rules for the considered points are included.

6.2.1 Hypervolume Indicator Gradient

To give a 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: $\mathcal{H}_\mathbf{F}(\mathbf{X}) \coloneqq H(\mathbf{F}(\mathbf{X})) = H(\mathbf{Y})$. 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 Equation 6.1, and represents the direction of steepest improvement of the HVI for the entire Pareto Front Approximation (PFA).

$$\nabla \mathcal{H}_\mathbf{F}(\mathbf{X}) = \left[\frac{\partial \mathcal{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{X}^{(1)}}^\top, \ldots, \frac{\partial \mathcal{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{X}^{(\mu)}}^\top \right]^\top . \tag{6.1}$$

Subsequently, Equation 6.2 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.

$$\frac{\partial \mathcal{H}_\mathbf{F}(\mathbf{X})}{\partial \mathbf{X}^{(i)}} = \left[\frac{\partial \mathcal{H}_\mathbf{F}}{\partial x_1^{(i)}}, \ldots, \frac{\partial \mathcal{H}_\mathbf{F}}{\partial x_d^{(i)}} \right]^\top . \tag{6.2}$$

Each subgradient can then be computed as:

$$\frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial x_j^{(i)}}(\mathbf{X}) = \sum_{k=1}^{m} \frac{\partial H}{\partial y_k^{(i)}}(\mathbf{Y}) \times \frac{\partial f_k(\mathbf{x}^{(i)})}{\partial x_j^{(i)}} \,. \tag{6.3}$$

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:

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

Note that strictly greater (smaller) values are subtracted, 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 Equation 6.3 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.

6.2.2 Normalisation

A limitation of the HVI gradient method is the so-called *creepiness* behaviour, as analysed in [51]. 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 Equation 6.4, before using them to update the original points.

$$G_{norm} = \left[\frac{\frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial \mathbf{x}^{(1)}}}{\left\| \frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial \mathbf{x}^{(1)}} \right\|}^{\top}, \ldots, \frac{\frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial \mathbf{x}^{(\mu)}}}{\left\| \frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial \mathbf{x}^{(\mu)}} \right\|}^{\top} \right]^{\top}, \quad \text{where} \quad \left\| \frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial \mathbf{x}^{(i)}} \right\| = \sqrt{\sum_{j=1}^{d} \left(\frac{\partial \mathcal{H}_{\mathbf{F}}}{\partial x_j^{(i)}} \right)^2} . \quad (6.4)$$

6.2.3 Step Size Adaptation

In [98], the authors mentioned that the normalised subgradients may lead to oscillatory (even divergent) behaviour. To mitigate this effect, the step size adaptation mechanism that has been proposed in [97] is adopted as follows. In Equation 6.5 $\langle \cdot, \cdot \rangle$ 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.

$$I_t^{(i)} = \left\langle \left(\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} \right)_{t-1}, \left(\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} \right)_{t} \right\rangle , \qquad i = 1, \ldots, \mu , \qquad t = 1, 2, \ldots . \quad (6.5)$$

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 (Equation 6.6). The accumulation coefficient $0 < c < 1$ controls how much new information will be incorporated.

$$p_t^{(i)} \leftarrow (1 - c) \times p_{t-1}^{(i)} + c \times I_t^{(i)} , \qquad i = 1, \ldots, \mu , \qquad t = 1, 2, \ldots . \quad (6.6)$$

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 Equation 6.7. The parameter α controls the rate of change for updates to the step size.

$$\sigma_{t+1}^{(i)} = \begin{cases} \sigma_t^{(i)} \times \alpha & \text{if } p_t^{(i)} < 0, \\ \sigma_t^{(i)} & \text{if } p_t^{(i)} = 0, \\ \sigma_t^{(i)} / \alpha & \text{if } p_t^{(i)} > 0. \end{cases} \qquad 0 < \alpha < 1 . \quad (6.7)$$

The parameters $c = 0.7$ and $\alpha = 0.8$ are used here as they were used in [97], but should ideally be tuned for the specific problem. Both $\left(\frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}^{(i)}} \right)_{t-1}$ and $p_{t-1}^{(i)}$ are initialised to zeros, such that the starting position is neutral.

6.2.4 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 [66] the authors defined a new search direction according to which they suggested to move dominated points.

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 [98]. 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 [97], 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 [32], 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 [97] is used here, as shown in Equation 6.8. 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,

$$\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \sigma^{(i)} \frac{\partial \mathcal{H}_{\mathbf{F}}(\mathbf{X})}{\partial \mathbf{x}_j^{(i)}} , \qquad i = 1, \ldots, \mu , \qquad j = 1, \ldots, d . \tag{6.8}$$

6.3 Algorithms

Each subsection here describes one of the considered algorithms. First the HIGA-MO-SC approach, as adapted from the standard HIGA-MO [97] algorithm. Second the SMS-EMOA-SC algorithm, previously introduced in [15]. Finally, a combination of the two in the form of a memetic algorithm, MEMO-SC, is considered.

6.3.1 HIGA-MO-SC

The HVI gradient method adapted to the context of building spatial design is described in Algorithm 4. There, HIGA-MO [97] is adjusted to work with the supercube representation and forms the HIGA-MO-SC algorithm. An initial population is generated with the problem specific initialisation procedure introduced in Section 5.3.1. Following this, the population is sorted according to nondominated sorting [32]. Each front is then updated separately as follows. First, the HVI gradient is computed for the continuous subspace as previously described in Section 6.2.1. Second, the HVI gradients are normalised using Equation 6.4. Third, step sizes are updated by employing Equations 6.5, 6.6, and 6.7. Finally, the old points are replaced with new points generated according to the normalised HVI gradients and the updated step sizes.

Algorithm 4 HIGA-MO-SC

```
1: input: μ, λ, σ, c, α, h
2: output: PFA based on all evaluated solutions
3: Initialise population X of μ parents as in Section 5.3.1
4: while Stop condition not met do
5:     while X ≠ ∅ do
6:         X_nds ← NDS_1(X)          ▷ Where NDS_1 returns the first front after nondominated sorting
7:         X ← X \ X_nds
8:         Compute the HVI gradient for X_nds according to Section 6.2.1
9:         Normalise HVI gradient of X_nds according to Equation 6.4
10:        Update step size of X_nds according to Equations 6.5, 6.6, and 6.7
11:        Move X_nds according to Equation 6.8
12:        X' ← X' ∪ X_nds
13:    end while
14:    X ← X'
15: end while
```

Note that HIGA-MO-SC as used here differs from HIGA-MO from [97] in two aspects. 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. Therefore, 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 [40].

6.3.2 SMS-EMOA-SC

Algorithm 5 SMS-EMOA-SC

input: μ, MT, MC
output: PFA based on all evaluated solutions
Initialise population X of μ parents as in Section 5.3.1
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 $\leftarrow 1$ ▷ Local move
        else
            n_steps $\leftarrow 3$ ▷ Explorative move
        end if
        Mutate binary variables in $\mathbf{x}'$ with n_steps as in Section 5.3.2
    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 μ individuals from $X \cup \mathbf{x}'$
end while

The SMS-EMOA SuperCube (SMS-EMOA-SC) algorithm is the result of tuning the tailored SMS-EMOA in Section 5.5.3. Through the use of problem specific initialisation and mutation operators SMS-EMOA-SC as described in Algorithm 5 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. Consequently, disconnected feasible areas can be reached as well. For mutation of the continuous variables, polynomial mutation [32] is applied with probability $MC = 0.4381$. Both MT and MC are used with values as found by parameter tuning in Section 5.5.3, 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 Section 4.4.3) to the same volume to be able to make a sensible comparison between them. Therefore, any 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.

6.3.3 MEMO-SC

Algorithm 6 shows how the SMS-EMOA-SC Section 6.3.2 and the HIGA-MO [97] 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.

Algorithm 6 MEMO-SC

```
1: input: μ, MT, MC, λ, σ, c, α, h
2: output: PFA based on all evaluated solutions
3: Initialise population X of μ parents as in Section 5.3.1
4: while Stop condition not met do
5:     if eval ≥ eval_max × frac then
6:         Generate a new population as in Algorithm 4
7:     else
8:         Generate a new population as in Algorithm 5
9:     end if
10: end while
```

Although different hybridisation strategies are possible, here a relay hybrid [94] is chosen. This is favoured 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 relay hybrid, with appropriate consideration for the noted pitfalls, may still be worth investigating in future work.

6.4 Experiments

For the comparison of the three algorithms (SMS-EMOA-SC, HIGA-MO-SC, and MEMO-SC) two objectives are considered as discussed in Subsection 6.4.1. Following that, Subsection 6.4.2 describes the experimental setup.

6.4.1 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 computed through simulations [23, 25]. Settings for each of the simulation models are described briefly in the following.

Note that for both structural design (SD), and building physics (BP), improved metrics are used here compared to the previous chapters (4, 5). For the SD objective, the number of wind load cases has been reduced to four and the magnitudes of the loads have changed. The BP objective now uses realistic heating and cooling performance like in [23], instead of only a measure of the outer surface area. Moreover, error control has been introduced in the solver of the BP simulations to prevent possible erroneous results compared to [23].

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 developed for 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 [25].

The following SD grammar has been defined for the studies in this work: For every surface in the spatial design a concrete slab is added with thickness $t = 150\,\mathrm{mm}$ (millimetre), Young's modulus $E = 30\,000\,\mathrm{N\,mm^{-2}}$ (newton per square millimetre), and Poisson's ratio $\nu = 0.3$. 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 $p_{live} = 5.0\,\mathrm{kN\,m^{-2}}$ (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: $p_{w,p} = 1.0\,\mathrm{kN\,m^{-2}}$ for pressure, $p_{w,s} = 0.8\,\mathrm{kN\,m^{-2}}$ for suction, and $p_{w,sh} = 0.4\,\mathrm{kN\,m^{-2}}$ 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 of 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 is applied 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 the 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 [50]. 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.

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 $T_g = 10\,°\mathrm{C}$. The temperature data of the weather is obtained from real world measured data by KNMI (Koninklijk Nederlands Meteorologisch Instituut) at De Bilt, The Netherlands [61]. 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 $C_s = 3600\,\mathrm{J\,K^{-1}\,m^{-3}}$ (joule per kelvin per cubic metre), a heating set point $T_h = 18\,°\mathrm{C}$, a cooling set point $T_c = 20\,°\mathrm{C}$,[1] 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 $\varrho = 2400\,\mathrm{kg\,m^{-3}}$ (kilogram per cubic metre), specific heat capacity $C = 850\,\mathrm{J\,K^{-1}\,kg^{-1}}$ (joule per kelvin per kilogram), and thermal conductivity $k = 1.8\,\mathrm{W\,K^{-1}\,m^{-1}}$ (watt per kelvin per metre). Additionally, each external surface is assigned insulation on the outside with thickness $t = 150\,\mathrm{mm}$, density $\varrho = 60\,\mathrm{kg\,m^{-3}}$, specific heat capacity $C = 850\,\mathrm{J\,K^{-1}\,kg^{-1}}$, and thermal conductivity $k = 0.04\,\mathrm{W\,K^{-1}\,m^{-1}}$ (values based on stone wool [13]). 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 [63], 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) [25]. This system is solved using time steps of 15 minutes using the error controlled explicit Runge-Kutta-Dopri5 solver by odeint [2]. 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

[1] Although these set points are close together, this does not result in issues relevant for the presented optimisation problem. Cooling may become active when it is colder outside than inside, but due to the 15 minute time steps will not result in a temperature drop below the heating set point. Further, these simulations result in a somewhat distorted view with regard to the quantitative energy performance (which is anyway not accurate because things like solar irradiation are not considered), but the qualitatively performance between spatial designs matches reality. This is sufficient because only qualitative comparisons between spatial designs are considered here.

in a space, otherwise 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.

6.4.2 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 Section 4.6 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 (per space, so 81 in total), 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 $]3, 20]$ (all in metres). 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 kept within one cubic millimetre of $V_0 = 300\,\mathrm{m}^3$ (cubic metre), as described in [23].

For these experiments each algorithm is executed 35 times with an evaluation budget of 10 000. 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 an insufficient evaluation budget left to generate a new point (in this case 10 evaluations), it ultimately uses five evaluations less than the two other algorithms. However, this should have no significant impact on the results.

Settings for the SMS-EMOA-SC algorithm are given in Table 6.1. 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 Section 6.3.2 for details). A reference point of $(1.1e9, 1.1e9)$ is used as in Section 5.5. The settings for HIGA-MO-SC are available in Table 6.2, details on their values are available in Section 6.2. Finally, MEMO-SC uses settings from either of the other two algorithms, depending on whether it is in the global or local search phase.

μ	MT	MC
25	0.4993	0.4381

Table 6.1: Settings for SMS-EMOA-SC.

μ	λ	σ	c	α	h
25	25	0.0025	0.7	0.8	0.01

Table 6.2: Settings for HIGA-MO-SC.

6.5 Results

In Figure 6.1 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 suffi-

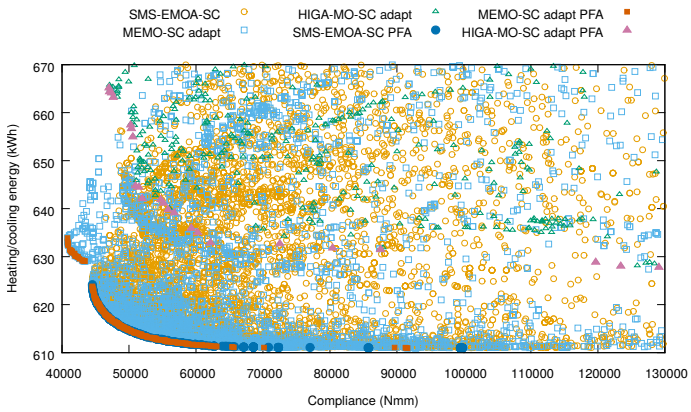
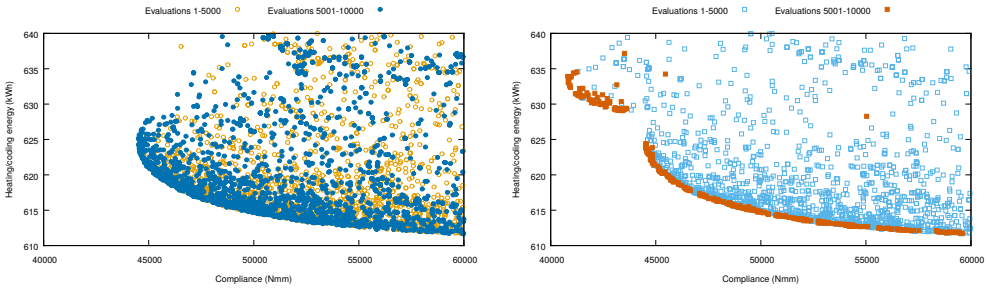


Figure 6.1: Scatter plot of the PFA region for a single execution of the SMS-EMOA-SC, adaptive HIGA-MO-SC, and adaptive MEMO-SC approaches.

ciently close to the PFAs) are shown. Evidently, SMS-EMOA-SC and MEMO-SC seem to perform similarly well. In contrast to these results – 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.



(a) SMS-EMOA-SC. **(b)** Adaptive MEMO-SC.

Figure 6.2: Scatter plot of the PFA region for a single execution.

Figures 6.2a and 6.2b 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 Figure 6.2a, and an extra part in the upper-left corner of Figure 6.2b. 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.

A visual comparison of the results over multiple repetitions is done using median attainment curves [49]. Figure 6.3 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 in a mixed-integer environment. 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.

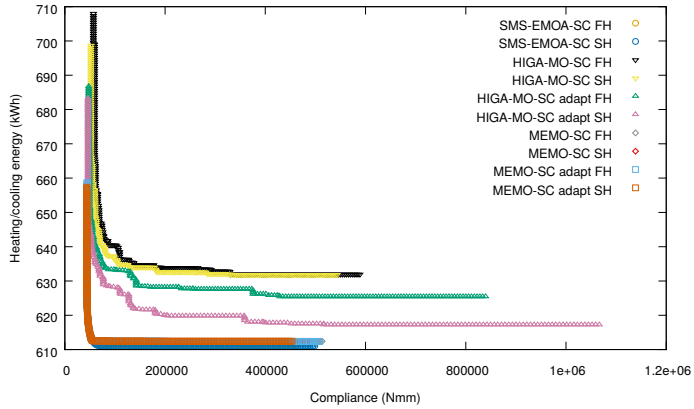


Figure 6.3: Median attainment curves per algorithm (35 repetitions each), first halves (FH) and second halves (SH).

When zoomed in on the knee point area of the median attainment curves in Figure 6.4, 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.

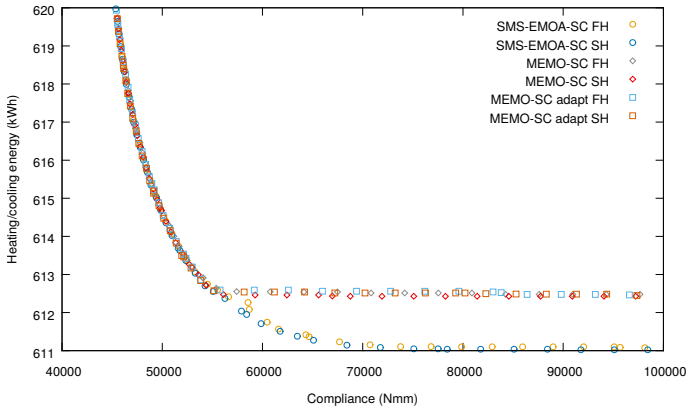


Figure 6.4: Median attainment curves per algorithm (35 repetitions each), first halves (FH) and second halves (SH); zoomed in on the knee point area.

To understand what is happening Figure 6.5 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 Figure 6.4, 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 Figure 6.5 it is also possible to visualise the trade-off between the two objectives. In Figure 6.6 three example solutions are shown, together with some details on their features in Table 6.3. 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 energy efficiency is found by using a cubic shape and some spaces as padding to the outside, in order to provide insulation.

The union of all solutions found over all repetitions of all the approaches is taken. Based on the nondominated solutions of this collection (Figure 6.7), the objective values are normalised. From these nondominated solutions it is found that for compliance a range of $[0, 500\,000]$ can be considered, while in energy use a range of $[610, 660]$ is sufficient. All objective values then are normalised from those ranges to a $[0, 1]$ range.

Given the normalised objective values, statistics over the hypervolume indicator

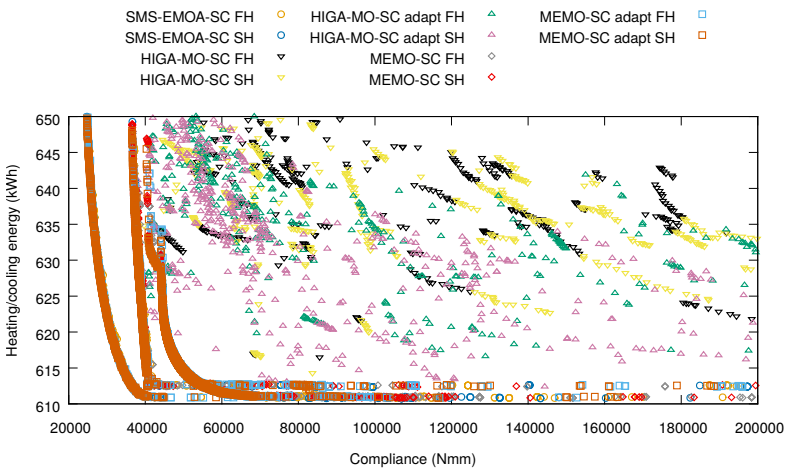


Figure 6.5: Nondominated solutions from each of the 35 repetitions per algorithm, first halves (FH) and second halves (SH); zoomed in.

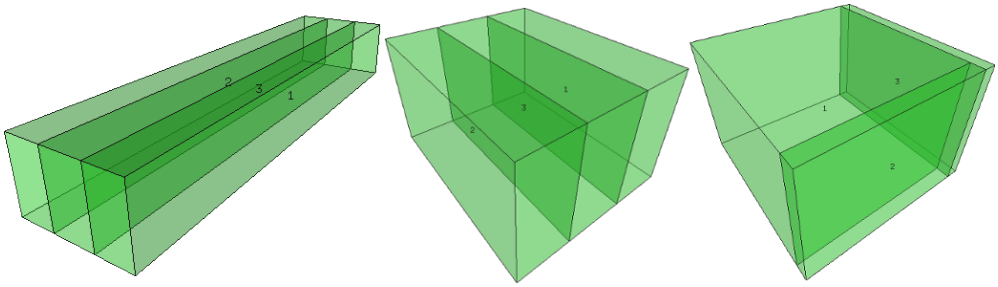


Figure 6.6: Example solutions: optimal compliance (left), a knee point solution (centre), and optimal energy efficiency (right).

(HVI) can be computed, with reference point (1,1). Table 6.4 shows these results per algorithm for the first half of the optimisation process. 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.

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

	Compliance (N mm)	Heating/cooling energy (kW h)	Surface area (m)	Soil surface (m)	Height (m)	Longest edge (m)	Shortest edge (m)
Optimal compliance	24478.9	655.7	251	100	3.0	20.2	4.9
Knee point	37816.4	611.7	216	60	5.0	8.6	7.0
Optimal energy efficiency	435965.0	610.7	215	58	5.2	7.6	7.6

Table 6.3: Details on the features of the optimal compliance, knee point, and optimal energy efficiency spatial designs.

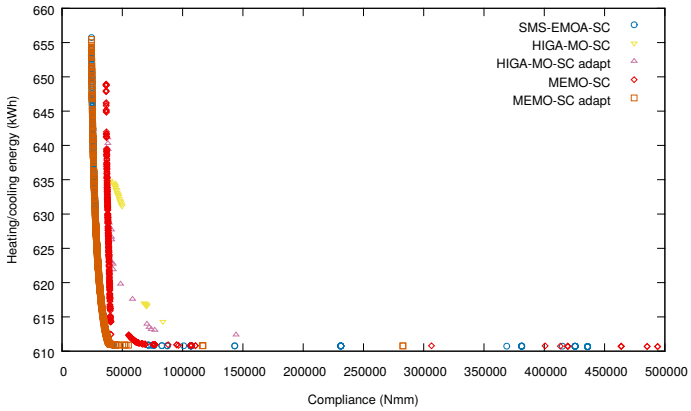


Figure 6.7: Nondominated solutions over all 35 repetitions per algorithm, second halves only.

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 difficult to reach – subspaces.

Table 6.5 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 6.5 to those in Table 6.4 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 Figures 6.4 and 6.5, 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

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

Table 6.4: 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	**0.86337**	**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 6.5: Statistics of the normalised HVI per algorithm for the second half, over 35 repetitions, best values in bold.

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, for the problem here, the quality of the found PFA depends more on the discrete, than on the continuous decision variables.

6.6 Conclusion

6.6.1 Summary

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. Considering the mixed-integer nature of the problem, this is 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. As such, the answer to whether local search can improve over the results found during global search (RQ3) is positive, but whether it is worth the effort in practice will be situation dependent.

6.6.2 Future Work

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, should also be investigated. 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, surrogate models (approximations of the objective functions that

can be cheaply evaluated) can be considered. For instance, surrogate models are used for the mixed-integer case in [68]. Use of the HVI gradient is possible in this case by taking the HVI gradient of the surrogate model, 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 [23] can lead to an optimisation strategy covering all levels of the building spatial design problem. Specifically, it would allow exploration with a free representation using co-evolutionary design simulation (as in e.g. [23, 52]), followed by global search with the evolutionary algorithm, and local search with the HVI gradient method when using the superstructure representation.

Finally, 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.