A novel encoding for separable large-scale multi-objective problems and its application to the optimisation of housing stock improvements

Alexander E. I. Brownlee[a], Jonathan A. Wright[b], Miaomiao He[b],
Timothy Lee[c], Paul McMenemy[d]

[a]*alexander.brownlee@stir.ac.uk, Division of Computing Science & Mathematics, University of Stirling, UK*
[b]*School of Architecture, Building & Civil Engineering, Loughborough University, UK*
[c]*School of Engineering and the Built Environment, Birmingham City University, UK*
[d]*Division of Computing Science & Mathematics, University of Stirling, UK*

Abstract

Large-scale optimisation problems, having thousands of decision variables, are difficult as they have vast search spaces and the objectives lack sensitivity to each decision variable. Metaheuristics work well for large-scale single-objective optimisation, but there has been little work for large-scale, multi-objective optimisation. We show that, for the special case problem where the objectives are each additively-separable in isolation and share the same separability, the problem is not separable when considering the objectives together. We define a problem with this property: optimisation of housing stock improvements, which seeks to distribute limited public investment to achieve the optimal reduction in the housing stock's energy demand. We then present a two-stage approach to encoding solutions for additively-separable, large-scale, multi-objective problems called Sequential Pareto Optimisation (SPO), which reformulates the global problem into a search over Pareto-optimal solutions for each sub-problem. SPO encoding is demonstrated for two popular MOEAs (NSGA-II and MOEA/D), and their relative performance is systematically analysed and explained using synthetic benchmark problems. We also show that reallocating seed solutions to the most appropriate sub-problems substantially improves the performance of MOEA/D, but overall NSGA-II still performs best. SPO outperforms a naive single-stage approach, in terms of the optimality of the solutions and the computational load, using both algorithms. SPO is then applied to a real-world housing stock optimisation problem with 4424 binary variables. SPO finds solutions that save 20% of the cost of seed solutions yet obtain the same reduction in energy consumption. We also show how application of different intervention types vary along the Pareto front as cost increases but energy use decreases; e.g., solid wall insulation replacing cavity wall insulation, and condensing boilers giving way to heat pumps. We conclude with proposals for how this approach may be extended to non-separable and many-objective problems.

1. Introduction

Large-scale global optimisation (LSGO) has attracted a lot of attention in recent years among evolutionary computation researchers, with varied real-world applications and diverse approaches appearing in the literature [1–5]. Much of the focus has been on single-objective LSGO, although recently there has been growing interest in multi-objective LSGO problems [6–10]. There is no universal definition of a LSGO problem, but typically one involves thousands or more decision variables. Such problems are difficult because of their correspondingly very large search spaces, and lack of sensitivity of the objectives to each decision variable.

The LSGO problem forming the focus of this paper is regional-scale building stock optimisation. This aims to find the best way to spend limited public funds on improving housing, to reduce energy demand and the corresponding emissions. Housing is responsible for more than a quarter of total energy consumption in the UK [11] and, given a typical house's lifespan, any significant improvements must focus on refurbishments to the existing stock [12]. We consider the multi-objective problem of identifying the trade-off between investment cost and resulting energy savings for the refurbishments of housing stock at the city or regional scale. The objectives for this problem are *additively separable*, meaning that the variables can be grouped such that dependencies exist within but not between the groups. An interesting property of housing stock optimisation is that both the cost and energy objectives have the same separability: the problem for each house (or group of houses if they share, for example, a district heating system) is independent of the others. This property is shared by employee scheduling for large companies [13], specifically where employees are scheduled at a regional level, and regions are independent of each other. For single-objective additively-separable problems, the search complexity can be reduced by considering sub-problems in isolation [14]; however, for multi-objective problems, we show that there is an additional complication: when considering the objectives together the problem is, in effect, non-separable where the objectives share the same additive separability. We term these types of problems *combinatorially-separable*.

This paper resolves the problem of low objective sensitivity, and the loss of separability for large-scale problems, by introducing a two-stage approach to encoding the solutions called Sequential Pareto Optimisation (SPO). The approach is applicable to the special case of multi-objective optimisation where each objective is additively-separable, and the objectives share the same separability (i.e., combinatorially-separable). In Stage 1, the Pareto-optimal solutions for each sub-problem are obtained. The solutions are sorted with respect to their values for one objective, providing a natural ordering for the second stage search.
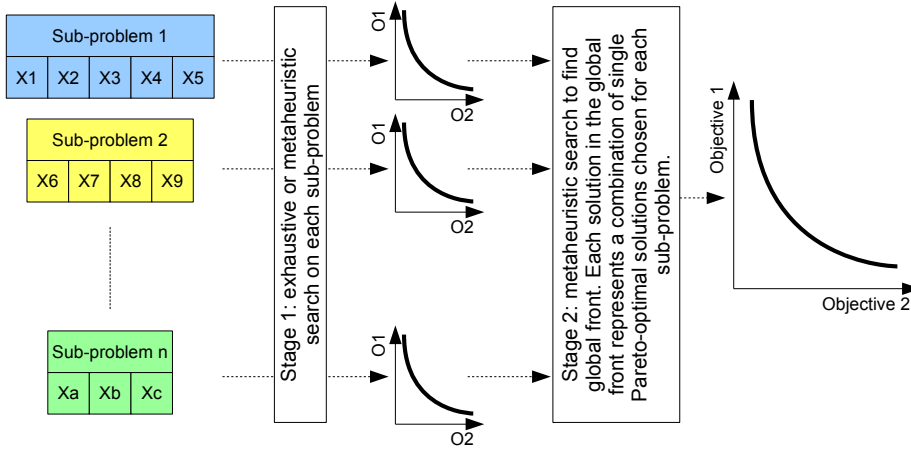
2

Figure 1: Overview of the SPO approach

In Stage 2, an approximation to the Pareto-optimal set for the global problem is found by searching different combinations of Pareto-optimal solutions to the sub-problems. Thus, SPO re-encodes the global problem into a search over Pareto-optimal solutions for each sub-problem. A high-level overview of SPO is given in Figure 1.

As well as improving overall search efficiency, SPO has several other strengths. The encoding is independent of the search algorithm used at each of the two stages, inducing adaptivity to the search spaces of different problems. Costly function evaluations can be conducted at Stage 1 and then cached for Stage 2, potentially making the large-scale global search faster. Sub-problem level constraints can also be considered at Stage 1 so that Stage 2 only searches across feasible solutions.

We demonstrate the SPO approach by application to the multi-objective building stock optimisation problem, using synthetic variants of this problem constructed by forming a stock of identical houses. SPO is shown to be robust and offers a substantial improvement over a naive encoding when using two different search algorithms. We also apply SPO to a real-world housing stock optimisation problem, with exploration of the insights that the approach can reveal to the decision maker. The results show that it is possible to find solutions savings of 20% of the cost, yet yielding the same reduction in energy consumption, compared to the solutions that seeded the search. We show how use of different interventions vary along the Pareto front as cost increases and energy use decreases: e.g., solid wall insulation replacing cavity wall insulation, and condensing boilers giving way to heat pumps. This thorough analysis extends previous preliminary results for a version of the problem that was smaller in scale but used more time-consuming energy modelling [15, 16].

This work makes several novel contributions:

(1) we show that additively-separable, single-objective problems can become

3

non-separable when combined into a multi-objective problem;

(2) a new Sequential Pareto Optimisation method for additively-separable, large-scale problems is described and demonstrated for the housing stock optimisation problem;

(3) the paper thoroughly compares and explains the relative performances of two well-known evolutionary algorithms (NSGA-II and MOEA/D) under a variety of configurations using the SPO encoding. Implementation and performance of the two algorithms in concert with SPO are discussed: in particular, we show that reallocating seed solutions to the most appropriate sub-problems substantially improves the performance of MOEA/D;

(4) this is the first approach that attempts to simultaneously optimise the refurbishment of numerous houses for multiple objectives, and as such is the first to potentially provide information on the trade-off between capital investment and energy demand reduction in decision-making at a regional or national level;

(5) the synthetic problems are new test problems developed here to demonstrate and examine the scalability of the SPO approach. No benchmarks currently exist that exhibit the combinatorially-separable property, where each objective shares the same additive-separability.

The rest of this paper is structured as follows: Section 2 discusses related work on large-scale optimisation problems. The problem is formally defined in Section 3, where issues around separability for multi-objective problems and two approaches to encoding (a naive approach and SPO) are outlined. The real-world problem of housing stock optimisation that motivates this work is described in Section 4. Experimental results and analysis of the naive and SPO approaches are presented in Sections 5 and 6.2, where the NSGA-II and MOEA/D are applied to a synthetic problem, with Section 6.2 presenting results for a much larger scale synthetic problem. Section 7 reports the results and analysis of an optimisation run performed on the real-world problem, with Section 8 drawing conclusions and proposing some possible directions for future work.

2. Related work

There has recently been increasing interest in large-scale optimisation among evolutionary computation researchers. This is highlighted in recent reviews [1, 2] and the release of large-scale benchmark problems [3–5]. Most published examples focus on problems with one to a few thousand variables, though there are some examples with millions of variables or more [17–20]. This paper presents synthetic problems with 4675 and 74 800 binary variables (numbers derived from the real-world problem), and a real-world problem with 4424 variables, placing these problems in the realm of large-scale optimisation.

4

As one might expect, the most successful approaches for large-scale problems exploit characteristics of the application used to improve search efficiency. These include features such as separability, known partial solutions, and constraints which reduce the search space [17–20]. Li et al. [21] describe an approach in which problems are divided into their separable and non-separable components: these are then searched using a two-level algorithm incorporating multiple surrogate fitness models. Li et al. [22] further explore the concept of non-separability in several popular benchmark functions. In Zhang et al. [9], the decision variables are divided into those related to convergence and those related to diversity, and are separately targeted by the algorithm. Manicassamy et al. [23] report a technique known as "gene suppression" for large-scale optimisation problems, the technique being a means of repairing infeasible solutions, and provide a demonstration on medium to large knapsack problem instances having $1000 – 20\,000$ objects and $2 – 5$ knapsacks. Cai et al. [24] combine a problem-specific greedy approach with particle swarm optimisation for large-scale social network clustering. Kabán et al. [25] propose an estimation of distribution algorithm for large-scale continuous global optimisation problems, using random projections to low dimensions to improve scalability. Other mathematical approaches have also recently shown some success [26].

Cooperative co-evolution [27, 28] has been used on several occasions to tackle large-scale problems, working with co-evolved sub-populations that tackle smaller-scale sub-problems. More recently, Yang et al. [29] and Omidvar et al. [30] developed the use of cooperative evolution, adaptively decomposing a large-scale problem with unknown separability into simpler sub-problems, and starting with randomly grouped variables. Li and Yao [31] built on these examples to develop cooperative co-evolving PSO, where the problem is divided into subcomponents, the size of which are co-evolved. This was applied successfully to problems with up to 2000 variables. Sun et al. [32] used cooperative co-evolution, with an extended differential grouping method to find separable groups of variables. The method was shown to work well with problems of 1000 variables from the CEC benchmarks. Cooperative co-evolution in combination with fitness inheritance has also been shown to perform well on problems with up to 1000 variables [33].

A two-stage approach to large-scale optimisation, different to the SPO method proposed here, is described by [34] and [35]. The idea is to locate a promising region of the search space in the first stage, then focus heavily on exploitation in the second. Adaptive differential evolution has also been found to work well on continuous optimisation problems with 1000 variables (e.g., [36]).

All of the above research focuses on single-objective problems:, however, there is some evidence [37, 38] that commonly-used evolutionary multi-objective algorithms do not scale well with the number of decision variables, emphasising the importance of techniques which can perform well with larger numbers of decision variables. As far as the authors are aware, there have been very few attempts (exceptions being studies by [6–9]) at multi-objective optimisation of problems with thousands of decision variables. SPO represents a novel approach to adapting existing multi-objective algorithms for large-scale problems. Furthermore,

while many of the above approaches to large-scale single-objective problems use some form of decomposition, these tend to be problem specific (though Antonio and Coello [7], Qiu et al. [39] use randomised grouping of variables with some success). An interesting method in [10] used a problem transformation scheme to reduce dimensionality of search space, whereby weights were applied to decision variables in groups and these were optimised to approximate the global Pareto front. Cao et al. [40] proposed a cooperative coevolutionary algorithm for large-scale multi-objective problems in which variables are grouped according to their class (e.g., by distance, position or mixed), but the techniques they used relied upon the decision variables being numeric (discrete or continuous) to allow exploration of their individual impacts. Furthermore, the approaches using cooperative coevolutionary methods require a large number of function evaluations (150 000 to 2 000 000 evaluations for 200–5000 variables). However, this is impractical for real-world problems such as the one we focus on, where fitness depends upon some model or simulation rather than a cheap benchmark function.

In contrast, the SPO approach is generic: as long as separable groups of decision variables can be identified, it will work. Furthermore, existing approaches do not consider how variables that are separable for a single objective become non-separable when considering the objectives together, because the trade-off involves balancing the contribution to each objective from each of the sub-problems. SPO overcomes this issue by determining the Pareto-optimal front for each sub-problem, then combining the best solutions from each sub-problem to generate optimal solutions for the global problem.

3. Problem definition and solution approaches

3.1. Combinatorially-separable problems

Table 1 lists the algebraic terms used in this section and throughout the rest of the paper. We consider an additively-separable, single-objective optimisation problem, with the aim of finding a solution (or configuration) $\boldsymbol{x} \in X$ that minimises some objective function F, where $\boldsymbol{x}$ is a vector of individual decision variables in the global search space X. .

There exists M independent nonlinear sub-problems G which partition both the global problem and the global search space, with each independent sub-problem denoted by $G(X_m)$. Thus the global problem can be expressed as an additively separable function [41]:

$$F(X) = \sum_{m=1}^{M} G(X_m).$$ (1)

The additively-separable property means that minimising $F(X)$ is simply a case of minimising each $G(X_m)$. This is particularly helpful in the case of large-scale optimisation problems, as the large search space can be dramatically reduced by consideration of only the sub-problems.

Variable	Explanation
X	The global search space of all variables x_j, $j \in \{1, 2, ..., J\}$
$|X|$	The cardinality of global search space, $|X| = J$
$F(X)$	Global optimisation problem defined over X
N	Number of objective functions across $F(X)$
$F_n(X)$	Individual objective function of $F(X)$, $n \in \{1, 2, ..., N\}$
$\boldsymbol{x}$	A particular configuration of X representing one solution to the problem
M	Number of sub-problems partitioning X
X_m	The independent partition of X covered by sub-problem $m \in \{1, \ldots, M\}$
X_{m_i}	A single decision variable in partition X_m, $i \in \{1, \ldots, |X_m|\}$
$G(X_m)$	Sub-problem of $F(X)$ defined over partition $X_m \subset X$
$G_n(X_m)$	Sub-problem of objective $F_n(X)$ defined over an independent partition $X_m \subset X$
$\mathcal{F}_m$	The Pareto-optimal solutions from sub-problem $G(X_m)$
Y	List of integers Y_m, each of which acts as a pointer to one member of each $\mathcal{F}_m$
Y_m	Index representing the chosen solution for sub-problem m
$\mathcal{Y}$	Global solution to $F(X)$ chosen from Y

Table 1: Definitions of variables used in SPO approach

Now consider a multi-objective optimisation problem (MOP), in which we seek to find the solution, or multiple Pareto solutions $\boldsymbol{x}$ (each $\boldsymbol{x}$ being a configuration of X), that minimises all N objectives F_n in F, where $n \in \{1 \ldots N\}$. When each objective F_n shares the same separability, i.e., F_n is a sum of sub-problems G_n, and each G_n applies to the same subsets X_m of X. This can be expressed as

$$F_n(X) = \sum_{m=1}^{M} G_n(X_m).$$
(2)

However, the situation is considerably more difficult for MOPs than for single-objective problems, even where each objective shares the same separability as above (which we term *combinatorially-separable*). When a MOP has conflicting objectives, multiple solutions will exist for each sub-problem $G(X_m)$, where $G(X_m)$ is the MOP across all $G_n(X_m)$.

The global problem seeks to apply a value to each of the variables $x_j \in X$, resulting in choosing exactly one solution for each sub-problem $G(X_m)$ (recalling that X is a concatenation of all X_m). It then follows that the global solution can be obtained by choosing one solution for each of the sub-problems such that, when combined together, they strike a balance that minimises the global objectives.

If the global objectives conflict with each other, this may result in finding a Pareto front in which each solution has a different combination of solutions to

the sub-problems. Since the global search space is the product of the Pareto-optimal fronts of all the sub-problems, the global Pareto-optimal front cannot be obtained by only adding the Pareto fronts for the sub-problems. Indeed, there is no clear concept of what "adding" the fronts for the sub-problems will result in.

This concept is illustrated using a 2-objective minimisation problem, with three additively-separable sub-problems. The encoding of the sub-problems is irrelevant here, all that matters is that they do not overlap in the variable space. For example, these might be non-overlapping groups of bits in a bit-string representation. For simplicity, we present three bi-objective sub-problems that, although they occupy different parts of the global search space, have resulted in identical objective values for their three sub-problem, Pareto-optimal solutions A, B and C:

A: $\forall m \in \{1, 2, 3\}$, $F_1(X_m) = 5$, and $F_2(X_m) = 1$, denoted as solution A(5,1);

B: $\forall m \in \{1, 2, 3\}$, $F_1(X_m) = 3$, and $F_2(X_m) = 4$, denoted as solution B(3,4);

C: $\forall m \in \{1, 2, 3\}$, $F_1(X_m) = 1$, and $F_2(X_m) = 5$, denoted as solution C(1,5).

Intuition might say that since all solutions to the sub-problems are Pareto-optimal, the global Pareto-optimal solutions can be formed as a linear combination of the sub-problem solutions, with each of the objective values separately formed according to Equation (1). That is, the global Pareto-optimal solutions are (AAA), (BBB) and (CCC), which would lie at (3,15), (9,12) and (15,3) in the objective space. We term these solutions *reference points*, and refer to the front formed in this manner as the *reference front*. However, when we consider the full set of possible combinations of the sub-problem Pareto fronts, we see that the reference points are not the global Pareto-optimal front. The full global search space of 27 is illustrated in Figure 2, "All points". Of the 27, there are only 10 unique solutions, the remaining 17 being duplicates.

Figure 2 illustrates that when all 10 unique combinations of solutions for the global problem are considered, the reference point at (9,12) – BBB – is dominated. Therefore, even in the specific case that all the sub-problems are identical, the Pareto-optimal front cannot simply be formed by a simple combination of all the sub-problem solutions. The global front is still made up of solutions that contain Pareto-optimal solutions to the sub-problems, but from different parts of each sub-problem's front. *It follows that considering the objectives together, the sub-problems are not independent of each other; i.e. the global problem is no longer additively separable. Rather, we call it combinatorially-separable.* This also means that while the reference front serves as a guide to the location of the global Pareto-optimal front, the two are not equivalent. Furthermore, as is far more likely to be the case in practice, if the Pareto fronts for the sub-problems are not identical, it would be impossible to construct a reference front in this manner. This is the case in the building stock problem that we consider as our application.

We now describe two possible approaches to encoding large-scale separable MOPs. Both approaches are independent of the search algorithm. For simplicity

Sub-problem solutions	$G_1(X)$	$G_2(X)$	RF	PF
AAA	15	3	1	1
AAB	13	6	0	1
AAC	11	7	0	1
ABA	13	6	0	1
ABB	11	9	0	0
ABC	9	10	0	1
ACA	11	7	0	1
ACB	9	10	0	1
ACC	7	11	0	1
BAA	13	6	0	1
BAB	11	9	0	0
BAC	9	10	0	1
BBA	11	9	0	0
BBB	9	12	1	0
BBC	7	13	0	0
BCA	9	10	0	1
BCB	7	13	0	0
BCC	5	14	0	1
CAA	11	7	0	1
CAB	9	10	0	1
CAC	7	11	0	1
CBA	9	10	0	1
CBB	7	13	0	0
CBC	5	14	0	1
CCA	7	11	0	1
CCB	5	14	0	1
CCC	3	15	1	1

Plot — objective space, horizontal axis $F_1(X)$, vertical axis $F_2(X)$. Legend: All points (×), Global Pareto-optimal front (○), Reference front (●).
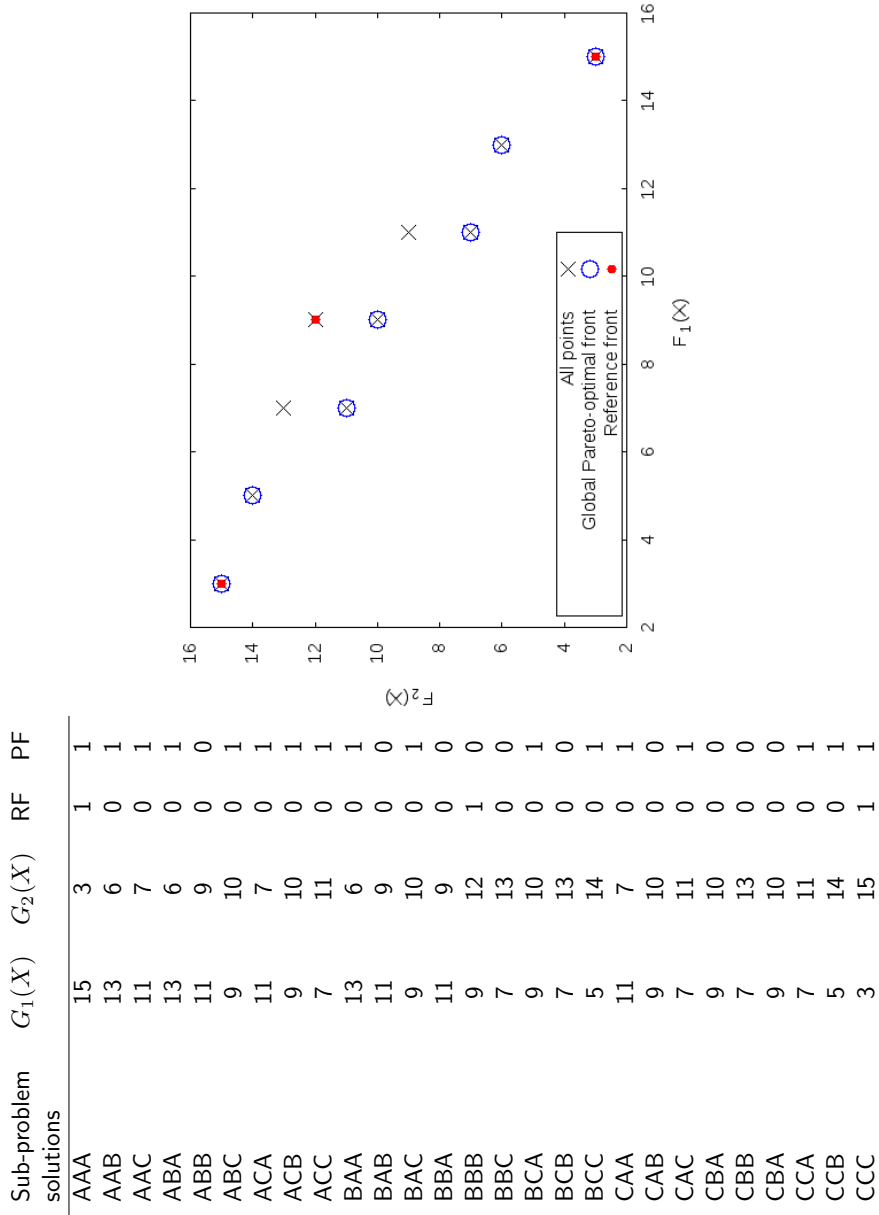
Figure 2: The identical Pareto-optimal fronts of three sub-problem solutions are combined by taking their product results in a new front. The new front dominates the "reference" front that was created by simply combining the solutions that lay in the same place in the original sub-problem fronts. The plot shows the solutions in objective space; the table on the right lists the combinations of the 3 sub-problem solutions A(5,1), B (3,4) and C(1,5) making up each global solution, with the resulting global objective values. Labels 'RF', 'PF' identify whether each solution is on the reference front and/or the global Pareto front. For example, the second row (solution AAB) provides global objective values of $G_1 = (5 + 5 + 3) = 13$ and $G_2 = (1 + 1 + 4) = 6$, which is on the global Pareto front but not the reference front.

9

of the representation, the examples consider the sub-problems to be over binary variables, but they are applicable to any representation. Note that only high-level pseudocode of the SPO process is given: we are not proposing new algorithms but simply an alternative way to formulate or encode combinatorially-separable large-scale problems.

3.2. Naive approach

One approach might be to consider the sub-problems together in a global optimisation: here called the naive approach. The overall encoding for the global problem is a concatenation of the bit-strings for all sub-problems into a single bit-string:

$$Y = X_1 \cup X_2 \cup \ldots \cup X_m \tag{3}$$

where each X_M is a list of variables $X_{m_i} \in \{0, 1\}$.

Figure 3b illustrates an example global problem comprising 4 sub-problems. Figure 3a shows the concatenated bit-string for an example global problem, where the naive encoding results in a global search over 17 bits, labelled x_1–x_{17}. While the naive approach is simple to implement, there are several obvious problems. Firstly, solutions returned by the optimisation process will almost certainly contain solutions for individual sub-problems that are not Pareto-optimal for that sub-problem. This is because the influence of a single variable on a single sub-problem will only make a small impact on the global objective values. As a result, there is unlikely to be enough selective pressure to drive the search towards the global optima for all sub-problems. Secondly, the lack of domain-specific neighbourhood structure to the search space could impact negatively on the performance of the optimisation algorithm. Thirdly, handling any mutually exclusive values for variables or other constraints in each sub-problem are very difficult to solve at the global scale. This is because feasible solutions for each sub-problem will need to be found in parallel with the search for the global Pareto-optimal front. Finally, it ignores the separability of the problem for each objective, substantially increasing the global search space to the product of the search space sizes for the sub-problems.

3.3. Sequential Pareto Optimisation (SPO)

An alternative approach is possible that exploits the fact that the sub-problems are independent (that is, in each objective, the sub-problems are *separable*). The problem is divided into two stages. At **Stage 1**, a search of the space for each sub-problem, $G(X_m)$, is performed in isolation, with the Pareto-optimal front, $\mathcal{F}_m$, determined for each sub-problem. The encoding for Stage 1 is problem-specific: a binary encoding is used in the example shown in Figure 3 but could, in principle, be anything. If the sub-problem is small enough, this search can be exhaustive, guaranteeing that the solutions for the sub-problems are optimal. Were the sub-problems to be larger, they could also be solved using metaheuristics or another suitable search method.

10

Global problem F(X)

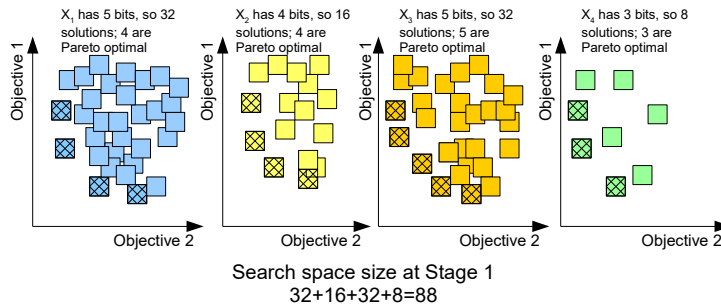| x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1

Search space size
2^{17}=131072

(a) Naive approach: exemplar, global problem encoded as a string of 17 bits

Sub-problem $G(X_1)$					$G(X_2)$				$G(X_3)$					$G(X_4)$		
X_{1_1}	X_{1_2}	X_{1_3}	X_{1_4}	X_{1_5}	X_{2_1}	X_{2_2}	X_{2_3}	X_{2_4}	X_{3_1}	X_{3_2}	X_{3_3}	X_{3_4}	X_{3_5}	X_{4_1}	X_{4_2}	X_{4_3}
0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1

2^5=32 2^4=16 2^5=32 2^3=8

(b) The variables X_{M_m}, applicable to each sub-problem $G(X_M)$, and their respective search space sizes

X_1 has 5 bits, so 32 solutions; 4 are Pareto optimal

X_2 has 4 bits, so 16 solutions; 4 are Pareto optimal

X_3 has 5 bits, so 32 solutions; 5 are Pareto optimal

X_4 has 3 bits, so 8 solutions; 3 are Pareto optimal

Search space size at Stage 1
32+16+32+8=88

(c) Stage 1: exhaustive search across each sub-problem to identify Pareto-optimal solutions, illustrated using hashed boxes which are then sorted by one of the objectives ready for Stage 2. Each of these solutions is a specific assignment of values to the sub-problem variables

Y_1	Y_2	Y_3	Y_4
1-4	1-4	1-5	1-3

Search space size at Stage 2
4x4x5x3=240
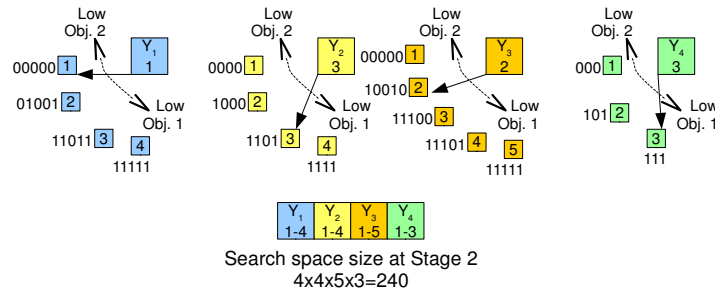
(d) Stage 2: one integer variable is an index identifying a specific Pareto-optimal solution for each sub-problem. The global solution is a concatenation of selected solutions for each sub-problem. Here, $Y = (1, 3, 2, 3)$ which corresponds to a solution $\boldsymbol{x} = (00000|1101|10010|111)$

Figure 3: An example optimisation problem over 17 binary variables, with the resultant encodings using the naive approach (Fig. 3a), then grouped into 4 sub-problems (3b), and SPO (3c, 3d) approach. Each sub-problem operates on a distinct sub-set of variables that partitions the global problem. With the naive approach, the variables for the sub-problems are simply concatenated together; with SPO, Stage 1 independently solves each sub-problem, and Stage 2 assigns a variable to indicate the choice of Pareto optimal solution for each sub-problem. The global solution is the concatenation of each selected solution for each the sub-problem

11

Data: $F(X)$, the global optimisation problem
Data: X, The global search space
Data: G, the subproblems of $F(X)$
Data: M, the number of subproblems partitioning X
foreach $m \in \{1, \ldots, M\}$ **do**   /* Stage 1: solve sub-problems */
    $\mathcal{F}_m \leftarrow \arg\min_{X_m} G(X_m)$ /* Pareto front of $G(X_m)$ */
    $\mathcal{F}_m = sort(\mathcal{F}_m)$ /* $\mathcal{F}_m$ ascending by Obj. 1 */
    $setDomain(Y_m) = \{1, 2, ..., |\mathcal{F}_m|\}$
end
$\mathcal{Y} \leftarrow \arg\min_Y F$ /* Stage 2: solve global problem */
return $\mathcal{Y}$

Algorithm 1: The overall SPO procedure

Stage 1 is illustrated in Figure 3c, where each $\mathcal{F}_m$ are shown as hashed boxes. Each $\mathcal{F}_m$ comprises several solutions for each sub-problem, $G(X_m)$, and each solution in each $\mathcal{F}_m$ is then assigned an index corresponding to its rank-order in $\mathcal{F}_m$ after sorting (see Figure 3d).

The encoding for the global problem is a vector of M integers (that is, one integer for each sub-problem):

$$Y = Y_1, \ldots, Y_M , \tag{4}$$

where each Y_m represents an index value pointing to a solution in the Pareto-optimal front for each sub-problem $G(X_m)$. Stage 2 can be seen as weighing up which balance of the objectives (i.e., which Pareto-optimal solution) for each sub-problem should be combined to build up the solution for the global problem.

SPO is summarised in Algorithm 1. SPO is illustrated for our 4-sub-problem example in Figure 3d. Each of the variables in the global problem ($Y1$–$Y4$) represent a "pointer" to one of the Pareto-optimal solutions for each sub-problem.

SPO has several advantages over the naive approach. Firstly, it reshapes the problem to be considered over only Pareto-optimal solutions for each sub-problem. This has the benefit that, once the optimisation run is complete, any individual sub-problem solution within the final set of solutions is approximately Pareto-optimal for that sub-problem. Pareto-optimality for the sub-problems can be guaranteed if the search at Stage 1 is exhaustive or uses an exact solver. This is particularly important for our application: a homeowner is unlikely to want a solution that is sub-optimal for their house. Secondly, the sorting of each $\mathcal{F}_m$ also provides an implicit neighbourhood relationship over the Pareto-optimal solutions for each sub-problem. These have a natural ordering in terms of the objectives that corresponds to their integer values: in Figure 3 incrementing the value for any Y_m will decrease objective 1's and simultaneously increase objective 2's value. The use of variation operators designed for manipulating ordered values such as integers or real numbers should allow the global space to be searched more efficiently than using the naive approach. Thirdly, constraints at the level of individual sub-problems can

be considered at the first stage, so the global search only considers feasible solutions. Fourthly, the overall search space for the global problem is decreased to the product of sizes of the Pareto-fronts for the sub-problems. Finally, the objective values obtained for the sub-problems at Stage 1 can be cached in memory, allowing the global search to be conducted using fast lookups to this cache rather than re-running potentially costly function evaluations.

4. Optimisation of the housing stock

We now turn to the application that motivates this work, and which will be used to illustrate the technique. Climate change is a significant issue facing society today. In addressing the problem, considerable effort is being made to reduce energy consumption and carbon emissions related to housing, which is responsible for more than a quarter of total energy consumption in the UK [11]. The long lifespan of a typical house means that it is important to invest in reducing the energy demand of existing housing. Improvements include improved insulation, new heating systems, or renewable energy systems like solar thermal and photovoltaic cells. Consequently, national and local authorities seek to maximise improvements to the housing stock using the limited financial resources at their disposal. This means that efficient optimisation methodologies for identifying the trade-off between refurbishment costs and the reduction in energy demand across the housing stock have become crucial.

However, even when houses in the stock are grouped into similar types and conditions, there can be thousands of groups, each associated with a range of refurbishment options. As such, the simultaneous optimisation of all houses results in a very large search space, and one in which the problem objectives have a very low sensitivity to changes in the refurbishment of a single house, the low sensitivity making the problem difficult to solve. Furthermore, while each objective is additively-separable between each house, when considering the multiple conflicting objectives together, the global problem covering all houses is not additively-separable (rather, combinatorially-separable). *In applied terms, if money is being spent to improve one house, it cannot be spent on another.* We now show how this problem maps to the generic large-scale optimisation problem described in the previous section.

4.1. Definition of the building stock optimisation problem

Following the definitions in Section 3, $G(X_0), \ldots, G(X_M)$ are sets of sub-problems, each of which corresponds to one house in the stock.

R is the set of all possible refurbishments that can be applied to any house and, for any given sub-problem, a subset $R_m \subset R$ comprises the possible refurbishments that can be applied. For example, houses with solid walls will not be suitable for cavity wall insulation, and houses with newly installed heating systems will not gain any benefit from a boiler replacement.

The naive approach considers all the decisions for potential refurbishments across all houses together. Domain-specific knowledge introduces a subtle level

13

of structure into the search space, in that more refurbishments (that is, more variables are set to true) means increased cost and lower energy. As noted in Section 3.2, constraints are more difficult to solve with this approach. Thus to be able to handle the mutually exclusive options for each house either needs a variation on the encoding (using categorical values for each of the options), or a global constraint measure. Furthermore, if any additional constraints are placed on the search (for example, a threshold on estimated comfort for a house's occupants, or a limit on expenditure for refurbishments on any one house), these will prove very difficult for the global search to solve. This is because there would be a large number of these constraints (assuming one per house), each of which would by insensitive to most of the variables in the global problem.

In Stage 1 of SPO, the search is conducted for each sub-problem. Constraints such as mutually-exclusive refurbishments or summer over-heating risk are considered at this stage, so that only solutions meeting the constraints are passed to the next stage as members of a Pareto-optimal front, $\mathcal{F}_m$. Once the Pareto-front for each house is found, the global search explores the trade-off of investments between the houses in the stock. Furthermore, the number of potentially long-running simulations to compute energy consumption and summer over-heating is reduced: for the example in Figure 3, from a naive approach search space of 131 072 to 88 evaluations at Stage 1 for SPO.

4.2. Naive and SPO encodings

It may be helpful to explain the two approaches in the context of this application. For the naive approach, each solution is represented by a bit-string where each bit represents a decision about whether to apply a particular refurbishment to a particular house. For SPO, at Stage 1, each sub-problem corresponds to one house. Again, the solutions for the sub-problems are bit-strings where each bit represents a decision about whether to apply a particular refurbishment to that house. Each solution applying values to these bits is a set of refurbishments, a particular strategy for improving the house. The difference for SPO is that the sub-problem for each house is solved in isolation to all the other sub-problems. The Pareto-optimal strategies for each house are then assigned integer IDs. Solutions for Stage 2 are a set of integers, one per house, to choose the strategy for each house.

4.3. Seeding

It is well known [42–44] that seeding the initial population with known good solutions can improve the performance of metaheuristics. For both naive and SPO approaches we can generate seeds that approximate the trade-off between the objectives. In all runs, we can ensure that at least two solutions are present in the population: those that minimise one objective and form the ends of the global Pareto-optimal front. This can be done by combining the minimal solutions for each objective in turn across all the sub-problems. The ease with which these can be identified is dependent on the problem and its formulation. In the case of the housing stock optimisation problem, the minimum cost solution

Data: *Count*, number of seeds required
Data: M, the number of sub-problems, $G(X_m)$
Result: S, set of seeds generated;
$S \leftarrow \emptyset$;
for $c = 1$ **to** *count* **do**   /* generate *Count* seeds */
    $\sigma = c/(Count + 1)$
    $Y = $ newSolution   /* instantiate new seed */
    for $m = 1$ **to** M **do**   /* traverse all sub-problems */
        $Y_m = round(\sigma * |\mathcal{F}_m|)$ /* round to nearest integer */
        $Y \leftarrow (Y, Y_m)$ /* concatenate solutions */
    end
    $S \leftarrow S \cup Y$ /* add to set of seeds */
end

Algorithm 2: Method for generating seeds with SPO

corresponds to the base-case (no-refurbishment) scenario, while the maximum cost solutions can be determined from inspection of the housing stock model (and corresponds to a scenario close to all possible refurbishment options being applied to a particular house – with only the poorer of mutually exclusive options excluded). The method of generating seed solutions between these points then differs depending on the optimisation approach taken.

For the naive approach, we can use the domain specific knowledge that there is a reasonably close relationship between the number of bits set true and the cost objective. This means that seed solutions can be generated by varying the rate at which bits in the solutions are set true. Rather than the conventional initialisation method of setting 50% of bits true, some solutions are generated with 10%, some with 20% and so on. This will produce solutions that ought to represent increasing refurbishment cost and decreasing total energy use, since, in this problem, the higher the bit-count the more refurbishment options are applied, and so the higher the cost. As this method is domain-specific it would not necessarily apply to another problem.

In contrast, with SPO it is possible to devise a seeding method that spreads the seeds evenly along the reference front between the objectives. This approach is formally presented in Algorithm 2. Recall that in the second stage of the problem, each variable Y_m takes an integer in range 0 to $|\mathcal{F}_m|$, a pointer into the sorted set of Pareto-optimal solutions $\mathcal{F}_m$ for each sub-problem $G(X_m)$. To generate a seed, we start with a value σ, where $0 < \sigma < 1$. We use σ to choose a value for each Y_m from approximately the same part of the Pareto-optimal front for each $G(X_m)$. In this manner, we produce solutions that ought to lie near the global Pareto-optimal front.

5. Experiments with a synthetic problem

In this section, we compare SPO and the naive approach, with different seeding schemes and algorithms for the second stage global optimisation. In
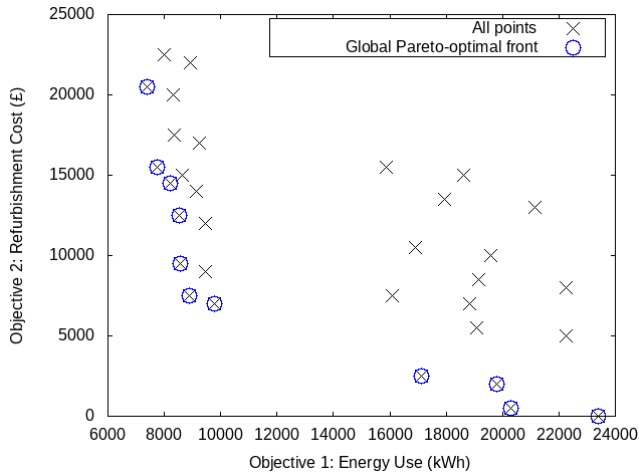
Figure 4: The exhaustive set of 32 solutions for the base sub-problem, which is duplicated 935 times to create the synthetic problem. Pareto-optimal solutions for the sub-problem are highlighted with blue circles.

order to test and compare the approaches, we have developed a synthetic version of the stock optimisation problem consisting of identical sub-problems (each sub-problem being one house). This was formulated by duplicating a single sub-problem 935 times (the full housing stock problem studied later has 935 houses). This sub-problem has 5 binary variables, so the global problem has 4675 binary variables in total. The Pareto-optimal front of solutions for the sub-problem contains 11 solutions. Figure 4 shows the set of all 32 possible solutions for the sub-problem plotted in the objective space, with the 11 Pareto-optimal solutions highlighted. To allow for simpler analysis at this stage, all 32 solutions for the synthetic problem are feasible. For reproducibility, the solutions are detailed in full in the Appendix. Constructing a problem using identical sub-problems means that we can approximate the region of the global Pareto front by scaling up objective values for the sub-problem (multiplying each by 935) to create a reference front in the same way as in Section 3. As noted in Section 3, the true Pareto-optimal front for the global problem will not perfectly match this reference front. It will almost certainly have many more solutions between the members of the reference front, as well as some which may dominate those in the reference front. However, since some of the solutions at least, will be globally Pareto-optimal, we use it here as a reference against which to judge the convergence of the second stage of the SPO procedure.

5.1. Experiments

SPO is an algorithm-independent approach to encoding separable large-scale multi-objective problems. As such, we compare SPO with the baseline of a

16

naive encoding, using two different algorithms to demonstrate the improvement it offers with both.

Two well-established and popular evolutionary algorithms are used as the basis for implementing the approaches: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [45] and the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [46]. NSGA-II selects solutions for reproduction according to their dominance of other members of the population, using the "crowding distance" to give preference to solutions in sparsely populated parts of the objective space. The algorithm is strongly elitist, keeping only the best solutions in the union of offspring and parents from one generation to the next. It has been found to perform well in a wide range of application areas. MOEA/D decomposes the multi-objective problem into many single-objective problems, with different members of the population allocated to different sub-problems. When generating a new population, solutions can be restricted to be only recombined with others that have similar sub-problems. It has also been applied to a wide range of problems, with particular success in many-objective problems (as the number of objectives increases, dominance becomes less effective for distinguishing them).

Each algorithm was applied to the synthetic problem using both the naive and SPO configurations. For the naive encoding, the algorithms were applied to a search over a space of solutions represented by 4675 bits. For SPO, Stage 1 was carried out using an exhaustive search on the 32 solutions for each sub-problem, so that the Pareto fronts for each sub-problem passed to Stage 2 were guaranteed to be optimal. The algorithm (NSGA-II or MOEA/D) then searched over the Stage 2 problem, where each solution comprised 935 integers. Each integer represents a choice of one solution from the Pareto front for the corresponding sub-problem.

The initial population for all runs was seeded with the minimal and maximal cost solutions. For runs identified as "seeded", we also included 9 solutions between these extremes, generated by the methods described in Section 4.3. An odd number was chosen so that one of the seeds lies at approximately the midpoint of the trade-off. All remaining solutions in the initial populations were generated by drawing random values for the variables from a uniform distribution. Overall this totals eight experimental runs (all combinations shown in Table 3). Each run was repeated for 30 independent trials using different random number generator seeds.

The hypervolume measure [47] was computed for the Pareto front output by each separate run. This is the volume of the space enclosed by the Pareto-optimal solutions and a reference point, which in these experiments was located at $(21\,878\,155, 21\,037\,500)$, the maximal values for the objectives in the base sub-problem multiplied by 935. Objective values in the Pareto-optimal fronts were normalised to between 0 and 1, using the reference point as an upper bound and the point with $(6\,913\,350, 0)$ as a lower bound (both being the minimal values for the sub-problem multiplied by 935). Hypervolume provides a balance between measuring multiple traits of a Pareto front: convergence to the true front, solution spread, and extent of the front. Higher hypervolume values are

Parameter	Naive NSGA-II	Naive MOEA/D	SPO NSGA-II	SPO MOEA/D
Population size	468	108	495	129
Tournament size	10	n/a	8	n/a
Mutation rate	0.01	$0.02/n$	$0.01/n$	$0.27/n$
Poly. real n_m	n/a	n/a	98.83	10.64
Crossover rate	0.30	0.29	0.70	0.47
C gene rate	0.54	0.14	0.71	0.81
C SBX n_c	n/a	n/a	0.98	0.09
Max evaluations	500 000	500 000	500 000	500 000

Table 2: Algorithm parameters, as chosen by SMAC runs with each algorithm variant

desirable.

Besides the approach taken to the global problem and the presence of seeding, other implementation details for the algorithms were as follows. Selection of parents for recombination and variation was as in the original papers describing each algorithm [45, 46]. NSGA-II used tournament selection to choose pairs of solutions. MOEA/D used the immediate neighbourhood of solutions (that is, solutions representing neighbouring decomposed sub-problems). Runs were terminated after 500 000 unique solutions had been evaluated. Algorithms using the naive approach with binary encoded solutions had uniform crossover [48] for recombination and bit-flip mutation for variation. Algorithms using SPO with integer encoded solutions had simulated binary (SBX) crossover [49] and polynomial real mutation [50]. Recombination was applied to a given pair of solutions according to a fixed probability. Variation was applied to all solutions generated, with a fixed probability on each variable.

So that the comparisons were fair, the parameters for each algorithm configuration were tuned using Sequential Model-based Algorithm Configuration (SMAC) [51], aiming to maximise hypervolume, with each algorithm limited to 30 000 evaluations. SMAC uses models based on random forests to estimate likely good configurations. These models are refined as more configurations are tested, until no further improvement can be found or a cap on computational time is reached. In our experiments, the reference Java implementation[1] was used. The resulting parameters are given in Table 2. The 'mutation rate' is the probability of a single variable being changed. The 'crossover rate' is the probability that a pair of solutions have crossover applied. 'C gene rate' is the probability that, when crossover is being applied, a single variable is crossed (either swapped for binary encoding or blended for integers using SBX). 'Poly. real n_m' and 'C SBX n_c' are the distribution indices for polynomial real and SBX crossover respectively.

---

[1]http://www.cs.ubc.ca/labs/beta/Projects/SMAC/

	Naive NSGA-II	Naive MOEA/D	SPO NSGA-II	SPO MOEA/D
Unseeded	$0.7661_{0.0008}$	$0.5177_{0.0074}$	$0.8039_{0.0004}$	$0.7505_{0.0021}$
Seeded	$0.7682_{0.0009}$	$0.5920_{0.0038}$	$0.8122_{0.0001}$	$0.7580_{0.0023}$

Table 3: Hypervolumes for synthetic problem. These are means over 30 independent runs, with standard deviations given in subscript. Higher values are preferable.

5.2. Results

Table 3 gives hypervolumes for the eight algorithm configurations. An ANOVA applied to the hypervolumes for each algorithm found a p-value of < 0.001, rejecting the null hypothesis that there is no statistically significant difference between the groups. A post-hoc Tukey's HSD Test [52] was performed on these results to test for significance in the difference of the means between the configurations. It returned a p-value of < 0.001 for all pairs (rejecting the null hypothesis that there is no statistically significant difference), with the exception of Naive NSGA-II unseeded vs. Naive NSGA-II seeded, for which $p = 0.193$.

Figure 5 shows summary attainment curves [53] for the different algorithm configurations. For each, the median attainment curve (that is, the front in the objective space reached by at least half of the repeat runs of the algorithm) is shown as a solid or dashed line. Around this line is a shaded area in the same colour, bounded by the minimal and maximal attainment curves (reached by only one repeat run and by all repeat runs respectively). These are intended to illustrate the spread of results found by each algorithm. The straight lines that are parallel to the axes in the fronts for MOEA/D represent gaps in the fronts returned by each run, leaving large parts of the objective space unreached. It is also worth noting that at the minimal O1 (energy) / maximal O2 (cost) end of the seed solutions, the SPO seeds dominate the naive seeds. This is because, as illustrated in Figure 4, the minimal energy solution is not the maximal cost "do everything" scenario. Some of the refurbishments conflict, resulting in increased energy use. In contrast, both sets of seeds have a common minimal O2 (cost) solution, corresponding to the minimal O2 (cost) / maximal O1 (energy) solution in the sub-problem. We can also see that the seeds for the naive approach are inferior to those for SPO. This is because the combination of sub-problems into one global problem means that the variables set true are evenly distributed across the sub-problems. The solution to each sub-problem may be at a different point in its own search space: some will be low-O1 (energy), some low-O2 (cost), and this will average out across the global problem, minimising neither. The scheme to generate seeds for SPO ensures that the solutions to all sub-problems are at a similar point within their sub-problem level Pareto-optimal fronts, improving the overall values at the global level for each seed.

There are three clear conclusions from these results. Firstly, SPO offers a marked improvement over the naive approach in the Pareto-optimal fronts obtained. Secondly, seeding the initial population is also crucial to obtaining
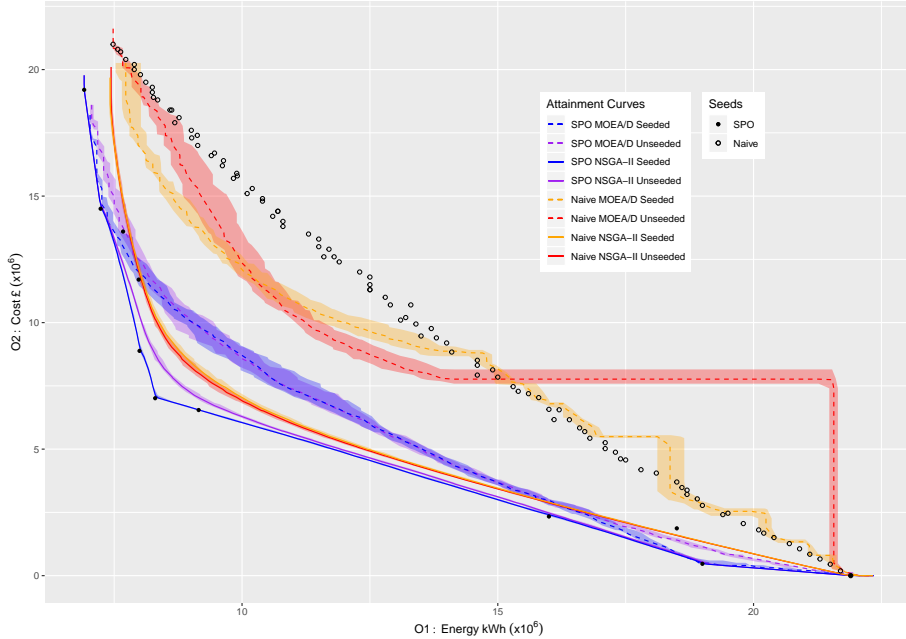
19

Figure 5: Summary attainment curves for the eight algorithm configurations on the synthetic problem. The dashed/solid lines represent the median attainment curve (the region of objective space reached by half of the runs). The shaded area around each line shows the variation in the runs: the upper extent of shading being the front reached by all runs, and the lower extent of shading being the front reached by only one of the runs. Also shown are the seed solutions for both approaches.

a good set of final solutions, particularly for NSGA-II. The seeds for SPO also appear to be a better starting point for the search. Thirdly, it would also appear that NSGA-II outperforms MOEA/D for this problem, and produces more consistent results. The first two conclusions are as we would expect. The reduced overall search space and re-encoding mean that searches using SPO are more efficient. The seed solutions also give any search algorithm a head start, with a population that is closer to the true Pareto-optimal front. It is, perhaps, unexpected that MOEA/D shows such poor performance with respect to NSGA-II, when it has proven highly competitive for many other problems [46, 54, 55]. In the next section, we consider what causes the poor performance of MOEA/D relative to NSGA-II in more detail.

5.3. Improving the performance of MOEA/D

To understand why NSGA-II performs better than MOEA/D, we consider what drives the search in both algorithms. MOEA/D decomposes the multi-objective problem into scalar (single objective) problems, each a weighted sum of the objectives, distributed between extremes minimising only one objective.
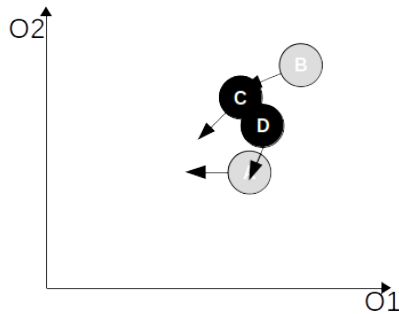
Figure 6: Recombination of a seed and a randomly generated solution.

Each member of the population is associated with one scalar problem. New solutions are generated by recombining neighbouring solutions (those having a small Euclidean distance between their weight vectors), and are retained if they improve their scalar problem. This does not necessarily mean that they dominate the solutions that they replace, and no test for dominance is made. Overall, this spreads the solutions evenly along the trade-off between the objectives, and works well for problems with many objectives where most solutions tend to be non-dominated. In contrast, NSGA-II generates new solutions by recombining solutions from throughout the population, with a bias towards non-dominated and uncrowded solutions. Non-dominated sorting is applied to the combination of the population and the newly generated solutions, with only the top-ranked half being retained for the next generation. This rapidly pushes the population towards consisting entirely of non-dominated solutions, and only solutions which dominate part of the existing population are likely to be kept.

Two characteristics of MOEA/D are of relevance: the lack of dominance checking, and the allocation of the initial solutions to scalar problems. To explain these further, we consider several scenarios for MOEA/D, illustrated in Figures 6 and 7. Solutions are rendered as black circles, plotted in the objective space (the objectives being $O1$ and $O2$ for simplicity). The scalar problem for each solution is represented by an arrow showing the weights applying to both objectives. Solution A has been assigned to the scalar problem with weights $(O1 = 1, O2 = 0)$, B to $(O1 = 0.75, O2 = 0.25)$, C to $(O1 = 0.5, O2 = 0.5)$, D to $(O1 = 0.25, O2 = 0.75)$ and E to $(O1 = 0, O2 = 1)$.

Lack of Dominance checking. In Figure 6, seed solutions A and B have been recombined to generate solutions C and D. Solution C has a lower value for X, and so takes the place of solution A for the scalar problem $(O1 = 1, O2 = 0)$. Solution D offers an improvement in both objectives over B, so takes the place of B for the scalar problem $(O1 = 0.75, O2 = 0.25)$. This means that the seed A would be lost, despite not being dominated by either of the two newly created solutions. In contrast, NSGA-II would keep A and C, the two non-dominated solutions of the four. Zhang and Li [46] do note that an external population

21

(a) Random starting population.

(b) Population after several generations.

(c) Seeded starting population, randomly allocated to scalar problems.

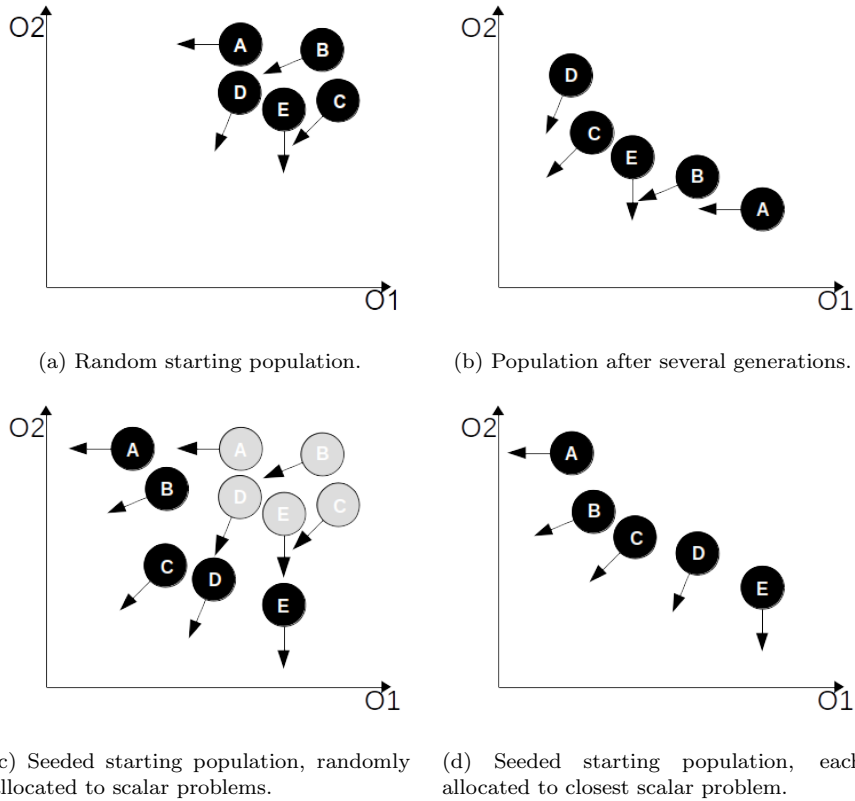(d) Seeded starting population, each allocated to closest scalar problem.

Figure 7: Placement of seed and randomly generated solutions for MOEA/D.

could be maintained to keep non-dominated solutions resulting from the whole search. This would prevent the seed solution being lost from the final output of the search but, without a mechanism to reinsert a solution from the external population into the main population, the search will still be impeded by the loss of A from the main population.

Allocation of the initial solutions to scalar problems. Figure 7a shows a randomly generated population, with solutions assigned to scalar problems at random. After many algorithm iterations, the solutions will spread out towards the global Pareto-optimal front (Figure 7b). Figure 7c shows an initial population of seed solutions, allocated to scalar problems at random. The seeds extend over a wide area of the objective space, so some solutions must move a long distance to reach the part of the Pareto-optimal front that minimises their scalar problems. For a large-scale problem like building stock optimisation this movement consumes many iterations of the algorithm because changing a single variable has a tiny impact on the objectives. Relocating the seed solutions so that they are allocated to more appropriate scalar problems (Figure 7d) will

22

reduce the need for this movement, allowing the algorithm to immediately start improving on the seeds. This can be achieved by allocating to each scalar problem the seed which minimises the weighted sum of objectives for it, which we will consider later in this section.

Effect on the synthetic problem. Figure 8 illustrates the population of one run each of MOEA/D and NSGA-II on the synthetic problem. The figure shows the initial population, and that after the first and tenth generations for both algorithms, plotted in the objective space[2]. Black points show the current population, with red points for previous generations (lighter red being older). There are more points for NSGA-II because of the larger population size following parameter tuning. The seeds are spread out along a trade-off between the two objectives, and the randomly generated solutions are clustered in one region, dominated by the seed solutions. Due to the lack of dominance checking when generating new offspring, many of the seeds are lost by MOEA/D and the population as a whole reduces in optimality within the first ten generations. In contrast, while NSGA-II keeps some of the randomly generated solutions, many are lost in the first two generations because they are dominated by the seeds, which are all kept. NSGA-II proceeds to fill the gaps between the seeds, and further optimise the front as solutions which dominate them are found.

An obvious solution to the loss of seed solutions by MOEA/D is to use only seeds for the initial population. MOEA/D was re-run with an initial population comprising only 129 seed solutions, again stopping after 10 iterations. The seeds were duplicates of the 11 used previously, created using the method in Section 4.3. Figure 9 shows the population after the first and tenth generations as before; the initial population is omitted because it appears identical to Figure 8a. The population now remains closer to the seeds because there is no draw towards randomly generated solutions, but there is still some loss of non-dominated solutions. This is because of the other issue discussed above, random allocation of the seeds to scalar problems means that iterations are wasted as seeds move towards the "correct" locations. The result is that by the tenth generation, solutions at the extremes of the front have been lost (red rather than black), and several solutions in the middle of the front lie at positions dominated by the seeds.

Having performed this analysis, we reran the 30 repeats of SPO MOEA/D using two modifications. The initial population for each run comprised 129 seed solutions, and no randomly generated solutions (leaving the only remaining use of random numbers being the variation and recombination operations). The seeds were relocated to the scalar problem that they achieved the lowest value for. The results from these runs were compared with those of NSGA-II and the original version.

Table 4 gives the mean hypervolume over the 30 runs, with the figures for

---

[2]Animations showing the convergence over the full run of both algorithms are available from
`http://www.cs.stir.ac.uk/~sbr/stockopt/`

(a) MOEA/D initial generation. (b) NSGA-II initial generation.

(c) MOEA/D after generation 1. (d) NSGA-II after generation 1.

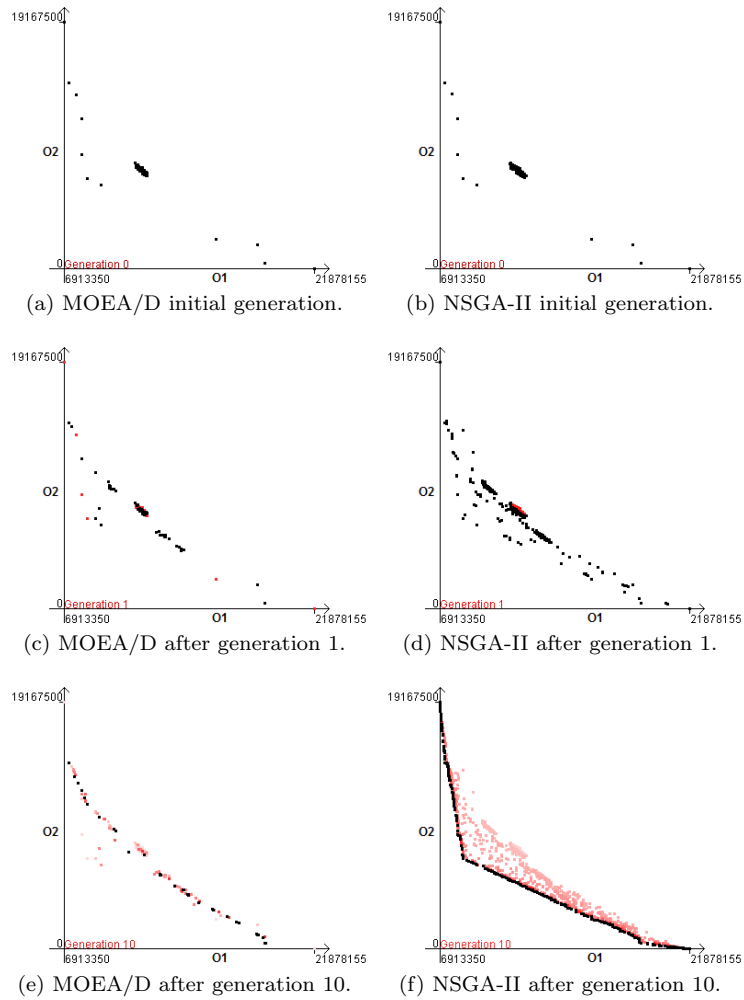(e) MOEA/D after generation 10. (f) NSGA-II after generation 10.

Figure 8: Early stages of the search for MOEA/D and NSGA-II, showing that MOEA/D spreads evenly among the population and NSGA-II has a strong bias towards the non-dominated solutions. Black points are the current generation, red shaded points are all previous generations.
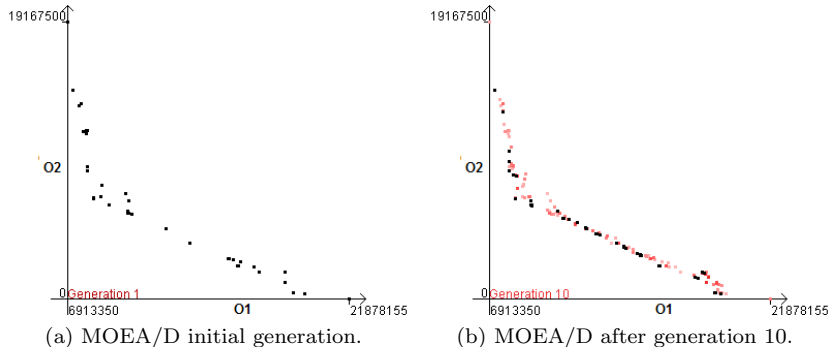
24

(a) MOEA/D initial generation.

(b) MOEA/D after generation 10.

Figure 9: Early stages of the search for MOEA/D, having used only seed solutions for the initial population. Less ground is lost because there are no randomly generated solutions to pull the whole population back, but progress is still less rapid than for NSGA-II, as MOEA/D tries to better distribute solutions along the Pareto-optimal front. Note, the red points in generation 1 are completely covered by black points, because all seeds have survived to the second generation.

Variant	Hypervolume
NSGA-II	$0.8122_{0.0001}$
Original MOEA/D	$0.7580_{0.0023}$
MOEA/D with 129 seeds	$0.8064_{0.0005}$
MOEA/D with 129 seeds, and relocation	$0.8040_{0.0003}$

Table 4: Hypervolumes for variations of MOEA/D on the synthetic problem, with previous results for seeded SPO MOEA/D and NSGA-II reproduced for comparison. As in Table 3 These are means over 30 independent runs, with standard deviations given in subscript. Higher values are preferable.

the original MOEA/D and NSGA-II reproduced for convenience. An ANOVA found $p < 0.001$; Tukey's HSD test for significance on the differences in these means found $p < 0.001$ for all pairs of algorithms. Thus, we can reject the null hypothesis that there is no statistically significant difference between the groups. Figure 10 shows the summary attainment curves for the revised version of MOEA/D. It is clear that only using seeds for the initial population makes a substantial difference to the performance of the algorithm. The amended approach produces fronts much closer to those coming from NSGA-II, even improving on them slightly around the knee point. These observations on the use of seed solutions should be noted when using MOEA/D for other applications. However, for this specific problem, there is still greater variation in the fronts produced by MOEA/D than for NSGA-II. They are mostly dominated by those from NSGA-II, resulting in slightly, but still significantly, lower hypervolumes. For these reasons, we will use SPO NSGA-II with seeding for application to the larger scale and full real-world problems in the rest of the paper. It should be noted that a recent development, MOEA/DD [56], incorporates dominance and
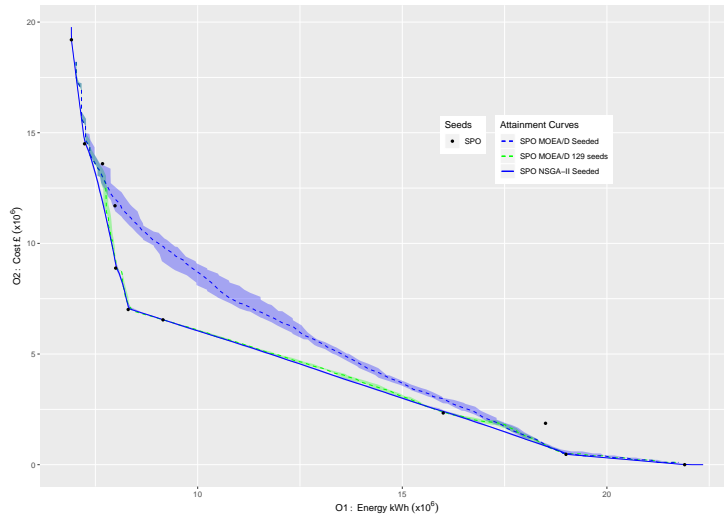
Figure 10: Summary attainment curves for the variants of MOEA/D, with those for the original MOEA/D and NSGA-II copied for reference. As before, the dashed/solid lines represent the median attainment curve (the region of objective space reached by half of the runs). The shaded area around each line shows the variation in the runs: the upper extent of shading being the front reached by all runs, and the lower extent of shading being the front reached by only one of the runs. Also shown are the seed solutions for SPO.

decomposition in the MOEA/D framework and may resolve the issues discussed above. However, the primary focus of this study is on the difference between the naive approach and SPO, so this is not considered further here.

6. Further experiments with synthetic problems

6.1. Additional synthetic problems

To draw more general conclusions, we repeat the experiment from the previous section by generating 10 further synthetic problems using exemplar houses in the CHM data. As before, the sub-problem representing one house is duplicated 935 times to create the global problem. The specific houses are identified in Table 5. For each of the problems, the duplicated sub-problem has 5 binary variables (i.e., 32 variants), so the global problem has 4675 binary variables in total. The Pareto-optimal front of solutions for the sub-problem vary from 6 to 18 solutions. These problems are not replicated in full in the paper, but are available at the URL given at the end of the paper. The lower and upper bounds (min and max values found over all runs) used for normalising the objectives prior to computing hypervolume were (1 770 728, 26 993 766) for energy and (0, 27 712 000) for cost.

The algorithms were applied to each problem without further tuning. Algorithm parameters were as given in Table 2; but the improved version of MOEA/D from

26

Problem	House ID	Solutions in Pareto-front for 1 house
1	G1531103	15
2	G1543406	6
3	H1454110	8
4	H1461214	9
5	H1463202	12
6	H1464204	18
7	H1493210	16
8	H1494209	10
9	H1494401	14
10	H1494409	13

Table 5: Base houses used for additional 10 synthetic problems.

Section 5 was used, with 129 seeds and relocation. The mean hypervolumes of 30 runs of each algorithm on each problem are given in Table 6.

As before, an ANOVA applied to the hypervolumes for each algorithm found a p-value of < 0.001, rejecting the null hypothesis that there is no statistically significant difference between the groups for all 10 problems. A post-hoc Tukey's HSD Test was performed on these results to test for significance in the difference of the means between the configurations. For each problem, it returned a p-value of < 0.001 for all pairs (rejecting the null hypothesis that there is no statistically significant difference), with the exception of the groups of algorithms identified by * and ⁺.

For all problems, NSGA-II with SPO and seeding was found to return the fronts with the highest hypervolumes, though occasionally MOEA/D with SPO and seeding produced hypervolumes that were not significantly different to NSGA-II. In all cases, seeding improved every algorithm variant, and SPO outperformed the naive encoding.

6.2. Larger-scale synthetic problem

In order to test how well our approach scales to larger problems, we scaled up the synthetic problem from Section 5. As for the smaller version of the problem, a representative sub-problem[3] was selected. This has eight applicable refurbishment options, two of which are mutually exclusive. Thus each sub-problem has 128 possible solutions, of which 18 are non-dominated. The sub-problem was duplicated 9350 times, ten times the number in the previously examined problem, resulting in a global optimisation problem over 74 800 binary variables.

NSGA-II with the SPO encoding and seeds were applied to this problem, using the same parameters as in Table 2. Strictly speaking, the parameters

_____

[3]Specifically house ID G1461110 in the Cambridge Housing Model [57]

Problem	Naive NSGA-II		Naive MOEA/D		2-stage NSGA-II		2-stage MOEA/D	
	unseeded	seeded	unseeded	seeded	unseeded	seeded	unseeded	seeded
1	$0.7512_{0.0022}$	$0.8071_{0.0002}$	$0.6851_{0.0023}$	$0.7539_{0.0009}$	$0.8163_{0.0001}$ *	**$0.8184_{0.0000}$** *	$0.7900_{0.0030}$	$0.8159_{0.0002}$ *
2	$0.9281_{0.0025}$	$0.9448_{0.0022}$	$0.8365_{0.0031}$	$0.8871_{0.0017}$	$0.9868_{0.0001}$ *	**$0.9875_{0.0000}$** *	$0.9595_{0.0018}$	$0.9864_{0.0001}$ *
3	$0.4991_{0.0065}$	$0.6331_{0.0002}$	$0.3578_{0.0066}$	$0.5219_{0.0025}$	$0.6420_{0.0001}$ *	**$0.6433_{0.0001}$** *✚	$0.6262_{0.0013}$	$0.6404_{0.0002}$ *✚
4	$0.7355_{0.0023}$	$0.7961_{0.0001}$	$0.6289_{0.0026}$	$0.7126_{0.0010}$	$0.8015_{0.0001}$ *	**$0.8030_{0.0000}$** *	$0.7637_{0.0024}$	$0.7996_{0.0001}$ *
5	$0.5896_{0.0042}$	$0.6720_{0.0002}$	$0.4918_{0.0034}$	$0.5955_{0.0015}$	$0.6865_{0.0001}$ *	**$0.6890_{0.0000}$** *	$0.6595_{0.0018}$	$0.6863_{0.0002}$ *
6	$0.6305_{0.0038}$	$0.7167_{0.0001}$	$0.5596_{0.0040}$	$0.6617_{0.0011}$	$0.7305_{0.0001}$ *	**$0.7334_{0.0000}$** *	$0.7060_{0.0020}$	$0.7314_{0.0002}$ *
7	$0.6659_{0.0030}$	$0.7486_{0.0003}$	$0.5954_{0.0043}$	$0.6901_{0.0012}$	$0.7589_{0.0001}$	**$0.7625_{0.0000}$** *	$0.7380_{0.0018}$	$0.7610_{0.0001}$ *
8	$0.6366_{0.0035}$	$0.7118_{0.0002}$	$0.5055_{0.0030}$	$0.6085_{0.0010}$	$0.7160_{0.0002}$ *	**$0.7195_{0.0001}$** *	$0.7480_{0.0025}$	$0.7170_{0.0001}$ *
9	$0.5078_{0.0036}$	$0.5981_{0.0001}$	$0.3855_{0.0042}$	$0.5053_{0.0018}$	$0.6073_{0.0002}$ *	**$0.6104_{0.0000}$** *	$0.5749_{0.0015}$	$0.6072_{0.0001}$ *
10	$0.6511_{0.0034}$	$0.7473_{0.0003}$	$0.5746_{0.0038}$	$0.6819_{0.0010}$	$0.7604_{0.0001}$ *	**$0.7634_{0.0001}$** *	$0.7395_{0.0022}$	$0.7611_{0.0001}$ *

Table 6: Hypervolumes for synthetic problem. These are means over 30 independent runs, with standard deviations given in subscript. Higher values are preferable. For each problem, the differences in the means for each algorithm were found to be statistically significant; the exceptions for each problem (i.e., groups of algorithms where the differences in means were not significant) are marked with * and ✚
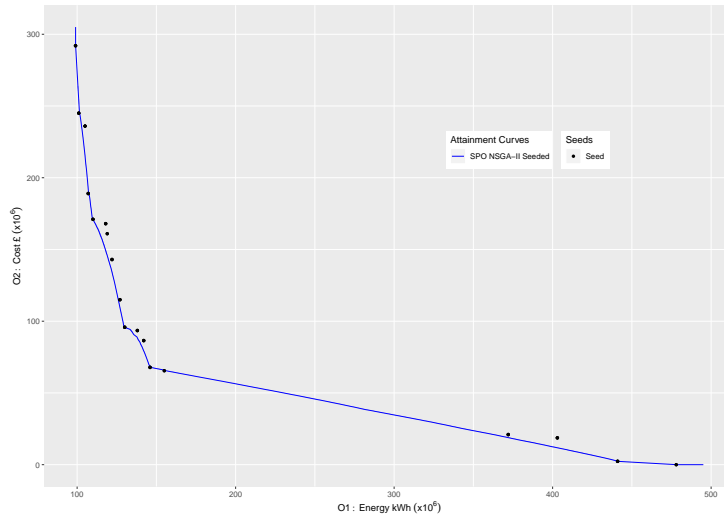
28

Figure 11: The median attainment curve for SPO NSGA-II with seeding on the large-scale problem, with the minimal and maximal attainment curves shown by shading, and the original seeds marked as points.

ought to be retuned for the new problem, but here our main concern is scalability of the approach rather than comparisons across algorithms, retuning was not applied. Figure 11 shows the attainment surfaces with respect to reference front.

The results are highly consistent over all 30 runs, the minimal and maximal attainment curve showing very small variation on the median curve. Several of the seeds are dominated by the fronts, while some seeds have proven difficult to improve on much. Most search effort has been spent on filling in the gaps between the seeds, providing useful information about the trade-off between applying refurbishments to different houses. These results demonstrate that SPO can scale to a much larger problem, though further testing is needed to gain much insight into general scalability. This is also important for our application area since, as more effort is put into finding ways of refurbishing the existing housing stock, there will be a larger number of possible refurbishments per-house, and a larger number of houses for which refurbishments are possible. Both factors will mean that the scale of the real-world building stock optimisation problem will tend to increase.

7. Application to the full housing stock optimisation problem

We now consider a full real-world instance of the housing stock optimisation problem: optimisation of refurbishment options for the housing stock in the North-East of England. Data for these houses is taken from the English Housing Survey (EHS) [12], an annual survey commissioned by the UK Government. The survey data contains detailed information on the condition and energy efficiency

29

Refurbishment	Cost (£)
Cavity wall insulation (CWI)	500
Loft insulation (Loft)	250
Double glazing (DG)	5000
Condensing boiler (Cond)	2500
Solid wall insulation (SWI)	8000
Air source heat-pump (ASHP)	7000
Ground source heat-pump (GSHP)	10000
Biomass heat (BH)	10000
Photovoltaic cells (PV)	8000
Solar hot water (SHW)	2000

Table 7: Refurbishment options and costs. Abbreviations shown in brackets are used later in the paper.

for housing throughout England. This includes: age band; dwelling type; region; dimensions; window area and glazing type; wall, roof and floor construction; loft and other insulation; and built form. There are 935 houses in the EHS for the North-East region of England. In practice, each "house" represents several hundred real houses of a similar type, age and refurbishment state: the 935 houses represent 1.2 million homes in reality. For simplicity we refer to these as single houses.

The EHS data for 2009 is embedded in the Cambridge Housing Model (CHM) [57]. This is coupled to a SAP-based energy calculator [58], to estimate energy consumption and CO_2 emissions for all homes in England, broken down by final use. Some earlier work with a smaller stock optimisation problem used a more time-consuming, dynamic energy model [15, 16, 59]. The dynamic model is not used here because it currently models a limited number of house types and refurbishment options. In estimating the costs, there is no single source of cost information that covers all retrofit measures and the range of cost values varies between different sources [60]. Table 7 shows the 10 refurbishment options that are considered in this problem, with their associated costs; the costs have been obtained from various sources, predominantly [61] and [62]. The full model of cost and energy use for houses with the different refurbishments applied is available from (URL TBC on paper acceptance).

Note that not all 10 refurbishments are applicable to all houses. For example, some houses already have loft insulation installed, or have solid rather than cavity walls. The rules used to determine where refurbishments could be applied are listed in Table 8. Condensing boiler, ground source heat-pump (GSHP), air source heat-pump (ASHP), and biomass boiler are mutually exclusive, but dwellings are allowed to have both solar hot water (SHW) and photo-voltaic (PV) panels. Of the 10 possible refurbishments, the number applicable to any one house within the stock ranges from 2 - 8. Overall, there are 4424 individual decisions about whether or not to apply a particular refurbishment

Refurbishment	Eligibility Rules
Cavity wall insulation	Unfilled cavity wall
Loft insulation increased to 300mm	Loft insulation < 250mm
Double glazing	<100% double glazed; if pre-1850, secondary glazing only
Condensing boiler	Non-condensing boiler; all houses in town centres
Solid wall insulation	Uninsulated solid wall
Air source heat pump	All dwellings
Ground source heat pump	All detached houses, semi-detached houses not in town centres, terraced houses in villages and hamlets
Biomass heat	All houses without gas, detached houses not in town centres, all houses in villages and hamlets
Photovoltaic cells	All dwellings with a roof and built after 1850
Solar hot water	All dwellings with a roof and built after 1850

Table 8: Refurbishment options and the rules for determining the dwellings to which they might be added.

to a particular house, an average of 4.73 per house over the 935 houses. After considering mutually exclusive options, there are 27 150 house variants, these including the un-refurbished and all possible refurbished states for each house; this gives an average of 29 states for each house.

Running an exhaustive search at Stage 1 of the optimisation revealed that there were between 2 and 34 Pareto-optimal solutions for each house, and 10 777 in total across all houses (an average of 11.52 per house).

Given that NSGA-II has proved to be the most robust of the two algorithms for this class of problem, we now apply only the SPO NSGA-II with seeding to Stage 2 of the full stock problem (using the same parameters as in Table 2). The global Pareto-optimal front found by the algorithm is given in Figure 12. Along the length of the front, the solutions found represent an improvement on the seeds. In the middle of the front, these offer a saving 20% of the cost of the seed solutions having same energy use.

Figure 13 shows the trends in application of refurbishments along the length of the Pareto-optimal front. Here, the solution numbers run from 0 (the base-case zero-cost scenario) to 494 (the minimum-energy / maximal-cost scenario). Loft insulation rises sharply at the start and at the end, probably because different levels of insulations already exist in the current housing stock. The early changes will be from 0/12mm up to 300mm, which are very cost effective, whereas the changes from 250-300 mm are much less cost effective and so these will only be optimal at the higher cost end.

Cavity wall insulation rises steadily from the start along the length of the front. Solid wall insulation, on the other hand, only starts to appear from two thirds (around 335) along the front, due to its high cost and relatively small savings. Double glazing shows a similar trend.
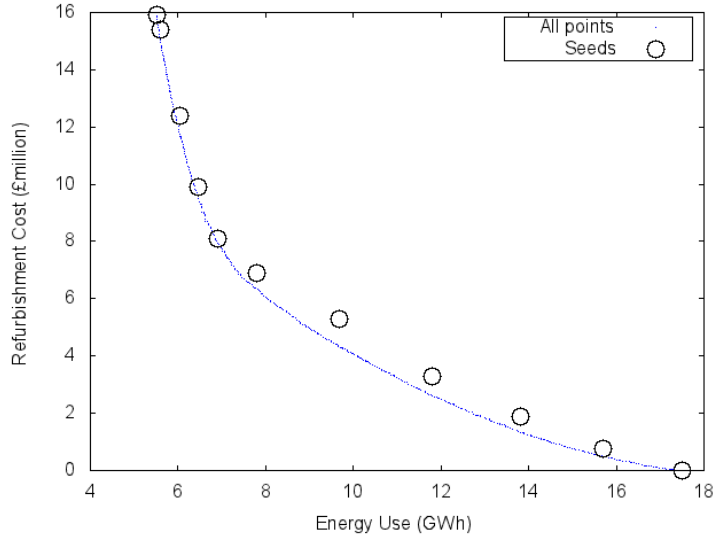
Figure 12: The Pareto-optimal front for the full real-world stock optimisation problem, found by NSGA-II using SPO.

Use of condensing boilers rises steadily from the start but drops to nearly zero around solution 350. This indicates, at the higher end of refurbishment costs, all condensing boilers are being replaced by systems with higher efficiencies, e.g. ASHP or GSHP. ASHP shows the most prominent growth among all the options starting from the low cost end, to the highest point at about 850 installations, but drops slightly at the high cost end of the trade-off to around 800 installations. In contrast, GSHP only starts to appear towards to mid-point of the trade-off (solution no. 250), and continue to rise to approximately 100 installations. There are two reasons for this. Firstly, the cost for installing a ASHP is much cheaper than a GSHP, although the coefficient of performance for a ASHP is slightly lower than that of a GSHP. Secondly, the installation of a GSHP is limited by the rules for refurbishments. As stated in Table 8, GSHP only allowed for all detached houses, semi-detached houses not in town centres, and terraced houses in villages and hamlets.

PV starts to rise sharply at two third of the length (around solution no. 335), and it continues to rise to its highest point at the high-cost end of the front, to more than 800 installations. This indicates that, although expensive, PV is a cost-effective option when the capital allows for it.

SHW shows an interesting trend along the front. It rises from the low-cost end of the trade-off to its first plateau at solution no. 200, then drops to a low point at solution no. 278, and rises again to its second plateau at solution no. 400, and finally continues to drop to nearly zero at the high-cost end. This is correlated with the provision of hot water. As hot water systems become more
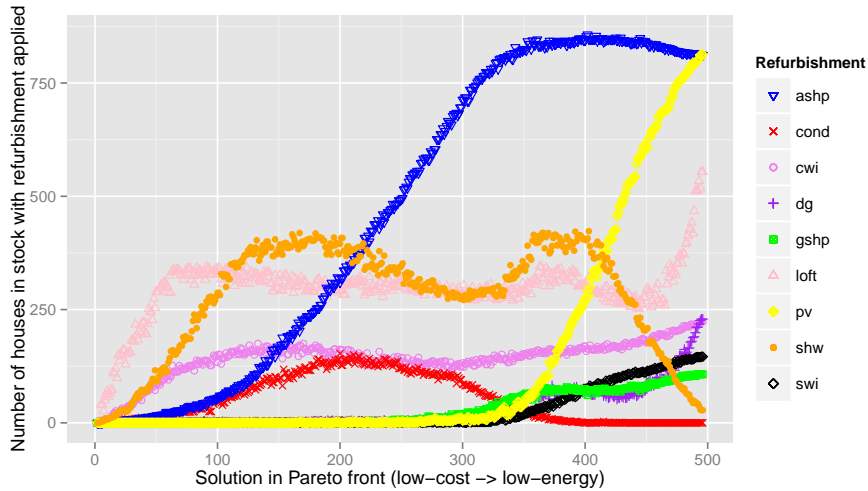
32

Figure 13: The trends of individual refurbishment options along the length of the Pareto-optimal front. Each point shows the number of houses in the stock with a particular refurbishment applied, for a given solution in the Pareto-optimal front. To the left are solutions at the low-cost end of the front, to the right are solutions at the low-energy end of the front. Biomass heat is not plotted as it was zero for all solutions in the front.

efficient, the energy used in producing the hot water reduces, this decreasing the saving due to solar hot water. As SHW only impacts on the energy use, it therefore becomes cost-ineffective and falls out of the optimal solutions.

It is worth noting that no biomass boiler is found in the Pareto-optimal solutions. Biomass is probably excluded as the boiler is as expensive to buy as a GSHP, but will have a lower "efficiency" than the GSHP – and hence will be sub-optimal in energy use.

8. Conclusion

We have presented Sequential Pareto Optimisation (SPO), a two-stage approach to encoding separable large-scale multi-objective problems, such as optimisation of the housing stock. This uses a unique decomposition of the global problem into two stages. There has previously been very little work on multi-objective optimisation for problems with thousands of variables, and none on the optimisation of buildings at the regional stock scale. This represents a new category of problem which we term combinatorially-separable; the synthetic problems we have generated represent a new set of benchmarks sharing this property. SPO implementations using NSGA-II and MOEA/D for the second stage were applied to two sizes of synthetic optimisation problem. This was compared to a naive approach using a single stage and a binary encoding, which was found to perform poorly. NSGA-II performs better than MOEA/D for this problem, resulting

33

in Pareto-optimal fronts which largely dominate those of MOEA/D. We have shown that the performance of MOEA/D can be improved by using only seeds in the initial population, with relocation of the seeds to closely matching scalar sub-problems. These observations should be noted when using MOEA/D for other applications. Seeding the population was shown to greatly improve the final results for both algorithms.

We applied SPO using NSGA-II to a real-world housing stock optimisation problem and conducted some analysis of the results from an applied perspective to show how this may be used in practice. These results are important: this method can be used to allow decision makers to make better-informed choices in targeting refurbishments to the housing stock. This has the potential to make a significant impact on the energy consumption and related emissions.

This work has also opened up three additional lines of future research, which will make the SPO approach more general.

Firstly, the present work has proven the concept for a problem with known separability. Of course, where such characteristics of a problem are known, they should be exploited by the algorithm designed; a growing field termed Grey-box optimisation [63], where known structures in problems such as boolean satisfiability can be exploited for substantial efficiency gains. There are few examples of real-world multi-objective problems that have this property. For this to be the case an application will be like the housing stock problem where there are two scales: sub-problems where there is a trade-off, but these sub-problems interact at the global scale. Another example is employee scheduling [13], where regions might be scheduled to both meet the target workload and minimise the required workforce; but increasing the workforce in one region will require a reduction in the workforce for another. Identification and classification of such problems is an interesting direction to pursue. Additionally, often features like separability are unknown. It will be of interest to consider integrating approaches for detecting separability in the objectives automatically based on linkage detection, similar to the approach of [8], or random grouping [7, 39].

Secondly, with the additively-separable problems described in this paper, the sub-problems have small enough search spaces that they can be explored exhaustively. It would be interesting to consider much larger search spaces for the first stage of the process. In terms of the housing application, this means a larger number of options per-house, or collected groups of houses (for example, with shared facilities such as community heating systems). This could be tackled by using a non-exhaustive search method (perhaps a second metaheuristic) in Stage 1, before proceeding to the integer-encoded global optimisation over the Pareto-optimal fronts for each house in Stage 2.

Thirdly, expansion of the method to more than two objectives can also be considered. Focussing on two-objective optimisation means that the Pareto-optimal solutions for each sub-problem can be sorted into ascending Objective 1 / descending Objective 2. This yields a natural ordering over the Pareto-optimal solutions for each sub-problem so they could be translated to integers for the Stage-2 optimisation, and regular integer variation operators could be used. This means that, for one sub-problem, changing the index by a small

amount should lead to a small change in the objectives and changing it by a large amount should result in a larger change in the objectives. This respects the proximate optimality principle [64]. Extending the approach to three and many-objective problems would need the following modifications. Stage 1 would remain the same: an exhaustive or efficient search over the space for each sub-problem, to find a Pareto-optimal front for each sub-problem. At Stage 2, new mutation and crossover operators would need to be chosen that respect the shape of the Pareto-optimal surface per sub-problem. For two objectives it is possible to use operators that respect the natural ordering of integers to efficiently explore 2-objective Pareto fronts for the sub-problems. Pareto-optimal surfaces for 3-objectives and higher would need new operators that can "walk" in the neighbourhood of similar Pareto-optimal solutions in the surface.

Exploration of these latter issues will lead to a more general framework for large-scale optimisation, building on this paper's work to develop a method that works for many-objective problems with a much lower degree of separability than the house stock optimisation problem. This will have the potential for application in a variety of other application domains.

Data Access Statement

All data and source code arising from this research is available from (URL TBC on paper acceptance).

Appendix: Synthetic problem

The synthetic problem is based on house "H1501108" in the Cambridge Housing Model data. This house is representative of the "average" house in the stock in two ways. It has 5 refurbishment options, close to the average number per house in the whole stock of 4.73. This means that over the whole stock, the synthetic problem has 4675 decisions about whether or not to particular refurbishments to individual houses. The Pareto-optimal front of solutions for house H1501108 contains 11 solutions, also close to the average for the stock of 11.53. Additionally, all of the options apply, so all 32 solutions for the house are feasible, making analysis simpler.

The full set of solutions for this house are given in Table 9. The full problem is formed by duplicating the 5 refurbishment options 935 times. Objective

evaluations for the full stock simply sum the objective values for the separate houses. The Pareto-optimal solutions for each sub-problem are identified by a non-zero Pareto Index (which ascends with O1). For Stage 2, the decision variables for the global problem select solutions at the sub-problem level based on this index.

X0	X1	X2	X3	X4	O1	O2	Pareto Index
0	0	0	0	0	23399.09617	0	10
1	0	0	0	0	20277.82204	500	9
0	1	0	0	0	22270.98959	5000	-
1	1	0	0	0	19093.89017	5500	-
0	0	1	0	0	9787.96717	7000	6
1	0	1	0	0	8878.598432	7500	5
0	1	1	0	0	9464.11987	12000	-
1	1	1	0	0	8534.231879	12500	3
0	0	0	0	1	19793.33607	2000	8
1	0	0	0	1	17103.94735	2500	7
0	1	0	0	1	18829.17604	7000	-
1	1	0	0	1	16097.72663	7500	-
0	0	1	0	1	9461.298231	9000	-
1	0	1	0	1	8551.85455	9500	4
0	1	1	0	1	9137.796621	14000	-
1	1	1	0	1	8207.866941	14500	2
0	0	0	1	0	22258.82169	8000	-
1	0	0	1	0	19137.54756	8500	-
0	1	0	1	0	21130.7151	13000	-
1	1	0	1	0	17953.61569	13500	-
0	0	1	1	0	8647.692686	15000	-
1	0	1	1	0	7738.323949	15500	1
0	1	1	1	0	8323.845387	20000	-
1	1	1	1	0	7393.957395	20500	0
0	0	0	1	1	19584.13113	10000	-
1	0	0	1	1	16894.74241	10500	-
0	1	0	1	1	18619.97111	15000	-
1	1	0	1	1	15888.5217	15500	-
0	0	1	1	1	9252.093294	17000	-
1	0	1	1	1	8342.649612	17500	-
0	1	1	1	1	8928.591684	22000	-
1	1	1	1	1	7998.662003	22500	-

Table 9: All solutions for one house within the synthetic problem. In terms of the application, variables X0 - X4 map to Cavity Wall Insulation, Double Glazing, Air-source Heat Pump, Photovoltaic Cells and Solar Hot Water respectively. O1 and O2 are the energy and cost objectives.

References

[1] A. LaTorre, S. Muelas, J. Peña, A comprehensive comparison of large scale global optimizers, Information Sciences 316 (2015) 517–549, ISSN 0020-0255, doi:10.1016/j.ins.2014.09.031.

[2] S. Mahdavi, M. E. Shiri, S. Rahnamayan, Metaheuristics in large-scale global continues optimization: A survey, Inform. Sci. 295 (2015) 407 – 428, ISSN 0020-0255, doi:10.1016/j.ins.2014.10.042.

[3] X. Li, K. Tang, M. Omidvar, Z. Yang, K. Qin, Benchmark Functions for the CEC'2013 Special Session and Competition on Large Scale Global Optimization, Tech. Rep., Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.

[4] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, Test Problems for Large-scale Multi-and Many-objective Optimization, IEEE Transactions on Cybernetics 47 (12) (2017) 4108–4121.

[5] University of Birmingham, Large-Scale Global Optimization Repository, URL `http://www.cercia.ac.uk/projects/lsgo/`, 2016.

[6] L. M. Antonio, C. A. C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, 2013 IEEE Congress on Evolutionary Computation doi:10.1109/cec.2013.6557903.

[7] L. M. Antonio, C. A. C. Coello, Decomposition-Based Approach for Solving Large Scale Multi-objective Problems, in: Parallel Problem Solving from Nature – PPSN XIV, Springer Nature, 525–534, doi:10.1007/978-3-319-45823-6_49, 2016.

[8] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for Multiobjective Optimization Problems With Large-Scale Variables, IEEE T. Evolut. Comput. 20 (2) (2016) 275–298, doi:10.1109/tevc.2015.2455812.

[9] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A Decision Variable Clustering-Based Evolutionary Algorithm for Large-scale Many-objective Optimization, IEEE Transactions on Evolutionary Computation (2016) 1–1 doi:10.1109/tevc.2016.2600642.

[10] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation, IEEE Transactions on Evolutionary Computation 22 (2) (2018) 260–275, ISSN 1089-778X, doi:10.1109/TEVC.2017.2704782.

[11] DECC, Energy Consumption in the United Kingdom, URL `https://www.gov.uk/government/statistics/energy-consumption-in-the-uk`, 2014.

[12] DCLG, English housing survey, URL `https://www.gov.uk/government/collections/english-housing-survey`, 2015.

[13] K. N. Reid, J. Li, J. Swan, A. McCormick, G. Owusu, Variable Neighbourhood Search: A case study for a highly-constrained workforce scheduling problem, in: Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, IEEE, 1–6, 2016.

[14] M. N. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative Co-Evolution With Differential Grouping for Large Scale Optimization, IEEE Transactions on Evolutionary Computation 18 (3) (2014) 378–393, ISSN 1089-778X, doi:10.1109/TEVC.2013.2281543.

[15] M. He, T. Lee, S. C. Taylor, A. E. I. Brownlee, J. A. Wright, Multi-objective optimization for a large scale retrofit program for the housing stock in the North East of England, in: 6th International Building Physics Conference, vol. 78 of *Energy Procedia*, 854–859, 2015.

[16] M. He, A. E. I. Brownlee, J. A. Wright, S. C. Taylor, Multi-dwelling Refurbishment Optimization: Problem Decomposition, Solution, and Trade-off Analysis, in: Proc. IBPSA Building Simulation Conference, IBPSA, Hyderabad, India, 2015.

[17] K. Deb, A. R. Reddy, G. Singh, Optimal Scheduling of Casting Sequence Using Genetic Algorithms, Mater. Manuf. Processes 18 (3) (2003) 409–432.

[18] K. Sastry, D. E. Goldberg, X. Llora, Towards billion-bit optimization via a parallel estimation of distribution algorithm, in: Proc. GECCO, GECCO '07, ACM, New York, NY, USA, ISBN 978-1-59593-697-4, 577–584, doi:10.1145/1276958.1277077, 2007.

[19] Y. Semet, M. Schoenauer, An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling, in: Evolutionary Comp., 2005. The 2005 IEEE Congress on, vol. 3, 2752 – 2759 Vol. 3, doi:10.1109/CEC.2005.1555040, 2005.

[20] K. Deb, C. Myburgh, Breaking the Billion-Variable Barrier in Real-World Optimization Using a Customized Evolutionary Algorithm, in: Proc. GECCO, ACM, doi:10.1145/2908812.2908952, 2016.

[21] E. Li, H. Wang, F. Ye, Two-level Multi-surrogate Assisted Optimization method for high dimensional nonlinear problems, Applied Soft Computing 46 (2016) 26–36, doi:10.1016/j.asoc.2016.04.035.

[22] K. Li, M. N. Omidvar, K. Deb, X. Yao, Variable Interaction in Multi-objective Optimization Problems, in: Parallel Problem Solving from Nature – PPSN XIV, Springer Nature, 399–409, doi:10.1007/978-3-319-45823-6_37, 2016.

[23] J. Manicassamy, S. S. Kumar, M. Rangan, V. Ananth, T. Vengattaraman, P. Dhavachelvan, Gene Suppressor: An Added Phase Towards Solving Large Scale Optimization Problems In Genetic Algorithm, Applied Soft Computing doi:10.1016/j.asoc.2015.06.017.

[24] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, L. Jiao, Greedy discrete particle swarm optimization for large-scale social network clustering, Information Sciences 316 (2015) 503–516, ISSN 0020-0255, doi:10.1016/j.ins.2014.09.041.

[25] A. Kabán, J. Bootkrajang, R. J. Durrant, Toward Large-Scale Continuous EDA: A Random Matrix Theory Perspective, Evolutionary Computation (2015) 1–37ISSN 1530-9304, doi:10.1162/evco_a_00150.

[26] R. Rashedi, T. Hegazy, Capital renewal optimisation for large-scale infrastructure networks: genetic algorithms versus advanced mathematical tools, Structure and Infrastructure Engineering 11 (3) (2014) 253–262, ISSN 1744-8980, doi:10.1080/15732479.2013.866968.

[27] M. Potter, K. D. Jong, A cooperative coevolutionary approach to function optimization, in: Proc. 3rd Conf. Parallel Problem Solving Nat., 249–257, 1994.

[28] L. M. Antonio, C. A. C. Coello, Coevolutionary Multiobjective Evolutionary Algorithms: Survey of the State-of-the-Art, IEEE Transactions on Evolutionary Computation 22 (6) (2018) 851–865.

[29] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Information Sciences 178 (15) (2008) 2985 – 2999, ISSN 0020-0255, doi:10.1016/j.ins.2008.02.017.

[30] M. Omidvar, X. Li, X. Yao, , Z. Yang, Cooperative co-evolution for large scale optimization through more frequent random grouping, in: Proc. IEEE CEC, 1754–1761, 2010.

[31] X. Li, X. Yao, Cooperatively Coevolving Particle Swarms for Large Scale Optimization, IEEE Trans. Evol. Comp. 16 (2) (2012) 210–224, ISSN 1089-778X, doi:10.1109/TEVC.2011.2112662.

[32] Y. Sun, M. Kirley, S. K. Halgamuge, Extended Differential Grouping for Large Scale Global Optimization with Direct and Indirect Variable Interactions, in: Proc. Genetic & Evolutionary Comp. Conf., ACM, New York, NY, ISBN 978-1-4503-3472-3, 313–320, doi:10.1145/2739480.2754666, 2015.

[33] A. Hameed, A. Kononova, D. Corne, Engineering Fitness Inheritance and Co-operative Evolution Into State-of-the-Art Optimizers, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, doi:10.1109/ssci.2015.238, 2015.

[34] Y. Wang, B. Li, Two-stage based ensemble optimization for large-scale global optimization, in: Proc. IEEE CEC, 2010.

[35] Y. Wang, J. Huang, W. S. Dong, J. C. Yan, C. H. Tian, M. Li, W. T. Mo, Two-stage based ensemble optimization framework for large-scale global optimization, European Journal of Operational Research 228 (2) (2013) 308–320, ISSN 0377-2217, doi:10.1016/j.ejor.2012.12.021.

[36] Z. Yang, K. Tang, X. Yao, Scalability of generalized adaptive differential evolution for large-scale continuous optimization, Soft Computing 15 (11) (2011) 2141–2155, ISSN 1432-7643, doi:10.1007/s00500-010-0643-6.

[37] J. J. Durillo, A. J. Nebro, C. A. Coello Coello, F. Luna, E. Alba, A comparative study of the effect of parameter scalability in multi-objective metaheuristics, IEEE World Congress on Computational Intelligence doi:10.1109/cec.2008.4631047.

[38] J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. Garcia-Nieto, F. Luna, E. Alba, A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems, IEEE Trans. Evol. Computat. 14 (4) (2010) 618–635, ISSN 1089-778X, doi:10.1109/tevc.2009.2034647.

[39] F. Qiu, L. Mo, B. Jiang, L. Wang, Multi-objective particle swarm optimization algorithm using large scale variable decomposition, Chin. J. Comput. 39 (12) (2016) 2598–2613.

[40] B. Cao, J. Zhao, Z. Lv, X. Liu, A Distributed Parallel Cooperative Coevolutionary Multiobjective Evolutionary Algorithm for Large-Scale Optimization, IEEE Transactions on Industrial Informatics 13 (4) (2017) 2030–2038.

[41] H. Mühlenbein, T. Mahnig, A. O. Rodriguez, Schemata, Distributions and Graphical Models in Evolutionary Optimization, Jnl. of Heuristics 5 (2) (1999) 215–247, doi:10.1023/A:1009689913453.

[42] B. A. Julstrom, Seeding the population: improved performance in a genetic algorithm for the rectilinear Steiner problem, in: Proc. of the 1994 ACM symp. on Applied computing, SAC '94, ACM, New York, NY, USA, ISBN 0-89791-647-6, 222–226, doi:10.1145/326619.326728, 1994.

[43] R. Evins, Configuration of a genetic algorithm for multi-objective optimisation of solar gain to buildings, in: Proc. GECCO, ACM Press, New York, NY, USA, ISBN 978-1-59593-697-4, 2003–2006, 2010.

[44] J. A. Wright, A. E. Brownlee, M. M. Mourshed, M. Wang, Multi-objective optimization of cellular fenestration by an evolutionary algorithm, J. Build. Perform. Sim. 7 (1) (2014) 33–51, doi:10.1080/19401493.2012.762808.

[45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE T. Evolut. Comput. 6 (2) (2002) 182–197.

[46] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE T. Evolut. Comput. 11 (6) (2007) 712 –731, ISSN 1089-778X, doi:10.1109/TEVC.2007.892759.

[47] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 1999.

[48] K. A. De Jong, Evolutionary Computation: A Unified Approach, MIT Press, ISBN 0-262-04194-4, 2006.

[49] K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9 (3) (1994) 1–15.

[50] K. Deb, S. Agrawal, A Niched-Penalty Approach for Constraint Handling in Genetic Algorithms, Artificial Neural Nets and Genetic Algorithms (1999) 235–243 doi:10.1007/978-3-7091-6384-9_40.

[51] F. Hutter, H. Hoos, K. Leyton-Brown, Sequential Model-Based Optimization for General Algorithm Configuration, in: LION-5, LNCS, 507–523, 2011.

[52] K. Carlson, J. Winquist, An Introduction to Statistics: An Active Learning Approach, Sage Publications Inc, Thousand Oaks, CA, ISBN 978-1-4522-1743-7, 2014.

[53] J. Knowles, A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers, in: Proc. Conf. on Intel. Syst. Design and Appl., ISDA '05, IEEE, Washington, DC, USA, ISBN 0-7695-2286-06, 552–557, doi:10.1109/ISDA.2005.15, 2005.

[54] W. Peng, Q. Zhang, H. Li, Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem, in: Multi-objective memetic algorithms, Springer, 309–324, 2009.

[55] H. Li, Q. Zhang, Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 284–302, doi:10.1109/tevc.2008.925798.

[56] K. Li, K. Deb, Q. Zhang, S. Kwong, An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition, IEEE Trans. Evol. Computat. 19 (5) (2015) 694716, ISSN 1941-0026, doi:10.1109/tevc.2014. 2373386.

[57] Cambridge Architectural Research Ltd , A Guide to The Cambridge Housing Model v2.7, UK Government Dept of Energy and Climate Change and Cambridge Architectural Research Ltd, 2011.

[58] Building Research Establishment, The Governments Standard Assessment Procedure for Energy Rating of Dwellings, 2009 Edition, URL `http://www.bre. co.uk/filelibrary/SAP/2009/SAP-2009_9-90.pdf`, retrived 1/10/2015, 2010.

[59] M. He, T. Lee, S. Taylor, S. Firth, K. Lomas, Dynamic modelling of a large scale retrofit programme for the housing stock in the North East of England, in: 2nd International Conference on Urban Sustainability and Resilience, UCL, London, UK, 2014.

[60] S. Porritt, Adapting UK Dwellings for Heat Waves, Ph.D. thesis, De Montfort University, Leicester, UK, 2012.

[61] Element Energy, The growth potential for Micro generation in England, Wales and Scotland, URL `http://webarchive.nationalarchives.gov.uk/+/ http:/www.berr.gov.uk/files/file46003.pdf`, 2008.

[62] T. Lee, R. Yao, P. Coker, An analysis of UK policies for domestic energy reduction using an agent based tool, Energy Policy 66 (2014) 267–279, doi:10.1016/j.enpol. 2013.11.004.

[63] D. Whitley, A Gray Box Manifesto for Evolutionary Combinatorial Optimization, SIGEVOlution 12 (1) (2019) 35, doi:10.1145/3328473.3328474, URL `https:// doi.org/10.1145/3328473.3328474`.

[64] D. Niizuma, K. Yasuda, A. Ishigame, Multipoint Tabu Search based on proximate optimality principle – application of parts concept, IEEJ Transactions on Electrical and Electronic Engineering 2 (6) (2007) 635–642, ISSN 1931-4981, doi: 10.1002/tee.20219.