Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation

Alexander E.I. Brownlee [a],[*], Jonathan A. Wright [b]

[a] *Division of Computing Science and Mathematics, University of Stirling, UK*
[b] *School of Civil and Building Engineering, Loughborough University, UK*

ARTICLE INFO

ABSTRACT

Reducing building energy demand is a crucial part of the global response to climate change, and evolutionary algorithms (EAs) coupled to building performance simulation (BPS) are an increasingly popular tool for this task. Further uptake of EAs in this industry is hindered by BPS being computationally intensive: optimisation runs taking days or longer are impractical in a time-competitive environment. Surrogate fitness models are a possible solution to this problem, but few approaches have been demonstrated for multi-objective, constrained or discrete problems, typical of the optimisation problems in building design. This paper presents a modified version of a surrogate based on radial basis function networks, combined with a deterministic scheme to deal with approximation error in the constraints by allowing some infeasible solutions in the population. Different combinations of these are integrated with Non-Dominated Sorting Genetic Algorithm II (NSGA-II) and applied to three instances of a typical building optimisation problem. The comparisons show that the surrogate and constraint handling combined offer improved run-time and final solution quality. The paper concludes with detailed investigations of the constraint handling and fitness landscape to explain differences in performance.

© 2015 Published by Elsevier B.V.

1. Introduction

The building and construction industry is one which offers large potential for reduced environmental impact and improved sustainability. Buildings are a large contributor to world carbon emissions, particularly due to their heating, cooling and lighting energy demands (for example, over 50% of UK carbon emissions are related to building energy consumption [1]). Usual practice, driven by expectation and building function, is for each individual building to be designed afresh, losing the benefit of mass-production where a single optimisation process applies to thousands or millions of units. This means that improvements in the building design process to aid designers in reducing building carbon emissions are an important area of research. Computer-based optimisation methods are of rapidly-increasing importance in this area.

Several characteristics of building energy optimisation present a challenge. Each specific problem has a large decision space, with a mixture of continuous and discrete variables determining varied aspects of a building's design. Typically there are multiple conflicting objectives and constraints: reducing energy demand conflicts with reducing the building's capital cost, and both conflict with increasing the comfort of the building's occupants. These factors motivate the use of evolutionary algorithms (EAs), successfully applied to a number of applications in this domain [2], including: heating, ventilating, and air conditioning (HVAC) system sizing [3]; fenestration [4]; building envelope [5,6]; and building form [7,8]. Further to the issues above, the building performance simulations (BPS) used to estimate energy consumption, occupant comfort and factors affecting capital cost are time-consuming, taking minutes to hours depending on building complexity and the modelling accuracy required. This limits the practicality of EAs for real-world use in this domain; a highly competitive industry where submission of design solutions are usually time-critical and optimisation runs must be completed over a night or weekend.

There are many approaches to improving the run-time of EAs: parallelisation [9], reducing simulation complexity, and fitness approximation methods such as surrogates [11]. Parallelisation of fitness evaluations is easily adopted and can be applied alongside the latter methods, for a multiplicative speed gain. Reducing the simulation complexity requires problem-specific knowledge, and has the negative impact of less accurate simulation results possibly leading the EA to false optima. Surrogate fitness approximation, where some solutions (including the final returned optima)

are evaluated by the full simulation while others are evaluated by a much cheaper model offers potential to improve run times without losing accuracy. There are, however, a limited number of works featuring surrogates for problems with discrete decision spaces, and still fewer looking at problems with constraints or multiple objectives. This has led to minimal adoption of surrogates in building optimisation, where the problems often exhibit these characteristics. The need for greater efficiency of EAs for building optimisation motivates further research into surrogates that can simultaneously handle constraints, multiple objectives and discrete or mixed variable types. If successful, such surrogates will improve the practicability of EAs for this application area.

Given the above points, this paper explores the use of radial basis function networks (RBFN) to reduce the number of calls to the full building performance simulation, by approximating objective and constraint values. Several contributions are made. The first is that an existing approach [12] is adapted for multi-objective and constrained problems, with a change to the way that categorical variables are handled. A separate RBFN for each constraint and objective allows for retraining on each independently. The paper's second contribution is Deterministic Infeasibility Sorting (DIS), a scheme where infeasible solutions are retained in the population to deal with two issues. One issue is that estimation of constraints is difficult because an approximation error can mean that the algorithm discards feasible solutions that are in truth feasible: particularly problematic with many constraints. The second issue is that approximation errors can also affect the dominance relationship between solutions in multi-objective problems. The paper's third contribution is the comparison of several possible approaches to integrating the surrogate and DIS within Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [13]. Several variants of NSGA-II are applied to three versions of a mixed-integer, constrained, multi-objective building optimisation problem. Comparisons are made in both the quality of the final solution sets and convergence speed. It is found that the best performing NSGA-II variant offers a performance improvement for two of the three problems (with no significant difference on the third). The final contributions are a detailed analysis of the constraint handling, to allow greater understanding of the impact of the techniques, and a study of the problem's fitness landscape to aid analysis of the results.

The rest of the paper is structured as follows. Section 2 describes related work in this area. Sections 3 and 4 describe the building optimisation problem and the proposed modifications to NSGA-II: the RBFN surrogate; the constraint handling approach; a number of minor changes to improve performance; and the resulting variants of NSGA-II. Section 5 describes and discusses experiments tuning and comparing the algorithm variants. Sections 6 and 7 provide deeper analysis of the constraint handling and fitness landscape, and Section 8 concludes the paper.

2. Related work

There is a rapidly growing body of work on approximating fitness [14,15,11], whether to: reduce calls to the evaluation function and decrease running time or other cost; counteract noise; or overcome discontinuities in the fitness function. Techniques include: fitness inheritance [10,16]; co-evolved fitness predictors [17]; fuzzy matching against an archive [18]; artificial neural networks [19–21]; linear and polynomial regression [22,23]; Gaussian processes or Kriging [24,25] and probabilistic distributions [26].

Of particular interest for the present work is the combination of fitness approximation for problems with discrete variables, multiple objectives and constraints. Several of the above papers reported success with surrogates for multi-objective problems. A recent review [27] on constraint handling approaches in

evolutionary algorithms (EAs) suggests that the use of surrogate approaches for constraints remains quite rare, and all for problems with continuous decision variables.

There has, however, been some work on fitness approximation for discrete problems without constraints, including [28–32]. Tresidder et al. [33] and Zemella et al. [34], make use of surrogates in EAs for discrete building design problems. Bajer and Holeňa [35] also used a RBFN surrogate, for a mixed continuous and discrete problem. In contrast with the present paper, the RBFs were defined over only the continuous variables, with centres located in clusters determined by Hamming or Jaccard distance on the discrete variables.

In [12], an RBFN is used as a surrogate for a mixed-integer search space. The model is used as a filter; too many offspring are created each generation and the model is used to choose promising ones which are then evaluated with the true fitness function and retained. The technique was applied to intravascular ultrasound image analysis; this approach is the basis of the method proposed in this paper.

This paper also describes an operator which allows infeasible solutions to remain in the population as a means of improving performance for constrained problems. There are many examples of work in this area, including [36–40].

3. Application

Variants of NSGA-II were applied to a real-world simulation-based building optimisation problem; this will now be described in detail. The aim is to find the trade-off between operational energy use (for heating, cooling and lighting) and construction cost for a mid-floor of a small commercial office building (Fig. 1), subject to constraints relating to the comfort of the building's occupants. The floor is divided into five rooms, treated as separate zones for energy and thermal comfort simulation. The two end zones are 24 m × 8 m, and the three middle zones 30 m × 8 m. Floor to ceiling height throughout is 2.7 m. Building energy consumption strongly depends on local climate and the experiments in this work placed the building in three locations: Birmingham (UK), Chicago (IL, USA), and San Francisco (CA, USA). A building situated in Chicago (hot summer, cold winter) would be expected to need better insulation and more artificial heating and cooling than Birmingham, and even more so San Francisco (more temperate, flatter climates). Consequently these represent three quite different instances of the same optimisation problem, albeit sharing the same objectives, variables and constraints.

The optimisation takes place at the building's detailed design stage, with overall form and layout fixed. Construction variables to be optimised are building orientation, glazing type and window-to-wall ratio, and wall/floor material types. The optimisation also includes variables for the operation of the building's HVAC systems. This allows for the final design to be tuned to a particular operational pattern, for example exchanging an increased capital
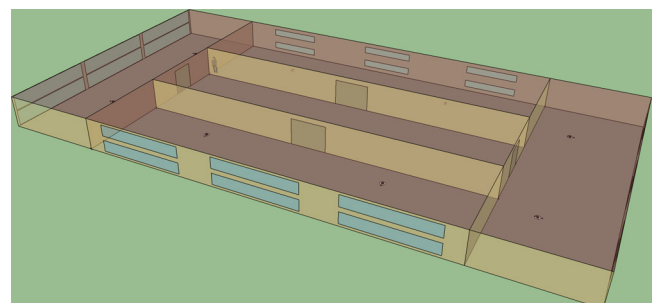

Fig. 1. The building floor for the optimisation problem for illustration.

Table 1
Summary of constraints; the abbreviations are used in subsequent references to individual constraints in the paper.

Abbreviation	Description	Limit
NSOH WSOH ESOH SSOH ISOH	Summer overheating: number of hours per year where zone temperature exceeds 28 °C, for North, West, East, South and Internal zones respectively	≤30
NPMV WPMV EPMV SPMV IPMV	Predicted mean vote: number of hours per year where the predicted mean vote for zone occupants falls below −0.5, for North, West, East and South zones respectively	≤30
NCO2 WCO2 ECO2 SCO2	CO_2: maximum concentration of carbon dioxide (parts per million) in the air of North, West, East and South zones respectively	≤1500
NACH WACH EACH SACH	Maximum air change rate: a practical limit to the air change rate arising from natural ventilation; the constraints are on the number of hours per year where a zone requires an air change rate of more than 8 air changes per hour	≤0

cost for a building which can be economically run at a lower temperature range. These are the temperature and air flow set-points (thresholds beyond which HVAC systems are switched on) and start-stop times for heating and cooling systems. Overall there are 8 integer, 30 continuous and 12 categorical variables (the latter taking one of 2 or 3 values), summarised in Table 14 (A). Gray coding is used for crossover and mutation operations, allowing the same operators to apply across all variables. This means that the continuous variables are discretised for mutation and crossover; the step size is also noted in the table.

Energy use is determined by the freely-available whole-building energy simulation package EnergyPlus (E+) [41]. This models heating, cooling and lighting loads required to meet predefined set-points for measures such as temperature and light level. Energy use covers a whole year of operation given the weather patterns at each location. The construction cost objective uses a simple function based on the Spon's [42] costs for the materials used and equipment size (determined by E+). A run of the E+ simulation takes 1–2 min on an Intel i7 2.7 GHz CPU, depending on the specific building configuration. Allowing up to six simulations in parallel, a budget of 5000 evaluations completes in approximately 24 h (a practical duration for real-world use).

The optimisation is also subject to 18 inequality constraints based on occupant comfort, determined by E+, and summarised in Table 1. They cover hours of overheating during summer months, carbon dioxide (CO_2) concentration ("freshness" of internal air), predicted mean vote, and maximum air change rate due to natural ventilation. Predicted mean vote is an estimation of the mean response of a group of people according to the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) thermal sensation scale, in which +3 is hot, zero is neutral and −3 is cold.

Several assumptions are made about the final working conditions for the building that impact on operational energy use and comfort. Working hours are 9:00 to 17:00, outside which the building is unoccupied. Each zone has typical design conditions of 1 occupant per 10 m² floor area and equipment loads of 11.5 W/m² floor area. Lighting is controlled to provide an illuminance of 500 lux at two reference points located in each perimeter zone; visible as points on the floor of Fig. 1. Infiltration is 0.1 air changes per hour, and ventilation rates are 8 l/s per person. Heating and cooling are modelled using an idealised system that provides sufficient energy to offset the zone loads and meet the zone

temperature set-point during hours of operation; free-cooling is available through natural and mechanical ventilation.

A simplified version of the same problem formed the basis for experiments describing an approach to multi-criteria decision making and visualisation in [43].

4. Algorithms

This section describes the proposed surrogate, constraint handling technique, a number of minor changes to improve performance, and methods by which all of these can be integrated within NSGA-II.

4.1. RBFN surrogate

Radial basis function networks (RBFN) are fully connected feed-forward artificial neural networks which can be used for function interpolation [44], providing a mapping from a set of real-valued inputs to a set of real-valued outputs. They have been applied as surrogates for evolutionary algorithms in several previous works including [45,12,46,47]. Li et al. [12] noted that an RBFN can be used for approximation in many different decision spaces, as long as an appropriate distance measure can be defined. This makes RBFNs particularly suitable for a problem with mixed variable types such as the example building problems.

For the distance measure, the present research adopts the Heterogeneous Euclidean-Overlap Metric (HEOM) approach suggested by Wilson and Martinez in [48] to combine distances for different parameter types. Li et al. [12] also used the measure within a RBFN meta-model for a mixed-integer search space. Solutions are divided into three components; real/continuous, integer and non-numeric or categorical parameters. Continuous and integer parameters are normalised to the range (0,1) prior to calculating the distance metric. The total raw distance Δ is the combination of the Euclidean Δ_r, Manhattan Δ_z and Hamming Δ_d distances between these parameters respectively:

$$\Delta = \sqrt{\Delta_r + \Delta_z + \Delta_d}. \tag{1}$$

In normalising the variables to (0,1), categorical variables are over-weighted compared to integer and continuous variables. While the mean distance between two normally distributed random variables with a range of (0,1) is 1/3, if the variables are restricted to 0 or 1, the mean distance is 1/2. To correct for this, a weight that will be termed the *Hamming weighting* ω is introduced, changing the distance measure to (2).

$$\Delta = \sqrt{\Delta_r + \Delta_z + \omega\Delta_d} \tag{2}$$

ω is 2/3 so the mean contribution to the total distance that a categorical variable makes is 1/3, equal to that for the other variable types. This could also apply to the integer variables but the problem is much reduced because the lowest cardinality for the integer variables in the example building optimisation problem is 7, for which the mean difference between a pair of variables is 0.38. An experiment exploring the impact of Hamming weighting further is described in Section 5.4.

4.1.1. Training

Training of the RBFN takes three steps: determining the centres for the hidden layer nodes, the widths for the Gaussian functions, and the output layer weights. This work adopts the common practice that the first two steps are unsupervised and the last step is supervised. For determining the ideal cluster size and determining when to retrain the RBFN, the fitness prediction correlation (FPC) [49] is used to measure the model's accuracy. The FPC is Spearman's

```
1:  R_t = P_t ∪ Q_t                        #combine parent+offspring populations
2:  F =fast-non-dominated-sort(R_t) #F = (F_1, F_2, . . .), R_t's nondom fronts
3:  P_{t+1} = ∅ and i = 1
4:  repeat
5:     crowding-distance-assignment(F_i)  #calculate crowding distance in F_i
6:     P_{t+1} = P_{t+1} ∪ F_i        #include ith nondominated front in parent pop
7:     i = i + 1                          #check next front for inclusion
8:  until P_{t+1} + F_i ≤ N                    #until parent pop filled
9:  sort(F_i ≺_n)                          #sort in descending order using ≺_n
10: P_{t+1} = P_{t+1} ∪ F_i[1 : (N − |P_{t+1}|)]   #choose first (N − |P_{t+1}|) elems of F_i
11: Q_{t+1} =make-new-pop(P_{t+1})     #apply selection, crossover and mutation
12: t = t + 1                                 #increment generation counter
```

Fig. 2. Core of NSGA-II [13].

rank correlation [50], between the true constraint/objective values and the surrogate predicted values, for a solution set.

Typically, a clustering algorithm such as *k*-means is used to locate the centres for RBFN hidden layer nodes, or the centres are simply the set of training points as in [12]. However, training time for the output layer weights increases rapidly with the number of training points. Karakasis and Giannakoglou [51] also suggested that using the training patterns as centres is suboptimal for multi-objective problems. The present work uses a clustering algorithm to determine the centres. *k*-medoid rather than *k*-means clustering is used, so the centres represent real points in the discrete decision space. This is necessary as it is impossible to interpolate between points separated by categorical values.

To find a suitable number of clusters, 1000 solutions were generated at random, and the network was repeatedly trained on random subsets of solutions (subset sizes matched the population sizes in Section 5). The FPC was calculated for objective and constraint predictions of the remaining solutions, and was highest when the number of clusters was one third of the population size.

The width for each hidden node RBF is the root mean square distance from that node's centre to the centres of the two nearest neighbours. Least squares fitting is used to determine the output layer weights.

4.1.2. Integrating the surrogate with NSGA-II

For convenience, the core of NSGA-II [13] is reproduced in Fig. 2. The important modification is at step 11: too many offspring are generated, and the surrogate filters promising solutions for inclusion in the new population (Fig. 3).

The surrogate is initially trained using the starting population P_0. Then, in place of the original offspring creation step, standard crossover and mutation operators are used to generate a *surrogate population S*, of size |*S*|, a multiple of the population size |*P*|. The surrogate approximates objective and constraint values in *S*, and NSGA-II's fast-non-dominated-sort ranks *S* by feasibility and non-dominance, with the highest ranking solutions inserted into the offspring population *R*. *R* is evaluated using the full fitness function (the building performance simulation), and model accuracy is checked to determine if retraining is needed. NSGA-II continues to the next generation as usual.

4.1.3. Model updates

A separate RBFN is used for each objective and constraint, allowing each to be retrained as needed. FPC determines model accuracy for each objective and constraint every generation, using the filtered offspring solutions (Q_{t+1}), after evaluation by the full simulation. (This approach is termed evolution control [14].) The threshold FPC for retraining was 0.7, representing a strong statistical correlation between predicted and true values. The solutions of the most recent generation (both parents and offspring) were used as training data.[1]

4.2. Constraints

In applying an EA to a constrained problem, an important consideration is how constraints are handled alongside the objective values of solutions. This is further complicated by the possibility of approximation error caused by the surrogate resulting in solutions being incorrectly classified as (in)feasible.

Surveys of EA constraint handling approaches [52,53,27] have identified four categories: penalty functions, decoders, special operators and separation of objectives and constraints, the last being used in NSGA-II (classed as a feasibility rule in [27]). The concept (originally proposed by Deb [13,54]) is simple; when comparing solutions (either in tournament selection, or elitism):

1. Of two feasible solutions, choose the one with the best objective function(s).
2. Of an infeasible and a feasible solution, choose the feasible one.
3. Of two infeasible solutions, choose the one with the lowest sum of normalised constraint violations.

Mezura-Montes and Coello [27] describes several new categories of approach, and of these, the present work builds on stochastic ranking and use of multi-objective concepts.

A number of studies have shown that inclusion of infeasible solutions in the population is beneficial (see Section 2). The argument can be made that often infeasible solutions which are almost feasible are valuable stepping stones between feasible regions of the search space. Further, when approximating constraint values, there is a risk that some solutions will be incorrectly classified as infeasible, and it is better to increase the chance that they will be retained.

The method proposed here takes a deterministic approach, referred to as Deterministic Infeasibility Sorting (DIS). The elitist component of NSGA-II was modified to ensure that a fixed proportion α of the population is taken from the set of infeasible solutions, unless the set is smaller than the α-proportion, in which case all are copied into the new population. Like [38,39], after the population is evaluated, it is divided into feasible and infeasible solutions, and the feasible solutions are non-dominating sorted (with crowding distance applied). The infeasible solutions are sorted using NSGA-II's fast non-dominated sort, ignoring the magnitude of their constraint

---

[1] A small number of runs of NSGA-II-S (see Section 4.3) with retraining based on the previous four generations' worth of solutions found no significant difference between that and using only the previous generation.

1: $S =$ make-new-pop(P_{t+1}) #create surrogate pop S, where size $|S| > |P|$
2: objective-and-constraint-approx(S) #approximate obj & con values in S
3: $F_S =$ fast-non-dominated-sort(S) #$F_S = (F_{s1}, F_{s2}, \ldots)$, S nondom fronts
4: $i = 1$
5: **repeat**
6:      crowding-distance-assignment(F_{s_i}) #calc crowding distance in F_{s_i}
7:      $Q_{t+1} = Q_{t+1} \cup S_i$ #include ith nondom front in offspring pop
8:      $i = i + 1$ #check next front for inclusion
9: **until** $Q_{t+1} + F_{s_i} \leq N$ #until offspring pop is filled
10: sort$(F_{s_i} \prec_n)$ #sort in descending order using $\prec_n$
11: $Q_{t+1} = Q_{t+1} \cup F_{s_i}[1 : (N - |Q_{t+1}|)]$ #choose first $(N - |Q_{t+1}|)$ elements of F_{s_i}
12: objective-and-constraint-calc(Q_{t+1}) #full obj & con calc on offspring
13: **for all** objectives and constraints **do**
14:     compute FPC for constraint / objective, the correlation between predicted and true values in Q_{t+1}
15:     **if** FPC < 0.7 **then**
16:         update-model$(P_{t+1} \cup Q_{t+1})$ #retrain model with current generation
17:     **end if**
18: **end for**

Fig. 3. Amended step 11 in NSGA-II.

violations (so, using only objective values). The top $\alpha|P|$ solutions from the ranking are combined with the highest ranking feasible solutions to form the next population of size $|P|$.

This is similar to the approach in [38,39], although simpler. DIS does not substitute constraint violation with an extra objective, instead ignoring constraint violation completely in the infeasible population. Additionally, selection for reproduction still favours feasible solutions. DIS can be applied in two places; in the elitist step of the core algorithm, and in the surrogate filtering step. Both approaches are explored.

4.3. Algorithm variants

The experiments covered several variants of NSGA-II, including different approaches to surrogate and constraint handling. *NSGA-II* refers to the original algorithm of Deb et al. [13]. Use of the surrogate is denoted with suffix -S. DIS, if present in the elitist step of NSGA-II, is denoted by subscript c, and if present in the surrogate filtering step, is denoted by subscript d. Overall, we have:

1. NSGA-II-S: NSGA-II, with RBFN surrogate, and original NSGA-II constraint handling;
2. NSGA-II$_c$: NSGA-II, with DIS at elitist step;
3. NSGA-II-S$_c$: NSGA-II, with RBFN surrogate and DIS at elitist step;
4. NSGA-II-S$_d$: NSGA-II, with RBFN surrogate and DIS at surrogate filtering step;
5. NSGA-II-S$_{cd}$: NSGA-II, with RBFN surrogate and DIS at both elitist and surrogate filtering steps.

Minor changes were made to the base NSGA-II algorithm and all variants. To maximise efficient use of the 5000 evaluation quota, population size was small. An external archive was added to the algorithm to store the Pareto front, allowing a bigger front than the population size to be returned. The algorithm attempts to insert every solution evaluated into the archive, and the same dominance rules used by NSGA-II are used to determine which solutions stay in the archive.

An internal cache of previously-evaluated solutions was maintained for each run, to avoid re-evaluating identical solutions.

Solutions drawn from the cache did not count towards the limit of 5000. Six threads were used to conduct evaluations in parallel.

Optimisation variables were handled as Gray-encoded bitstrings during crossover and mutation to simplify operator choice (bit-flip mutation and uniform crossover). The surrogate operated directly on the unencoded continuous, integer and categorical variable values.

5. Experiments and results

Three sets of experiments compared the performance of the different approaches to constraint handling and the surrogate. Firstly, the algorithm variants were tuned on the Birmingham problem. Secondly, all variants were compared on the Birmingham problem. Thirdly, a subset were compared on the Chicago and San Francisco problems. All runs were restricted to 5000 true evaluations, and comparisons were made of the final Pareto fronts returned. The goal was to find the algorithm setup returning the highest hypervolume and lowest spread metric for the same number of evaluations (i.e., better quality solutions for the same effort). Also given are comparisons of the effort required to reach the same quality of solutions.

5.1. Parameter tuning

Preliminary runs aimed to find good parameters for the algorithm and variants: the parameters tuned and the value for each yielding the highest mean hypervolume is given in Table 2. Each run was limited to five repeats (limited by the time needed for a single run: around one day).

5.2. Comparisons for Birmingham

Each NSGA-II variant was run 30 times: Table 3 gives the mean hypervolume and spread for the Pareto fronts found. To avoid issues with multiple *t*-tests, a Dunnett test for significance using plain NSGA-II as the base-case was performed: the resulting *p*-values are also given. Analysis of variance also found that $P < 0.05$ for the null hypothesis that the results from all runs were drawn from the

Table 2

Parameter setting experiments. The range of parameters, and the value producing the best hypervolume for all algorithms.

Parameter	Values tried	Best value
Population size	15, 20, 30, 60	20
Mutation rate[a]	$0.5/n$, $1.0/n$, $2.0/n$, $4.0/n$	$4.0/n$ (NSGA-II, NSGA-II$_c$), $1.0/n$ (NSGA-II-S variants)
α for constraint method at elitist step	0.1, 0.2, 0.3, 0.4, 0.5	0.2
α for constraint method at surrogate filtering	0.1, 0.2, 0.3, 0.4, 0.5	0.3
Number of offspring for filtering by surrogate (multiple of pop size)	3, 5, 10, 20, 100	3

[a] Mutation rates are $1/n$, n being the number of bits in the Gray encoding.

Table 3

Optimisation results for the Birmingham problem. The mean hypervolume and spread (over 30 runs) of the Pareto front found by the NSGA-II variants. Standard deviations are in subscript. Also given are the p-values from a Dunnett test comparing the results of the variants with plain NSGA-II. Bold figures indicate statistical significance ($p < 0.05$).

Algorithm variant	Hypervolume	p	Spread	p
NSGA-II	$0.849_{0.028}$	n/a	$0.796_{0.088}$	n/a
NSGA-II$_c$	$0.845_{0.022}$	0.969	$0.792_{0.070}$	1.000
NSGA-II-S	$0.856_{0.028}$	0.783	$0.802_{0.092}$	0.998
NSGA-II-S$_c$	$0.860_{0.024}$	0.338	$0.789_{0.076}$	0.996
NSGA-II-S$_d$	$0.881_{0.031}$	**<0.001**	$0.827_{0.052}$	0.342
NSGA-II-S$_{cd}$	$0.867_{0.027}$	**0.034**	$0.779_{0.059}$	0.841

same distribution. Fig. 4 gives the corresponding median summary attainment curves [55]: the curve of points in the objective space reached by at least half of the repeat runs. A comparison was also made with random search, seeking the set of non-dominated feasible solutions from 5000 randomly generated solutions. None of 30 runs of random search found a feasible solution, whereas all runs of the NSGA-II variants found solutions meeting all constraints within 1984 evaluations.

Although an improvement in final solution quality is useful, at least as important is reduced overall run-time while achieving the same quality of solutions. This is particularly true for the building industry, where timely production of plans for tendering is critical, and a speedup of 10–20% could allow an optimisation run to complete overnight rather than postponing a morning's work. For this reason, the original results were reprocessed, taking the median hypervolume reached by NSGA-II (0.8486) as a target, and measuring the number of evaluations taken by each algorithm to find a

Fig. 4. Summary attainment curve for the Birmingham problem.

Table 4

Optimisation results for the Birmingham problem. SR is the percentage of runs for each algorithm variant that were successful (reaching the target hypervolume within the 5000 evaluation budget), and Evals is the mean number of evaluations taken by successful runs to reach a front with the target hypervolume.

Algorithm variant	SR (%)	Evals
NSGA-II	50	4017
NSGA-II$_c$	40	4026
NSGA-II-S	83	3817
NSGA-II-S$_c$	63	4002
NSGA-II-S$_d$	83	3184
NSGA-II-S$_{cd}$	73	3340

Pareto set with that hypervolume. The results are given in Table 4; where a run did not reach the target hypervolume within 5000 evaluations, it was classed as failed.

There are several key points to note. None of the methods improve or worsen spread significantly. This is important because it would be less useful to have a set of solutions with improved hypervolume if they were unevenly distributed, reducing the choices available to the decision maker.

DIS (in NSGA-II$_c$) reduces the performance of NSGA-II, with lower mean hypervolume and higher mean spread, although neither difference is statistically significant. This is reflected in a lower success rate (fewer runs reached the target hypervolume within 5000 evaluations). Further, the median attainment curve for NSGA-II$_c$, while dominating part of that for NSGA-II, is dominated at the high capital cost end by the latter. This point of the front is where the comfort constraints are more likely to be broken, further analysed in Section 7. This indicates that DIS improves search efficiency in parts of the search space with a balanced mixture of feasible and infeasible solutions, but DIS decreases efficiency where the local search space is mostly infeasible. Given that all algorithm variants found feasible solutions within a few hundred evaluations, it would appear that the feasible region of the search space is large and contiguous enough (further observed in Section 7) to avoid needing to retain infeasible solutions to help jump gaps in the feasible region. Instead, DIS adds drag to the search: removing feasible solutions from the population effectively reduces its size with no countering benefit, lowering the search speed.

The mean hypervolume of the sets found by NSGA-II-S is higher than for NSGA-II, though not significantly. This is reflected by an increased success rate, and the median attainment curve of NSGA-II-S dominating that of NSGA-II.

More importantly, NSGA-II-S was further improved by adding DIS – particularly at the filtering stage (NSGA-II-S$_d$), where the difference over NSGA-II is significant. This would appear to be because DIS compensates for constraint approximation errors made by the surrogate: further explored in Section 6. Adding DIS at the elitist step (NSGA-II-S$_c$) also improves final hypervolume, but not significantly. Adding DIS at both the filtering and elitist steps (NSGA-II-S$_{cd}$) is sub-optimal, except at the low-cost end where the curve for NSGA-II-S$_{cd}$ dominates that for NSGA-II-S$_d$ (similar to the slight benefit shown for NSGA-II$_c$ vs NSGA-II). A similar effect occurs when DIS is applied at the elitist step, but when it is already applied at the filtering stage this benefit is lost due to the drag effect described earlier. The median attainment curves show no similar benefit for NSGA-II-S$_c$. It would appear that the issues related to constraint approximation have more impact on efficiency than retaining infeasible solutions in the population.

In practical terms, the real-world benefit of the improved fronts found by NSGA-II-S$_d$ is substantial. Taking the median summary attainment curves for NSGA-II and NSGA-II-S$_d$, and choosing a capital cost about half way along the curve (£380,000), NSGA-II found a solution with an annual energy use of 487,65 kWh, and NSGA-II-S$_d$ found a solution with an annual energy use of 422,34 kWh. This
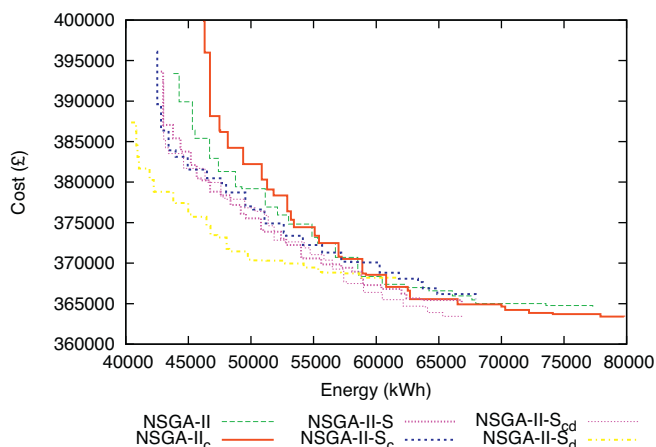
Table 5
Optimisation results for Chicago, formatting matches Fig. 3.

Algorithm variant	Hypervolume	p	Spread	p
NSGA-II	$0.625_{0.107}$	n/a	$0.779_{0.099}$	n/a
NSGA-II-S	$0.716_{0.090}$	**0.001**	$0.805_{0.057}$	0.345
NSGA-II-S_d	$0.784_{0.095}$	**<0.001**	$0.703_{0.055}$	**<0.001**
NSGA-II-S_{cd}	$0.695_{0.093}$	**0.016**	$0.779_{0.060}$	0.999

Table 6
Optimisation results for San Francisco, formatting matches Fig. 3.

Algorithm variant	Hypervolume	p	Spread	p
NSGA-II	$0.727_{0.067}$	n/a	$0.759_{0.070}$	n/a
NSGA-II-S	$0.730_{0.063}$	0.992	$0.810_{0.100}$	0.069
NSGA-II-S_d	$0.733_{0.064}$	0.869	$0.755_{0.069}$	0.999
NSGA-II-S_{cd}	$0.727_{0.041}$	0.990	$0.795_{0.075}$	0.422

saving of 6531 kWh (13%) per annum will reduce carbon emissions and running costs over the lifetime of the building.

5.3. Comparisons for Chicago and San Francisco

Further experiments applied NSGA-II, NSGA-II-S, NSGA-II-S_d and NSGA-II-S_{cd} to the same building, using the Chicago and San Francisco weather, for 30 independent runs. As noted in Section 3, while the basic problem is similar, the search space and optima for these problems are expected to be quite different to those for Birmingham. Compared to the Birmingham climate, Chicago is more demanding, requiring high levels of heating in winter and active/mechanical cooling in summer (the building has to work across the range). San Francisco has a local micro-climate resulting in almost no need for heating and mechanical cooling, the building remaining comfortable with passive operation. Tables 5 and 6 give the mean hypervolume and spread of the Pareto sets found by each algorithm within the 5000 evaluation budget for the two problems. Table 7 give the success rate for each algorithm (proportion of runs achieving the median hypervolume reached by NSGA-II: 0.6095 for Chicago and 0.7334 for San Francisco). Summary attainment curves for Chicago are given in Fig. 5; those for San Francisco are excluded as they were all overlaid over each other and add little to the discussion.

For Chicago, all NSGA-II-S variants found fronts with statistically significantly higher mean hypervolumes than NSGA-II, and NSGA-II-S_d found fronts with significantly lower (better) spread values. This is particularly clear in the median attainment curves; the curves for all variants dominate that for NSGA-II. As with the Birmingham problem, NSGA-II-S_d performs best, with NSGA-II-S and NSGA-II-S_{cd} performing similarly, both better than NSGA-II. The issue seen with Birmingham where the attainment curves coincided for low cost solutions does not appear for Chicago. This is because the more extreme weather conditions require all the

Table 7
Optimisation results for Chicago and San Francisco. SR is the percentage of runs for each algorithm variant that were successful (i.e., reaching the target hypervolume within the 5000 evaluation budget), and Evals is the mean number of evaluations taken by successful runs to reach a front with the target hypervolume.

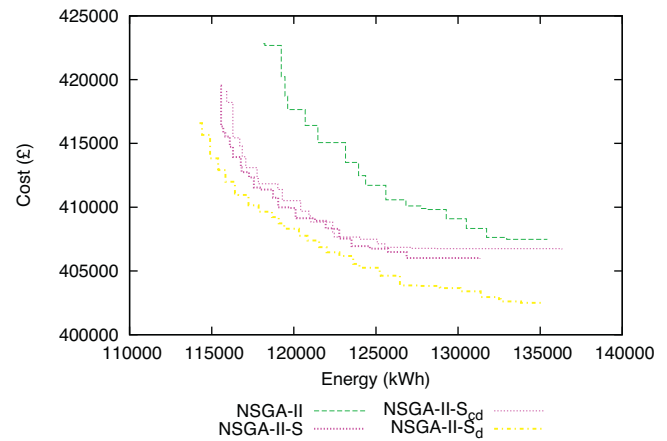| Algorithm variant	Chicago		San Francisco	
	SR (%)	Evals	SR (%)	Evals
NSGA-II	50	3892	50	3620
NSGA-II-S	83	3557	50	3684
NSGA-II-S_d	97	3085	57	3747
NSGA-II-S_{cd}	80	3603	47	3898

Fig. 5. Summary attainment curve for the Chicago problem.

buildings to have heavier construction types (e.g. all solutions have heavy weight external walls, rather than a variety) in order to meet the comfort constraints. This moves the buildings from running close to constraint violation with the lighter construction types to running safely within the constrained region with heavier types, making the impact of DIS less than for Birmingham. This also comes with a higher capital cost for all solutions in the trade-off than for Birmingham.

For San Francisco, all algorithm variants performed similarly. There is no statistically significant difference in the hypervolumes of the fronts found, and success rates are also similar (although NSGA-II-S_d found the target hypervolume within the 5000 evaluation budget slightly more often than the other variants). Indeed, all the algorithms converge at approximately the same rate, demonstrated for NSGA-II and NSGA-II$_d$ in Fig. 6.

In part, the difference in algorithm behaviour for the three problems is due to the constraints. Table 8 shows the mean number of evaluations taken by runs of NSGA-II to find solutions meeting the individual constraints (this is not the time taken to find feasible solutions, which is much longer as all constraints must be simultaneously met). Most constraints for Chicago are harder to solve than for Birmingham, and in turn are harder than those for San Francisco. This implies that where the constraints are easiest, there is least to gain from use of the surrogate.
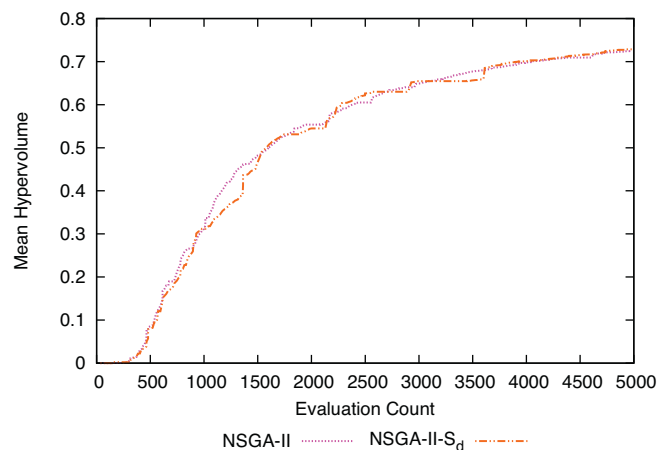

Fig. 6. Hypervolume reached over function evaluations for the San Francisco problem.

Table 8
The number of function evaluations taken by NSGA-II to find solutions meeting the individual constraints for the three problems (B)irmingham, (C)hicago and (S)an Francisco. (constraints, left to right, are in the order given in Table 1).

Constraints 1–9								
B	14	25	23	37	24	89	142	129	82
C	31	38	33	44	27	382	475	528	350
S	11	11	9	18	8	23	39	28	9

Constraints 10–18								
B	13	8	15	16	8	112	38	28	43
C	43	8	8	10	8	120	82	64	58
S	9	8	13	16	8	61	62	142	61

Table 9
The mean hypervolume and spread (over 30 runs) of the Pareto front found by plain NSGA-II, and NSGA-II-S having Hamming weightings of 2/3 and 1. Standard deviations are in subscript. The first two rows duplicate part of Table 3, repeated here for convenience.

Algorithm variant	Hypervolume	Spread
NSGA-II	$0.849_{0.028}$	$0.796_{0.088}$
NSGA-II-S with $\omega = 2/3$	$0.856_{0.028}$	$0.802_{0.092}$
NSGA-II-S with $\omega = 1$	$0.851_{0.024}$	$0.821_{0.078}$
NSGA-II-S_d with $\omega = 2/3$	$0.881_{0.031}$	$0.827_{0.052}$
NSGA-II-S_d with $\omega = 1$	$0.876_{0.037}$	$0.810_{0.075}$

Table 10
p-values for unpaired t-tests between hypervolumes for NSGA-II and NSGA-II-S with the two Hamming weightings. Different p-value (to Table 3) for NSGA-II vs NSGA-II-S arises from use of unpaired t-test rather than Dunnett test.

Algorithm variants	p
NSGA-II vs NSGA-II-S ($\omega = 2/3$)	0.347
NSGA-II vs NSGA-II-S ($\omega = 1$)	0.710
NSGA-II-S ($\omega = 2/3$) vs NSGA-II-S ($\omega = 1$)	0.524
NSGA-II vs NSGA-II-S ($\omega = 2/3$)	<0.001
NSGA-II vs NSGA-II-S_d ($\omega = 1$)	0.003
NSGA-II-S_d ($\omega = 2/3$) vs NSGA-II-S_d ($\omega = 1$)	0.545

5.4. Hamming weighting

The weighting for all NSGA-II-S variants was 2/3, justified in Section 4.1. The optimisation experiment for Birmingham was repeated with NSGA-II-S and NSGA-II-S_d, using a surrogate with a Hamming weighting of 1 (the same as [12]). The mean hypervolume and spread of the Pareto fronts found in 30 independent runs are given in Table 9, alongside results for NSGA-II and NSGA-II-S with the Hamming weighting of 2/3, repeated for convenience. t-tests between the hypervolumes are given in Table 10 (none of the differences for spread were significant).

While hypervolume is higher for the approaches with $\omega = 2/3$ compared to those with $\omega = 1$, the difference is not significant; the change also does not affect whether the hypervolume reached by NSGA-II-S or NSGA-II-S_d is significantly higher than for NSGA-II. There is no significant difference in the spread for the fronts found by any of the algorithms; for NSGA-II-S, setting $\omega = 2/3$ improves spread, but for NSGA-II-S_d setting $\omega = 2/3$ results in poorer spread.

The algorithms with $\omega = 2/3$ did converge more reliably to a front with the target hypervolume of 0.8486 within the 5000 evaluations budget. NSGA-II-S and NSGA-II-S_d reached the target in 83% of the 30 runs. With $\omega = 1$, this dropped to 57% and 67% respectively. This provides some justification beyond the theory in Section 4.1 for using the Hamming weighting of 2/3.

6. Analysis of the constraint handling

This section explores in more detail the success of the surrogate in handling constraints. Section 4.1.2 described how, within NSGA-II-S, the surrogate assigns values to each objective and constraint for each solution in the surrogate population. A fast-non-dominated-sort is conducted, and the top ranking N solutions are evaluated using the full simulation, then combined with the parent population by the elitist step of NSGA-II. To explore this further, for one run of NSGA-II-S on the Birmingham problem, the full simulation was run for all solutions evaluated by the surrogate. This amounted to an assessment of the surrogate's performance over 14,940 solutions. These solutions were divided into four categories for each constraint:

- *Correct fail* – predicted to have violated the constraint, which the simulation confirmed;
- *Incorrect fail* – predicted to have violated the constraint, but had not;
- *Correct pass* – predicted to have met the constraint, which the simulation confirmed;
- *Incorrect pass* – predicted to have met the constraint, but had not.

These can be compared with:

- *True fail* – violates constraint, according to the simulation;
- *True pass* – meets constraint, according to the simulation;
- *Predicted fail* – violates constraint, according to the surrogate;
- *Predicted pass* – violates constraint, according to the surrogate.

The counts of solutions in each category were used to calculate, for each constraint, two measures borrowed from the information retrieval community, precision P and recall R [56]:

$$\text{Precision} = \frac{\text{Correct passes or fails}}{\text{Total predicted passes or fails}} \tag{3}$$

$$\text{Recall} = \frac{\text{Correct passes or fails}}{\text{Total true passes or fails}} \tag{4}$$

Precision is the proportion of surrogate predictions which were correct, and recall is the proportion of true passes or fails that were predicted as such. Ideally, both measures should be 1 meaning that all true passes and fails were correctly predicted as such. P_p and R_p refer to precision and recall for passes, and P_f and R_f for fails. The results are given in Table 11, together with the FPC and number of times the surrogate for each constraint was rebuilt over the run. The abbreviated constraint names are given in Section 3.

P_p exceeds 0.85 for all constraints, except the PMV constraints, which have a low mean FPC (<0.3) and a high number of rebuilds (>95: around one rebuild every third generation). In contrast, P_f is low for most constraints; exceeding 0.5 for only three of the ACH constraints (note that for these, FPC is high and rebuilds are low). This is in part because there are far more true passes than fails, so with a similar error rate, more passes will be incorrectly classed as fails than the other way round. The converse of this is shown for the WPMV and EPMV constraints, where the number of true passes is much closer to the number of true fails, and P_p drops (with an increase in P_f).

The surrogate clearly performs better with some categories of constraint than others:

- For PMVs, pass precision is worse than for the other constraints but not low, and fail precision is not high but better than for SOH and CO_2.
- For SOH and CO_2, pass precision is high, and fail precision is low.
- For ACH, pass precision is high, and fail precision is not high but better than for SOH and CO_2.
- Where mean FPC is low, as expected, rebuild count is high.

Table 11
Summary of the surrogate predictions of the constraints over one run of NSGA-II-S. T_f and T_p are the number of true fails and passes for the constraint. T_p/T is the fraction of all solutions evaluated that passed the constraint (intended to give an impression of the constraint's difficulty). $P_{f,p}$ and $R_{f,p}$ are the precision and recall for solutions passing and failing to meet the constraint respectively. FPC is the mean fitness prediction correlation for the raw constraint value, and C is the model rebuild count for the surrogate on that constraint.

Con	T_f	T_p	T_p/T	P_f	R_f	P_p	R_p	FPC	C
NSOH	892	14,048	0.94	0.07	0.239	0.943	0.798	0.534	59
WSOH	1381	13,559	0.91	0.107	0.482	0.918	0.591	0.518	61
ESOH	1340	13,600	0.91	0.104	0.293	0.915	0.752	0.723	22
SSOH	2680	12,260	0.82	0.18	0.671	0.822	0.332	0.24	97
ISOH	1029	13,911	0.93	0.122	0.277	0.941	0.852	0.92	1
NPMV	3179	11,761	0.79	0.252	0.608	0.828	0.511	0.165	104
WPMV	6475	8465	0.57	0.447	0.78	0.609	0.262	0.149	99
EPMV	6235	8705	0.58	0.46	0.788	0.689	0.336	0.179	106
SPMV	2335	12,605	0.84	0.183	0.561	0.868	0.536	0.287	97
IPMV	542	14,398	0.96	0.097	0.314	0.972	0.89	0.545	53
NCO2	800	14,140	0.95	0.06	0.216	0.948	0.809	0.071	104
WCO2	134	14,806	0.99	0.018	0.06	0.991	0.97	0.045	109
ECO2	275	14,665	0.98	0.039	0.076	0.982	0.964	0.135	107
SCO2	409	14,531	0.97	0.03	0.11	0.973	0.9	0.139	108
NACH	1066	13,874	0.93	0.44	0.093	0.934	0.991	0.889	5
WACH	1308	13,632	0.91	0.825	0.255	0.933	0.995	0.841	9
EACH	1087	13,853	0.93	0.389	0.064	0.931	0.992	0.867	6
SACH	1121	13,819	0.92	0.213	0.06	0.928	0.982	0.856	8

For the ACH constraints, which have higher P_f and around the same P_p compared with the other constraints, the mean FPC is highest and the number of model rebuilds is low. The other constraints, which either have very low P_f (SOH and CO_2) or lower P_p (PMV), have lower mean FPC values, and correspondingly higher rebuild counts. This means that where the model is accurately predicting constraint passes and fails for solutions it is being kept, and where it is less successful it is being rebuilt in an attempt to improve accuracy. These results provide further evidence that the FPC is a reasonable measure of the surrogate's prediction capability, and suitable for choosing when to rebuild it.

6.1. Impact of DIS

As noted above, precision and recall are much higher for passes than fails. In simple terms, for this problem the surrogate is conservative in its prediction of constraint values, tending to falsely fail feasible solutions rather than allowing infeasible solutions into the population. While this may be preferable, it means a large number of solutions with useful genetic material are incorrectly labelled as infeasible and discarded. This motivated the introduction of DIS, which allows some of the solutions predicted infeasible, but also predicted high in the objective values, through the filtering on the good chance that they are (in reality) feasible.

The two DIS variants were: allowing some infeasible solutions to pass from one generation to the next at the elitist step (NSGA-II$_c$); and allowing some solutions predicted infeasible through the surrogate filtering step (NSGA-II-S$_d$). To explore the impact of the latter, the experiment described at the beginning of this section was repeated, but limited to solutions only included in the population because of DIS ("filtered-in" solutions), and the solutions which they replaced ("filtered-out" solutions). As there were far fewer solutions affected by DIS than were presented to the surrogate overall, these results were aggregated from all 30 runs of NSGA-II-S$_d$. Table 12 shows the success of the model in approximating constraints for these two groups of solutions as precision and recall figures. Figures in bold show where the value is lower than that for the whole run in Table 11.

Among filtered-out solutions, relative to the overall figures for precision and recall: 14/18 constraints had lower P_f; 12/18

Table 12
Summary of the surrogate predictions of the constraints for solutions that were lost because they were replaced with infeasible solutions by DIS. $P_{f,p}$ and $R_{f,p}$ are the precision and recall related for solutions passing and failing to meet the constraint respectively. Values in bold are lower than the corresponding values in Table 11.

Constraint	P_f	R_f	P_p	R_p	P_f	R_f	P_p	R_p
NSOH	0.143	**0.125**	0.978	0.981	0.143	**0.125**	0.978	0.981
WSOH	**0.096**	0.302	0.941	0.796	**0.095**	0.295	0.939	0.793
ESOH	**0.047**	0.150	0.971	0.902	**0.047**	0.150	0.971	0.902
SSOH	**0.091**	0.329	0.952	0.802	**0.091**	0.329	0.952	0.802
ISOH	**0.000**	**0.000**	0.974	0.984	**0.000**	**0.000**	0.974	0.984
NPMV	**0.226**	0.686	**0.818**	**0.375**	0.226	0.671	**0.805**	**0.371**
WPMV	**0.429**	0.732	**0.516**	**0.227**	0.429	0.732	**0.516**	**0.227**
EPMV	**0.423**	0.725	**0.552**	**0.254**	0.424	0.738	**0.565**	**0.253**
SPMV	0.267	0.695	**0.811**	**0.407**	0.267	0.686	**0.803**	**0.404**
IPMV	0.124	0.416	**0.956**	**0.813**	0.126	0.413	**0.954**	**0.810**
NCO2	**0.048**	**0.122**	0.963	0.905	**0.048**	**0.122**	0.963	0.905
WCO2	**0.000**	**0.000**	0.991	0.975	**0.000**	**0.000**	**0.990**	0.975
ECO2	0.042	0.077	0.991	0.982	0.042	**0.071**	0.990	0.982
SCO2	**0.000**	**0.000**	0.991	0.954	**0.000**	**0.000**	0.991	0.954
NACH	**0.143**	**0.077**	**0.932**	**0.965**	0.143	**0.076**	**0.931**	0.965
WACH	**0.500**	0.317	**0.921**	**0.962**	0.500	0.319	**0.922**	0.962
EACH	**0.143**	0.090	0.934	**0.960**	0.143	0.091	0.935	**0.960**
SACH	**0.029**	**0.012**	0.934	**0.973**	0.029	0.012	0.935	**0.973**

Table 13
Mean true (rather than surrogate predicted) values for overall constraint violation and the objectives, for filtered-in and filtered-out solutions. Standard deviations are omitted because they mean little – the solutions are from various parts of the Pareto front and have a wide range of objective values.

	Filtered-in solutions	Filtered-out solutions
True energy	97,378	98,468
True cost	399,910	40,224
True overall constraint violation	631,602	135,273

had lower R_f; 8/18 had lower P_p and 9/18 had lower R_p. Among filtered-in solutions, relative to the overall figures for precision and recall: 14/18 constraints had lower P_f; 13/18 had lower R_f; 8/18 had lower P_p and 9/18 had lower R_p. This means that for the solutions nearest to the feasible/infeasible boundary, most constraints were less accurately predicted by the surrogate than was the case for all solutions. This is hardly a surprise: boundary solutions were likely to be the point at which any error in approximation would manifest itself. It does, however, provide some justification for using DIS to swap some solutions at the feasible/infeasible boundary: if the model is less likely to accurately classify these solutions then they may as well be randomly labelled feasible or infeasible, avoiding any unhelpful bias introduced by modelling error.

As noted above, the most common error is predicting feasible solutions as infeasible. An effect of this is that not enough solutions are predicted feasible for DIS to filter them out (enough of the filtered population is already infeasible). Where DIS does filter solutions in and out, comparing true values for filtered-in solutions with those filtered-out showed that the former were higher in overall constraint violation (as intended), higher in the cost objective and lower in the energy objective. 71% of the DIS filtering operations led to an increase in the mean true overall constraint violation for the population; 52% led to a decrease in the mean true energy use of buildings in the population; 53% led to an increase in the mean true capital cost. Mean figures for all filtered-in and filtered-out solutions are given in Table 13. The lower energy objective values among filtered-in solutions appears to contribute to improved performance when using DIS (NSGA-II-S_d), compared with using the surrogate alone (NSGA-II-S). The summary attainment curve for the Birmingham problem (Fig. 4) showed that NSGA-II-S_c was able to find fronts with lower energy values but not as low capital costs as

NSGA-II-S. While the difference is slight, the impact of slightly better solutions would be amplified as the algorithm runs over many generations.

7. Analysis of the fitness landscape

It is helpful to understand the fitness landscape for the problem to determine how difficult it is to model accurately. For single-objective problems, common approaches include fitness distance plots [57] and autocorrelation [57]. Both use a single solution, typically the global optimum, as a starting point. For multi-objective problems, there are multiple solutions in the Pareto front rather than one single global optimum, so the picture is more complex. In the present study, a modified fitness distance plot is used to illustrate the area immediately around the Pareto front.

As the true Pareto front is unknown, a global Pareto front for the problem was constructed by taking the set of non-dominated solutions from the union of all Pareto fronts generated during the original experimental runs. These experiments covered approximately two million function evaluations (480 experimental runs of 5000 solutions, with some duplicate solutions). The resulting front consisted of 120 solutions. For each member of the front, all possible values were tried for each of its variables (given the discretisation caused by Gray encoding), one variable at a time. This amounts to an exhaustive walk around the locality of the Pareto front, covering 55,101 solutions. The results are given in Fig. 7. The points are separated using shape and colour into feasible, infeasible and the original front.

A deeper analysis of the constraints was performed by sorting the solutions generated by the stepwise traversal by ascending capital cost. This set was divided into 10 bands representing increasing cost. The fraction of solutions in each band that were feasible are plotted in Fig. 8.

A number of observations about the landscape can be made.

1. No large improvements in either objective can be found near the global Pareto front without breaking a constraint. This means that the "global" Pareto front is at least near to locally optimal and is useful for comparisons. However, without an (impractical) exhaustive search it is impossible to say that the front is truly, globally, optimal.
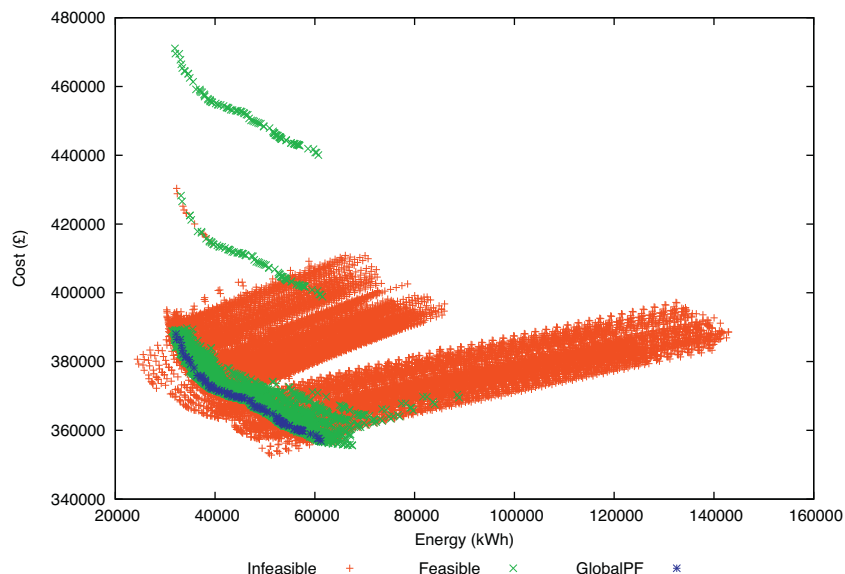

Fig. 7. Fitness distance plot around global Pareto front for the Birmingham problem.
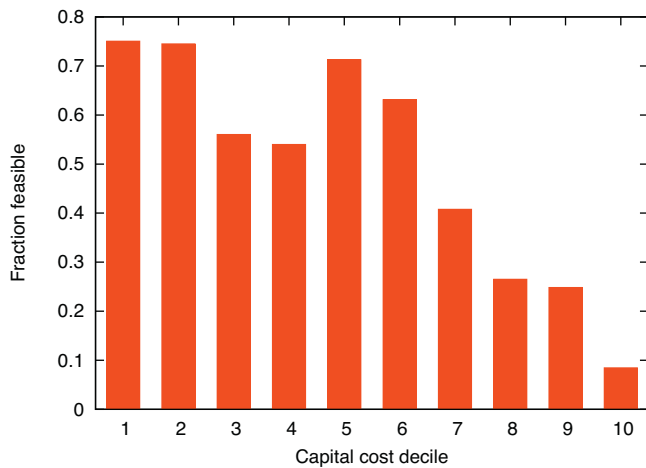
Fig. 8. Fraction of solutions that were feasible, when solutions were grouped into bins of ascending capital cost.

2. 27,896 (50.6%) of the 55,101 solutions in the immediate neighbourhood of the Pareto front are infeasible. This means that while considerable effort could be wasted exploring infeasible solutions, a large proportion of solutions are feasible. The problem would be much harder were the proportion of infeasible solutions higher.

3. The feasible region around the Pareto front is mostly contiguous. This provides some evidence for the conclusion (Section 5) that NSGA-II$_c$ did not offer improved performance because there were not enough gaps in the feasible part of the search space for including infeasible solutions in the population to give much benefit.

4. One variable (ceiling/floor construction type) has a large impact on cost, resulting in two duplicates of the Pareto front which are much higher in cost than most of the other solutions in the plot.

5. There are large changes in the objective values (particularly energy) by simply changing one variable. This means it is harder to model than a "flatter" landscape would be; so more difficult for a surrogate to reproduce.

6. Within the vicinity of the Pareto front, as capital cost increases, solutions are more likely to be infeasible. This is of relevance to the discussion of DIS in Section 5.2.

This experiment was repeated for the San Francisco problem, to attempt to further explain the different optimisation results. The plot is not shown for space reasons, but revealed two points. Firstly, the range in the energy objective across the solutions is much lower for the San Francisco problem meaning that once close to the Pareto front, crossover and mutation are less likely to generate solutions far from it (which would be wasted effort). Secondly, a slightly higher proportion (51.7%) of the solutions around the Pareto front are feasible. If these factors simplify the problem, there may be little room for improvement in efficiency over plain NSGA-II.

8. Conclusions

This paper has described a constrained, multi-objective problem, with mixed variable types, from the building design domain. This application area forms a crucial part of the global response to climate change, and given the time-constrained nature of the industry it is important that optimisation approaches are able to find a high-quality solution or set of solutions as quickly as possible. To meet this demand, the paper has proposed several variants of NSGA-II, incorporating RBFN surrogates and an approach to allowing infeasible solutions to remain in the population. These variants were applied to three instances of the building problem (using different locations for the building weather data). Optimisation runs were limited to a budget of 5000 function evaluations, and compared both on the quality of solutions found (how close the set was to the Pareto-optimal set) and on the rate of convergence relative to plain NSGA-II.

The paper has made a number of contributions. It has proposed using a separate RBFN surrogate for each constraint and objective to allow for retraining. To control the retraining the Fitness Prediction Correlation (FPC) is used. An existing approach to using RBFNs as surrogates has been modified using the Hamming weighting and k-medoid clustering, to better accommodate categorical variables. Deterministic Infeasibility Sorting (DIS), a method for including infeasible solutions in the population, has been proposed, and also used to mitigate surrogate approximation errors. It was found that the optimal configuration for this problem was NSGA-II-S_d: NSGA-II, with the surrogate, and DIS applied at the filtering step. This reliably found better sets of solutions than NSGA-II, more quickly, for two of the building problem instances. None of the algorithms showed much difference in performance for the third (easier) problem instance. The paper has also explored in more detail the effect on optimisation of the constraint handling approach and the Hamming weighting, and explored the landscape of the fitness functions to add context to the optimisation results.

There remain some open areas for future work to expand on that described here. Aside from the obvious routes of exploring more applications or alternative algorithms to NSGA-II, one area which will be more difficult to consider is adapting the surrogate based algorithm to cope with equality constraints. There are very few examples of this in the literature, one exception being [58]. It would be interesting to explore further the frequency with which model building for each of the objectives and constraints recurs and either use this to switch to different surrogate model types or to weight the constraints at the surrogate filtering stage to help mitigate for approximation error.

Acknowledgement

Appendix A. Decision variables

Table 14

Table 14
Summary of variables; ID numbers are used in subsequent references to individual variables in the paper.

ID	Name	Notes	Range &step size
Continuous variables			
1	HVAC heating set-point	Applies to all zones	18–21, step 0.5
2	HVAC cooling set-point		22–29, step 0.5
3	MechVentMinSPFrac	Controls the band of temperatures where mechanical rather than natural ventilation is used;	0–1.0, step 0.1
4	MechVentMaxSPFrac	applies to all zones	0–1.0, step 0.1
5	NatVent Delta	Minimum difference between indoor and outdoor temperature for ventilation to be switched on	0–10, step 0.5
6	Orientation	Building is symmetrical, so rotation of 90 degrees covers all orientations	0–90, step 5
7	North upper height ratio	Each perimeter zone has two horizontal windows, to simulate a single window which can	0.4–0.7, step 0.05
8	North lower height ratio	open at top and bottom to produce a "stack effect" to drive natural ventilation. These control	0.5–1.0, step 0.05
9	North upper area ratio	the height proportional to the floor-to-ceiling height, and the glazed area as a proportion of	0.1–0.9, step 0.05
10	North lower area ratio	the wall, for each window.	0.1–0.9, step 0.05
11	South upper height ratio		0.4–0.7, step 0.05
12	South lower height ratio		0.5–1.0, step 0.05
13	South upper area ratio		0.1–0.9, step 0.05
14	South lower area ratio		0.1–0.9, step 0.05
15	East upper height ratio		0.4–0.7, step 0.05
16	East lower height ratio		0.5–1.0, step 0.05
17	East upper area ratio		0.1–0.9, step 0.05
18	East lower area ratio		0.1–0.9, step 0.05
19	West upper height ratio		0.4–0.7, step 0.05
20	West lower height ratio		0.5–1.0, step 0.05
21	West upper area ratio		0.1–0.9, step 0.05
22	West lower area ratio		0.1–0.9, step 0.05
23	North open area	The proportion of each window which can open for ventilation	0.1–2.3, step 0.05
24	South open area		0.1–2.3, step 0.05
25	East open area		0.1–2.3, step 0.05
26	West open area		0.1–2.3, step 0.05
27	North mech vent rate	Mechanical ventilation rate in l/s/person	0.06–0.24, step 0.02
28	South mech vent rate		0.06–0.24, step 0.02
29	East mech vent rate		0.06–0.2, step 0.02
30	West mech vent rate		0.06–0.2, step 0.02
Integer variables			
31	Winter morning start	Operation times for HVAC systems in perimeter zones (24 h clock)	1–8, step 1
32	Winter evening stop		17–23, step 1
33	Summer morning start		1–8, step 1
34	Summer evening stop		17–23, step 1
35	Winter morning start	Operation times for HVAC systems in internal zone (24 h clock)	1–8, step 1
36	Winter evening stop		17–23, step 1
37	Summer morning start		1–8, step 1
38	Summer evening stop		17–23, step 1
Categorical and binary variables			
39	External wall type	From [59]	Heavy, medium, light weight[a]
40	Internal wall type	From [59]	Heavy, light weight
41	Floor and ceiling type	From [59]	Heavy, medium, light weight
42	North upper overhang	Presence or omission of shading overhangs above the two rows of glazing in each zone	Present, not present
43	North lower overhang		Present, not present
44	South upper overhang		Present, not present
45	South lower overhang		Present, not present
46	East upper overhang		Present, not present
47	East lower overhang		Present, not present
48	West upper overhang		Present, not present
49	West lower overhang		Present, not present
50	Glazing type	Types from [59]; applies to all zones	Standard, low-emissivity

[a] The weight refers to the effect that the materials have on the thermal response of the building, heavy being slow and light being fast.

References

[1] Part L Explained: The BRE Guide, HIS BRE Press, 2006.

[2] M. Wetter, J. Wright, A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization, Build. Environ. 39 (8) (2004) 989–999.

[3] J.A. Wright, H.A. Loosemore, R. Farmani, Optimization of building thermal design and control by multi-criterion genetic algorithm, Energy Build. 34 (9) (2002) 959–972, http://dx.doi.org/10.1016/S0378-7788(02)00071-3

[4] 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, http://dx.doi.org/10.1080/19401493.2012.762808

[5] L.G. Caldas, L.K. Norford, Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems, J. Sol. Energy Eng. 125 (3) (2003) 343–351.

[6] 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, 2010, pp. 2003–2006.

[7] D.A. Coley, S. Schukat, Low-energy design: combining computer-based optimisation and human judgement, Build. Environ. 37 (12) (2002) 1241–1247.

[8] P. Geyer, Component-oriented decomposition for multidisciplinary design optimization in building design, Adv. Eng. Inform. 23 (1) (2009) 12–31.

[9] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers, 2001.

[10] R.E. Smith, B.A. Dike, S.A. Stegmann, Fitness inheritance in genetic algorithms, in: SAC' 95: Proc. of the 1995 ACM Symp. on Applied Computing, ACM Press, New York, NY, USA, 1995, pp. 345–350, http://dx.doi.org/10.1145/315891.316014

[11] Y. Jin, Surrogate-assisted evolutionary computation: recent advances and future challenges, Swarm Evol. Comput. 1 (2) (2011) 61–70, http://dx.doi.org/10.1016/j.swevo.2011.05.001

[12] R. Li, M. Emmerich, J. Eggermont, E. Bovenkamp, T. Back, J. Dijkstra, J. Reiber, Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis, in: Proc. IEEE WCCI, 2008, pp. 2764–2771, http://dx.doi.org/10.1109/CEC.2008.4631169

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

[14] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Comput. 9 (1) (2005) 3–12.

[15] Y. Tenne, C.-K. Goh, Computational Intelligence in Expensive Optimization Problems, 1st Ed., Springer, Berlin, 2010.

[16] E.I. Ducheyne, B. De Baets, R.R. De Wulf, Fitness inheritance in multiple objective evolutionary algorithms: a test bench and real-world evaluation, Appl. Soft Comput. 8 (1) (2008) 337–349, http://dx.doi.org/10.1016/j.asoc.2007.02.003

[17] M.D. Schmidt, H. Lipson, Coevolution of fitness predictors, IEEE Trans. Evolut. Comput. 12 (6) (2008) 736–749.

[18] M. Davarynejad, C.W. Ahn, J. Vrancken, J. van den Berg, C.A. Coello Coello, Evolutionary hidden information detection by granulation-based fitness approximation, Appl. Soft Comput. 10 (2010) 719–729, http://dx.doi.org/10.1016/j.asoc.2009.09.001

[19] A. Syberfeldt, H. Grimm, A. Ng, R.I. John, A parallel surrogate-assisted multiobjective evolutionary algorithm for computationally expensive optimization problems, in: J. Wang (Ed.), Proc. IEEE WCCI, IEEE Press, Hong Kong, 2008, pp. 3177–3184.

[20] A.-C. Žavoianu, G. Bramerdorfer, E. Lughofer, S. Silber, W. Amrhein, E.P. Klement, Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives, Eng. Appl. Artif. Intell. 26 (8) (2013) 1781–1794, http://dx.doi.org/10.1016/j.engappai.2013.06.002

[21] R.G. Regis, Particle swarm with radial basis function surrogates for expensive black-box optimization, J. Computat. Sci. 5 (1) (2014) 12–23, http://dx.doi.org/10.1016/j.jocs.2013.07.004

[22] S.H. Han, H. Yang, Screening important design variables for building a usability model: genetic algorithm-based partial least-squares approach, Int. J. Ind. Ergonom. 33 (2) (2004) 159–171, http://dx.doi.org/10.1016/j.ergon.2003.09.004

[23] Z. Zhou, Y.S. Ong, M.H. Nguyen, D. Lim, A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm, in: Proc. IEEE CEC, 2005, pp. 2832–2839, http://dx.doi.org/10.1109/CEC.2005.1555050

[24] M. Emmerich, K. Giannakoglou, B. Naujoks, Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels, IEEE T. Evolut. Comput. 10 (4) (2006) 421–439, http://dx.doi.org/10.1109/TEVC.2005.859463

[25] I. Tsoukalas, C. Makropoulos, Multiobjective optimisation on a budget: exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty, Environ. Modell. Softw. (2015), http://dx.doi.org/10.1016/j.envsoft.2014.09.023 (http://www.sciencedirect.com/science/article/pii/S1364815214002874).

[26] A. Ochoa, Opportunities for expensive optimization with estimation of distribution algorithms, in: L.M. Hiot, Y.S. Ong, Y. Tenne, C.-K. Goh (Eds.), Computational Intelligence in Expensive Optimization Problems, Vol. 2 of Adaptation, Learning, and Optimization, Springer, Berlin, 2010, pp. 193–218, http://dx.doi.org/10.1007/978-3-642-10701-6.8

[27] E. Mezura-Montes, C.A.C. Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, Swarm Evol. Comput. 1 (4) (2011) 173–194, http://dx.doi.org/10.1016/j.swevo.2011.10.001

[28] D. Wallin, C. Ryan, in: Proc. IEEE CEC, Using over-sampling in a Bayesian classifier EDA to solve deceptive and hierarchical problems (2009) 1660–1667, http://dx.doi.org/10.1109/CEC.2009.4983141

[29] A.E.I. Brownlee, O. Regnier-Coudert, J.A.W. McCall, S. Massie, S. Stulajter, An application of a GA with Markov network surrogate to feature selection, Int. J. Syst. Sci. 44 (11) (2013) 2039–2056, http://dx.doi.org/10.1080/00207721.2012.684449

[30] A. Brownlee, J. McCall, Q. Zhang, Fitness modeling with Markov networks, IEEE Trans. Evolut. Comput. 17 (6) (2013) 862–879, http://dx.doi.org/10.1109/TEVC.2013.2281538

[31] S.-C. Horng, S.-Y. Lin, Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation, Inform. Sci. 233 (2013) 214–229, http://dx.doi.org/10.1016/j.ins.2013.01.024

[32] B. Yuan, B. Li, T. Weise, X. Yao, A new memetic algorithm with fitness approximation for the defect-tolerant logic mapping in crossbar-based nanoarchitectures, IEEE Trans. Evolut. Comput. 18 (6) (2014) 846–859, http://dx.doi.org/10.1109/TEVC.2013.2288779

[33] E. Tresidder, Y. Zhang, A.I.J. Forrester, Optimisation of low-energy building design using surrogate models, in: Proc. of the Building Simulation 2011 Conf, Sydney, Australia, 2011, pp. 1012–1016.

[34] G. Zemella, D.D. March, M. Borrotti, I. Poli, Optimised design of energy efficient building façades via evolutionary neural networks, Energ. Buildings 43 (12) (2011) 3297–3302, http://dx.doi.org/10.1016/j.enbuild.2011.10.006

[35] L. Bajer, M. Holeňa, Surrogate model for continuous and discrete genetic optimization based on RBF networks, in: Proc. of the 11th Int Conf. on Intelligent data engineering and automated learning, IDEAL'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 251–258, URL http://portal.acm.org/citation.cfm?id=1884499.1884530.

[36] T. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE T. Evolut. Comput. 4 (3) (2000) 284–294, http://dx.doi.org/10.1109/4235.873238

[37] T.P. Runarsson, Constrained evolutionary optimization by approximate ranking and surrogate models, in: Proc. of 8th Parallel Problem Solving From Nature, LNCS Vol. 3242, Springer, UK, 2004, pp. 401–410.

[38] A. Isaacs, T. Ray, W. Smith, Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems, in: Proc. IEEE WCCI, 2008, pp. 2780–2787, http://dx.doi.org/10.1109/CEC.2008.4631171

[39] T. Ray, H. Singh, A. Isaacs, W. Smith, Infeasibility driven evolutionary algorithm for constrained optimization, in: E. Mezura-Montes (Ed.), Constraint-Handling in Evolutionary Optimization, Vol. 198 of Studies in Computational Intelligence, Springer, Berlin, 2009, pp. 145–165, http://dx.doi.org/10.1007/978-3-642-00619-7.7

[40] R. Farmani, J. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Trans. Evolut. Comput. 7 (5) (2003) 445–455, http://dx.doi.org/10.1109/TEVC.2003.817236

[41] D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisherm, M. Witte, J. Glazer, EnergyPlus: creating a new generation building energy simulation program, Energy Build. 33 (4) (2001) 319–331.

[42] D. Langdon, Spon's Architects' and Builders' Price Book, 137th Ed., Taylor and Francis, 2012.

[43] A. Brownlee, J. Wright, Solution analysis in multi-objective optimization, in: Proc. Building Simul. and Optim. Conf., IBPSA-England, Loughborough, UK, 2012, pp. 317–324.

[44] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.

[45] A. Moraglio, A. Kattan, Geometric generalisation of surrogate model based optimisation to combinatorial spaces, in: EvoCOP, 2011, pp. 142–154.

[46] R. Regis, C. Shoemaker, Local function approximation in evolutionary algorithms for the optimization of costly functions, IEEE Trans. Evolut. Comput. 8 (5) (2004) 490–505, http://dx.doi.org/10.1109/TEVC.2004.835247

[47] A. Giotis, M. Emmerich, B. Naujoks, K. Giannakoglou, T. Bäck, Low cost stochastic optimisation for engineering applications, in: Proc. Intl Conf. Industrial Applications of Evolutionary Algorithms, Int. Center for Numerical Methods in Engineering (CIMNE), 2000.

[48] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, J. Artif. Int. Res. 6 (1997) 1–34, http://dl.acm.org/citation.cfm?id=1622767.1622768.

[49] A.E.I. Brownlee, J.A.W. McCall, Q. Zhang, D. Brown, Approaches to selection and their effect on fitness modeling in an estimation of distribution algorithm, in: Proc. IEEE WCCI, IEEE Press, Hong Kong, China, 2008, pp. 2621–2628.

[50] T. Lucey, Quantitative Techniques, 6th Ed., Thomson Learning, London, 2002.

[51] M.K. Karakasis, K.C. Giannakoglou, On the use of metamodel-assisted, multi-objective evolutionary algorithms, Eng. Optimiz. 38 (8) (2006) 941–957, http://dx.doi.org/10.1080/03052150600848000

[52] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evol. Comput. 4 (1996) 1–32, http://dx.doi.org/10.1162/evco.1996.4.1.1

[53] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Method Appl. Mech. 191 (11-12) (2002) 1245–1287, http://dx.doi.org/10.1016/S0045-7825(01)00323-1

[54] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Method Appl. Mech. 186 (2-4) (2000) 311–338, http://dx.doi.org/10.1016/S0045-7825(99)00389-8

[55] 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, 2005, pp. 552–557, http://dx.doi.org/10.1109/ISDA.2005.15

[56] I.H. Witten, E. Frank, Data Mining Practical Machine Learning Tools and Techniques, 2nd Ed., Morgan Kaufmann Series in Data Management Sys, Morgan Kaufmann, 2005.

[57] H. Hoos, T. Stützle, Stochastic Local Search: Foundations and Applications, Elsevier Science Inc., San Francisco, CA, 2005.

[58] G. Peconick, E. Wanner, R. Takahashi, Projection-based local search operator for multiple equality constraints within genetic algorithms, in: Proc. IEEE CEC, 2007, pp. 3043–3049, http://dx.doi.org/10.1109/CEC.2007.4424859

[59] ASHRAE, Handbook of Fundamentals, 2005, Ch. 30, tbls 19 and 22.