

Multi-objective design optimisation: getting more for less

1 **Ralph Evins** MEng

Research engineer, Buro Happold and the University of Bristol, UK

2 **Sam Conrad Joyce** MEng

Analyst, Buro Happold and research engineer, University of Bath, UK

3 **Philip Pointer** MSc

Senior engineer, Buro Happold, UK

4 **Shrikant Sharma** BEng, MTech, PhD, CEng, MIMechE

Associate director, Buro Happold, UK

5 **Ravi Vaidyanathan** MSc, PhD

Senior lecturer, Imperial College London, UK

6 **Chris Williams** MA, PhD, MStructE, CEng

Senior lecturer in structural engineering, University of Bath, UK



Multi-objective optimisation can help civil engineers achieve higher performance for lower costs in their designs. This is true whether 'performance' applies to structural strength or energy use, or whether 'cost' measures financial outlay or occupant satisfaction: if it can be quantified it can be optimised in some form. By exploring trade-offs between conflicting objectives and constraints, multi-objective optimisation enables informed decision-making. This paper outlines the principles and benefits of multi-objective optimisation and the means of implementation. The complementary aspects of parametric modelling and optimisation are discussed as an aid to the flexible design of buildings and structures. A range of real design problems are considered, including structural and environmental examples.

1. Introduction

Current civil engineering design practice is epitomised by an 'informed trial-and-error' approach to optimisation: 'designs are still optimised mostly through a manual iterative process' (Roy *et al.*, 2008). There is often the potential to add significant value by using more explicit methods to explore the design space. Other industries (e.g. aerospace) have long taken advantage of a more rigorous approach to engineering design optimisation and this trend is now beginning to take hold in civil engineering. Academic examples cover a wide range of applications, including optimising structural design (Koumoussis and Georgiou, 1994), geotechnical performance (Zolfaghari *et al.*, 2005), building form (Marks, 1997), fabric properties (Wang *et al.*, 2005), heating, ventilation and air-conditioning systems design (Fong *et al.*, 2006) and control (Huang and Lam, 1997).

All practicing civil engineers will recognise the description of 'a complex, multi-disciplinary engineering activity that requires making difficult compromises to achieve a balance between competing objectives' (Ren *et al.*, 2011). At a fundamental level, there is a

need to consider all sub-domains of the field and their impact on the overall design. At a broader level, there is a need for a holistic consideration of design and context. For example, the design of a new office building might address the impact of business practice on space requirements, commuting distances in relation to site selection and mixed-use development to allow a site-wide energy scheme to improve energy efficiency.

Applying multi-objective optimisation methods requires careful consideration of the system in question. It is not practical to consider all sub-systems and variables simultaneously; the formulation of system boundaries such that some things are varied while others are held constant is of critical importance. The system to be optimised is defined by objectives, variables to adjust and constraints that must be maintained.

2. Multi-objective optimisation

Consider a generic structural problem concerned with strength and cost. These two objectives are in conflict: a solution may be 'cheap but weak', 'strong but expensive', or anywhere in between. These

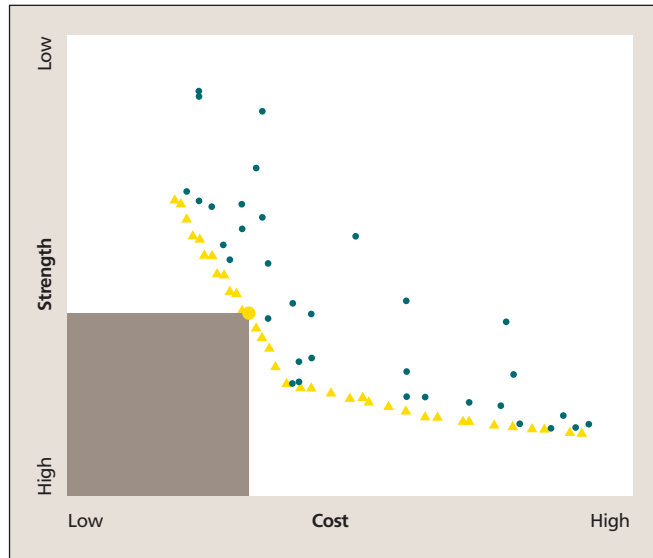


Figure 1. An example trade-off front or set of optimal points – yellow solutions are optimal; blue are not

two objectives are shown on the axes of Figure 1 (by convention, both are to be minimised). There exists a set of possible solutions all of which are optimal for some trade-off between strength and cost. The purpose of multi-objective optimisation is to find this set of optimal solutions (the yellow points in Figure 1), referred to as the trade-off front or Pareto front.

What makes optimal solutions distinct from non-optimal alternatives? For optimal solutions, there exists no other solution that is better in all objectives; that is, in this example, there is no solution that is both cheaper and stronger. If such a solution existed, it would clearly be preferred. In Figure 1, this is illustrated for the red solution: there are no points in the grey area, so this solution is part of the optimal set.

It is often not possible to say in advance where on the trade-off front it is most desirable to be. It is not always possible to specify the importance of each objective or to combine them into a single objective by applying weights. Exploration of the shape of the trade-off front allows informed decision-making regarding marginal benefits. The example in Figure 1 contains a distinct kink – the marginal increase in strength for a unit increase in cost changes

dramatically at this point. The aim of multi-objective optimisation is to discover the entire trade-off front; solutions should be well distributed along the front rather than occupying only a small niche.

There are many means of accomplishing the goals of multi-objective optimisation. The most well-known is the genetic algorithm, a form of evolutionary computation; other methods operate along similar lines, for example differential evolution, evolutionary strategies and genetic programming. These algorithms were inspired by Darwinian evolution or ‘survival of the fittest’. They mimic competition for survival among a ‘population’ of many ‘individuals’, each corresponding to a particular solution to the problem. Each individual possesses a certain ‘fitness’, which is measured against the objectives. Competition is enforced by eliminating individuals of predominantly poorer fitness, thus causing the fitness of the population to improve over time.

For a problem with a single objective, fitness can simply be proportional to the performance of the solution against the objective. Fitness assignment is more complicated for the multi-objective case. A variety of methods exist, generally using the principle of distance from the current trade-off front. One popular technique assigns the highest rank to solutions in the overall trade-off front, then removes these from contention and recalculates the next front, which is assigned the next rank (Deb *et al.*, 2002).

As well as a means of preferring solutions of higher fitness, individuals must also be encouraged to change over time in order to fully explore the problem domain. The genetic algorithm achieves this using two operators that mimic biological processes: the crossover between individuals and random mutation. The former involves splicing characteristics of two individual into new combinations to allow inheritance of good characteristics; the latter randomly alters values of an individual, in order to explore the search space more widely. By repeatedly performing the process of alteration and selection, the population improves with each subsequent ‘generation’; an ‘elite population’ can also be derived, consisting of the best individuals from all generations. A schematic illustration is shown in Figure 2.

There is great flexibility with regard to how to encode a problem for solving by such algorithms. Variables can consist of binary, integer or real numbers, or tree-structures that can represent operations, ordered graphs or computer programs. These algorithms have been implemented in many programming languages and platforms and both free and commercial packages are available. The algorithms can be configured to repeatedly call external programs that perform

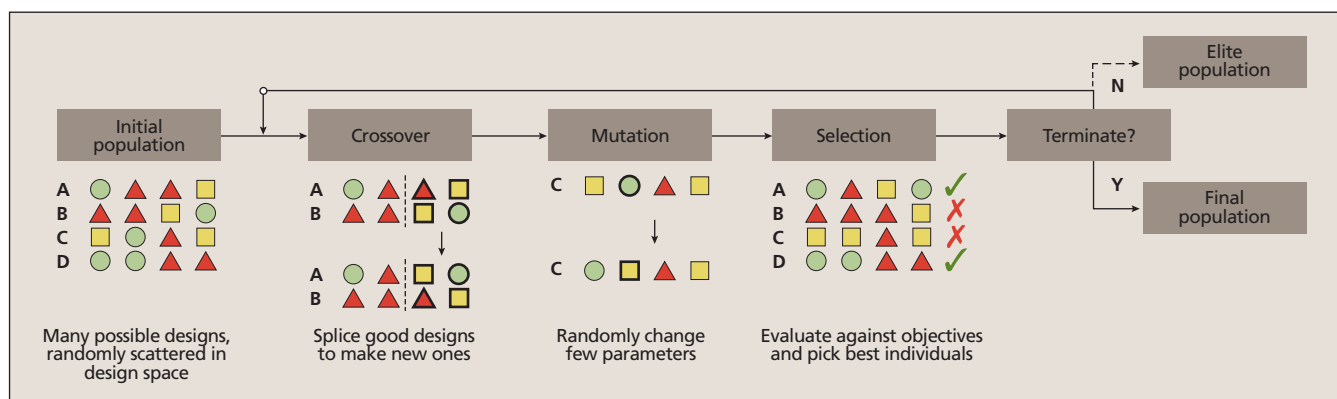


Figure 2. Schematic illustration of a simple genetic algorithm – the operators are illustrated for four individuals A–D

simulations for particular sets of variables and return output values that quantify performance against objectives.

3. Parametric design

Parametric design is a developing term used to encapsulate a method of design that involves using computational processes to define form. Its role within the design world is growing as firms are becoming increasingly aware of the benefits that automated techniques provide over other approaches. The capabilities of the computer provide a significant step change in the efficiency of the design process. It is now recognised that parametric design could further supplement current techniques by providing more holistic and adaptable tools that are aligned both with computational process and natural design thinking.

Computer-aided design (CAD) as a design tool has many advantages over traditional hand-draughting methods as it provides conveniences such as undo functions and cutting and pasting of information. Whereas these increase the speed of the drawing process, the CAD file is essentially a digital reproduction of conventional draughting information. The main issue with this is that the information is that of complete exception, where every mark is unique and the model has no intelligence about the relationships between items (Coenders, 2009).

Parametric design has the potential for a greater impact on the design process by capturing the design rationale rather than a static design drawing. It uses associative relationships to ensure that the logic of the design is embedded within the model – any readjustment and thus regeneration of the design uses this to automatically update the final output. The key concept is to create geometry that has logical associative links, such as the position of a beam being dependent upon the top points of the columns by which it is supported. Normally, this is implemented by way of a hierarchical system where basic geometry is built up and developed until a complex representative model is produced. This is done using

logical and geometric operations following computer programming principles and CAD software capabilities respectively.

The resulting power of this system over conventional CAD is twofold. First, the generation of a model can be linked to input values or parameters (hence the name). For example, a series of input parameters can be identified for various properties of a design such as the number of floors in a building, the length of shading overhang, the ratio between member length and diameter and so on. These parameters can then be modified, allowing for a high level of design flexibility, with options generated and modelled sequentially. This allows the user to design flexibility into a model where values are uncertain or variable.

The second feature is that the inbuilt logic of the design system does not change, irrespective of parameter values. The model will adapt based on the established rules and new variants produced with a change in parameters. The result is a ubiquitous and adaptive design system that can lever computational power to offer more possibilities in less time in comparison with conventional CAD.

Whereas the parametric design approach has been in existence for some time, the process has not yet been refined in its entirety. For example, parametric modelling was adopted for the generation of the roof of the British Museum in 2001, but this required highly skilled specialists with programming skills to make designs in this way (Williams, 2001). The complexity of this approach presents considerable barriers for employment of parametric methods to all but highly specialist teams.

The parametric approach has been made more accessible by computer programs that enable the creation of associative models in more intuitive ways. Generative Components (www.bentley.com/getgc) and Grasshopper (www.grasshopper3d.com), produced by Bentley and McNeel respectively, are implementations of parametric CAD software. They use network graphs to aid in the creation and visualisation of the associative links that govern the design (Figure 3) as well as allowing real-time update of the design model as changes are

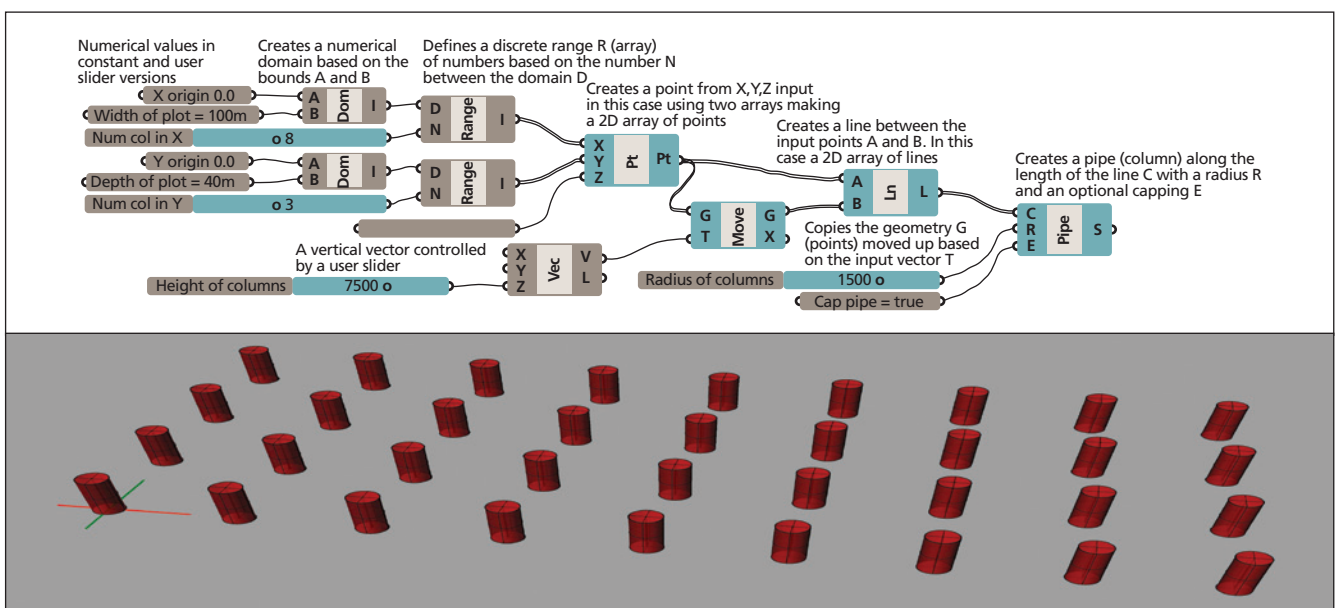


Figure 3. Example of an associative model visualised in parametric software. The model produces a regular grid of columns with number of columns in X and Y as well as their height and width as adjustable parameters. The output of the associative model for one configuration of parameters is shown below the model sketch

An optimisation based on a genetic algorithm was employed to generate the geometry of the trusses and then perform structural analysis under self-weight and wind loading conditions

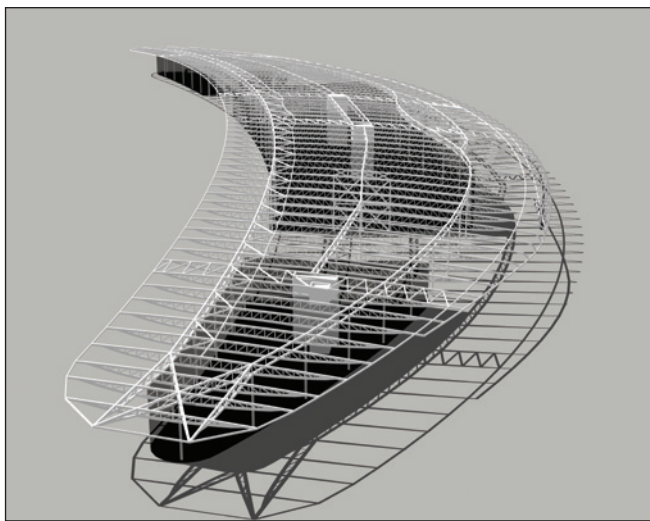


Figure 4. Perspective of truss geometry within ceiling cavity

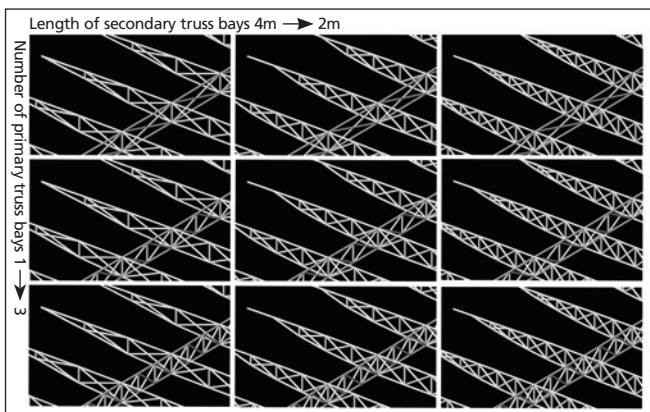


Figure 5. Two-dimensional mapping of examples of the two parameters of the model along the vertical and horizontal axis

made to parameters. This increased accessibility means that it is now practical for most companies to implement this method within their design processes, for either discrete elements or an overall design. This has translated into a greater adoption of parametric design processes on real projects such as the 2010 Aviva Stadium in Dublin, (Shepherd and Hudson, 2007) and others (Hesselgren *et al.*, 2007).

It is worth noting that, at the time of writing, the two principal programs for the generation of parametric designs are essentially free. It is thus now considerably more practical and expedient to learn and implement parametric modelling within design practice.

4. Case studies

4.1 Structural design

The first case study shows the use of parametric design to explore initial structural solutions for a large roof canopy. The positions of the truss elements were defined in plan based upon the requirement for coordination between the glazed facade and the roof. The structural elements needed to be situated in a predefined volume between roof and ceiling cladding. The aim was to produce an efficient truss, such as the one shown in Figure 4. Here, the secondary trusses can be seen as those that span along the short length of the roof and the primary trusses are those that intersect the secondary trusses (typically twice) and follow the glazing line of the building. For this specific design, the section sizes were already determined by previous constraints.

The design was driven by two principal parameters, both controlling the spacing of truss bays, which are defined here as one 'X' arrangement of webs between the chords. The first parameter was the number of bays on a primary truss span and the spacing of bays for the secondary trusses. This allowed a straightforward formulation as an optimisation problem with two variables: one continuous variable (bay spacing, between 2 m and 4 m) and an integer variable (number of bays per primary truss span, between one and four). This two-dimensional design space is represented in Figure 5. A parametric system was set up to define the problem geometrically; this allowed logical decisions to be encoded regarding design aspects of the truss, which would vary with the parameters, as well as generating a flexible automated model. One example of the in-built logic can be seen at the tips of the secondary trusses, which either terminate as points or beams depending on the minimum practical truss depth.

An optimisation based on a genetic algorithm was employed to generate the geometry of the trusses and then perform structural analysis under self-weight and wind loading conditions. The multi-objectives of weight and maximum deflection were chosen as fitness measures to be minimised. The final design chosen possessed the lowest weight for the truss structure, satisfying the allowable serviceability deflection limit and taking into account all possible combinations of main and secondary truss dimensions.

4.2 Low-carbon-dioxide housing design

The second case study concerns a residential project in Scotland subject to stringent carbon dioxide emissions and financial requirements. The development consisted of 1500 dwellings on a south-facing rural site. A mix of dwelling sizes and types and the layout of these based on architectural considerations provided a fixed development plan.

A key requirement was a 60% improvement over the carbon dioxide emissions target set by building regulations. There was also

a specific limit on the energy use for space heating to avoid solutions that had high energy use counterbalanced by high renewable energy provision. The developer obviously wished to meet the targets in as cost-effective a manner as possible and, in addition, it was necessary to ensure that there was not an excessive risk of overheating in the summer.

This problem was formulated as an optimisation of two objectives – carbon dioxide emissions and cost. The carbon dioxide objective was the percentage by which the dwelling emission rate exceeded the target emissions rate. Carbon dioxide emissions were evaluated using the Standard Assessment Procedure (SAP), the methodology used in England and Wales for building regulations compliance for domestic buildings (DECC, 2011). The cost objective combined capital cost with running costs over 20 years, and was based on data from the cost consultant to ensure it was appropriate to the project. Finally, constraints were imposed to ensure that the overheating risk for all designs was low or moderate and that the space heating requirement was met; both were calculated by SAPs.

The optimisation algorithm used was NGA-II (Deb *et al.*, 2002), one of the most popular multi-objective genetic algorithms. This was implemented in VBA for Microsoft Excel to facilitate interaction with the SAP calculations in Excel.

Seven variables were included in the optimisation, each taking a discrete value from a predetermined range; all other parameters were set to constant values. The variables chosen addressed

- fabric properties (areas of glazing, insulation and air-tightness)
- heating system (selected from four options – gas, air-source heat pump, solid fuel burner, community biomass)
- renewable energy provision by way of photovoltaics and solar thermal hot water.

The results presented give an example of an optimal set of designs; these are only valid for the context used here, as defined by the constants used for all other parameters in the methodology.

Figure 6 shows the trade-off front for the problem, from ‘expensive and low carbon dioxide’ to ‘cheap and high carbon dioxide’ (left to right). Figure 7 shows plots of each variable along the trade-off front, illustrating the nature of the trade-off front solutions. For example the heating system (Figure 7(a)) forms discrete sections, from gas to air-source heat pump to community biomass (as cost increases and emissions decrease); the solid fuel burner option does not appear, so is never an optimal design. With respect to photovoltaics (Figure 7(g)), none are required for up to a 50% carbon dioxide improvement; this then rapidly steps up to 7.5 m² per dwelling, the maximum allowable. Where periodic changes appear (e.g. window U-value and solar thermal), performance is improved until a change elsewhere allows the specification to be backed-off again.

5. Issues to be resolved

The examples in Section 4 demonstrate that it is now practical to perform automated optimisation on certain aspects of design problems. However, these methods are not a replacement for designers: the approaches still require an underlying system or model to optimise. Nevertheless, designers are now able to introduce a greater level of flexibility where appropriate and allow optimisation methods to perform the evaluations. This does, however, require designers to fully understand what they desire as an outcome so that

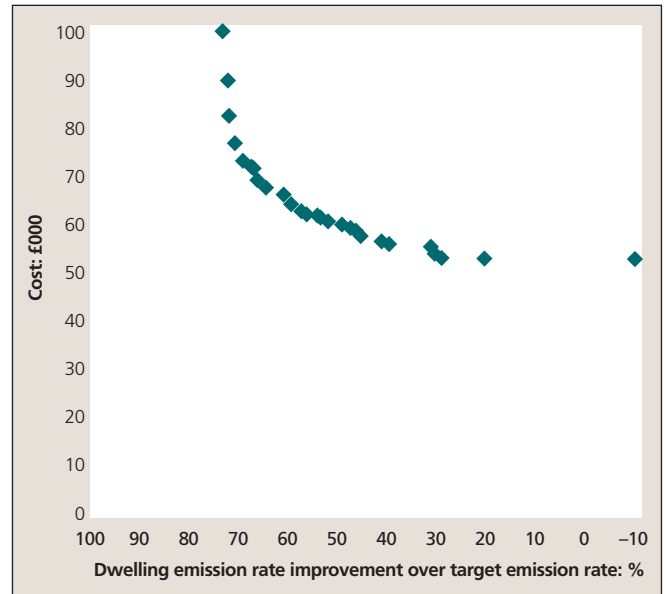


Figure 6. Trade-off front for a low-carbon-dioxide housing problem

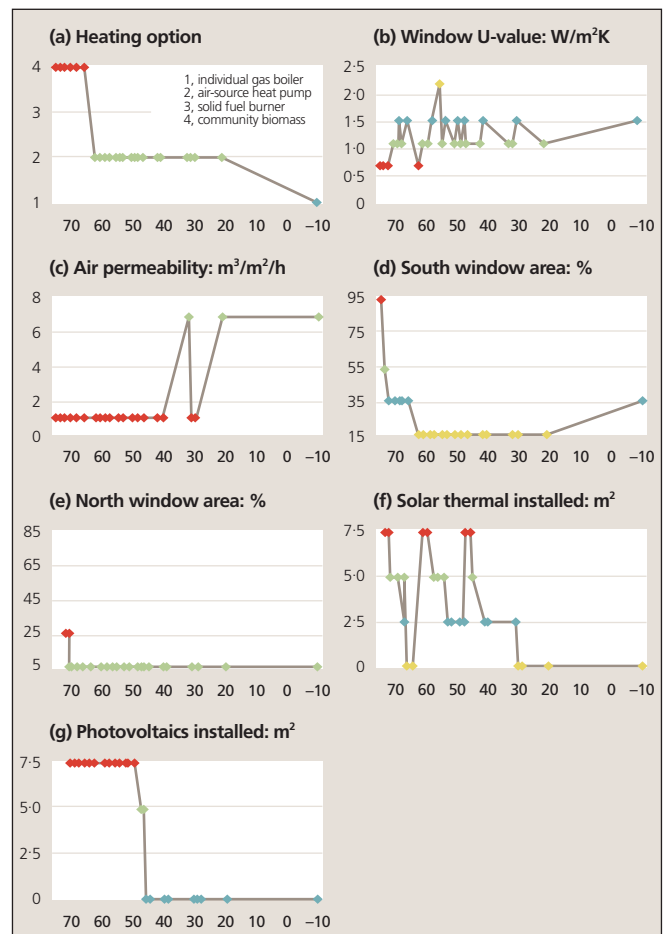


Figure 7. Variations among optimal solutions for a low-carbon-dioxide housing problem. The x-axis is the percentage dwelling emission rate improvement over the target emission rate and the y-axis is the variable value

they can correctly formulate the problem. It is common to perform multiple optimisations to answer different facets of the same problem. The exploratory and questioning nature of the designer is thus still at a premium even in this automated process.

The computation time required for the models to run is not trivial. Whereas simple rules of thumb can be introduced to allow quick appraisals, large run times may be required for detailed structural, thermal or fluid simulations. Poor communication methods between different programs can also limit the level of automation that is possible. These form the main limitations to these techniques: it may only be possible to apply them to simplified sub-sets of the overall problem or it may be necessary to use simplified analysis methods available in packages with good interoperability.

6. Conclusions

The approaches outlined in this paper have found wide application in industries other than civil engineering. With the introduction of readily accessible tools for creating parametric design models as well as the emergence of standard multi-objective optimisation algorithms, barriers to adoption in the construction industry have been significantly reduced. Benefits over traditional methods include greatly reduced time per design option trialled (countered by increased set-up time), improvements in performance for complex problems and increased rigour in the design process. These advantages will be most significant on projects that push the boundaries of performance – and hence small improvements are important – or on projects with high repeatability – and thus savings are multiplied.

The range of uses of the techniques means it is difficult to generalise regarding their role in the design process. They can be used when the complex nature of the problem demands the use of advanced methods – and sufficient time and resources are available. They may be used when it is possible to abstract meaningful simplifications of a problem, for example examining a typical zone of a larger building. Alternatively, they may be used to examine many general problem cases, to develop design rules and guidelines that are then applied at project level.

Further information regarding multi-objective optimisation is available from conferences proceedings (e.g. the ACM Genetic and Evolutionary Computation Conferences, IEEE Congress on Evolutionary Computation, International Conference on Evolutionary Multi-Criterion Optimization), journals (e.g. *IEEE Transactions on Evolutionary Computation*, *Journal of Multi-Criteria Decision Analysis*) and the internet (www.lania.mx/~ccoello/ (evolutionary multi-objective optimisation repository), www.iitk.ac.in/kangal/codes.shtml (NSGA-II C code)).

Acknowledgements

Funding for this research was provided by the Engineering and Physical Sciences Research Council (EPSRC) and Buro Happold Ltd. Evins and Joyce are research engineers with Buro Happold and with the Industrial Doctorate Centre in Systems, Universities of Bristol and Bath, UK.

Benefits over traditional methods include greatly reduced time per design option trialled... improvements in performance for complex problems and increased rigour in the design process

References

- Coenders JL (2009) Exception as a rule in computational design. In *Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures*. International Association of Shell and Spatial Structures, Madrid, Spain. See <http://dspace.upv.es/manakin/handle/10251/6868> (accessed 28/01/2012).
- Deb K, Pratap A, Agarwal S and Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2): 182–197.
- DECC (Department of Energy and Climate Change) (2011) *The Government's Standard Assessment Procedure for Energy Rating of Dwellings*. BRE, Watford, UK. See http://www.bre.co.uk/filelibrary/SAP/2009/SAP-2009_9-90.pdf (accessed 17/02/2012).
- Fong K, Hanby V and Chow T (2006) HVAC system optimization for energy management by evolutionary programming. *Energy and Buildings* **38**(3): 220–231.
- Hesselgren L, Charitou R and Dritsas S (2007) The Bishopsgate Tower case study. *International Journal of Architectural Computing* **5**(1). See <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.112.3981> (accessed 28/01/2012).
- Huang W and Lam H (1997) Using genetic algorithms to optimize controller parameters for HVAC systems. *Energy and Buildings* **26**(3): 277–282.
- Koumousis V and Georgiou P (1994) Genetic algorithms in discrete optimization of steel truss roofs. *Journal of Computing in Civil Engineering* **8**(3): 309–325.
- Marks W (1997) Multicriteria optimisation of shape of energy-saving buildings. *Building and Environment* **32**(4): 331–339.
- Ren Z, Yang F, Bouchlaghem NM and Anumba CJ (2011) Multi-disciplinary collaborative building design – A comparative study between multi-agent systems and multi-disciplinary optimisation approaches. *Automation in Construction* **20**(5): 537–549. See <http://www.sciencedirect.com/science/article/B6V20-51YXM50-1/2/27d497b4ce5e8e4495b6180c3a77a242> (accessed 28/01/2012).
- Roy R, Hinduja S and Teti R (2008) Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals – Manufacturing Technology* **57**(2): 697–715.
- Shepherd P and Hudson R (2007) Parametric definition of Lansdowne road stadium. *Proceedings of International Association of Shell and Spatial Structures*. Vol 1, CD Rom.
- Wang W, Zmeureanu R and Rivard H (2005) Applying multi-objective genetic algorithms in green building design optimization. *Building and Environment* **40**(11): 1512–1525.
- Williams CJK (2001) The analytic and numerical definition of the geometry of the British Museum Great Court Roof. See <http://opus.bath.ac.uk/14111/> (accessed 28/01/2012).
- Zolfaghari AR, Heath AC and McCombie PF (2005) Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. *Computers and Geotechnics* **32**(3): 139–152.

What do you think?

If you would like to comment on this paper, please email up to 200 words to the editor at journals@ice.org.uk.

If you would like to write a paper of 2000 to 3500 words about your own experience in this or any related area of civil engineering, the editor will be happy to provide any help or advice you need.