Configuration of a genetic algorithm for multi-objective optimisation of solar gain to buildings

Ralph Evins
Buro Happold & University of Bristol
17, Newman St,
London. W1T 1PD. UK.
+44 (0) 207 927 9700
ralph.evins@burohappold.com

ABSTRACT
We report the formulation and implementation of a genetic algorithm to address multi-objective optimisation of solar gain to buildings with the goal of minimising energy consumption and hence limiting carbon emissions. Heuristic optimisation approaches hold significant promise to balance complex tradeoffs in building design; however the unique nature of each building optimization problem limits broader implementation. Parameter selection is very challenging with little or no correlation between different architectural configurations. We address this issue through 'calibration' on smaller scale problems with derivable optimal solutions. Various seeding, selection and fitness options were trialled, as well as different parameter values. The Pareto front of the global solution set was successfully reproduced for the calibration case. Varying climate produced no major change in the nature of the solution; however, building orientation forced reparameterization for an optimal solution. Future work will establish when calibration is useful, and aim to quantify the nature of the solution space.

Categories and Subject Descriptors
J.2.7 [**Computer Applications**]: Physical Sciences and Engineering – *Engineering.*

General Terms
Design, Experimentation.

Keywords
Carbon, buildings, solar gain, design optimisation, parameter selection.

1. INTRODUCTION
1.1 Low-carbon buildings and solar gain
There is a strong scientific consensus that we must find a way to limit man-made climate change by reducing carbon emissions. One way is to reduce the energy used in buildings for heating, cooling and lighting, which comprises around 40% of the carbon emissions of developed countries [1]. Furthermore, unlike many other sources of carbon emissions, the technology already exists to dramatically reduce the energy consumption of buildings.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
ACM 978-1-4503-0072-8/10/07.

Solar thermal gain (energy entering a building as solar radiation via windows etc) plays a significant role in the performance of low-carbon buildings, and leads to a trade-off: too much solar gain in summer causes overheating and increases the need for cooling; too little solar gain in winter increases the need for heating. In addition, if steps are taken to limit solar gain, the amount of daylight entering the room is also reduced, increasing the need for artificial lighting.

1.2 Optimisation of low-carbon buildings
A number of features of the problem of building design optimisation put it beyond the reach of calculus-based methods: it is hard or impossible to define differentiable objective functions; many variables are involved; there are many nonlinearities and interdependencies in the problem. There are other issues that favour evolutionary computational solutions: problems are often multi-objective, multi-modal and highly constrained; robust solutions are sought that account for uncertainty; designer interaction is desirable. Of the 25 optimisation methods listed by Roy [2], evolutionary computation is the only one that can fulfil all of these requirements.

There are academic precedents in the use of GAs for optimisation of low-carbon buildings with respect to solar gain. Kampf et al [3] optimised the layout of an urban area for optimal solar gain using the multiple objectives of irradiation offset by thermal losses and building volume. Manzan and Pinto [4] used a genetic algorithm to optimise the shade geometry and glazing transmittance for an office building for a single objective, annual energy consumption. Wright and Mourshed [5] discretised the façade into cells and found the optimum distribution of glazed cells for minimum energy use.

1.3 Aims of this paper
The building industry has not yet embraced optimisation: "designs are still optimised mostly through a manual iterative process" [2]. For evolutionary computational methods to be employed effectively there is a need for guidance in algorithm setup and parameter selection to enable real-world application. This paper applies a genetic algorithm to an optimisation problem involving solar gain. It aims to go beyond a simple example case to provide general guidance by establishing whether the calibration method used gives improvements for subsequent optimisations.

2. EXPERIMENTAL APPROACH
A range of parameters and options relating to the GA have been varied for a calibration case to determine a high-performing

combination. This combination was then applied to application cases, and performance assessed by means of a brute-force search to identify the global optimum solution set. The fitness of each candidate solution set was based on the discrepancy between the candidate solution set and the global optimum set. This was calculated from the set of minimum Euclidean distances between each member of the global optimum set and the solution in question (i.e. the distance to the nearest candidate set individual for each point in the global optimum set). This was averaged across the global optimum set, and divided by the average Euclidean distance to the centre of gravity of all points. For each trial, 100 optimisations were run and an average of solution discrepancy was taken.

2.1 Calibration

2.1.1 Calibration variables

A calibration case was used to assess the impact of the following variables, trialled groupwise (double-lines) in the order given.

Table 1. Calibration variables and options.

Seeding	Pseudo-random		Latin hypercube		Halton sequence			
Selection	Tournament				Roulette			
Fitness assignment	Ranked		Scored		Strength-Pareto			
Population size	5	10	20	30	40			
Max no. of generations	5	10	15	20	25	30	35	40
Crossover probability	0.1	0.2	0.5	0.8				
Mutation probability	0.05	0.1	0.2	0.5				

2.1.2 Calibration case

The room modelled was 5m by 8m by 3.5m. The internal gains to the room were lighting (18.5W/m^2), equipment (10W/m^2) and people (15m^2/person), each varied according to a plausible daily schedule. The heating setpoint was 18C; the cooling setpoint was 24C. Daylighting controls were used to dim the artificial lights based on a sensor set at 500lux. The simulation was performed for a London climate. The number of permutations was 7560.

Table 2. Optimisation variables for calibration case

Variable	Range	Step size
Horizontal shade width	0.5m to 3.0m	0.5m
Fin width	0.2m to 1.0m	0.2m
Fin angle	-45^0 to 45^0	15^0
Glazing ratio	20% to 80%	20%
Window construction	Single, double, triple	-
Wall construction	Heavy, medium, light	-

The optimisation objectives were annual energy demand in kWh (1) and construction cost (2) in £.

$$O_1 = D_{Cooling} + D_{Lighting} + D_{Heating} \quad (1)$$

$$O_2 = C_{Cooling} + C_{Heating} + C_{Lighting} + \left(C_{Window} * A_{Window}\right) + \left(C_{Wall} * A_{Wall}\right) \quad (2)$$

Where C denotes the construction cost (£) and A the area (m^2).

2.2 Application

2.2.1 Application case A

The first application case changed the climate in which the building was modelled from London to Madrid. This affects the solar performance of the building (due to changes in daylight hours, number of sunny hours and sun position) and hence also the optimum solution set.

2.2.2 Application case B

The second application case changed the orientation of the building by ninety degrees. Again, this changes the solar performance (due to changes in relative sun position affecting the amount of incoming solar radiation at different times of day) and the optimum solution set.

3. OPTIMISATION SETUP

3.1 Program setup

The algorithm was implemented in MatLab, originally based on code from the Genetic Algorithm Optimisation Toolbox [6]. The code was adapted to include multi-objective capability, to add the LHS and Halton seeding options, and to add the three fitness assignment methods. Evaluation of the objective function regarding building energy use required the algorithm to call the external program EnergyPlus, which completed the energy simulation for a given building design. MatLab codes have also been developed for: translation of a set of values of the optimisation variables into an EnergyPlus building model file; executing EnergyPlus via a script; reading in and post-processing the results of the simulation; calculating the other objective functions (including ray-casting for the view count function).

EnergyPlus was used to perform a one year simulation using a typical year of weather data. To do this, it was required to: calculate incoming solar levels; calculate shading to glazed surfaces; perform a heat balance every time-step (15mins); control the heating, cooling and lighting systems according to set points; calculate the energy required by these systems.

3.2 Genetic algorithm setup

A real-number encoding was used. The default seeding option used a standard pseudo-random number generator, with values scaled to appropriate bounds and rounded to appropriate step sizes for each variable. See §3.3 for advanced seeding options. Population size was set to the appropriate value (default 20).

Evaluation of the objective functions for each individual required the algorithm to call EnergyPlus (see §3.1). Fitness assignment then used the appropriate method to determine the probability of selection for each individual (default Strength Pareto).

A simple splice crossover was applied by selecting two individuals using the appropriate method, tournament or roulette

(default roulette). This was governed by a given probability of crossover (default 0.5); if the probability was not met, the individuals did not continue to the new population (failure to breed). Replacement of individuals into the population was used during crossover, allowing two copies of the same individual to be selected, and hence continue unchanged. The crossover process continued until the new population was full. Similarly mutation was applied at a random point in a random individual, governed by a given probability of mutation (default 0.1). Termination was according to a maximum number of generations (default 20).

3.3 Seeding options

Two advanced seeding methods were used to attempt to overcome the clustering which can occur with random sequences, and thus ensure an even sampling of the design space.

Latin hyper-cube sampling (LHS) is a pseudo-probabilistic technique which involves dividing each dimension of the design space into equal sections, randomly selecting a section from each, and then eliminating those sections until all others have been used. Preechakul and Kheawhom have successfully used LHS in the seeding of genetic algorithms [7].

Halton sequence sampling is based on a low-discrepancy sequence, which aim to give a uniform distribution of values; the most basic is the van der Corput sequence. The Halton sequence builds on the van der Corput sequence to give a uniform distribution in higher dimensions, necessary for use in seeding a genetic algorithm with more than two variables. The random-start Halton sequence initialises the sequence randomly, and has been successfully used by Kimura and Matsumura for seeding a genetic algorithm [8].

3.4 Fitness assignment options

The Ranked procedure was developed by Goldberg [9]. It proceeds iteratively, assigning a rank to all non-dominated solutions, removing them from the population, then assigning the next highest rank to the new non-dominated solutions.

The Scored procedure was developed by Fonesca and Fleming [10]. Individuals were scored according to the number of individuals that dominate them, with a rank of 1 being given for the least number of dominating individuals.

The Strength Pareto procedure used was outlined by Zitzler and Thiele [11]. All non-dominated solutions are moved to an external population, and scores are assigned to these based on the number of individuals *dominated by* the individual in question. For dominated individuals, a score is assigned based on the sum of the scores of all non-dominated individuals which *dominate it*. Selection was from the combined current and external populations. Clustering was used to limit the size of the external population by means of pruning, using the average linkage method as in [11].

4. RESULTS

4.1 Calibration results

Data from a brute-force search was used as the basis of comparison for calibrations (Figure 1). The Pareto front (17 individuals), shown in black, is the global optimum solution set.

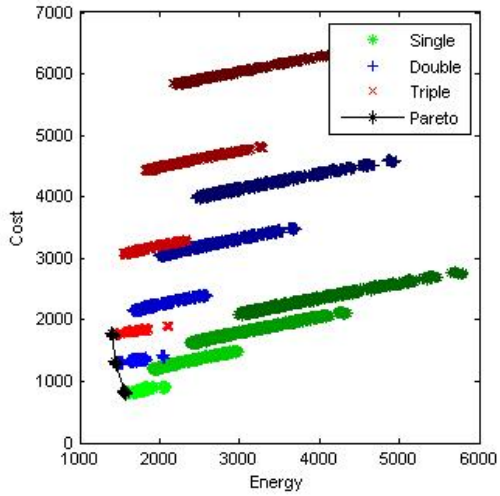

Figure 1. Brute-force search data (calibration case).

Table 3. Normalised discrepancy values for seeding options.

Random	LHS	Halton
0.78	0.96	0.49

For seeding options, the Halton method achieved the closest fit to the optimum solution set, with LHS performing least well.

Table 4. Normalised discrepancies for fitness/selection options.

	Pareto Strength	Sorted	Ranked
Roulette	0.37	1.54	1.54
Tournament	0.78	2.57	2.98

The Pareto strength method is the best-performing fitness assignment, with Roulette being the slightly better selection method. Tournament selection performed worst with all three fitness methods. Sorted and Ranked fitness methods performed more or less equally.

For population size and number of generations, there is a tradeoff between accuracy and number of evaluations. The benefits of more evaluations gradually diminish; the values of P=20 and G=20 have been taken as a suitable tradeoff of quality (normalised disrepancy ~0.2) against runtime (200 evaluations).

Table 5. Normalised discrepancies for operator parameters.

		P_c			
		0.1	**0.2**	**0.5**	**0.8**
P_m	**0.05**	0.34	0.54	0.45	0.26
	0.1	0.29	0.15	0.10	0.08
	0.2	0.08	0.09	0.08	0.08
	0.5	0.08	0.08	0.08	0.08

For crossover and mutation, as long as both probabilities are not too low, performance is much the same. Low values for P_m are more harmful than low values for P_c. The values of $P_m = 0.2$ and $P_c = 0.5$ have been chosen for subsequent optimisations.

4.2 Application results

4.2.1 Discrepancy results

Table 6. Discrepancy results comparison.

	Discrepancy	Average No. of Evaluations	Corresponding Random Search Discrepancy
Calibration	0.36	230	1.82
App A	0.13	234	1.25
App B P=20 G=20	15.8	238	16.5
App B P=40 G=40	3.9	705	11.2

The discrepancy result for application case A was excellent in comparison to both the calibration case and random search. For application case B, the GA using the calibrated options (P=20, G=20) performed surprisingly poorly. Increasing the population size and number of generations to 40 improved the result, however it is still much higher than the calibration case. The GA with P=20, G=20 was only marginally better than a random search; for the GA with P=40, G=40, the improvement over random search was considerable.

4.2.2 Brute-force search data

Comparison of the solution space for the calibration with that of application case A shows no major change in the nature of the solution space or optimum solution set. The optimum solution set has been shifted to higher energy values, and the total range of energy values is higher. However, comparison of the calibration case with application case B shows several differences: the Pareto front is longer for case B, and there is more variance in the design variables (more clusters; no single minimum for glazing ratio).

As would be expected, the better discrepancy results were obtained for the cases with 'easier' solution spaces. In both the calibration case and application case A, the solution space was relatively straightforward (implied by the low discrepancy values for the random search method and confirmed visually), and good discrepancy results were obtained. This implies that (moderate) changes in climate can be tolerated, allowing calibration for one location for use in another. For application case B, there is clearly some facet of the problem that renders it 'difficult' for the GA (and for random search) which is not present in the other cases. This resulted in poor discrepancy results for the GA and for random search, and showed calibration to be of little use.

5. CONCLUSIONS

It has been established that genetic algorithms can perform well in this area of building design optimisation. To our knowledge, this is the first successful implementation with the goal of scalability based on calibration of search parameters. Key contributions of this paper include:

- The first implementation of a genetic algorithm for multi-objective optimization specifically for solar gain.
- Development and implementation of a calibration method to parameterize a genetic algorithm for solar gain problems, which is essential for practical industrial use.
- Comparison of algorithm performance to brute-force data and random-search results using the discrepancy measure through a non-dimensional method of contrasting optimum populations of variable size.
- Investigation into the features (e.g. climate) for which the calibrated algorithm is transferable, those (e.g. orientation) for which it is not.

The fact that closely related cases can vary significantly in difficulty is a problem for the industrial deployment of such algorithms when brute-force search data is unavailable. In its current state, where it can be shown that the nature of solution spaces does not vary, calibration of the algorithm does provide a suitable means of approaching the problem and scoping the search of the genetic algorithm. Future work will be aimed at developing calibration procedures to attempt to predict and account for the optimisation problem difficulty as well as scale.

6. ACKNOWLEDGEMENTS

Funding has been provided by EPSRC and Buro Happold Ltd. The author is a Research Engineer with the Industrial Doctorate Centre in Systems, Universities of Bristol and Bath. Supervisors: Dr Ravi Vaidyanathan & Philip Pointer.

7. REFERENCES

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and Buildings*, vol. 40, 2008, pp. 394-398.

[2] R. Roy, S. Hinduja, and R. Teti, "Recent advances in engineering design optimisation: Challenges and future trends," *CIRP Annals - Manufacturing Technology*, vol. 57, 2008.

[3] J.H. Kämpf, M. Montavon, J. Bunyesc, R. Bolliger, and D. Robinson, "Optimisation of buildings' solar irradiation availability," *Solar Energy (In Press, Corrected Proof)*.

[4] M. Manzan and F. Pinto, "Genetic Optimization of External Shading Devices," *Proceedings: Building Simulation 2009*, 2009.

[5] J. Wright and M. Mourshed, "Geometric Optimization of Fenestration," *Proceedings: Building Simulation 2009*, 2009.

[6] C. Houck, J. Joines, and M. Kay, "A Genetic Algorithm for Function Optimization: A Matlab Implementation," 1995.

[7] C. Preechakul and S. Kheawhom, "Modified genetic algorithm with sampling techniques for chemical engineering optimization," *Journal of Industrial and Engineering Chemistry*, vol. 15, 2009.

[8] S. Kimura and K. Matsumura, "Genetic Algorithms using Low Discrepancy Sequences," *Proceedings: GECCO 2009*, 2009.

[9] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.

[10] C. Fonseca and P. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," *Genetic Algorithms: Proceedings of the Fifth International Conference*, Morgan Kaufmann, 1993, pp. 423, 416.

[11] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, 1999.