Building Simulation 2005

Ninth International IBPSA Conference
Montréal, Canada
August 15-18, 2005

THE ROBUSTNESS OF GENETIC ALGORITHMS IN SOLVING UNCONSTRAINED BUILDING OPTIMIZATION PROBLEMS

Jonathan Wright, and Ali Alajmi
Department of Civil and Building Engineering,
Loughborough University,
Loughborough, Leicestershire, LE11 3TU
UK

ABSTRACT

This paper investigates the robustness of a Genetic Algorithm (GA) search method in solving an unconstrained building optimization problem, when the number of building simulations used by the optimization is restricted. GA search methods can be classified as being probabilistic populations based optimizers. The probabilistic nature of the search suggests that GA's may lack robustness in finding solutions. Further, it is a common perception that since GA's iterate on a population (set) of solutions, they require many building simulations to converge. It is concluded here that a particular GA was robust in finding solutions with 1.4% mean difference in building energy use from that for the best solution found in all trial optimizations. This performance was achieved with only 300 building simulations (in any one trial optimization).

INTRODUCTION

The benefit of the simulation-based optimization of building design has been demonstrated through a potential reduction in building energy use by as much as 30% of that resulting from a benchmark design (Wetter and Wright, 2003).

Recent research examined the performance of several different optimization algorithms in solving a non-smooth, single-criterion, non-linear, unconstrained building optimization problem (Wetter and Wright, 2004). The optimization problem included elements of window geometry, the setpoints for the control of window shading devices, night-setback setpoints for the zone temperature, and the design supply air temperature setpoint. The performance of twelve different optimization algorithms was examined; the algorithms included, direct search methods (pattern and simplex searches), a gradient based method, probabilistic populations based methods (particle swarm and genetic algorithms), and hybrid methods (particle swarm and pattern search).

It was concluded that the direct and gradient based methods could converge far from the optimum when the objective function was non-smooth (which was the case for the example building optimization problem). Conversely, the probabilistic population-based algorithms were more robust in finding near-optimum solutions, with a hybrid particle-swarm pattern-search algorithm having the most consistent performance. A simple genetic algorithm also proved to be robust in finding near-optimum solutions.

Although robust, the hybrid particle-swarm pattern-search algorithm required in the order of 765 building simulations before converging on a solution. In comparison, the genetic algorithm required in the order of 585 simulations before convergence. However, the "convergence" of the genetic algorithm was determined by arbitrarily limiting the maximum number of iterations of the algorithm. In contrast, the number of iterations of the hybrid algorithm was governed, in-part, by the characteristics of the search space and the formally defined convergence of the pattern search algorithm. In this respect, the pattern-search is guaranteed to have converged onto a local optimum, whereas the arbitrary convergence criteria of the genetic algorithm, means that convergence onto a local optimum cannot be guaranteed.

Although previous research, indicates that genetic algorithms have the potential to solve unconstrained building optimization problems, the number of algorithm iterations required to find a solution remains unclear. Further, genetic algorithms are probabilistic optimizers and as such their behavior varies probabilistically with the choice of starting conditions and algorithm control parameters. Although previous research has examined the range of solutions found during a particular search (Coley and Schukat, 2002; Caldas and Norford, 2002), no research has been conducted to examine the effect of the algorithm starting conditions or control parameters on the robustness of the search in finding near-optimum solutions.

This paper examines the robustness of a genetic algorithm in solving an unconstrained non-smooth building optimization problem (Wetter and Wright, 2003). The robustness of the algorithm is examined in relation to the range of the solutions found for different starting conditions and algorithm control parameters. In particular, the ability of the search to find a solution with a limited number of trail simulations is examined (the computational overhead associated with simulating the building performance being the dominant computational element of the optimization).

GENETIC ALGORITHM DESIGN

A large volume of literature on the form and operation of Genetic and other Evolution Algorithms exists (among others; Bäck, 1996; Bäck et al, 2000; Deb, 2000). In brief, Genetic Algorithms (GA's), iterate on a "population" (set), of solutions. Having first, randomly initialized the population with solutions (each variable being randomly assigned a value within its bounds), there are five main operations in the iteration:

1. **Fitness assignment:** the evaluation of the solutions objective function (often followed by some scaling or rank ordering).
2. **Selection:** of individuals (solutions), from the existing population for further operation. The selection is normally probabilistic with the probability of selection increasing with fitness (in this case of a minimization problem, increasing as the objective function values decreases).
3. **Recombination:** the "mixing" of solutions to produce new solutions. Recombination is performed probabilistically, and as such may not take place.
4. **Mutation:** the probabilistic perturbation of variable values.
5. **Replacement:** of old solutions in the population with the new solutions resulting from selection, recombination, and mutation.

Further, GA's operate with an encoding of the problem variables know as a "chromosome". The encoding can be in the form of a concatenated string of binary numbers, or a vector of real (or integer), values. Recombination and mutation operate on the chromosome "gene" values. In the case of the binary encoding, a gene is represented by a single bit in the binary string (the value of a problem variable being represented by several bits). In contrast, real encoded GA's, operate directly on the real value of the problem variable.

Form of Chromosome

In contrast to a real vector chromosome, a binary encoding has potentially greater exploratory power than a real vector chromosome, and naturally lends itself operating with both discrete and continuous variables. Building optimization problems are mixed-integer problems. For example, alternative wall constructions might be identified by an integer index that points to a particular combination of construction materials, whereas a supply air temperature setpoint may be treated as being continuous.

Both continuous and discrete variables can be encoded in a binary chromosome through controlling the number of bits assigned to a given variable (a three bit encoding resulting 8 discrete values for the variable). In practice the number of bits can be set to give a real value precision up to the limit of the machine precision, although in practice, the number of bits is set to provide a practicable resolution in variable precision. The inherent encoding of mixed-integer problems and the associated control of variable precision, lends a binary encoding to the solution of building optimization problems.

A Gray binary encoding has been adopted for use in this research. This encoding improves the continuity of the encoded search space, since it limits the occurrence of "Hamming cliffs" (these occur when a sequential change bit values results in large, discontinuous, changes in the value of the decoded problem variable).

Fitness

In this study, we seek to minimize the building energy use and therefore, the lower the energy use, the higher the fitness of an individual. In this research, the solutions have been rank-ordered on the energy use objective function, the fitness of a solution being given by its rank.

Selection

A characteristic of GA's is that, through the choice of algorithm operators, and their control parameter values, it is possible to influence the behavior of the search. A common consideration in choosing the algorithm operators and parameters, is the balance between the convergence reliability and the convergence velocity (or "exploration" versus "exploitation"; Bäck, 1996). One of the principal operators governing this balance is the selection mechanism. The selection operator is used to select solutions from the current population that will be used to form the next population of solutions (this being the basis for the next iteration of the algorithm). If for example, the selection method simply selected the best solution of the population to form the basis of the next population, then the search would soon converge towards the current best solution. However, if the selection method randomly selected solutions from the population, then it is likely that no one solution would dominate the search direction, and therefore, the search would be more random in nature.

In this research, we seek robust convergence with as few building simulations as possible (that is, reliable convergence with a high convergence velocity). In order to examine this, the "tournament" selection method has been adopted for use in this study, since it allows control of the "selection pressure" on the best solutions.

The tournament operator randomly selects *n*, solutions from the population, the winner of the tournament and solution carried forward for recombination, being the solution having the highest fitness of all solutions in the tournament. The probability of a solution having a high fitness being selected, depends on the number of individuals in the tournament, and the size of the population (that is, the percentage of the population selected for the tournament).

One measure of the "selection pressure" is the "takeover time" (Bäck, 1996). This is the number of generations (algorithm iterations), for the population to fill with the best solution found in the initial generation, in the absence of recombination and mutation (in the absence of recombination or mutation, no new solutions are produced, the search only then sampling solutions from the randomly generated initial population). The takeover time for a tournament selection is approximated by (Bäck, 1996):

$$\tau = \frac{1}{\ln(n)}\left(\ln(q) + \ln(\ln(q))\right) \qquad (1)$$

where, τ is the takeover time, *q* the number of individuals in the population, and *n*, the number of individuals in the tournament.

In this research, we examine the effect of selection pressure on the performance of the search by varying the population size (*q*), for a tournament size (*n*). Table 1, gives the takeover times for a binary tournament (*n*=2) and three populations sizes. The table also includes the number of tournaments necessary to fill the population with the best of the initial solutions (taken as, $\tau' = \tau q - q$).

Table 1
Takeover times for a binary tournament

Population Size, *q* (-)	**Takeover Time**	
	τ **(-)**	τ' **(-)**
5	3.0	10
15	5.3	65
30	6.7	170

The effect of population size on the selection pressure is clear from Table 1, with a population size of 5 individuals converging in under half the number of iterations required for a population of 30. Perhaps what is more significant is the difference in number of tournament selections (τ'), since, this reflects the impact of the population size and selection pressure on the likely number of building simulations required to reach convergence (with larger populations requiring significantly more simulations before the search begins to converge on a solution).

Recombination

The recombination operator controls the mixing of "genetic information" from paired individuals through a process know as "crossover" (each individual in the pair resulting from a separate tournament selection). In the case of a binary chromosome, crossover take place by swapping bit values between the two individuals. In the "uniform crossover" operator used here, each pair of bits are swapped with a 50% probability (an average of 50% of the bits will be swapped). A 50% probability of bit crossover gives the greatest mixing of genetic information between paired individuals. However, in this research, a further probability parameter has been used to control the occurrence of chromosome crossover. If the probability of chromosome crossover is such that crossover takes place, the bit values are swapped between chromosomes with a 50% probability. The effect of chromosome crossover probability on the performance of the search, is examined here by experiment.

Mutation

A probabilistic bit-wise mutation, in which a given gene value if flipped from 0 to 1, or visa versa, has been adopted in this study. The effect of mutation probability on the performance of the search is examined here by experiment.

Replacement and Elitism

It is common to control the proportion of solutions replaced in the population at each iteration of the algorithm. In this case, we replace all solutions, except the solution having the highest fitness (the "elite" solution). This guarantees that the search does not diverge to a solution having a higher objective function than that already found by the search. The replacement individuals are a result of selection, recombination, and mutation (although, since recombination and mutation are probabilistic, it is possible that some selected solutions remain unchanged when replaced in the population).

Convergence and Automatic Restart

GA's are conventionally stopped after a fixed number of algorithm iterations (generations). However, since the aim of this research is to study the convergence behavior of the GA in relation to the number of building performance simulations, the search is stopped here after a fixed number of building simulations. Since, the recombination and mutation operators are probabilistic, it is possible that a selected solution is simply copied from one generation to the next (this also occurs for the “elite” individual). When this occurs, the objective function value is taken from memory so that the need to re-simulate the building performance is avoided. Therefore, all of the building simulations performed are guaranteed to be unique.

Further, in order to guarantee that the search is able to continue until the specified number of simulations is reached, the search is automatically re-initialized if the population collapses onto a single solution. Such convergence can be measured in terms of the problem variables (the “genotype”), or the objective function (the “phenotype”). In this case we choose to identify the collapse of the population in terms of the objective function, as should the objective function have a low gradient in the region of the optimum, convergence of the objective function may not be reflected by the same degree of convergence in the problem variables. Collapse of the population has been defined in terms of:

$$\alpha = \left(\frac{f_{\max}(\cdot) - f_{\min}(\cdot)}{f_{\min}(\cdot)} \right) \times 100 \qquad (2)$$

where, $f_{\max}(\cdot)$, is the maximum objective function value found in the current population; $f_{\min}(\cdot)$, the minimum (best) objective function value in the current population; α the convergence parameter.

Should the current population have collapsed, then the next population is first re-seeded with the elite (best) solution, with the remaining individuals being randomly initialized within their bounds. This strategy is normally applied to a “micro-GA” (for example; Caldas and Norford, 2002); micro-GA use small population sizes, that due to the high selection pressure have a tendency to converge prematurely. In this case, the re-initialization is applied regardless of population size. Note that, to some extent, the re-initialization of the population can mask the apparent convergence and therefore the effect of the selection pressure. A value of α <=1% has been used in this study to indicate convergence and trigger re-initialization of the population.

EXPERIMENTS

There are two elements to the experiments performed here, the example building optimization problem, and the choice of GA control parameter sets.

Example Building and Optimization Variables

The minimization of annual energy use of a mid-floor five zone office building (Wetter and Wright, 2004) has been used as a basis for this study. All exterior zones have daylight control, and the zone conditions are maintained by a VAV system. The performance of the building is simulated by EnergyPlus, which also auto-sizes the capacity of the HVAC system. All experiments have been performed using version 1.1.0.020 of EnergyPlus, on a MS Windows platform (Pentium 4 chip set). The building location (and associated weather data, is taken as being Chicago, IL, USA.

The optimization problem variables are given in Table 2. The normalized window geometry variables determine the window width and height (with a value of 0.0 corresponding to window area of 20.4% of the façade, and 1.0 for 71.3% of the façade). The normalized window overhang variables govern the depth of overhang above each window (with a value of 0.0 corresponding to an overhang of 0.05m and 1.0 to 1.05m). The window shading setpoints control the use of an external shading device (the shading device being activated when the total insolation on the window exceeds the setpoint). The zone air temperature setpoints govern night operation of the HVAC system, while the supply air temperature setpoint is used to in the auto-sizing of the HVAC system capacity.

Note that the increment in each variable, together with the lower and upper bounds, are used to discretize the search space during the binary encoding of the variables. The discrete increment in each variable has been set based on reasonable engineering tolerance for each class of variable. The total discretized search space has a size of 1.94 x 10^{22}.

Genetic Algorithm Control Parameters

In this research, we aim to examine the robustness of a GA in solving the example optimization problem for a fixed number of building simulations. In a previous study, a solution to the example optimization problem, that had an objective function value within 0.4% of the best solution, was found by a particle swarm algorithm with only 317 trial simulations; all other algorithms in the study that found a solution with an objective function value within 1% of the best required more function evaluations (Wetter and Wright, 2004). It would therefore seem that in the order of 300 trial simulations is required to solve the example problem. Therefore, in this study, each optimization is performed for 300 trial simulations.

The behavior of the GA has been examined for 12 different sets of control parameters, consisting of 3 alternative population sizes, 2 crossover probabilities, and 2 mutation probabilities (Table 3).

Table 2
Optimization problem variables

Index	Variable	Lower Bound	Upper Bound	Increment	Best Solution
1	North Window Geometry (-)	0.0	1.0	0.02	1.0
2	West Window Geometry (-)	0.0	1.0	0.02	0.44
3	West Window Overhang (-)	0.0	1.0	0.02	0.34
4	East Window Geometry (-)	0.0	1.0	0.02	0.36
5	East Window Overhang (-)	0.0	1.0	0.02	0.48
6	South Window Geometry (-)	0.0	1.0	0.02	0.94
7	South Window Overhang (-)	0.0	1.0	0.02	0.48
8	West Window Shading Setpoint (W/m^2)	100.	600.	5.	285.
9	East Window Shading Setpoint (W/m^2)	100.	600.	5.	365.
10	South Window Shading Setpoint (W/m^2)	100.	600.	5.	285.
11	Zone Night Temperature Setpoint, Winter (^oC)	20.0	25.0	0.2	24.6
12	Zone Night Temperature Setpoint, Summer (^oC)	20.0	25.0	0.2	25.0
13	Supply Air Temperature Setpoint (^oC)	12.0	18.0	0.2	12.8

Table 3
GA Control parameter sets

Control Parameter	Values
Population Size (-)	[5,15,30]
Probability of Chromosome Crossover	[0.7, 1.0]
Probability of Mutation	[0.01, 0.02]

Population sizes in the order of 80 individuals are commonly used in GA's. However, since we have a limit of 300 simulations, this is likely to result in less than 5 iterations of the GA (assuming that, say, more than 60 of the 80 individuals are new in each generation). Considering that the larger the population, the lower the selection pressure and convergence velocity, it is unlikely that 5 iterations is sufficient to find a solution. For this reason, we adopt small population sizes that allow 10 or more iterations of the algorithm. Further, the alternative population sizes have been chosen to provide a range of selection pressures, as indicated by their respective takeover times in Table 1. In the unlikely event that each population was filled with new solutions, a population size of 30 would result in the minimum of 10 iterations. Previous research (Wetter and Wright, 2004), also indicated that a population in the order of 15 individuals was able to find a solution to the optimization problem, and finally, a population of 5 individuals is typical of that used in a micro-GA (for which, the automatic re-initialization of the algorithm is required).

The probability of chromosome crossover controls the mixing of “genetic” information between selected individuals. In general, crossover rates in the order of 0.7 or higher, improve the chance of good solutions being found. Given the high selection pressures resulting from the small population sizes, the higher degree of mixing of solutions is also likely to maintain the exploratory power of the search. In this respect, we examine the behavior of the search for two high, chromosome crossover rates of 0.7 and 1.0 (70% and 100% probability).

Mutation perturbs the gene values, and as such has a high probability of introducing new solutions to the search. However, too high a mutation rate can result in a random search. It is common for binary mutation rates to be set in terms of the number of bit mutations per chromosome. The encoded binary chromosomes used in this problem are 78 bits long, so that a 0.01 probability of mutation is likely to result in less than one bit mutation per chromosome, and a probability of 0.02, of greater than one mutation per chromosome. A probability of one mutation per chromosome is commonly used, so that here, we examine the effect of a mutation rates that are slightly higher and lower than normally used. It is expected that the higher mutation rate is important in terms of maintaining the exploratory power of the search, given the high selection pressure due to the small population sizes.

Finally, since GA's are probabilistic optimizers, for each of the 12 GA parameters sets, we conduct 9 separate experiments, with each experiment started with a different randomly generated population of solutions. Given 9 experiments for each of the 12 parameter sets and each experiment having 300 trial simulations, we perform a total of 32,400 simulations in this study.

RESULTS AND DISCUSSION

In this study, we seek to obtain good algorithm performance in two potentially competing criteria; robustness in finding solutions, and minimization of the number of building simulations.

Algorithm Robustness

The robustness of the algorithm is examined in terms of the variability of the final solutions from each set of experiments. Table 4, gives the results for the trail optimizations (each statistic relating 9 separate optimizations). There is no statistical difference between any of the solutions, although subjectively, the smaller populations appear to have a greater chance of finding a better solution. Further, although not statistically measurable, for the small population sizes (5 and 15), the better solutions result from the higher probability of crossover and mutation.

The variability (spread) in the solutions is illustrated in Figure 1. The spread is illustrated in terms of the distance from the best solution found in all trail optimizations, the horizontal axis being the distance in the problem variable domain, and the vertical axis the distance of the objective function. The normalized Euclidean distance $d(X^*, X)$, in the problem variables is given by:

$$d(X^*, X) = \sqrt{\sum_{i=1}^{nv} \Delta x_i^{\,2}} \qquad (3)$$

where

$$\Delta x_i = \frac{x_i - x_i^*}{u_i - l_i} \qquad (4)$$

and $X = (x_1, ..., x_{nv}) \in \Re^{nv}$, is a solution and $X^* = (x_1^*, ..., x_{nv}^*) \in \Re^{nv}$, the best solution found in all experiments; nv is the number of problem variables, and l_i , and u_i are the variable bounds such that $l_i \le x_i \le u_i,\ \forall i \in (1, ..., nv)$ where $l_i, u_i \in \Re^{nv}$ and $l_i < u_i,\ \forall i \in (1, ..., nv)$.

The difference in the objective function values is given by:

$$d\left(f\left(X^*\right), f(X)\right) = \left(\frac{f(X) - f\left(X^*\right)}{f\left(X^*\right)}\right) \times 100 \qquad (5)$$

Figure 1 illustrates that the majority of the solutions have an objective function value within 2.5% of the best solution, the mean difference being 1.4%. The extent to which this is considered sufficiently "robust", should be judged in relation to difference in building energy use to that of the energy use for a "standard" design solution, as well as the inherent uncertainty in the simulation and design process. Unfortunately, such an analysis is beyond the scope of this paper.

Figure 1
Solution spread

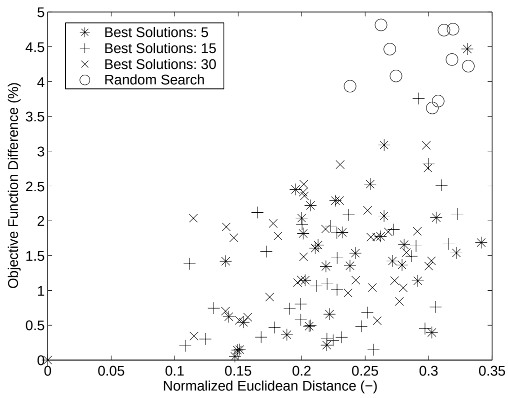


Figure 1, also indicates that there is some overlap in the optimized solutions with those from a randomly generated set (these random solutions resulting from the GA initialization procedure). However, given that the mean objective function difference, for all 270 random solutions, was 28.4%, the overlap is insignificant.

Finally, in terms of the consistency of the solutions, Figure 1, indicates that the similar objective function values can be obtained from different solutions. This can occur for two reasons, first, the optimization problem may be insensitive to the problem variables (that is, the objective function gradient is low),

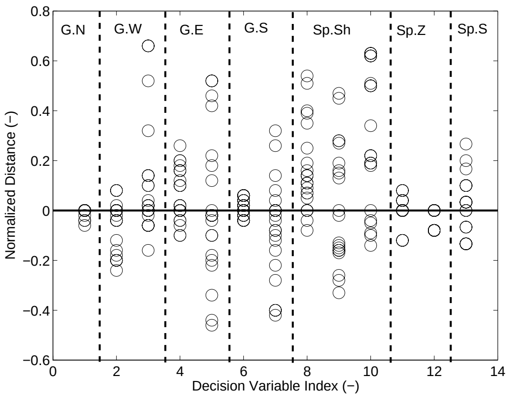
Table 4
Best solutions

Population Size, *q* (-)	Probabilities (-)		Objective Function (kWh/m^2)			
	Crossover	Mutation	Minimum	Maximum	Mean	Standard Deviation
5	0.7	0.01	122.53	125.45	124.39	0.94
5	0.7	0.02	122.80	127.83	124.25	1.42
5	1	0.01	122.42	126.14	123.98	1.10
5	1	0.02	122.36	125.16	123.77	0.94
15	0.7	0.01	122.73	126.95	123.92	1.35
15	0.7	0.02	122.73	125.43	124.08	0.90
15	1	0.01	122.71	125.81	123.79	0.92
15	1	0.02	122.54	124.60	123.54	0.79
30	0.7	0.01	122.78	125.45	124.42	0.88
30	0.7	0.02	123.39	125.79	124.49	0.85
30	1	0.01	123.05	124.85	123.89	0.64
30	1	0.02	123.11	126.13	124.37	0.85

and/or, the optimization problem is highly multi-modal. Figure 2, illustrates the spread in variable values for solutions that have an objective function difference (Equation 5), of 0.5% or less. The decision variable index in Figure 2 relates to that given in Table 2, and the normalized distance by Equation 4. The variable groups G.N, G.W, G.E and G.S relate to the window area and overhang size for the north, west, east, and south window respectively; the group Sp.Sh the window shading setpoints; Sp.Z the zone air temperature night setpoints; and Sp.S the design supply air temperature. It is postulated here that a large scatter in a variable indicates a low sensitivity, or high multi-modality of the objective function to the problem variable. The least spread variables relate to the window areas (indexes 1, 2, 4 and 6), the zone night setpoints (indexes 11 and 12), and the design supply air temperature (index 13). The variables having the most spread relate to the size of the window overhangs (indexes, 3, 5, and 7), and the shading setpoints (indexes 8, 9, and 10).

It is probable that the large spread in size of window overhang indicates a low sensitivity of the objective function (energy use), to overhang size for the example building. This may also be the case for the shading setpoint. In both cases however, further research is required to confirm the sensitivity of the objective function to these variables, and in particular their impact on the multi-modality of the objective function.

Figure 2
Solution spread in the region of the optimum



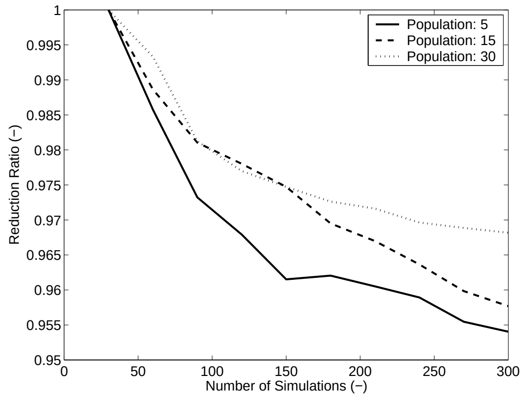
Convergence Velocity

Figure 3, illustrates the convergence for three of the trial optimizations (each optimization used the same random number sequence, and had a 1.0 probability of crossover and 0.02 probability of mutation). In order to obtain a true comparison between the convergence of each population, the convergence is given as a reduction ratio:

$$r\left(\bar{f}(U_1),\ \bar{f}(U_i)\right) = \frac{\bar{f}(U_i)}{\bar{f}(U_1)} \qquad (6)$$

where: a set of *np* solutions, is given by $U = \{X \mid (X_1, ..., X_{np})\}$ and the $\bar{f}(U_1)$ is the mean objective function value of the set. Since the largest population size used here contains 30 solutions, the objective function means have been calculated for every 30 new trial simulations.

Figure 3,
Rate of convergence



In this respect, the smaller populations would have already completed several iterations (generations), before reaching 30 simulations. Even though this is the case, the initial mean objective function value is lower for the 30 random solutions of the 30 individual population (Table 5). This in part explains the higher rate of convergence of the small populations (the divisor in Equation 6 is higher). Further, the population of 30 converged onto a worse solution, than that found for either of the two smaller population sizes, which had the effect of increasing the final reduction ratio. It can be concluded, that there is some evidence that the population size of 5 individuals has a higher convergence velocity than the larger two populations, although this requires further investigation.

Table 5
Mean solutions after 30 simulations

Population size	5	15	30
$\bar{f}(U_1)$	129.8	129.0	128.5

CONCLUSIONS

The aim of this research was to examine the robustness of a GA in finding solutions to an unconstrained building optimization problem, given a limited number of trial simulations. Experiments where performed for twelve different sets of GA control parameters, with nine different trial optimization completed for each parameter set.

It was concluded, that the GA was insensitive to the choice of GA control parameters, there being no statistically significant difference in solutions found between any of the parameter sets. However, the better solutions where obtained using small population sizes (5 and 15 individual), with high probabilities of crossover and mutation (100% and 2% respectively). The mean difference in objective function values, from that for the best solution found, was 1.4%, with the majority of solutions being within 2.5% of the best solution found. Given that there is no significant difference between the sets of solutions, it can be concluded that the GA was robust in finding solutions.

It was also concluded that it is possible to find near-optimum solutions with a competitive (low) number building performance simulations (in this case 300 simulations). This may, in part, be due to the high convergence velocities resulting from the low population sizes, although this requires further research.

Further research is also required to examine the robustness of the GA, particularly in relation to the characteristics of the problem and search that result in poor solutions being found. Research is also required to investigate the solution accuracy that is expected in relation to that achievable given normal design tolerances and the uncertainty in results from building performance simulation.

REFERENCES

Bäck.,T, 1996, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York, ISBN 0195099710.

Bäck,T., Fogel, D.B., Michalewicz, Z., 2000, Evolutionary Computation 2: Advanced Algorithms and Operators (Evolutionary Computation), Institute of Physics, ISBN 0750306653.

Caldas., L.G., Norford., L.K., 2002, A Design Optimization Tool Based on a Genetic Algorithm, *Automation in Construction*, 11, 173-184, ISSN 0926-5805.

Coley, D.A., Schukat. S., 2002, Low-Energy Design: Combining Computer-Based Optimisation and Human Judgement, *Building and Environment*, 37, 1241-1247, ISSN 0360-1323.

Deb, K. D., 2001, "Multi-Objective Optimization using Evolutionary Algorithms", John Wiley and Sons Ltd, ISBN 0-471-87339-X.

Wetter, M., Wright, J.A., 2003, Comparison of a Generalized Pattern Search and a Genetic Algorithm Optimization Method, Proceedings of the 8th International IBPSA Conference, Eindhoven, Netherlands, 1401-1408.

Wetter, M., Wright, J.A., 2004, A Comparison of Deterministic and Probabilistic Optimization Algorithms for Nonsmooth Simulation-based optimization, *Building and Environment*, 39, 989-999 ISSN: 0360-1323.