International Journal of Sustainable Built Environment (2014) 3, 18–26

HOSTED BY ELSEVIER

Gulf Organisation for Research and Development

International Journal of Sustainable Built Environment

ScienceDirect
www.sciencedirect.com

Original Article/Research

Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem

Ali Alajmi [a,*], Jonathan Wright [b]

[a] Department of Mechanical Engineering, College of Technological Studies, Mishref 40189, Kuwait
[b] Department of Civil and Building Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK

Received 9 August 2013; accepted 15 July 2014

Abstract

Effective optimization of unconstrained building optimization problem involves coupling a building energy simulation program with an optimization evolutionary algorithm such as the genetic algorithm (GA). The aim of this paper is to find the most appropriate GA set that obtains the optimum, or near optimum, solutions in a reasonable computational time (less numbers of simulations). Twelve control parameter sets of binary encoded GA are tested to solve unconstrained building optimization problems that are coupled with EnergyPlus simulation program.

The results show that population size is the most significant control parameter and that the crossover probability and mutation rate have insignificant effects on the GA performance. In general, a binary encoded GA with small population sizes can be used to solve unconstrained building optimization problems by around 250 building simulation calls. In particular, the smaller population size of about 5 individuals helps reach the optimum solution faster than larger population sizes.

Keywords: Genetic algorithms; Building Simulation program; EnergyPlus; Simulation-based building optimization problem; GA control parameters

1. Introduction

Energy consumption in buildings accounts for a considerable proportion of energy consumption in urban areas, and this sector will increase in the coming decades (EU, 2002). For instance, in Europe the demand of electricity in residential buildings will increase from 1% to 2% annually over the next decade. Therefore, energy saving in buildings is an important aspect for reducing national energy consumption, and consequently reducing greenhouse effect. For example, the energy consumption in buildings has a potential to be reduced by 22% in EU countries (Caldas, 2001). Building simulation provides an excellent way to study such potential and opportunities to achieve optimum building consumption.

Traditionally, parametric studies were done on building optimization problems such as the study on the effect of window size on the building performance while holding = other effective design variables constant (Jo and Gero, 1998; Guillemin and Molteni, 2002; Guillemin and Morel, 2001; West and Sherif, 2001a,b). Later, new optimization techniques were created which could handle

* Corresponding author.
E-mail address: af.alajmi@paaet.edu.kw (A. Alajmi).
Peer review under responsibility of The Gulf Organisation for Research and Development.

http://dx.doi.org/10.1016/j.ijsbe.2014.07.003
2212-6090/

a limited number of design variables (Wright and Farmani, 2001; Caldas and Norford, 2002, 2003). Fortunately, new techniques were found (Evolutionary Algorithms, EAs) which can handle efficiently a larger number of parameters. Also, the EAs have the advantages of being less sensitive to the problem characteristics, such as the availability of the objective function in a closed form mathematical expression and the design of variable types (discrete or continuous).

In particular, genetic algorithms (GAs) have been found to be robust in finding the optimum solutions for various engineering optimization problems (Wetter and Wright, 2003).

Recent research has shown that more efficient buildings can be designed using simulation-based building optimization (Wetter and Wright, 2003; Caldas and Norford, 2001) that couples an optimization algorithm, such as GA (genetic algorithms), with a building energy simulation program such as the "state-of-the-art" EnergyPlus (Crawley, 2001). The building design control parameters are entered to the simulation program and simultaneous changing within these parameters will lead to different possible solutions that can be systematically searched by an optimization algorithm.

The form and operation of genetic and other evolution algorithms are extensively studied (Bäck, 1996; Bäck et al., 2000; Deb, 2001). In brief, genetic algorithms (GA's) iterate on a set of solutions "population" that are randomly initialized. Each solution consists of all variables that are assigned a value within its lower and upper bounds. Then the process of generating new solutions commences after assigning fitness values for each solution (chromosome) accomplished by main operators. These main operators are known as: selection, recombination (crossover), and mutation. In addition, to ensure the solution does not become totally random the best solution will remain in the new generation. This process is known as "replacement".

The problem variables are combined together in a term known as a "chromosome" and part of it is named as "gene". Practically, the chromosome(s) is encoded in a concatenated string of binary numbers (binary encoding), or a vector of real values (real encoding). 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 GAs operate directly on the real value of the problem variable.

Although the GAs showed effectiveness in handling building optimization problems, the GA's main operators, such as population size, crossover, and mutation rate, have not been fully examined in building optimization problems (Alajmi and Wright, 2006). In fact, selecting appropriate GA operators is a trade-off between fast convergence and maintaining the exploratory power of the algorithm (to prevent false convergence).

In this paper, a GA and its alternative operator forms will be selected for solving a whole building optimization problem with 23 design variables (building envelop variables) without limiting the search space (constraints). The convergence behavior of the GA in relation to the number of required calls of the building simulation program is mainly considered.

2. Selection of genetic algorithm structure

Genetic algorithm structure incorporates the five main operations in iteration to create the new 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. This is harmonious with the nature of building optimization problems that have mixed-integer parameter 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 set point may be treated as being continuous.

Choosing the algorithm operators and parameters is a 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.

2.1. Binary encoding

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 will result in 8 discrete values for the variable). The inherent encoding of mixed-integer problems and the associated control of variable precision, make a binary encoding very useful in the solution of building optimization problems.

2.2. Fitness assignment

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 selection. Then, solutions (obtained from EnergyPlus simulation program runs) will be ranked-ordered (stochastic ranking) based on the energy use obtained from objective function. Hence the stochastic ranking will simply rank all solutions based on their objective function values alone. It sorts the solutions in order of the "best" to the "worst".

2.3. Selection

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). In this research, we seek robust convergence with as few building simulations as possible (that is, reliable convergence with a high convergence velocity). The tournament operator randomly selects n-solutions from the population, the winner solutions, which have better ranking, out of the tournament which are carried forward for recombination.

One measure of the "selection pressure" is the "takeover time" (Bäck, 1996). This is the number of generations (algorithm iterations), for the population to be filled with the best solutions found in the initial generation, in the absence of recombination and mutation. The takeover time for a tournament selection is approximated by Bäck (1996):

$$\tau = \frac{1}{\ln(n)}(\ln(q) + \ln(\ln(q))) \tag{1}$$

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). In Table 1, the takeover times for a binary tournament ($n = 2$) and three different populations sizes are shown. The effect of population size on the selection pressure is clear from Table 1; with a population size of 5 individuals, convergence is achieved in under half the number of iterations required for a population of 30.

The table also includes the number of tournaments, τ', necessary to fill the population with the best of the initial solutions (taken as, $\tau' = \tau\, q - q$).

2.4. Crossover

The recombination operator controls the mixing of "genetic information" selected from paired individuals through a process known as "crossover" (each individual in the pair resulting from a separate tournament selection). It takes place by swapping bit values between the two individuals. In the "uniform crossover" operator used in this research, each pair of bits is swapped with a 50% probability (an average of 50% of the bits will be swapped). The effect of chromosome crossover probability on the performance of the search is examined here by applications.

2.5. Mutation

A probabilistic bit-wise mutation, in which a given gene value if flipped from 0 to 1, or vice versa, was adopted in this study. The effect of mutation probability on the performance of the search is examined here through applications (trials).

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

2.6. Replacement and elitism

There are many methods of keeping the "best solution so far" in the population at each iteration of the process. In this research, all solutions were replaced, except the solution having the highest fitness (the "elite" solution) of the current solutions will be re-added to the next generation solutions. This guarantees that the search does not diverge to a solution having a higher value of the objective function than that already found by the search.

2.7. Convergence and automatic restart

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 work we choose to identify the collapse of the population in terms of the objective function. Collapse of the population is defined by:

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

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

If the current population has collapsed, then the next population is first re-seeded with the elite (best) solution, and remaining individuals will be randomly initialized within their bounds. This strategy is normally applied to a "micro-GA" as in Caldas and Norford (2002); because micro-GA use small population sizes, which due to the high selection pressure have a tendency to converge prematurely. A value of $\alpha \leqslant 1\%$ was used in this study to indicate convergence and trigger re-initialization of the population.

2.8. Recall simulated solutions

GAs are conventionally stopped after a fixed number of algorithm iterations (generations). However, since one 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 is guaranteed to be unique. The GA optimization algorithm structure is summarized in Fig 1.

binary variable encoding;
stochastic ranking fitness assignment;
tournament selection;
uniform crossover;
bit-wise mutation;
replacement with single elite;
automatic reseeding for premature convergence;
convergence defined by the number of new trial simulations;
recall simulated solutions.

Figure 1. Selected GA structure.

3. Setup of EnergyPlus and genetic algorithm

3.1. Coupling GA and EnergyPlus

In order to implement the simulation-based building optimization problem that coupled the genetic algorithm (GA) with the building simulation program (EnergyPlus), a JAVA code has been created to handle the simulation building problems. The code is structured based on the selected operators and to test different GA control parameters. A generic template input file is designed to read all GA parameters such as population size, selection operator type, crossover probability, mutation rate, number of allowed function calls, and number of random sequence generators. The input file also contains all the design variables for each application (problem) type (unconstrained in this case). Also, EnergyPlus input template files have been prepared for the building under consideration (a typical office building of five-zone offices). In this example all the required building simulation parameters are defined for both the system model (HVAC system components) and the air side model (duct and air terminal units).

As shown in Fig. 2 these two input files are used to initiate the GA optimization process to create the EnergyPlus input file (design variables by GA and EnergyPlus template). Then the building will be simulated by the EnergyPlus and generate the output file. Finally, GA will read these output values to evaluate the building energy consumption. This process will be looping until the stopping criterion is satisfied, which is 500 simulation calls.

3.2. Selection of control parameters

Naturally, GAs have many parameters that need to be tuned in order to find the best performance for any optimization problem. The main control parameters are the population size, selection operator, the crossover probability p_c,and the mutation probability p_m.In order to identify suitable control parameter values for solving building optimization, the performance of the GA will be evaluated for several different sets of these parameters, see Table 2.

From the combination of these control parameters, twelve different sets of GA control parameters will be tested. These will be implemented on the unconstrained building optimization problem. As it is known that GAs are a probabilistic optimizer, repeated runs are required (10 different runs) to eliminate the effects of the initial random selected solution.

4. Description of studied building

In this paper, a typical office building layout was chosen as an example to test the GA performance. The building is virtually located at Chicago, Illinois (42° Latitude, −88° Longitude). This building is designed as a mid-floor in a multiple story building which consists of five zones. Each of the external zones (North, South, East and West) has an exterior wall along its perimeter with a single window

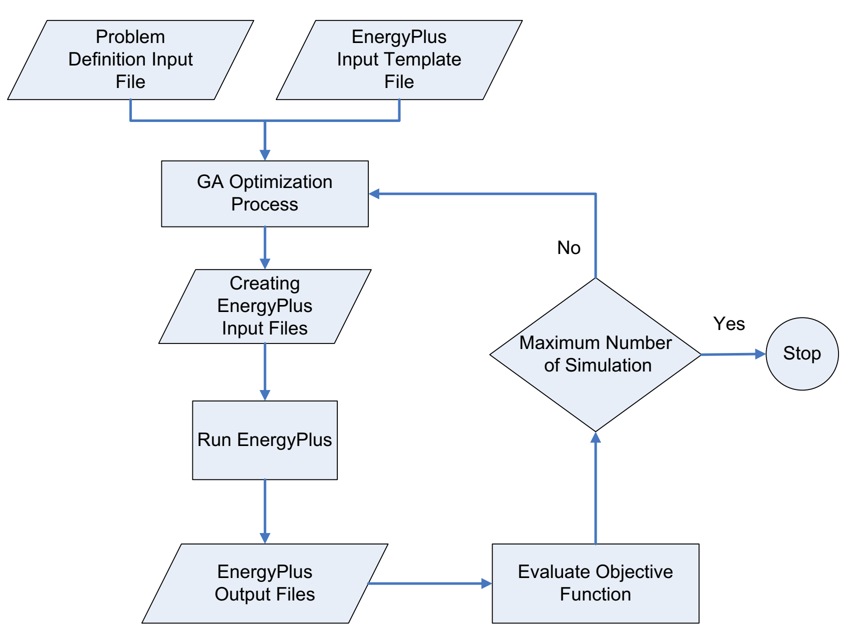

Figure 2. Implementation of model-based building optimization problems on this research.

Table 2
GA control parameter sets.

Control parameter	Values
Population size (–)	[5,15,30]
Probability of Crossover	[0.7,1.0]
Probability of Mutation	[0.01,0.02]

and overhang shading. The internal zone "I" is bounded by partition walls of perimeter zones.

There are four interior doors that connect the zones. The total floor area of the building is ($46 \times 24 = 1104\ m^2$), and all zones have a height of 2.7 m. The longer side of the building is initially oriented toward a northerly direction as shown in Fig. 3.

The HVAC system was a unitary direct exchange package unit (DX) with central electrical heating. The air side of this HVAC system is selected to be a single duct with a variable air volume (VAV) terminal and reheat element for each zone. The zone load is met by varying the zone air flow rate until the set point temperature is maintained within the capacity of the HVAC system.

4.1. The building design variables

The design variables are the parameters that have influences on the optimization problem. As the variables increase, the problem complexity increases. In this paper, the design variables with their lower and upper bounds and increment are listed in Table 3. The discrete increment in each design variable is set based on engineering tolerance and the variable characteristics.

In Table 3, the list represents the envelope design variables that influence the building energy consumption. These variables are the window area sizes, which are represented by the window's height and width to form a glazed area starting from 10% of the wall area to full glazing façade area (indices 1–8). Associated with the window area variables are the window' overhang over the North, West, East, and South windows, respectively (indices 9–12). These overhangs were allowed to vary from a minimum depth of 0.0 m (measured from the façade base) up to 1.5 m in steps of 0.05 m. Also, the window construction is formed from three layers, internal window material, window gas-filling types, and external type's material. Internal and external window materials have four different alternative window materials (see Table 3) whereas the window gas-filling contains five different gas types. In Table 3 variable indices from 16 to 19 represent the envelope construction, i.e. external walls, internal walls, ceiling and floors, respectively. These variables alternate between three material construction types: heavy, medium or light. Also, three insulation thicknesses, varying from 0.05 to 0.2 m, for each wall type are defined (indices 20–22). Lastly, the building orientation is permitted to change from 0° "North" direction to 90° "East" direction (index 23).

4.2. Objective function

In this study the objective function is the annual primary energy consumption of the studied building. It comprises three main energy consuming elements of a building: lighting, fan(s), and cooling and heating systems. This objective function of the annual energy consumption, $f(x)$, can be expressed as follows:

$$F(x) = [Q_h(x) + Q_c(x) + E_l(x)]/3.6 \times 10^6 \quad (3)$$

where $Q_h(x)$ and $Q_c(x)$ are the zones' annual heating and cooling consumptions (Joules), respectively, and $E_l(x)$ is the zones' electricity consumption for the lighting and fan. The denominator is a conversion factor to get the energy in kilo Watt hours (kWh) instead of Joules.

5. Evaluation of GA performance

Each parameter set shown in Table 3 was run 10 times, each time with different initial starting search points to

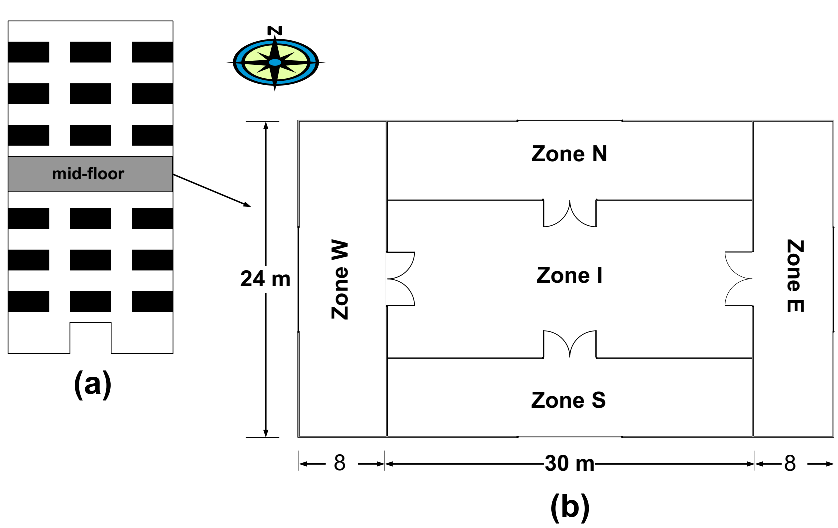

Figure 3. Example building (a) front view of office building and (b) top view layout of the five-zone studied floor.

Table 3
Building envelop design variables.

Index	Variable	Lower bound	Upper bound	Increment
1	North window width (m)	0.5	29	0.1
2	South window width (m)	0.5	29	0.1
3	East window width (m)	0.5	23	0.1
4	West window width (m)	0.5	23	0.1
5	North window height (m)	0.5	2.1	0.1
6	South window height (m)	0.5	2.1	0.1
7	East window height (m)	0.5	2.1	0.1
8	West window height (m)	0.5	2.1	0.1
9	North window overhang (m)	0.0	1.5	0.05
10	South window overhang (m)	0.0	1.5	0.05
11	East window overhang (m)	0.0	1.5	0.05
12	West window overhang (m)	0.0	1.5	0.05
13	Window layer 1 specification (–)	0	3	1
14	Window gas types (–)	0	5	1
15	Window layer 2 specification (–)	0	3	1
16	External wall construction (–)	0	2	1
17	Internal wall construction (–)	0	2	1
18	Ceiling construction (–)	0	2	1
19	Floor construction (–)	0	2	1
20	Heavy wall insulation thickness (m)	0.05	0.2	0.05
21	Medium wall insulation thickness(m)	0.05	0.2	0.05
22	Light wall insulation thickness (m)	0.05	0.2	0.05
23	Building Azimuth (°)	0	90	5

eliminate the effect of initial randomness selected solutions. The minimum, maximum, mean, and standard deviation of the objective function for each parameter set is summarized in Table 4. The second column of this table gives the parameter sets as follows: population size, crossover rate, and mutation rate, respectively.

5.1. Statistical hypothesis and T-test

A statistical hypothesis has been utilized to compare these parameter sets (the null hypothesis and/or alternative hypothesis). The null hypothesis, H_o presumes that there is no difference in the mean values between the compared samples, whereas the alternative hypothesis, H_1, presumes that there is a difference in the mean values existing between the compared samples. A total of 66 comparisons were made between the parameter sets, as shown in Table 5.

In this table the value in the intersect cell between the two compared parameter-sets represents the t-test value. The critical t value ($t_{critical}$) for the 95% confidence interval is determined ($t_{critical} = 2.26$). The two compared parameter sets are considered to be statistically different when their t-value exceeds $t_{critical}$.

As shown in Table 5, the highest population sizes (30) show a significant percentage difference to the small populations, in particular with a mutation rate ($p_m = 2$) and with a population size of 15 and a mutation rate of 1. Also, the population size (15) shows a slight percentage difference with small populations, low crossover probability, and a mutation rate of 2.

Also, the data in Table 5 can be used to determine the number of times that a parameter set is statistically worse than any other parameter set. For example, the parameter set [15,0.7,2] results in 4 worse solutions which gives a 36% probability of this parameter set giving a worse result

Table 4
Final best objective function values (annual energy consumption).

Index	Parameter set	Min. (MWh)	Max. (MWh)	Mean (MWh)	Std. Dev.
1	[5,1.0,1]	78.0	79.6	78.6	0.49
2	[5,1.0,2]	78.0	79.3	78.5	0.40
3	[5,0.7,1]	78.2	79.3	78.6	0.36
4	[5,0.7,2]	78.2	79.6	78.7	0.43
5	[15,1.0,1]	77.9	79.8	78.7	0.57
6	[15,1.0,2]	78.0	80.5	78.9	0.67
7	[15,0.7,1]	78.0	79.3	78.5	0.38
8	[15,0.7,2]	78.6	79.7	79.0	0.39
9	[30,1.0,1]	78.2	79.8	78.9	0.50
10	[30,1.0,2]	78.8	79.8	79.2	0.35
11	[30,0.7,1]	78.2	80.0	79.0	0.50
12	[30,0.7,2]	78.9	80.0	79.3	0.36

Table 5
Paired t-values.

	Index	2	3	4	5	6	7	8	9	10	11	12
Index	Parameter sets*	[5,1.0,2]	[5,0.7,1]	[5,0.7,2]	[15,1.0,1]	[15,1.0,2]	[15,0.7,1]	[15,0.7,2]	[30,1.0,1]	[30,1.0,2]	[30,0.7,1]	[30,0.7,2]
1	[5,1.0,1]	0.56	0.43	0.29	0.13	1.15	0.98	1.79	1.13	3.97	1.36	4.78
2	[5,1.0,2]		0.12	0.81	0.59	1.50	0.43	2.49	1.31	4.29	2.72	4.12
3	[5,0.7,1]			0.90	0.74	2.04	0.58	2.72	1.39	4.20	2.36	5.53
4	[5,0.7,2]				0.12	1.37	1.60	5.25	0.75	3.95	1.54	2.73
5	[15,1.0,1]					0.81	1.05	1.33	0.72	2.76	1.94	3.77
6	[15,1.0,2]						2.24	0.50	0.30	1.48	0.31	1.36
7	[15,0.7,1]							3.53	2.18	6.15	2.88	4.60
8	[15,0.7,2]								0.87	1.50	0.03	1.25
9	[30,1.0,1]									2.56	0.69	2.19
10	[30,1.0,2]										1.02	0.21
11	[30,0.7,1]											1.09

*Population size, crossover probability, and mutation rate.

Table 6
Effect of population sizes.

Index	Parameter sets	Percentage difference
1	[5,1.0,1]–[15,1.0,1]	−0.13
2	[5,1.0,2]–[15,1.0,2]	−0.51
3	[5,0.7,1]–[15,0.7,1]	0.13
4	[5,0.7,2]–[15,0.7,2]	−0.38
5	[5,1.0,1]–[30,1.0,1]	−0.38
6	[5,1.0,2]–[30,1.0,2]	−0.89
7	[5,0.7,1]–[30,0.7,1]	−0.51
8	[5,0.7,2]–[30,0.7,2]	−0.76
9	[15,1.0,1]–[30,1.0,1]	−0.25
10	[15,1.0,2]–[30,1.0,2]	−0.38
11	[15,0.7,1]–[30,0.7,1]	−0.64
12	[15,0.7,2]–[30,0.7,2]	−0.38

comparison to the other parameter sets (each set has been compared with 11 others so that $4/11 \times 100 = 36\%$. Similar calculations for all parameter were conducted showing 64%, 55%, and 27% worse probability for [30,1.0,2], [30,0.7,2] and [30,0.7,1], respectively.

The comparison shows that most of the parameter sets that contain a population size of 30 in their control parameters (except for [30,1.0,1]) are the ones that have highest probability of a worse solution compared with others.

5.2. Effect of crossover probability and mutation rate

A further analysis is performed by studying the impact of population size on the performance while the other parameters (crossover probability and mutation rate) were equated so as not to influence the compared parameter sets, see Table 6. In this comparison, twelve different parameter sets were compared to study the population size effect.

From this table it can be noticed that a smaller population size always has positive impact on GA performance by finding a better objective function (less energy consumption of the tested building). Also, it can be observed that the significant statistical difference appeared when the population size of 30 takes part in the control parameter set (indices 5–12). The shaded cells represent the significant percentage difference between the compared parameter sets (indices 2, 6, 7, 8, and 11).

The data listed in Table 7 show that the crossover probability shows no statistical significance on the GA's performance. It was noticed that high crossover probability showed poor performance relative to the low crossover probabilities, particularly with a small mutation rate.

The effect of mutation rate on GA performance is insignificant, except with mid size population at low crossover rate. However, the low mutation rates showed better performance than the high ones as can be seen in Table 7.

Table 7
Effect of crossover probability and mutation rate.

Crossover probability			Mutation rate		
Index	Compared sets	Percentage differences	Index	Compared sets	Percentage differences
1	[5,1.0,1]–[5,0.7,1]	0.00	1	[5,1.0,1]–[5,1.0,2]	0.13
2	[5,1.0,2]–[5,0.7,2]	−0.25	2	[5,0.7,1]–[5,0.7,2]	−0.13
3	[15,1.0,1]–[15,0.7,1]	0.25	3	[15,1.0,1]–[15,1.0,2]	−0.25
4	[15,1.0,2]–[15,0.7,2]	−0.13	4	[15,0.7,1]–[15,0.7,2]	−0.64
5	[30,1.0,1]–[30,0.7,1]	−0.13	5	[30,1.0,1]–[30,1.0,2]	−0.38
6	[30,1.0,2]–[30,0.7,2]	−0.13	6	[30,0.7,1]–[30,0.7,2]	−0.38

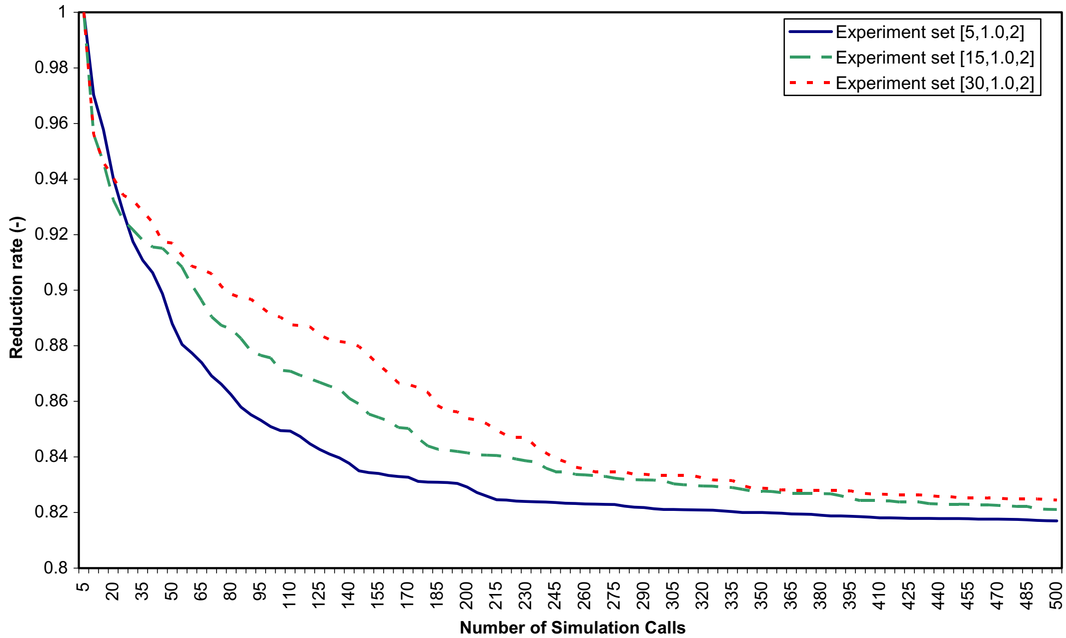

Figure 4. Effectiveness of population sizes on GA performance of crossover probability 1.0 and mutation rate 2 based on the best solution in every generation.

This is in agreement with the earlier findings of the authors (Alajmi and Wright, 2006).

Based on the above statistical analysis on the unconstrained building optimization problem, a conclusion can be drawn that the parameter sets that compromise the largest population size (30) have the worst performance. However, to find which of the remaining population sizes perform better than the others, the number of simulation calls required to reach convergence (convergence velocity) will be examined in the following section.

5.3. Convergence velocity

As the stopping criteria in the Genetic optimization algorithm (GA) in this work was considered as the number of simulation calls, it is good to study how each population size affects the GA performance until it converges. This can be done by comparing the reduction ratio from the initial solution until the optimization process is stopped (500 simulation calls). The reduction ratio is given by the following formula;

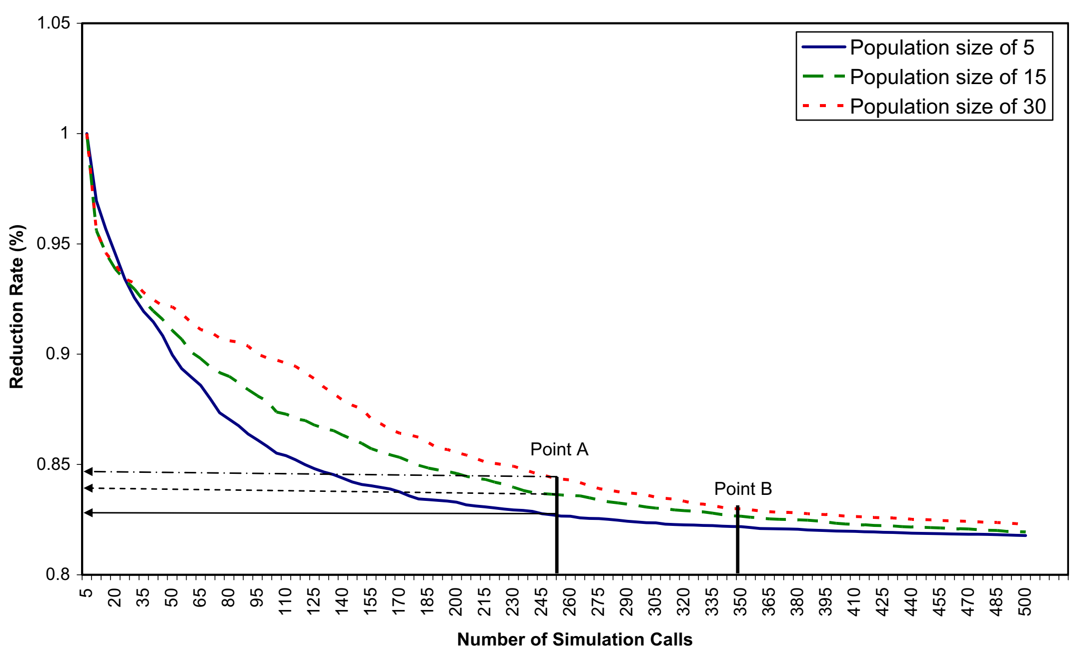

Figure 5. Effect of population size on the GA convergence velocities for the averaged data of the tested 12 sets (4 sets are averaged for the same population size in each curve).

$$r(\bar{f}(U_i), \bar{f}(U_1)) = \frac{\bar{f}(U_i)}{\bar{f}(U_1)} \quad (4)$$

where: U_i is the mean of a set of solution given by $U_i = \{X|(X_1, \cdots, X_{np})\}$, np is the number of solutions considered in each set. Since the smallest population size of parameter sets that are used is a size of 5 solutions, the objective function means have been calculated for every 5 new trial solutions ($nb = 5$) while, $\bar{f}(U_1)$ is the mean objective function value of the initial 5 solutions. Note that this analysis is performed on the sequence of new objective function values rather than the sequence of all function evaluations, the number of simulation calls (new function evolutions) is the focus criterion of this research. Since a previously evaluated solution appear in a particular population, then it is not considered in this analysis (as it will be simply recalled from memory and not require a further simulation). It is worth noting that basing the reduction ratio on the mean of 5 solutions has some effect on the results "smoothing the results".

The effect of population size on the GA convergence velocity and the reduction rate of the best solutions, while the crossover probability and mutation rate are kept constant, is compared in Fig. 4. Note that these curves are for the best solution found after the given number of simulation calls (500). Clearly, the small population size (5) shows a better reduction rate and faster convergence toward the optimum solutions. Similar trends to those in Fig. 4 are experienced with different parameter sets. An average reduction ratio for the four parameter sets in each of the population size is given in Fig. 5.

From Figs. 4 and 5, the population size of 5 shows its superiority over the other population sizes (15 and 30). It can be also seen that a less number of simulations are required if the building designer does not mind to scarify some of the ultimate reduction ratio that was reached by the high number of simulation calls (500). The number of simulation calls is dramatically reduced to 250 with a relatively close value of reduction rate, see point A and B in Fig. 5. An assumption has been made that the building designer will terminate GA at 250 or 350 number of simulation calls. The reduction rate reached at those cross points is much lower for the smallest population size (5) than it is for the population size of 15 and 30.

6. Conclusion

The results show that GA performance is not-sensitive to most control parameter values, such as crossover probability and mutation rate, since there was no statistically significant difference between the optimum solutions. However, the population size was the control parameter that had the most significant effect among the other control parameters. The small population sizes show better results than the large population size. Further, the smallest population sizes showed a greater reduction rate for a smaller number of simulation calls (less than 350) compared with the other population sizes.

Although the crossover probability and mutation rate showed less sensitivity on the GA performance, there are some evidences that suggested that they are performing better with the higher crossover probability (1.0) and lower mutation rate (1).

Ultimately, a general conclusion that was possibly drawn at the end of this study is that the small population size, high crossover probability, and low mutation rate are the most appropriate control parameter sets for the unconstrained building optimization problem.

References

Alajmi, A., Wright, J., 2006. Efficient optimisation of building design using a genetic algorithm, Loughborough University.

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), Inst. Phys. ISBN 0750306653.

Caldas, L.G., 2001. An evolution-based generative design system: using adaptation to shape architectural form, thesis, Massachusetts Institute of Technology.

Caldas, L.G., Norford, L.K., 2001. Architectural constrains in a generative design system: interpreting energy consumption levels, 7th International IBPSA.

Caldas, L.G., Norford, L.K., 2002. A design optimization tool based on a genetic algorithm. Autom. Constr. 11 (2), 173–184.

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

Crawley, D.B., 2001. EnergyPlus: the future of building energy simulation. U.S. DOE replaces DOE-2 and BLAST. HPAC. Heating Piping Air Cond. Eng. 73 (11), 65–67.

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

EU, 2002. European Union – The new Directive on the Energy Performance of Buildings. First ed., EU.

Guillemin, A., Molteni, S., 2002. An energy-efficient controller for shading devices self-adapting to the user wishes. Build. Environ. 37 (11), 1091–1097.

Guillemin, A., Morel, N., 2001. An innovative lighting controller integrated in a self-adaptive building control system. Energy Build. 33 (5), 477–487.

Jo, J.H., Gero, J.S., 1998. Space layout planning using an evolutionary approach. Artif. Intell. Eng. 12 (3), 149–162.

West, A., Sherif, S., 2001a. Optimization of multistage vapour compression systems using genetic algorithms. Part 2: Application of genetic algorithm and results. Int. J. Energy Res. 25 (9), 813–824.

West, A., Sherif, S., 2001b. Optimization of multistage vapour compression systems using genetic algorithms. Part 1: Vapour compression system model. Int. J. Energy Res. 25 (9), 803–812.

Wetter, M., Wright, J.A., 2003. Comparison of a generalized pattern search and genetic algorithm optimization method. 8th International IBPSA Conference.

Wright, J.A., Farmani, R., 2001. The simultaneous optimization of building fabric construction, HVAC system size, and the plant control strategy. 7th International IBPSA Conference, IBPSA, pp. 1111–1112.