Implementation of a genetic algorithm for energy design optimization of livestock housing using a dynamic thermal simulator

Maria Elena Menconi, Massimo Chiappini, David Grohmann

Department Uomo e Territorio, University of Perugia, Italy

Abstract

A Genetic Algorithm (GA) is an optimization process inspired by natural systems ability of surviving in many different environments through the mechanisms of natural selection and genetics. The pairing of GA-based optimization techniques with dynamic energy models is a common and effective practice to find energy efficient design solutions. In this paper is implemented an optimization tool that use a GA and a dynamic energy model. Efficiency of GAs depends largely on the coding strategy and on the parameters selection. In order to test the code and to find the best combination of parameters, a parametric analysis of GA's performances is carried out. The algorithm, coded in Matlab, works with populations of strings. Each string, that represents a complete design solution, is initially randomly generated by the GA and evaluated in terms of energy performances by the dynamic thermal simulator. A new population is then generated using three different GA stochastic operators: reproduction, crossover and mutation, by selecting, mixing and randomly modifying the fittest solutions of the previous generation. Each generation is energetically evaluated and thus the fitness of the strings, that represent the energy efficiency of the design solutions, improves every cycle till eventually converge to the best solution. This whole methodology is well documented and applied in residential buildings design but can be easily extended to livestock housing. In this paper the algorithm is coded to be applied on a simple sheepfold model in order to optimize only passive design solutions.

Introduction

Objectives of the research

In this work is investigated the interaction between various parameters of an implemented Genetic Algorithm (GA) for the analysis of the energy efficiency of the external envelope of sheepfolds, with the aim to identify the best combinations of these parameters.

The interaction between the parameters depends mainly on the type of function to be optimized (Hart W.E. and Belew R.K., 1991). Many studies have been conducted to find the optimal parameter setting of GAs (De Jong K., 2007), some of them take into consideration the interaction between only two parameters and analyze the variations that the pairing generates in the algorithm performance (Goldberg D.E. *et al.*, 1992). In some cases, empirical studies have been conducted (Wu. A. *et al.*, 1997) or complex stochastic models like the Markov chains (Chakraborty U. *et al.* 1996; Chakraborty U and Janikow C.Z., 2003) have been used.In this paper, which is inspired by a work of Kalyanmoy Deb (1998), we addressed the effects induced by five parameters of the GA: population size, crossover probability, mutation probability, encoding and reproduction strategy.

Genetic Algorithm

The GAs are research and optimization systems which are inspired by the evolution of natural systems and by their ability to adapt to many and varied external conditions in effective and efficient manner.

Their strength, compared to other search systems, is given by three factors: the GA do not operate on individual points but on populations of points, reducing the likelihood of running into local minima; they act directly on the code and not on the functions, making them difficult to deceive; they make use of semi probabilistic operators, enabling a highly exploratory research (Goldberg D.E., 1989).

Before a GA can be developed it is necessary to choose the type of encoding to be used for the representation of the chromosome’s parameters. The quality of a GA depends largely on the coding strategy adopted and on its interaction with the operators of crossover and mutation, in respect of the variables and constraints of the problem. Binary type is one of the most used encodings.

As schematically represented in Figure 1, the basic element of a GA, the individual, is constituted by a string, chromosome, containing a set of sequenced parameters, said genes. Each individual represents a possible solution to the given problem, more individuals form a population.

The algorithm is initialized with the random generation of the first population. The iterative loop begins with the evaluation of the quality of an individual (that is how good the solution is to the problem) which is measured by a fitness function.

At each generation, the individuals of the population are stochastically selected in agreement with the value of the fitness function and are paired to generate better individuals.

The evolutionary process is carried out by three basic operators: reproduction, crossover and mutation:

- with the reproduction the strings are selected according to their fitness value and are copied to the mating pool to generate the next population;
- through crossover couples of chromosomes are randomly paired to generate new individuals;

Correspondence: Maria Elena Menconi, University of Perugia, Uomo e Territorio Department
e-mail of corresponding author: mariaelena.menconi@unipg.it

Key words: Genetic algorithm, building energy optimization, dynamic thermal model, parametric analysis.

©Copyright M.E. Menconi et al., 2013
Licensee PAGEPress, Italy
Journal of Agricultural Engineering 2013; XLIV(s2):e38
doi:10.4081/jae.2013.s2.e38

This article is distributed under the terms of the Creative Commons Attribution Noncommercial License (by-nc 3.0) which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

- finally, the mutation involves the random variation of values of the genes of the individual and has the important task to prevent the premature convergence and therefore the loss of important genetic material (Haupt R. and Haupt S.E., 2004).

The purpose of the GA is the identification of chromosomes that optimize the fitness function.

Materials and methods

Flowchart of working methodology

The methodology adopted for the performance analysis of the implemented GA is shown in Figure 2.

The work can be divided into three main steps:

- analysis of the energy performance of the building;
- implementation of the GA;
- encoding of the statistical cycle and parametric analysis of the GA performance.

In optimizing the energy efficiency of building envelopes, the coupling between thermal simulation models and optimization algorithms is a widely tested practice(Caldas L.G. and Norford L.K., 2002; Znouda E. *et al.*, 2007). One of the most common methodologies involves the continuous exchange of data between the simulation and the optimization models, where the first carries out the task of calculating the energy performance of the building, and therefore the values assumed by the objective function, while the second identify the best technological and constructive solutions. In this study has been adopted a simplified interaction methodology between the simulation model and the energy optimization algorithm. In our case, in fact, using the simulation energy model, all the different construction types examined have been simulated in advance, analyzing the effects of a single construction element at a time, not taking into account the interaction between the different components of the building envelope. The different energy consumption values thus obtained were subsequently placed inside the optimization algorithm GA through a legend function that associates to each type of construction a corresponding sequence of integers.

Not considering the interactions between the various types of casing and being the simulation model and the optimization algorithm unable to exchange data in real time, make it impossible to conduct a rigorous analysis of the energy performance of the building, but only a qualitative analysis of the energy efficiency of three types of coating (vertical wall, windows and roof). However, for the purpose that has been set in this work, namely the analysis of the performance of the GA varying some significant parameters, the use of this simplified approach is more than justified, not going to affect any of the obtained results.

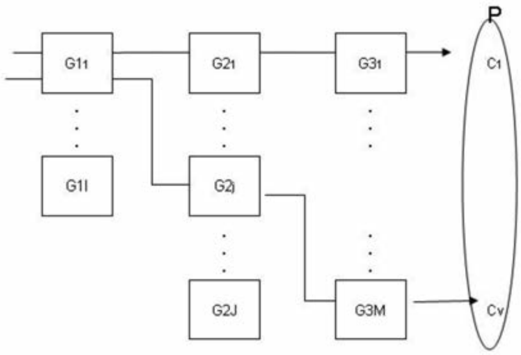


Figure 1. Representation of the constituent elements of the GA, genes, chromosomes and population. Genes: G1i, G2j, G3m, represent the different parameters that we want to optimize (in this case 3 parameters); variables counter i, j and m that vary respectively from 1 to I, J, M (the number of decision alternatives for each gene). Chromosome : Cv (G1i, G2j, G3m), represents a combination of the selected parameters, with v that varies from 1 to V (total number of chromosomes that constituting the population). Population: P (C1,…,CV) represents the set of chromosomes. Each iteration of the GA provides for the generation of new chromosomes, which will form a new population.

Energy simulation software and building model

The chosen software for the energy performance analysis of the building is Energy Plus, that can be downloaded from the following site: http://apps1.eere.energy.gov/buildings/energyplus/energyplus_download.cfm.

This software has been selected as it is free downloadable and is one of the most complete software for dynamic simulation of the energy performance available. In the present work it was decided to act only on existing sheepfolds, choosing between the range of passive solutions available, the most simple and immediate interventions on the exterior of the building. The building model includes the definition of its exact geographical location, its orientation, the geometric characteristics (3D model), a detailed description of the materials (properties and thickness) that compose the opaque and transparent surfaces, the modeling of infiltration, the description of the gain / consumption factors (people, animals, lights, electric equipment, air conditioning sys-

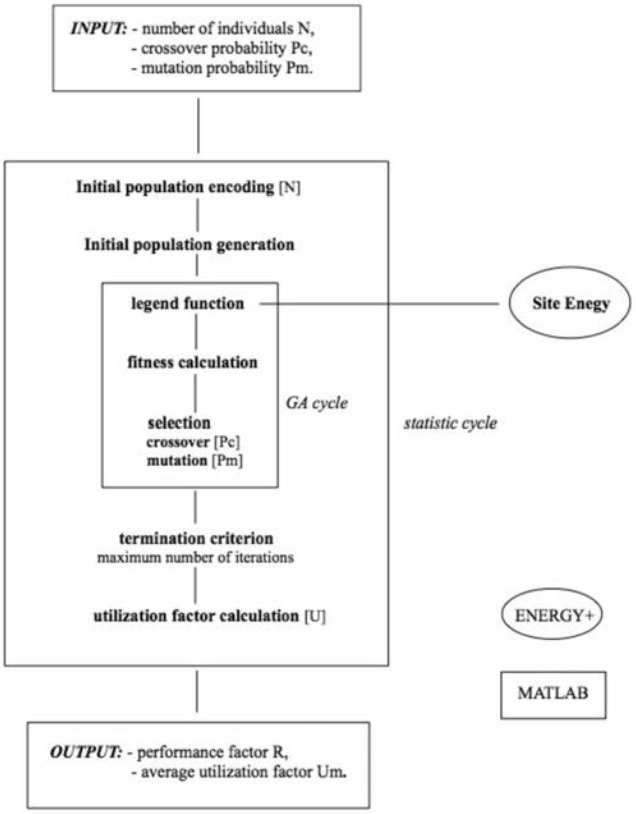

INPUT: - number of individuals N,
- crossover probability Pc,
- mutation probability Pm.
Initial population encoding [N]
Initial population generation
legend function
fitness calculation
selection
crossover [Pc]
mutation [Pm]
GA cycle
Site Enegy
statistic cycle
termination criterion
maximum number of iterations
utilization factor calculation [U]
ENERGY+
MATLAB
OUTPUT: - performance factor R,
- average utilization factor Um.

Figure 2. Flowchart of the adopted methodology. Starting from three GA parameters (input), number of individuals (N), mutation probability (Pm) and probability of crossover (Pc), have been obtained and analyzed the algorithm performance, by examining the performance parameter (R) and the average utilization parameter (Um).

tems). Between all the parameters listed above, those chosen to be varied are the genes of the GA developed in this work. The first step concerns the choice of the number of parameters to be varied in energy simulations (number of genes) and, for each parameter, the number of changes (for example the gene "wall's insulation" may differ depending on type, thickness or position of the insulation layer). The existence of an ideal HVAC system has been hypothesized to evaluate the energy savings that can be achieved by the introduction of an intervention on the building enclosure. The different Site Energy values, that are generated by the Energy Plus run varying the individual parameters, are the input values for the genes of the GA, according to which the chromosomes and the population are formed.

GA implementation

The GA developed for this study was coded using Matlab (MATLAB Version 7.1.0.124 (R14)), a programming language that works with matrices, strings and numeric operations. For the main functions of the GA a toolbox for Matlab developed by the University of Sheffield was used (Chipperfield A. *et al.*, 1995).

The first step is to choose the encoding strategy, in our specific case, having to deal with numbers that vary in discrete terms and that are included in well-defined intervals, was chosen a binary type encoding. All evolutionary processes take place through the manipulation of binary strings that represent, through sequences of integers, different configurations of the outer shell (Figure 3).

In this step, in addition to the type of encoding, all the data that characterize the structure of the population, such as the number and the length of the individuals are inserted. Following the coding of the population, the algorithm is initialized with the random generation of the first population. The iterative cycle begins with the assignment, to each gene, of the corresponding value of Site Energy, so that the value of the objective function can be calculated and its fitness assigned. The objective function was defined as the average of the three values of Site Energy associated with the genes of each chromosome, a function that the GA has the aim to minimize. The next step is the selection of individuals for the reproduction via Stochastic Universal Sampling, a selection method consisting of a semi-probabilistic roulette with X equally spaced arms, that, at every cycle, select X individuals arranged in sectors whose size is proportional to the owned fitness (Baker E., 1987). The selected individuals are then mated through single point crossover, a type of crossover through which the strings are broken in a single point and the resulting segments are crossed together. Following the crossover operator there is the mutation one which, randomly varying bits of individuals, introduce new genetic material. Both crossover and mutation do not automatically apply to every individual, but in accordance with determined probabilities.

To facilitate the algorithm convergence has been adopted an elitist reproduction strategy, that consists in keeping the best individuals of each generation and to insert them directly into the next one. This allows to avoid the destruction of valuable genetic material, thus improving the performance of the algorithm (Dumitrescu D. et al, 2000). In the implemented GA a 10% of individuals go directly to the next generation, the selection of these individuals is carried out based on the fitness value, in order to retain only those most efficient. The cycle ends when it reaches a predetermined number of generations.

Parametric analysis

In this paper we addressed the effects induced by five parameters of the GA: population size, crossover probability, mutation probability, encoding and reproduction strategy. Since in actual optimization and research problems, as in the case of energy optimization, the main source of expenditure of computational resources is the evaluation of the values of the objective function (Deb K. and Agrawal S., 1998), the maximum number of evaluations to be performed in the various cycles of the GA needed to reach the optimal solution is fixed in advance (S).

If S is such a number, a GA with a population of N individuals requires a maximum of T = S / N generations, since at each generation are evaluated N functions. The value of S depends essentially on the complexity of the function to be solved and on the length of the string, as evidenced by Heinz Muhlenbein (1992).

Since the global optimal solution is known beforehand (given, for each variable in the GA, by the sum of genes that have the lower value of Site Energy), is possible to set up a statistical analysis of the effectiveness of the implemented GA, by reiterating several times (Cs) the basic cycle of the GA, consisting of T generations, and then evaluate a performance factor, called R, and a Utilization Factor, called U.

R is an indicator of the ability of the algorithm to find the optimum solution within the predetermined number of T generations and is calculated as: R = O / Cs, where O is the number of times that the algorithm is able to identify the optimal solution.

U studies the degree of utilization of the evaluations of the objective function, since not all S evaluations are always used to find the optimal solution, and is given by: U = 1 - F / S, where F is the number of evaluations actually carried out to reach the optimal solution.

R is calculated at the end of the Cs iterations of the statistical loop, while U is calculated at the end of every GA cycle. Once the GA cycle is

Table 1. Energy plus IDF Editor: variable instantiation for option zero - Envelope. Sheepfold building located in central Italy (Umbria; lat 42.87, long 12.96), time zone 1 and elevation 974 m a.s.l. The total building area is 777.35 m^2. The weather file used is Perugia – ITA IGDG WMO#=161810 and the hours simulated are 8760.

Variable name	Range or value	
Orientation (degree from true North)	120	
Window-wall ratio (%)	4.89	
Window type	Simple glazing	
	U-factor (W/m2-K)	6
	Solar heat gain coefficient	0.7
Door type	Wood	
	Thickness (m)	0.009
	Conductivity(W/m-K)	0.14
	Density (Kg/m^3)	530
	Specific heat (J/Kg-K)	900
Floor type	Concrete	
	Thickness (m)	0.1
	Conductivity(W/m-K)	1.6
	Density (Kg/m^3)	2300
	Specific heat (J/Kg-K)	850
Wall type 1	Concrete block	
	Thickness (m)	0.2032
	Conductivity(W/m-K)	1.11
	Density (Kg/m^3)	800
	Specific heat (J/Kg-K)	920
Wall type 2	Metal	
	Thickness (m)	0.0008
	Conductivity(W/m-K)	45.28
	Density (Kg/m^3)	7824
	Specific heat (J/Kg-K)	500
Roof type	Fiber concrete	
	Thickness (m)	0.0065
	Conductivity(W/m-K)	0.35
	Density (Kg/m^3)	1500
	Specific heat (J/Kg-K)	1030

repeated Cs times, it is possible to calculate the average utilization factor Um, which represents the U average for the Cs simulations.

R can vary between 0 and 1 where R = 1 means that the algorithm always found the optimal solution in the Cs carried out simulations and 0 that it has never succeeded.

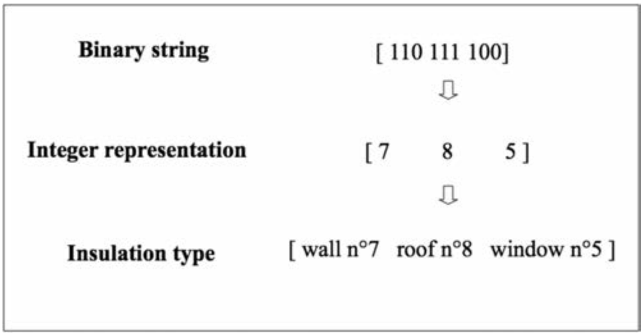
U can also vary between 0 and 1, with U = 1, the algorithm is able to identify the optimal solution in the first generation, while U = 0 means that S evaluations were not enough to identify the optimal solution.

Results

This section presents the results of the parametric analysis performed. The zero-building is a sheepfold located in central Italy, at an altitude of 974 meters above sea level (Figure 4). The building houses 200 sheep at full capacity and has two births periods per year, the operators are 2. The characteristics of the envelope are described in Table 1.

The GA structure was coded to represent the problem under consideration. The evaluation of the GA was made by varying, for the zero-building, the materials used for windows and for the insulation of the walls and cover, as shown in Table 2, in which are also reported the values of the Site Energy generated from different simulations, using the weather file of Perugia. From the dynamic energy simulation, the solution for the building which presents the best conditions for energy saving has shown that the best materials, respectively for glazed surfaces, insulation of the vertical walls and insulation of the roof are: aerogel, mineralized wood and polyurethane.

The 3 variables represented in table 2 are the 3 input variables in the GA. Each chromosome has then 3 genes of which the first, the windows, with values between 1 and 4, the second, the wall's insulation, and the third, the cover's insulation, with values between 1 and 8. Since with 2 bits is possible to represent 2^2 elements, and with 3 bits 2^3 elements, we have strings of 8 bits. Therefore 8 is the length of the population (L).



Binary string [110 111 100]

⇩

Integer representation [7 8 5]

⇩

Insulation type [wall n°7 roof n°8 window n°5]

Figure 3. Example of a string structure and decoding methodology: a binary string is converted into integer values that are then associated to insulation types.

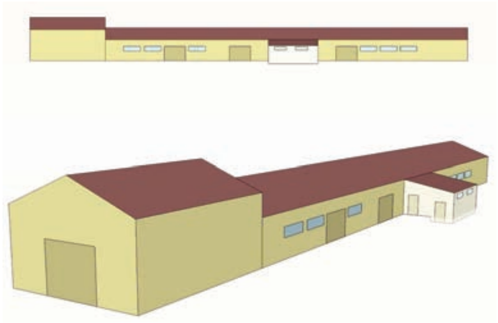


Figure 4. Sheepfold model.

Table 2. GA: variable instantiation.

Variable name	Variable type	Value	Site Energy (GJ)
Wintype (xi; i:1-4)	discrete	option zero	114.72
		low emission	113.26
		selective	113.06
		aerogel	112.43
Wall type: insulation (yj; j: 1-8)	discrete	option zero	114.72
outside layer: concrete block		sheep wool	103.57
layer 2: insulation		mineralized wood	102.34
(thickness: 0.05 m)		thatch	103.74
layer 3 : Lime-cement plaster (thickness: 0.01 m)		glass wool	103.94
		rock wool	103.24
		sintered polyester foam	103.48
		polyurethane	103.32
roof type: insulation (zm; m: 1-8)	discrete	option zero	114.72
outside layer: Fiber concrete (thickness: 0.0065 m)		sheep wool	41.97
layer 2: insulation		mineralized wood	45.13
(thickness: 0.05 m)		thatch	48.83
layer 3 : Lime-cement plaster (thickness: 0.01 m)		glass wool	43.78
		rock wool	42.37
		sintered polyester foam	41.69
		polyurethane	40.98

The simulations were conducted varying the number of the population N (4, 8, 12, 16, 20, 30, 60, 100, 150) and for various combinations of mutation probability (Pm) and crossover probability (Pc) respectively: 1/L – 0; 0.5/L – 0 ; 0 - 0.9; 1 / L - 0.9 ; 0.5 / L - 0.9 and 0.1 / L - 0.9. The combinations were selected so as to investigate the effects of the application of only crossover (for example the 0-0.9 combination), mutation alone (for example the 1/L -0 combination), or combinations of the two. S was set at 300, Cs at 600, this means that for each combination of N, Pc, Pm studied were carried out, for the calculation of R and Um, 300x600 = 180,000 evaluations of the objective function. From the analysis of the parameter R performance (Figure 5) it is possible to derive the algorithm's ability to identify the optimal solution: the algorithms with a high Pm behave very well and evenly from small to medium-high populations, 4 - 60 individuals, finding the optimal solution in 90% of cases. With decreasing of Pm, the curve loses its uniformity by presenting lower performance values, especially for medium-low populations. The algorithms based on crossover (high Pc) begin to give good results passed the threshold of 20 individuals, reaching peak performance, more than 90% success rate, at around the 60 individuals. Passed the peak the number of successes returns to decline with the increase of the population. This behavior can be explained considering that the crossover-based GA can rely solely on the genetic variability of the initial population, that clearly increases with a higher number of individuals present at the beginning of the simulation. All the analyzed types have a performance drop for large populations since, remaining unchanged the maximum number of evaluations of the objective function, decreases the number of cycles available to apply evolutionary operators, turning essentially from a semi-probabilistic to a random research method. The analysis of the U (Figure 6) enables to assess the efficiency of the algorithm in terms of speed (number of generations) of finding the optimal solution. For small populations (N = 4) all combinations have poor results (Um > 0.3). The combinations with low Pm tend to improve passed the threshold of 16 individuals, confirming previous observations. On the other hand, the combinations with high Pm tend to worsen with increasing number of individuals going from about 0.7, for 16 individuals, to about 0.57, for over 60 individuals. The best performances (Um > 0.7) were obtained by combining high Pc and Pm, in a population range between 8 and 30 individuals.

Further analysis was conducted on the effectiveness of the elitist strategy adopted (Table 3). Two series of simulations were carried out, one for a medium-low population (12) and another for a medium-high (60) one, in the absence of elitist strategy, that is letting participate to the processes of selection, reproduction and mutation all individuals of a given generation.

The parameters of performance and utilization were then compared with the relative series in the presence of elitist strategy. In both cases, the lack of this arrangement has led to a considerable decrease of the operating parameter, as shown in table 3, especially in combination

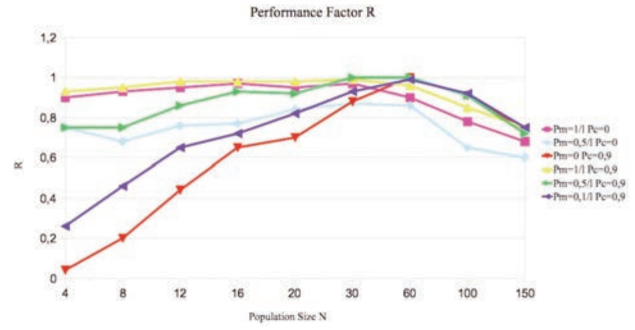


Figure 5. Performance Factor R varying population size for different combinations of crossover and mutation probability, Pc and Pm.

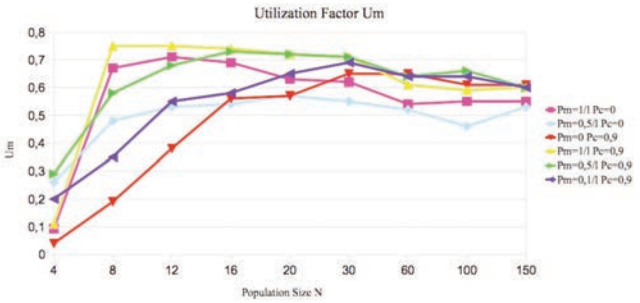


Figure 6. Average Utilization Factor Um varying population size for different combinations of crossover and mutation probability, Pc and Pm.

Table 3. Confrontation between different Um changing the number of chromosome reinsertion in the elitist strategy approach. Et=0,9 means 90% offspring reinsertion into the next generation, Et=1 means 100% offspring reinsertion (absence of elitist strategy).

N	Um		Pm	Pc
	Et 0,9	Et 1		
12	0,71	0,26	1/L	0
12	0,75	0,29	1/L	0,9
60	0,54	0,38	1/L	0
60	0,61	0,45	1/L	0,9
60	0,64	0,51	0,5/L	0,9
60	0,64	0,57	0,1/L	0,9

Table 4. Confrontation between Performance Factor R and Average Utilization Factor Um using Standard binary code and Gray code changing different input parameters.

	Input			Output	
	N	Pc	Pm	R	Um
Standard	8	0	1/L	0,93	0,67
Gray				0,99	0,76
Standard	8	0,9	0,5/L	0,75	0,58
Gray				0,98	0,76
Standard	8	0,9	0,1/L	0,46	0,35
Gray				0,76	0,49
Standard	12	0,9	0,5/L	0,84	0,68
Gray				0,99	0,76
Standard	12	0,9	0,1/L	0,6	0,49
Gray				0,84	0,62
Standard	16	0	1/L	0,97	0,69
Gray				0,99	0,76
Standard	16	0,9	0,5/L	0,93	0,73
Gray				0,99	0,75
Standard	16	0,9	0,1/L	0,72	0,58
Gray				0,9	0,62

with high Pm (1 / L). The effect tends to diminish in the series with the largest population, but it remains significant.

Finally, an analysis was performed on the type of encoding used. In general, the binary coding presents problems related to the so called Humming Cliff (Schaffer J.D. *et al.*, 1989). A coding problem that consists in phenotypic values (the values that assume the bit sequences of binary strings,) close to each other, described by totally different genotypic structures (the bit sequences of binary strings). To overcome the Humming Cliff problems is commonly used the Gray encoding (Caruana R.A. and Schaffer J.D., 1988), in which to variations of adjacent bits correspond minimal deviations of the values which those bits assume. In order to assess the effects of the binary encoding type on the performance of the algorithm, simulations were repeated in Gray code and then the performance and use of the parameters were compared with the relative simulations carried out in standard binary code. Going from a standard binary encoding to a Gray coding (tab.4) there are, for small and medium small populations (N = 4-16), improvements in both the performance and parameters utilization, in almost all the studied combinations. The improvement effect lowers with the increase of the population. With a population of 30 individuals are not observed substantial differences. The best results have been reported for small Pm (Pm = 0.5 / L or 0. 1 / L).

Conclusions

The carried out parametric analysis allowed us to measure the trend of the algorithm performance varying the size of the population (N), the probability of crossover (Pc) and mutation probability (Pc), in order to identify the best combination to be used in the energy optimization of the building enclosure.

We recall that the GA has been realized in standard binary code, with the use of an elitist strategy, the selection based on fitness using a Stochastic Universal Sampling, and the use of single point crossover.

From the obtained results (Figures 5 and 6), it can be deduced that the best overall performance (R and Um) can be attributed to an algorithm consisting of a population of 12 or 16 individuals, with Pc equal to 0.9 and Pm equal to 1 / L = 0.12, to which are associated a R equal to 0.98 and a Um of 0.75. Despite the small size of the space of solutions, 256 in total, however, it has been awarded the combination of parameters with the highest exploratory capacity (high values of both Pc and Pm).

The elitist strategy has certainly played an important role, in this mechanism, enabling not to waste the best solutions despite the high variability of the research process. This statement can be confirmed by observing the drastic reduction of Um, from 0.75 to 0.3, with the elimination of the elitist strategy (Table 3).

The overall picture changes going from a standard binary to a Gray encoding. Using a Gray encoding, the same performance (R = 0.99 and Um = 0.75) can also be obtained with the following combinations of N-Pc-Pm: 8-0-0,12 ; 8-0,9-0,06 ; 12-0,9-0,06 ; 16-0-0,12 ; 16-0,9-0,06 (Table 4). This result confirms that the Gray encoding is preferable in the development of GA similar to that used in this work.

This study is a preliminary work aimed at the realization of an energy optimization model for livestock production buildings. The next step would be to replace the simplified interaction methodology between the optimization algorithm and thermal simulation model used in this paper, with a complete methodology that allows the two systems to communicate simultaneously, taking into account all the interactions between the various components that contribute to the energy behaviour.

References

1. Baker E., 1987. Reducing bias and inefficiency in the selection algorithm. Proceedings of the 2nd International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 14–21.
2. Caldas L. G. and Norford L. K, 2002. A design optimization tool based on a genetic algorithm. Automation in Construction. 11, 173–184.
3. Caruana R. A., Schaffer J. D., 1988. Representation and hidden bias: Gray vs. binary coding for genetic algorithms. Proceedings of the Fifth International Workshop on Machine Learning, Eds. San Mateo, CA: Morgan Kaufmann, 153–161.
4. Chakraborty U., Deb K., Chakraborty M., 1996. Analysis of selection algorithms: A Markov chain approach. Evolutionary Computation. 4 (2), 132–167.
5. Chakraborty U., Janikow C.Z., 2003. An analysis of Gray versus binary encoding in genetic search. Information Science. 156, 253–269.
6. Chipperfield A., Fleming P., Pohlheim H., Fonseca C., 1995. Genetic Algorithm TOOLBOX For Use with MATLAB. User's guide, version 1.2. Retrieved from http://www.geatbx.com/ea_matlab.html
7. De Jong K., 2007. Parameter setting in Eas: a 30 year perspective, parameter setting in evolutionary algorithms. Studies in Computational Intelligence.54, 1–18.
8. Deb K., Agrawal S.,1998. Understanding interactions among genetic algorithm parameters, in Foundations of Genetic Algorithms V, Banzhaf W. and Reeves C., Eds. San Mateo, CA: Morgan Kauffman, 265–286.
9 Dumitrescu D., Lazzerini B., Jain L.G., Dumitrescu A., 2000. Evolutionary Computation, CRC Press LLC.
10. Goldberg D. E., 1989. Genetic algorithms in search, optimization, and machine learning. New York: Addison-Wesley.
11. Goldberg D. E., Deb K., Clark. J. H., 1992. Genetic algorithms, noise, and the sizing of populations. Complex Systems. 6, 333–362.
12. Hart W. E., Belew R. K., 1991. Optimizing an arbitrary function is hard for the genetic algorithm. Proceedings of the Fourth International Conference on Genetic Algorithms, 190–195.
13. Haupt R. L., Haupt S.E., 2004. Practical Genetic Algorithms. 2nd edition, John Wiley & Sons, Haboken, New Jersey.
14. Muhlenbein H., 1992. How genetic algorithms really work I: Mutation and hillclimbing. Foundations of Genetic Algorithms II, Eds. R. Ma:nner and B. Manderick, Amsterdam: North-Holland, 15-25.
15. Schaffer J. D., Caruana R. A., Eshelman L. J., Das R., 1989. A study of control parameters affecting online performance of genetic algorithms for function optimization. Proceedings of the Third International Conference on Genetic Algorithms. Eds. San Mateo, CA: Morgan Kaufmann, 51–60.
16. Wu A., Lindsay R. K., Riolo R. L., 1997. Empirical observation on the roles of crossover and mutation. Proceedings of the Seventh International Conference on Genetic Algorithms.362–369.
17. Znouda E., Ghrab-Morcos N., Hadj-Alouane A., 2007. Optimization of Mediterranean building design using genetic algorithms. Energy and Buildings. 39, 148–153.