Transactions on Ecology and the Environment vol 62, © 2003 WIT Press, www.witpress.com, ISSN 1743-3541

Energy management using genetic algorithms

F. Garzia, F. Fiamingo & G. M. Veca
Department of Electrical Engineering,
University of Rome "La Sapienza", Italy

Abstract

An energy management technique based on genetic algorithms is presented. This technique is able to evolve and adapt its behaviour according to the variation of the parameters of the controlled environment, ensuring a high flexibility and efficiency in energy management and a consequent energy waste reduction.

1 Introduction

The management and the control of the energy flows inside a building can be achieved using different system architectures [1,2]. Their differences lie on the features and performance and obviously on the overall cost.

Once chosen a hardware architecture [1,2] it is necessary to realize a proper software that implements and executes the desired energy management policy [3,4].

The choice of the energy policy needs to know in advance the exigencies of the final users together with their energy consumption time table, that is a certain number of data must be collected, in the most of the cases, for a long time.

This problem can be avoided using evolutionary strategies such as the one offered by the genetic algorithms [5,6].

Genetic algorithms (GAs) offer the great advantage of evolving their behaviour to match with the behaviour of the final users, using a mechanism that is very similar to the one used by nature.

The input data can be represented, for example, by the presence of people inside the room, the outside temperature, the inside temperature, the time, the date and other data that are useful to characterise the desired application and so on.

The output data are represented by the desired energy management strategies as a function of the input data installations that act directly on the electrical loads and the air conditioner.

Different genetic algorithm can be used to achieve the desired purpose, each characterised by peculiar features: as the number of inputs and their relations with output data varies, a genetic algorithm is more indicated with respect to the other algorithm. In fact every other management strategy would need a certain artificial intelligence, such as, for example, the one provided by neural networks, whose complexity and the consequent implementation duty grow with the number of input and output variables.

It is therefore possible to use very simple GAs to perform quite simple operations or it is possible to use advanced GAs to perform more complex operations.

Since the computation resources of the electronic module that controls the input data and the output devices are unavoidably limited, it is necessary to reduce as more as possible the number of input data and the complexity of the energy management strategy, that is find the most suitable genetic algorithm for this kind of application which has to perform, anyway, an advanced energy management program, that is the purpose of this paper.

2 The genetic algorithms

Genetic algorithms are considered wide range numerical optimisation methods, which use the natural processes of evolution and genetic recombination. Thanks to their versatility, they can be used in different application fields.

GAs are particularly useful when the goal is to find an approximate global minimum in a high-dimension, multi-modal function domain, in a near-optimal manner. Unlike the most optimisation methods, they can easily handle discontinuous and non-differentiable functions.

The algorithms encode each parameters of the problem to be optimised into a proper sequence (where the alphabet used is generally binary) called a gene, and combine the different genes to constitute a chromosome. A proper set of chromosomes, called population, undergoes the Darwinian processes of natural selection, mating and mutation, creating new generations, until it reaches the final optimal solution under the selective pressure of the desired fitness function.

GA optimisers, therefore, operate according to the following nine points:

1) encoding the solution parameters as genes;
2) creation of chromosomes as strings of genes;
3) initialisation of a starting population;
4) evaluation and assignment of fitness values to the individuals of the population;
5) reproduction by means of fitness-weighted selection of individuals belonging to the population;
6) recombination to produce recombined members;
7) mutation on the recombined members to produce the members of the next generation;
8) evaluation and assignment of fitness values to the individuals of the next generation;
9) convergence check.

The coding is a mapping from the parameter space to the chromosome space and it transforms the set of parameters, which is generally composed by real numbers, in a string characterized by a finite length. The parameters are coded into genes of the chromosome that allow the GA to evolve independently of the parameters themselves and therefore of the solution space.

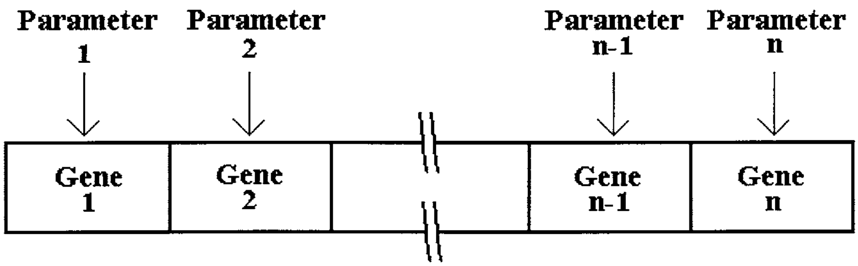

Figure 1: Encoding of the solution parameters as genes of a chromosome.

Once the chromosomes are created it is necessary to choose the number of them which composes the initial population. This number strongly influences the efficiency of the algorithm in finding the optimal solution: a high number provides a better sampling of the solution space but slows the convergence. A good compromise consists in choosing a number of chromosomes varying between 5 and 10 times the number of bits in a chromosome, even if in the most of situations, it is sufficient to use a population of 40-100 chromosomes and that does not depend of the length of the chromosome itself. The initial population can be chosen at random or it can be properly biased according to specific features of the considered problem.

Fitness function, or cost function, or object function provides a measure of the goodness of a given chromosome and therefore the goodness of an individual within a population. Since the fitness function acts on the parameters themselves, it is necessary to decode the genes composing a given chromosome to calculate the fitness function of a certain individual of the population.

The reproduction takes place utilising a proper selection strategy which uses the fitness function to choose a certain number of good candidates. The individuals are assigned a space of a roulette wheel that is proportional to they fitness: the higher the fitness, the larger is the space assigned on the wheel and the higher is the probability to be selected at every wheel tournament. The tournament process is repeated until a reproduced population of N individuals is formed.

The recombination process selects at random two individuals of the reproduced population, called parents, crossing them to generate two new individuals called children. The simplest technique is represented by the single-point crossover, where, if the crossover probability overcome a fixed threshold, a random location in the parent's chromosome is selected and the portion of the chromosome preceding the selected point is copied from parent A to child A, and from parent B to child B, while the portion of chromosome of parent A following

Transactions on Ecology and the Environment vol 62, © 2003 WIT Press, www.witpress.com, ISSN 1743-3541

the random selected point is placed in the corresponding positions in child B, and vice versa for the remaining portion of parent B chromosome.

If the crossover probability is below a fixed threshold, the whole chromosome of parent A is copied into child A, and the same happens for parent B and child B. The crossover is useful to rearrange genes to produce better combinations of them and therefore more fit individuals. The recombination process has shown to be very important and it has been found that it should be applied with a probability varying between 0.6 and 0.8 to obtain the best results.

The mutation is used to survey parts of the solution space that are not represented by the current population. If the mutation probability overcomes a fixed threshold, an element in the string composing the chromosome is chosen at random and it is changed from 1 to 0 or vice versa, depending of its initial value. To obtain good results, it has been shown that mutations must occur with a low probability varying between 0.01 and 0.1.

The converge check can use different criteria such as the absence of further improvements, the reaching of the desired goal or the reaching of a fixed maximum number of generations.

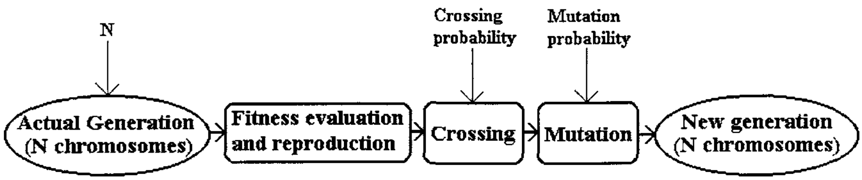

Figure 2: Operative scheme of GA iteration.

3 The genetic classifier

A classifier system is a machine learning system that learns syntactically simple string rules to guide its performance in an arbitrary environment. A classifier system consists of three main components:
1) rules and messages system
2) apportionment of credit system
3) genetic algorithm.

The rule and message system of a classifier system is a special kind of production system. A production system is a computational scheme that uses rules as its only algorithmic device. Although there is a wide variation in syntax between production systems, the rules are generally of the form if <condition> then <action>. The meaning of a production rule is that the action may be taken when the condition is satisfied. Even if this simple device for representing knowledge can seem to be too constraining, it has been shown that production system are computationally complete and also convenient, since a single rule or a small set of rules can represent a complex set of thoughts compactly. Classifier systems restrict a rule to a fixed-length representation. This restriction has two benefits: all strings under the permissible alphabet are syntactically meaningful and fixed string representation permits string operators of the genetic kind, letting possible a genetic algorithm search of permissible rules.

Classifier systems use parallel activation whereas traditional expert systems use serial rule activation. During each matching cycle, a traditional expert system activates a single rule. This rule-by-rule procedure is a bottleneck to increase productivity, and much of the difference between competing expert system architectures concerns the selection of the better single rule activation strategies for this or that type of problem. Classifier systems overcome this bottleneck, allowing parallel activation of rules during a given matching cycle. Thanks to this feature, classifier systems allow multiple activities to be coordinated simultaneously.

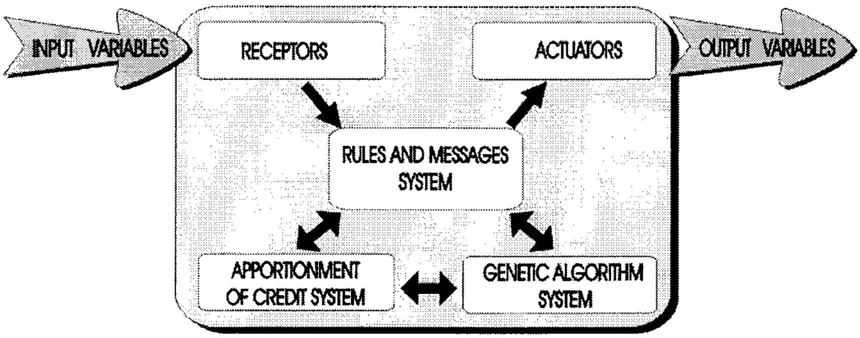

Figure 3: Scheme of a classifier system.

When choices must be made between mutually exclusive environmental actions or when the size of the matched rule set must be pruned to accommodate the fixed length message list, these choices are postponed to the last possible moment, and the arbitration is then performed competitively.

In traditional expert systems, the value or rating of the rule relative to the other rules is fixed by the programmer in conjunction with the expert group of experts being emulated. In a rule learning system, the relative value of different rules is one of the key pieces of information that must be learned. To facilitate this kind of learning, classifier system forces classifier to coexist in an information-based service economy. A competition is held between classifiers where the right to answer relevant messages goes to the highest bidders, with the subsequent payment of bids serving as a source of income to previously successful message senders. In this way a chain of rules is formed from the input of the system, represented by the detectors, to the output of the system, represented by the actuators. The competitive nature of the economy ensures that good rules, that are the more profitable, survive and bad rules, that are unprofitable, die off.

The apportionment of credit is very important in a classifier system. It uses a sort of internal currency that is exchanged and accumulated to provide a natural figure of merit. Using a classifier's bank balance as a fitness function, classifier may be reproduced, crossed, and mutated, according to the criteria illustrated in the previous paragraphs. Thus, not only can the system learn by ranking extant rules, but it can also discover new possibly better rules as innovative combinations of its old rules.

Apportionment of credit via competition and rule discovery using genetic algorithms constitute a reasonable basis for constructing a machine learning that is computationally convenient and efficient.

4 The used genetic system

Our used system is composed by an electronic microcontroller that controls one or more than one room. It is equipped with a certain number of input sensors and a certain number of output actuators, which manage the electric loads and the other energy sources.

The input can be represented, for example, by the presence of people inside a room, the outside temperature, the inside temperature, the time, the date and other data that are useful to characterise the desired application and so on. The output data are represented by the desired energy management strategies as a function of the input data that act directly on the electrical and the air conditioner installations.

An entry level system concentrates only on the presence information, switching on and off the electrical charges as a function of the occupation state of the controlled room, that is the system is capable of learning the occupation state and of switching in advance the electrical charges to let the people find a comfortable setting from the environmental point of view. It is also capable of learning when the room has been definitely left, so that it disconnects all the electrical charges. From this point of view the genetic system is capable of greatly reducing the energy consumption, ensuring an optimal comfort inside the controlled room.

The payoff information is represented from the efficiency in energy waste reduction: the higher this number the more the relative controlling rule is enforced, the lower this number the more the relative controlling rule tends to extinct.

The system must learn to predict when the room will be occupied basing on the previous occupancy state and on other parameters, that is to distinguish when it has momentarily left the controlled room from when it has definitely left the same room. This kind of system has already been studied [5], showing high potentialities in energy management.

To extend the potentiality of this kind of system we introduce a further development represented by the capability of controlling the air conditioning system from the user comfort point of view, that is to switch the air conditioner in advance or later with respect to a certain fixed time, as a function of the user occupation state of the room, of the desired inside temperature, of the outside temperature and of other environmental parameters. This potentiality allows the system not only to learn the room occupant behaviour, but also the room behaviour from the thermal point of view.

Transactions on Ecology and the Environment vol 62, © 2003 WIT Press, www.witpress.com, ISSN 1743-3541

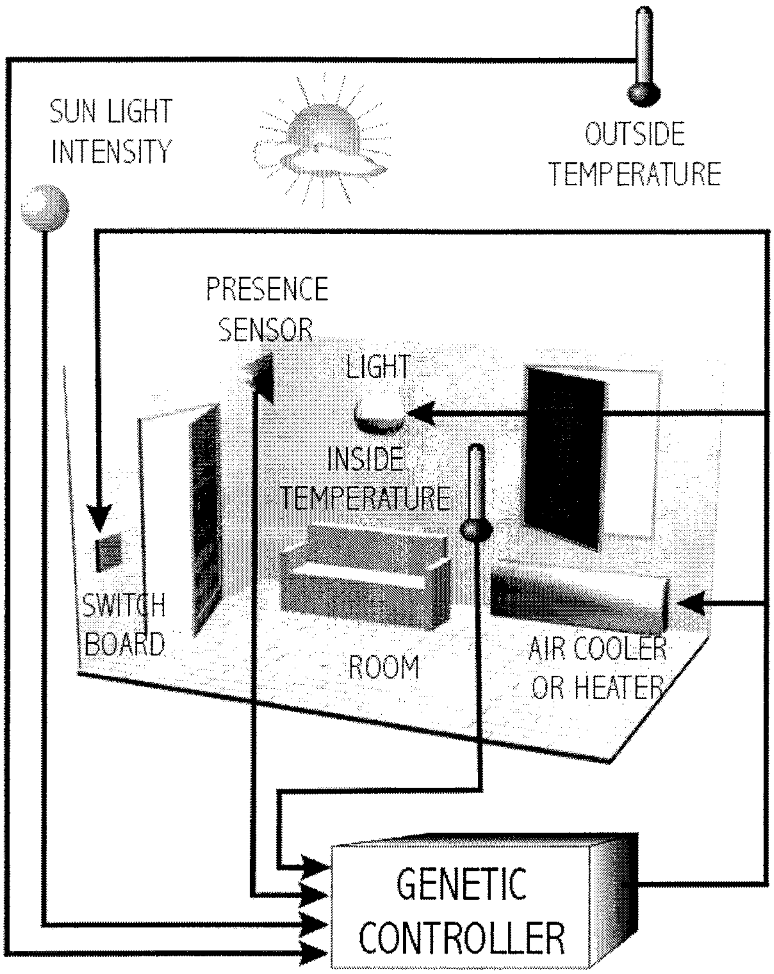

Figure 4: Scheme of the considered genetic system.

The considered input variables are the room occupation information, the time of the day, the day of the week, the day of the month, the month, the inside temperature, the outside temperature, and the outside light intensity. All the considered variables are thought to be essential in the determination of the occupation state of the room for the most of the use of the considered building (home, office, school, university, factory, museum, hospital, etc.).

The range of these variables together with their binary codification is shown in table 1.

The choice of these variables does not need further explications, with the exception of the room occupation information, that is composed by 144 binary

Transactions on Ecology and the Environment vol 62, © 2003 WIT Press, www.witpress.com, ISSN 1743-3541

samples corresponding to one sample every ten minutes for the previous 24 hours. This information is coded into the environmental message string as binary information, where the number 1 means that the room is occupied and the number 0 means that the room is unoccupied.

Considered variable	Variability range	Variable type	Number of bits
Room occupation information	0÷1 (*144)	Binary	144
Time (hour+minute)	0÷24 + 0÷59	Integer+Integer	5+6
Day of the week	1÷7	Integer	3
Day of the month	1÷31	Integer	5
Month	1÷12	Integer	4
Inside temperature	0 ÷+31	Integer	5
Outside temperature	-40 ÷+60	Integer	7
Outside light intensity	10^{-3}÷10^{6}	$M*10^{N}$ (M integer, N integer)	4+4

Table 1: Variables composing the message string and relative codification.

It is now necessary to define the message generated by the classifier to execute the switching on or off of the electrical loads and the air conditioners. Since the system must learn to predict at what time it is necessary to activate or deactivate the mentioned devices, the message has the following syntax:

message::=<time>:<load identification>:<load condition>,

where the time is coded in the same way of the time coded in the condition message, that is 5 bits for the hour information and 6 bits for the minute information, the load identification is used to act correctly on the desired device (a bit, that allows to represent two values, if it is necessary to operate on an electrical load and an air conditioner device) while the load condition is coded using only 1 bit, that is 1 when the load is switched on and 0 when the load is switched off.

Since the learning time of the net depends on the variability of the input data, that is similar input patterns need a low number of rules to be properly recognized while very different input patterns (owed, for example, to great weather variability that produces a great variability of the occupation state of the room) need a quite high number rules to be properly managed, it has been introduced a parameter called "daily presence variability" (DPV) that represents the variation degree between two subsequent days in term of room occupation. It consists, for all the 144 samples points used by the system for our example, in the calculation of the absolute value of the difference between the desired output $O_D(i)$ of the system on the actual day and the desired output $O_{D-1}(i)$ of the system on the previous day, both taken at the same sample time i:

$$\mathrm{DPV} = \frac{\sum_{i=1}^{144} \left| O_D(i) - O_{D-1}(i) \right|}{144}. \qquad (1)$$

From the given definition it is evident that if a considered day is characterized by a DPV equal to 1 it is totally different from the previous day (the system must switch on whereas in the previous day it had to switch off and vice versa) while if a considered day is characterized by a DPV equal to zero it is exactly equal to the previous day. The DPV parameter is very useful in characterizing the variability of the input data that strongly influences the learning time and the performance of the net.

The switching error, that is the error made by the system in switching on when it have to switch off and vice versa has already been studied [5] and it will not be reported here for brevity.

In this paper the authors are interested in studying the system from the thermal switching point of view that is to define a proper temperature error that is used to check the efficiency of the system in switching the air conditioner on exactly in time to let find a temperature that differs by ΔT from the desired temperature when the room occupants arrive. For this reason we consider a temperature error with margin ΔT defined as the number of times (related to 100 general events) that the temperature inside the room differs more than ΔT with respect to the desired temperature when the room occupants arrive.

5 Performance of the system and results

The performances of the proposed system have been studied in a simulated-real contest, obtaining interesting results.

It has been used, for the genetic system, optimal parameters equal to 0.7 for the crossing probability and 10^{-3} for the mutation probability.

The temperature error as a function of the number of training days for different values of ΔT, considering DPV constant and equal to 0.7, is shown in fig. 5, where it is possible to see that when the ΔT parameter decreases (higher toleration of temperature error), the number of training days decrease too.

The temperature error as a function of the number of training days for different values of DVP, considering ΔT constant and equal to 1, is shown in fig. 6, where it is possible to see that when the DVP parameter decreases (lesser daily variability of presence inside the controlled room), the number of training days decrease too.

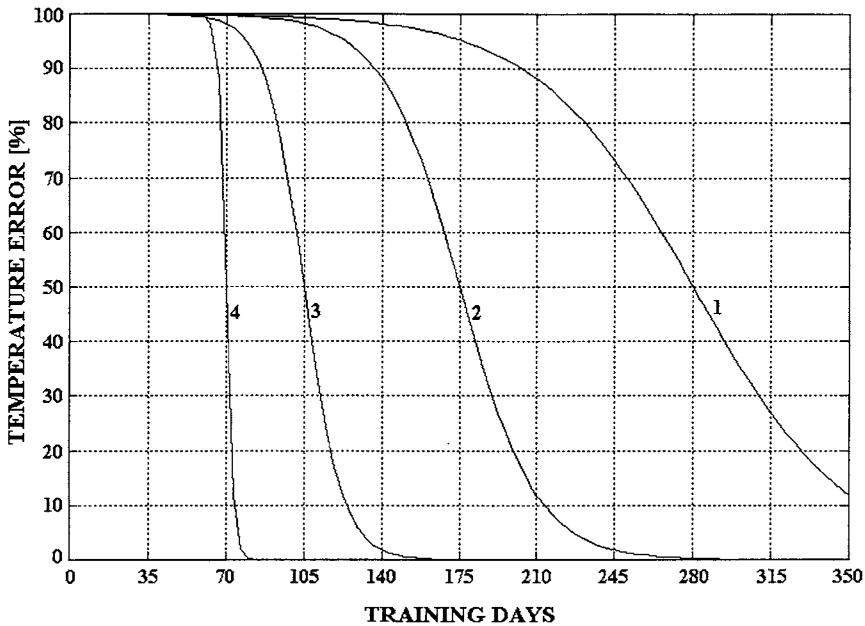

Figure 5: Temperature error as a function of the number of training days for different values of ΔT, for a constant value of DPV equal to 0.7.

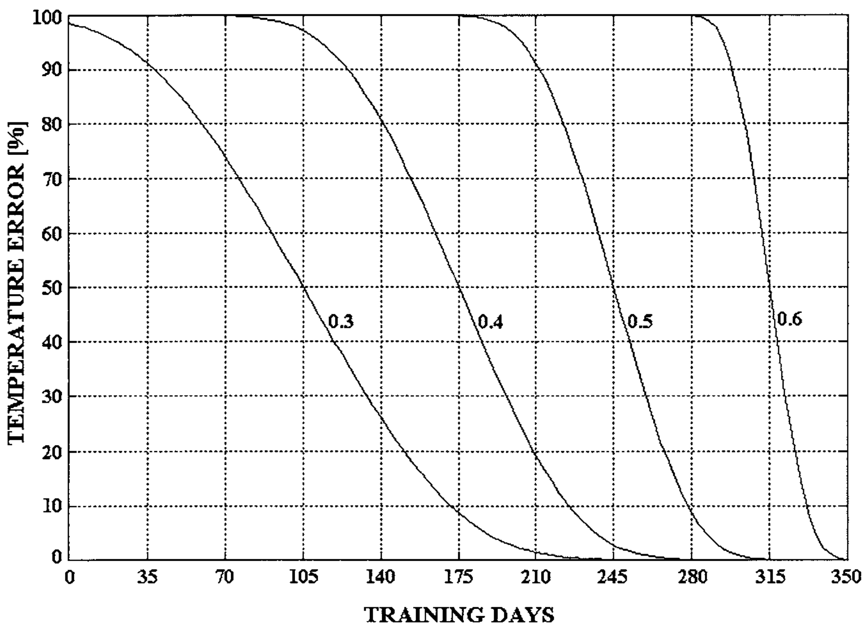

Figure 6: Temperature error as a function of the number of training days for different values of DVP, for a constant value of ΔT equal to 1 °C.

Transactions on Ecology and the Environment vol 62, © 2003 WIT Press, www.witpress.com, ISSN 1743-3541

6 Conclusion

An energy management technique based on genetic algorithms has been presented. The proposed system is able not only of evolving and adapting its behaviour according to the variation of the parameters of the controlled environment, but it is also able to ensure a correct comfort of room occupants, together with a high flexibility and efficiency in energy management, and a consequent energy waste reduction.

References

[1] F. Garzia, G.M. Veca, "Management and Control of Energy Flows in Building", TELESCON 97, Budapest (Hungary), pp.269–276, 1997.
[2] F. Garzia, G.M. Veca, "Smart Automatic Control of Energy Flows in Building", DUE 2000, Cape Town (South Africa), pp.267–272, 2000.
[3] F. Garzia, G.M. Veca, "Neural Techniques for Energy Management in Building", DUE 2001, Cape Town (South Africa), pp.291–297, 2001.
[4] F. Garzia, G.M. Veca, "Advanced Neural Techniques for Energy Management", DUE 2002, Cape Town (South Africa), pp.159–164, 2002.
[5] F. Garzia, G.M. Veca, "Evolutionary Computation and Genetic Algorithms for Energy Management and Conservation", Telescon 2002, Montreal (Canada), pp.386–393, 2002.
[6] D.E. Goldberg, "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley, 1989.