Design of an Adaptive Mutation Operator in an Electrical Load Management Case Study

Álvaro Gomes
DEEC – FCTUC / INESC-Coimbra
Polo II 3030-290 Coimbra
+351 239 79 62 80
agomes@deec.uc.pt

C. Henggeler Antunes
DEEC – FCTUC / INESC-Coimbra
Polo II 3030-290 Coimbra
+351 239 79 62 59
ch@deec.uc.pt

A. Gomes Martins
DEEC – FCTUC / INESC-Coimbra
Polo II 3030-290 Coimbra
+351 239 79 62 79
amartins@deec.uc.pt

ABSTRACT
An adequately designed and parameterized set of operators is crucial for an efficient behaviour of Genetic Algorithms (GAs). In GAs the mutation operator is commonly used with fixed mutation rates. However, in nature some genes mutate more often than others and mutation rates can be influenced by the environment. In this work a comparative analysis of the effects of using an adaptive mutation operator is presented in the operational framework of a multiobjective GA to design and select electrical load management strategies. It is shown that the use of a time/space varying mutation operator increases the efficiency and the efficacy of the algorithm.

Categories and Subject Descriptors
G.1.6 [**Optimization**]: *Global optimization.* J.2 [**Physical sciences and engineering**]: *Engineering*. I.6 [**Simulation and modelling**]: Applications.

General Terms
Algorithms, Experimentation, Theory.

Keywords
Adaptive control, genetic algorithms, multiobjective optimization.

1. INTRODUCTION
The identification and the selection of a suitable value or range of values for every parameter become an essential step when using meta-heuristics and particularly GAs. The choice of an adequately designed and parameterized set of operators is a crucial task for an efficient performance of GAs, namely in multiobjective (MO) problems, by making GAs better adapted to the current search space and allowing them to evolve according to the objectives being evaluated. Usually, two different approaches are used in the identification process of the parameter values. One, and perhaps the most common, consists in tuning the parameters through experimentation based on the analyst's expertise. Several runs are executed until the parameters are calibrated, and the process ends when the results produced are good enough according to the decision maker's (DM) preferences. This is an empirical process relying on the expertise of the actors involved (the DM and the analyst who mediates the communication of the DM with the computer tools). The second approach for parameter setting is through adaptive control. Instead of being constant over each simulation, like in the tuning process, the values of different parameters may vary with time [1], [2], [3]. This variation may appear in two forms. In the simplest one, the values are a function of time, generally of the number of generations. In the second form, the parameters are given the ability to evolve over generations (self-adaptive control), in which the parameter codification is made within the chromosomes of the individuals and thus are part of the evolution process. Therefore, if changes in the parameters are based on the results obtained in each generation rather than being a simple function of time (number of generations) this adaptive control can be a privileged manner of incorporating knowledge about the evolutionary process into the search, thus potentially contributing for the GA to work more effectively. This issue gains even more importance in a multiobjective setting, in which a prominent solution does not exist and the aim is to characterize a nondominated frontier and to select a nondominated solution or a set of solutions for further screening involving the DM's preference structure.

Mutation and crossover operators are used as a way for discovering new solutions, by contributing to focus the search into new regions of the search space. These operators give GAs the ability to go through the search space allowing, for example, to escape from local optima. In general, the conditions (environment) found by the individuals change over time (generations) and thus genetic operators should also evolve in order to play their role more effectively. In fact, there are some evolutionary algorithms's implementations in which genetic operators are adaptively controlled [1][2]. This behaviour is even more desirable when hard and large search spaces (particularly in combinatorial problems) must be explored.

In this paper, the influence of using adaptive control of the mutation operator in a GA for an MO problem is analyzed. The real-world problem herein reported deals with the identification and the selection of suitable control strategies to be applied to groups of electric loads, in the framework of load management in electric power systems. In the following section the case study is presented, while in section 3 the design of the mutation operator is described. The results obtained applying this approach are presented in section 4. In section 5, some conclusions are drawn about the merits of the proposed adaptive behaviour of the mutation operator.

2. THE SELECTION OF LOAD CONTROL STRATEGIES IN POWER SYSTEMS

Several activities encompassed by the so-called Load Management (LM) have been implemented by some electric utilities as a way of increasing the efficiency of the electric system or due to their potential economic attractiveness. In recent years the economic interests become the main goal [4][5] due to the volatility and spikes of wholesale electricity prices and reliability concerns. This kind of activities asks for the design and selection of suitable load shedding actions: on/off patterns to be applied over some groups of loads (usually air conditioners and electric water heaters). One control strategy encompasses the control actions (on/off pattern) to be applied to every group of loads. The minimum duration of each on or off time period may range from few minutes (for example, 5 minutes) to 1 or 2 hours. With shorter time periods the efficacy of the load management actions is improved. Traditionally, the identification of load control strategies is based on available knowledge from past experiences or from pilot programs carried out with the purpose of providing data for a suitable pre-evaluation study. Moreover, a cycling strategy is often used with constant on/off patterns. In such strategy a pre-determined on/off pattern is applied to the loads under control not taking into account eventual specific characteristics of each group of loads and eventually leading to a poor performance in some of the objectives pursuit. Usually, the demand of the loads is available on a daily basis and thus the power curtailment actions must be spread over several hours of the day.

The use of these activities in modern power systems structure asks for the capability of dealing with different objectives pursuit by diverse entities that appeared with the sector restructuring, such as regulators, distribution utilities, marketer, etc. Therefore, some of the objectives that a DM may consider when dealing with this kind of strategies are: minimize maximum power demand, maximize profits, minimize losses and minimize discomfort caused to customers. In this work the objectives explicitly considered in the mathematical model are: minimisation of peak power demand at three different demand aggregation levels (PA; PD1; PD2); profits maximisation (p); loss factor minimisation (l); minimisation of maximum time interval in which a state variable (usually associated with temperature) is over or under a pre-specified threshold (I); minimisation of the total time in which a state variable is over or under a pre-specified threshold (m). By evaluating the effects of LM programs at different demand aggregation levels and considering the multiple objectives referred above, it is possible to use this model in distinct market structures. Further details about the MO mathematical model can be found in [6].

Due to the potential size of the search space and also due to the number of evaluation dimensions, the selection of adequate load shedding actions to be implemented over sets of loads is a hard combinatorial problem to be faced by LM programs.

The identification of load curtailment actions to be applied to the different groups of loads under control is an iterative process encompassing the simulation of load demand and the run of the GA. The GA identifies a set of potential solutions (individuals in the population), which are then decoded into load control strategies (one strategy for each individual). These strategies are applied over the loads under control and their demand in the presence of the control strategies is simulated. The results of simulation, using a specific set of strategies corresponding to the actual population in the GA, are then used to evaluate every individual in the population. The GA proceeds with its normal steps: selection, crossover and mutation until a new population (generation) is obtained. The cycle repeats until the stop condition is verified. In figure 1 this iterative process is schematically described.

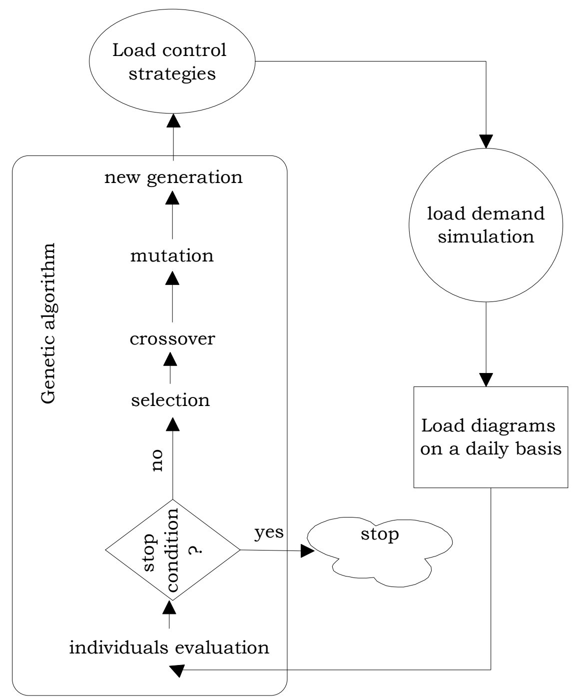

Figure 1 - Iterative process for the identification of load control strategies.

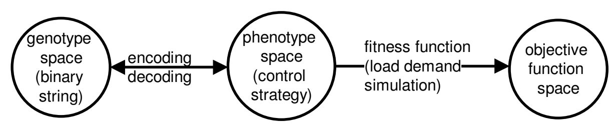

Figure 2 – Relationships between the different spaces.

The phenotype space is the set of all possible load-shedding strategies that can be used, which are mapped into the objective space through the simulation of load demand (Figure 2). This is a seven-dimension space whose structure depends on the control strategies applicable to loads. The codification of each control strategy in the genotype space is done through the use of binary matrices each one representing a control strategy applicable to all groups of loads. Each column in the matrix represents the control strategies applied to a group of loads. For *ng* groups of loads a control strategy is set up with *ng* columns in the matrix. Therefore, each individual in the population is a matrix whose size is 1440**ng* (1440 rows and *ng* columns). 1440 is the number of minutes in a day, in accordance with the time resolution used in the models that reproduce the demand of loads, since these models simulate the demand of groups of loads on a daily basis with a time resolution of one minute, for a more detailed representation of load behaviour [6]. In these matrices a "1" means that the load is shut-off during the corresponding period of time and a "0" means that the load is operating as usual, i.e. without any external control action.

The minimum period of time in which loads are cut-off is 5 minutes long and each period of normal operation lasts at least for 5 minutes. When crossover or mutation occurs, chromosomes are checked and eventually corrected in order to maintain these two minimum periods of time in which a load is shut-off or powered on. This time resolution requires some computational effort but the corresponding load switching maximum frequency (1/10 minutes) is not too high for loads of this size and type (thermostatic loads). Moreover, it also allows a more effective dispatch of the load under control to be achieved.

One individual is the codification of the on/off patterns applied over all the groups of loads under control, and the on/off patterns to be applied over a specific group is codified by a sequence of genes in the chromosome (Figure 3). Thus, the chromosome becomes the concatenation of the on/off patterns applied over all groups of loads.

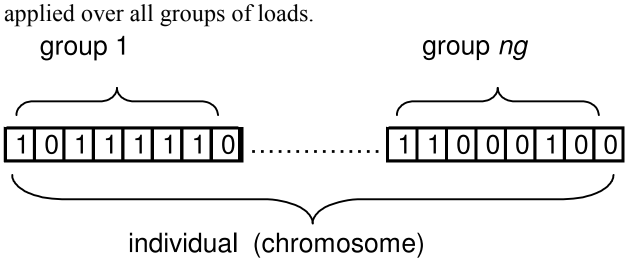

Figure 3 – One individual in the population.

3. DESIGN OF THE MUTATION OPERATOR

The performance of an algorithm depends on a good combination of all parameters. However, the mutation operator is sometimes managed as a secondary or background operator [7] in GAs and commonly used with fixed mutation rates [2][8]. Despite the apparent secondary role of the mutation operator, mainly due to the small values of the mutation probability, meaning that the individual produced by mutation is not very different from his ancestor (in the genotype space), this is a very important operator, allowing for the introduction of new genetic material in the population. An adequate mutation rate is essential for a good performance of the GAs, particularly in complex multiobjective combinatorial problems. Whenever too high rates are used, GAs become almost a random search throughout the search space, while with too low mutation rates GAs generally present very small rates of progress towards the Pareto-front leading to very time consuming and ineffective procedures. On the other hand, using fixed mutation rates is not in agreement with what occurs in nature, in which there are some genes that mutate more often than others and some mutations occur more frequently than others. Moreover, mutation rates may be influenced by the environment. Several studies show the importance of this operator [9][10].

In presence of both time varying functions and/or hard large search spaces, the use of fixed value operators may lead to poor performances when compared with implementations using adaptive operators, since a given value for the mutation probability may not perform well at different stages of the evolutionary process. In general, a given value for the mutation probability may be adequate in the initial phase of the search but may become very ineffective when the population is near the Pareto front. That is, the behaviour of the population during the evolutionary process must not be neglected and any available information about the optimisation process must be provided to the GA to improve its performance.

The idea proposed in this paper consists in allowing the mutation operator to be influenced by the environment (search space) and assess the importance of this behaviour in the GA's performance in the real-world case study described in section 2.

In the problem under study the binary alphabet is used which means that there are two different possible mutations that can occur in each gene: "0" mutates to "1" (*pm_0_1*) and "1" mutates to "0" (*pm_1_0*). In order to better adapt the mutation operator to the environment the two mutations that can occur in each gene may present different values. Also the probability of mutation in each gene can change, which is similar to what happens in nature. This means that:

- each gene can present a different value for the mutation probability when compared with others genes in the chromosome;
- each gene can suffer two different mutations, and each mutation can occur with different probabilities.

With the purpose of contributing to increase the efficiency of the GA, each of the individual values of the mutation probability is built up with contributions based on the performance of each individual in each objective. That is, in this methodological approach the mutation rates are modulated by the performance of the individuals, making the mutation probability to change accordingly. Thus, the probabilities of the two possible mutations can vary in different ways during the simulation, in the sense that when changes occur it may be interesting to increase the mutation rate in one direction but not in the other. For example, in a scenario in which simulation is going on in time interval *n* and in interval *n+1* profits increase, from the profit maximization perspective *pm_1_0* may increase and *pm_0_1* may decrease in interval *n+1* with respect to their values in interval *n*. If the maximum aggregate power demand decreases in interval *n+1* when compared with demand in interval *n* then *pm_1_0* may increase and *pm_0_1* may decrease with respect to their values in interval *n*. In this way the mutation rates for different individuals are different and change according to their performance in each objective function. These are simplified examples, in the sense that each time interval must not be taken separately, as the effects of changing power demand in one interval last for several subsequent intervals, depending on the amount of power controlled and other parameters that influence load operation. Thus, the mutation operator becomes a function of time within each simulation and also changes from one simulation to another and from one individual to another. The changes over time occurring in the individual's mutation probability are based on the information collected from the results obtained for each objective in the phenotype space by every individual in the population.

In order to increase the flexibility of this approach, the DM is given the capability to set weights for each objective, here perceived as degrees of importance, which are taken into account to compute the contribution of all objectives to the mutation probability.

The influence of the environment in the mutation operator is operationalized in the following way.

The objectives associated with the minimisation of peak power demand (objective functions i=1, 2, 3, for the aggregate level PA and the less aggregate levels PD1 and PD2, respectively) present a time varying behaviour (within each simulation and from one run to another). They have a contribution that is also time dependent, in a way closely related to the variation in time of the objective, and limited by a maximum threshold specified by the analyst: *pm_up*. The way each contribution varies in time is a function of the normalized difference between the current value of the objective function i ($D^i[n]$) and a percentage of its maximum value (MD^i) (90% is the current value used in the experiments).

$$pm^i\_0\_1[n] = \frac{\max\left(0; D^i[n] - 0.9 * MD^i\right)}{\max\limits_n\left(D^i[n] - 0.9 * MD^i\right)} \times pm\_up$$

i =1, 2, 3 (1)

$$pm^i\_1\_0[n] = \left[1 - \frac{\max\left(0; D^i[n] - 0.9 * MD^i\right)}{\max\limits_n\left(D^i[n] - 0.9 * MD^i\right)}\right] \times pm\_up$$

i= 1, 2, 3 (2)

In (1), the numerator gives the maximum difference between the current demand and the referred level (90%) of the maximum demand (in each level of load aggregation), at time interval n. The denominator is the maximum value of that difference when all time intervals are accounted for. $D^i[n]$ is the power demand at interval n and MD^i is the original maximum peak demand. $pm^i\_0\_1$ ($pm^i\_1\_0$) is the contribution of objective i to *pm_0_1* (*pm_1_0*) mutation probability. When the difference is negative (the actual demand is below the threshold level) the value 0 is considered for $pm^i\_0\_1$, meaning that, from this dimension's point of view, there is no interest in a mutation 0 to 1. For $pm^i\_1\_0$, the value *pm_up* is considered. That is, when power demand is low there is no interest in having load curtailments. When the current demand is over the threshold value, higher differences mean bigger mutation probability from the "no-load curtailment" to the "load curtailment" situation (from 0 to 1 in the genotype space). That is, the probability of occurring a load shedding rises when power demand increases. When the difference reaches its highest value, $pm^i\_0\_1$ is given the value pm_up and $pm^i\_1\_0$ is given the value 0. The contribution of this dimension (PA) for the mutation operator, within one generation and for one of the individuals, is depicted in figure 4. The load demand diagram is also presented and it can be seen that mutation from 0 to 1 increases substantially during higher demand periods.

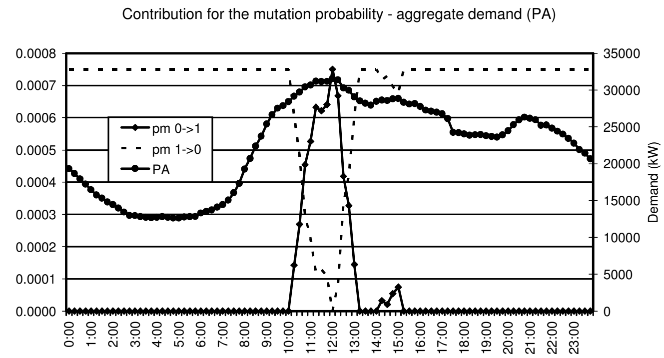

Figure 4 – The contribution of objective "minimize maximum value of aggregate demand" for the mutation probability.

Figures 5 and 6 show the contribution of the objectives related with power demand at the less aggregate levels (PD1 and PD2) for the mutation operator. The *rationale* is the same used for the more aggregate power level: to increase the probability of occurring load curtailments when power demand is above the threshold level and simultaneously to increase the probability of not occurring load control actions when the power demand is below the threshold level.

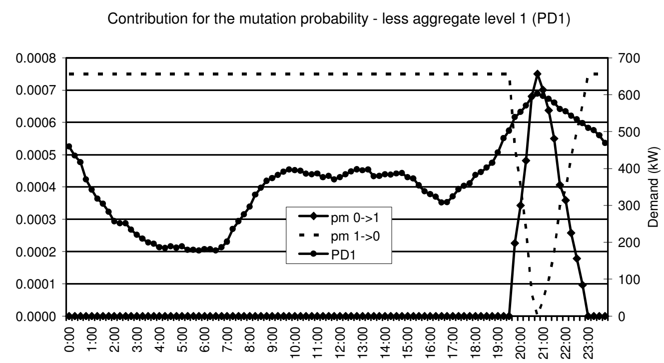

Figure 5 – Contribution of power demand at less aggregate level 1 (PD1) for the mutation probability.

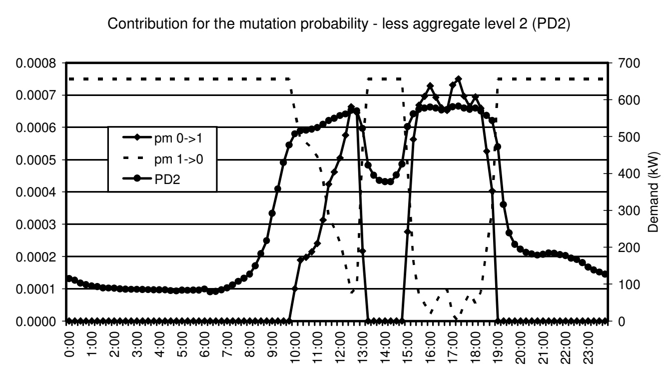

Figure 6 – Contribution of power demand at less aggregate level 2 (PD2) for the mutation probability.

The aspects related with consumers' comfort (objective functions 4 and 5, denoted by *I* and *m* in section 2) participate in the construction of the mutation operator with fixed values during the simulation. Their contribution is based on the *rationale* that to minimize the consumers' discomfort the number of load curtailments should be reduced. The analyst is asked to specify the maximum and minimum values desired for the mutation probability *(pm_max and pm_min)*. The values of the individual contribution of each objective ($pm^i\_0\_1$ and $pm^i\_1\_0$) are computed by:

$$pm^{i}\_0\_1 = pm\_min \quad , \qquad i= 4, 5$$

$$pm^{i}\_1\_0 = pm\_max \;, \qquad i= 4, 5$$

The objective function 6 (profits to be maximised) is in an intermediate situation. That is, the information chnages within each simulation but it is constant from one simulation to another one since 24 hours price forecasts are considered and its contribution for the *pm* values is based on those forecasts only. Profits are in the range [*min_profit*, *max_profit*] according to the forecasts and, in each time interval *n*, the contribution for *pm* is given by

$$pm^{i}\_0\_1[n] = \frac{Max\_profit - Profits[n]}{Max\_profit - Min\_profit} \times pm\_max \qquad i= 6$$

$$pm^{i}\_1\_0[n] = \left[1 - \frac{Max\_profit - Profits[n]}{Max\_profit - Min\_profit}\right] \times pm\_max \qquad i= 6$$

where *Profits[n]* is the value for profits in interval n, based on daily forecasts.

That is, in a way similar to the contribution of dimensions related with power demand, *pm* varies according to the normalized difference of current profits to their maximum value. When profits rise, contribution of this objective function to *pm_0_1* approaches 0 and their contribution to *pm_1_0* approaches *pm_max*. When profits reduce, the contribution for *pm_0_1* (*pm_1_0*) approaches *pm_max* (0). The *rationale* is to stimulate the occurrence of "operating as usual" rather than load curtailments in periods of time in which profits are higher. This behaviour is displayed in figure 7.

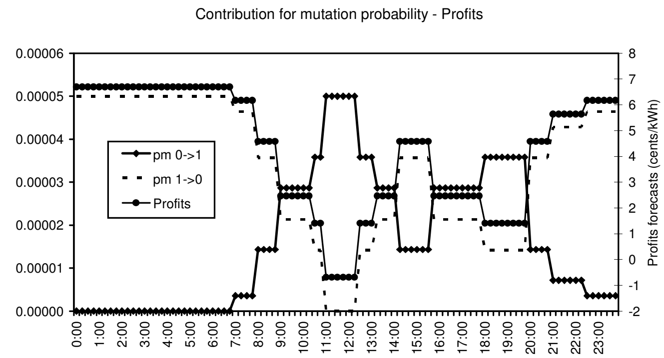

Figure 7 – Contribution of profits to mutation probability.

The loss factor (L, objective function 7) changes only from one simulation to another, and its contribution is computed as follows. Let ML, AL, and mL be the maximum, average and minimum values of the loss factor, respectively.

If L > AL then

$$pm^{i}\_0\_1[n] = \frac{L - AL}{ML - AL} \times pm\_max \quad i=7$$

$$pm^{i}\_1\_0[n] = \frac{ML - L}{ML - AL} \times pm\_max \quad i=7$$

else

$$pm^{i}\_0\_1[n] = \frac{AL - L}{AL - mL} \times pm\_max \quad i=7$$

$$pm^{i}\_1\_0[n] = \frac{L - mL}{AL - mL} \times pm\_max \quad i=7$$

The contribution of the loss factor to the mutation probability is displayed in figure 8.

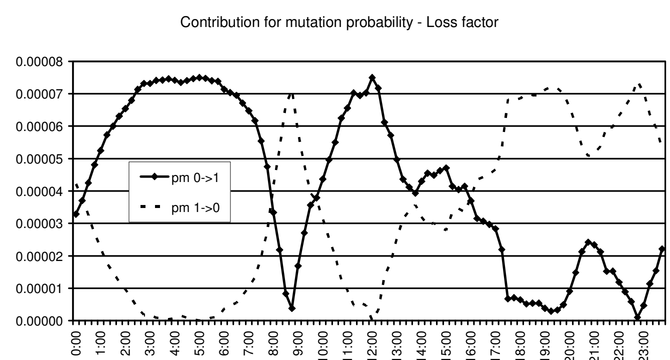

Figure 8 – Contribution of loss factor to the mutation probability

Figure 9 shows an example of the total mutation probability, taking into consideration the contributions of all objective functions (using equal weights ω_i, i=1,…,7), for a given group of loads. The aggregation of all contributions is $pm\_0\_1 = \sum_{i=1}^{7} pm^i\ 0\_1 . \omega_i$, and

$$pm\_1\_0 = \sum_{i=1}^{7} pm^i\ 1\_0 . \omega_i .$$

Figure 10 shows the mutation rates for an individual when there are three groups of loads under control. "ddc se" denotes the load diagram at the most aggregate level.

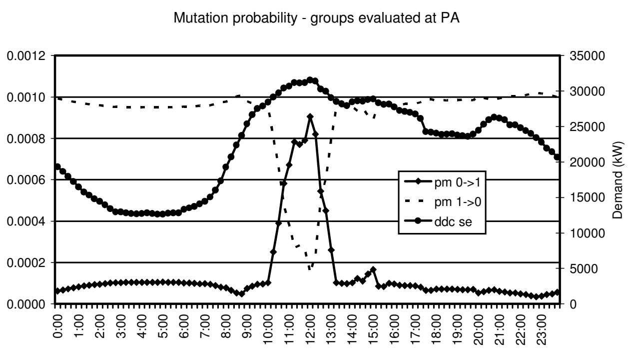

Figure 9 – Final values for mutation probability, for a given group of loads.

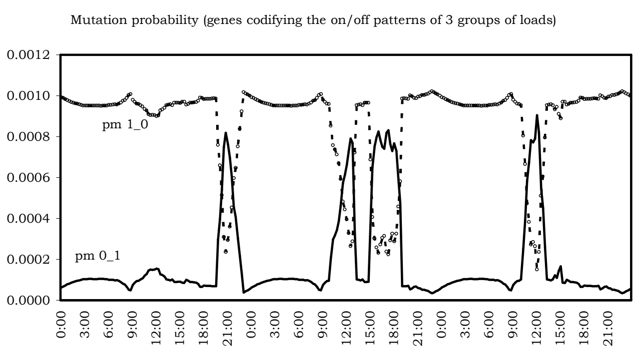

Figure 10 – Mutation rates for each gene in a chromosome (*ng* = 3).

Having in mind that mutation is an essential step in the evolution process, bringing new potential solutions to the search, there is a minimum value for the probabilities associated with it that is pre-specified by the DM/analyst *(pm_min)*. Thus, if the contribution of all objectives for the mutation operator is below that threshold level, then the value *pm_min* is assigned to *pm_0_1* or *pm_1_0*.

It must be emphasized that, despite the contribution for the mutation operator is built up based on differences between the results on the several objective functions and some pre-defined values of the objective functions, the mutation operator remains probabilistic and blind. There are three main reasons supporting this statement. The first one is that mutation still occurs with a given probability in each gene. The second reason is that despite the differences in mutation probability from one gene to another gene, this doesn't mean that if mutation in one gene occurs it contributes always for the improvement of the individual (sometimes the performance of the individual is better and sometimes it is worse). The third reason is that the loads under control are thermostatic ones, meaning that if the working cycle is changed only for a few minutes their demand is also changed in the following minutes. Thus, a mutation occurring in one gene may influence positively or negatively the performance of the individual.

4. CASE STUDY

In order to evaluate the performance of the GA with adaptive control of mutation operator, the results of the algorithm using fixed mutation rates and the results using variable mutation rates have been compared.

Some characteristics of the GA used in the simulations are:

- Population size: 20
- Maximum number of generations: 5000
- Crossover probability: 0.3
- One run (load demand simulation + one GA generation): 7.9 seconds (Pentium III, 1 GHz)

A key issue in MO optimisation is how to compare two different Pareto fronts resulting from two different algorithms or from the same algorithm with two different sets of parameters. Some criteria can be used, such as the number of non-dominated solutions, the distance from the Pareto front resulting from one run to the Pareto optimal front, the diversity and the spread of solutions throughout the Pareto front [13]. In the problem under study in this work, as in many real-world problems, the Pareto optimal front is unknown and the diversity and the spreading of solutions are not to be taken for granted. Therefore, the assessment of the adaptive control of mutation operator herein proposed is carried out by using the *surface attainment* concept [11][12]. [12] uses a pair of values *(a, b)*, in which *a* represents the percentage of the space (lines) in which the first algorithm performs better that the second algorithm and *b* represents the percentage of the space (lines) in which the second algorithm performs better than the first one. If the set of lines cover the whole front then the *surface attainment* method allows us to deal with the three issues raised in the performance assessment: distance to the optimal Pareto-front, distribution and diversity of the solutions. [12] further extended this analysis to more than two algorithms through the use of two "statistics", *unbeaten* and *beat all* by doing pairwise comparisons, as it is done for two algorithms, but now using all the points resulting from all the algorithms being compared for the "construction" of the Pareto front. *Unbeaten* is the percentage of the Pareto front where the algorithm is not beaten by any other algorithm, and *beats all* is the percentage of the front where the algorithm beats all other algorithms.

The results of using variable mutation rates have been compared with the results obtained with four different fixed values for mutation rates: 0.0001; 0.0005; 0.001 and 0.005. The Mann-Whitney non-parametric test at 95% confidence level and 15 runs for each situation have been used in the framework of the *surface attainment* method. Table 1 shows the percentage of space in which algorithms with fixed mutation rates perform better than the algorithm with variable mutation rates and the percentage of space in which the algorithm with variable mutation rates performs better. It was found that the latter implementation performs better than the implementation with fixed value (0.0001) in about 92% of the space and is unbeaten in all the space. When the fixed mutation rates are equal to 0.0005 or 0.001, the implementation with variable mutation rates performs better in about 68% of the space and is unbeaten in about 94% of the space. When compared with the values obtained with 0.005 for the mutation rate, it was found that using variable mutation rate performs better in more than 76% of the space and is unbeaten in more than 95% of then space.

Table 1 – Comparison of the two implementations.

Implementation	5 000 lines	10 000 lines
Fixed mutation rates (0.0001)	0%	0%
Variable mutation rates	91.6%	91.9%
Fixed mutation rates (0.0005)	14.6%	14.8%
Variable mutation rates	67.6%	67.9%
Fixed mutation rates (0.001)	6.6%	6.63%
Variable mutation rates	67.6%	67.4%
Fixed mutation rates (0.005)	4.6%	4.38%
Variable mutation rates	76.2%	76.2%

The populations resulting from each run of the algorithms have been grouped in pairs (one population from one run in which the mutation rate is fixed and one population resulting from using variable mutation rates – 4*15*15 pairs) and a Pareto-front was built with the non-dominated solutions of each pair. In average, in such Pareto fronts about 90% of solutions belong to the runs with variable mutation rates and 10% come from the populations in which fixed mutation rates have been used.

5. CONCLUSIONS

In this paper, it has been shown in the framework of a case study that allowing the individuals to be influenced by the environment is an important step for a more effective behaviour of the algorithm. In this study, the mutation operator presents an adaptive dynamic behaviour, changing according to the values achieved for the multiple objectives under analysis. The mutation operator has several characteristics that make the EA to work more effectively. The mutation probability depends on the mutation which is occurring: from load curtailment to no load curtailment or from no load curtailment to load curtailment. This operator has been split up into several parts, one for each dimension, making possible to consider distinct contributions from each objective for the global values of the operator. Finally, as it is possible to evaluate the effects of each LM action through the simulation of load demand, in each dimension in the phenotype space, this operator changes in time and from one solution to another solution in the population. According to the results obtained, this methodology implemented to make use of available information through an adaptive mutation probability results in an GA presenting a better performance and simultaneously the diversity of the population in each generation has increased substantially.

6. REFERENCES

[1] Michalewicz, Z., D. Fogel (2000), How to Solve it: Modern Heuristics, Springer, Berlin, 2000.

[2] A. E. Eiben, R. Hinterding and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms", IEEE Trans. on Evolutionary Computation, 3 (2), 124-141, 1999.

[3] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons Ltd, 2001.

[4] [4] G. C. Heffner and C. A. Goldman, "Demand Responsive Programs - An Emerging Resource for Competitive Electricity Markets?", Proceedings of the International Energy Program Evaluation Conference (2001 IEPEC), 2001.

[5] E. Hirst and B. Kirby, "Retail-Load Participation in Competitive Wholesale Electricity Markets", Edison Electric Institute, 2001.

[6] Gomes, A., C. H. Antunes, A. Martins (2004a), "A multiple objective evolutionary approach for the design and selection of load control strategies", IEEE Transactions on Power Systems,19 (2), 1173-1180, May 2004.

[7] D. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning; Addison-Wesley; 1989.

[8] Laumanns, M., G. Rudolph, H.-P. Schwefel, "Mutation Control and Convergence in Evolutionary Multi-Objective Optimization", In R. Matousek and P. Osmera (Eds.), Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Brno University of Technology, June 2001.

[9] Back, T., U. Hammel, H.-P. Schwefel (1997), "Evolutionary Computation: Comments on the History and Current State", IEEE Transactions on Evolutionary Computation, 1 (1), 3-17, April 1997.

[10] Fonseca, C., P. Fleming (1998b), "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part II: Application Example", IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 28 (1), 38-47, January 1998.

[11] Fonseca, C., P. Fleming (1996), "On the performance assessment and comparison of stochastic multiobjective optimizers", In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel (Eds), Fourth International Conference on Parallel Problem Solving from Nature (PPSN-IV), Berlin, Germany, 584-593, 1996. Springer.

[12] Knowles, J., D. Corne (2000), "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy", Evolutionary Computation Journal, 8, 2, 149-172, 2000.

[13] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE Trans. on Evolutionary Computation, 3 (4), 257-271, 1999.