ISAHP 2007, Viña Del Mar, Chile, August - 8-3, 2007.

USAGE OF GENETIC ALGORITHMS TO DEAL WITH INCONSISTENCY PROBLEM IN ANALYTIC HIERARCHY PROCESS

José Fabiano da Serra Costa
Universidade do Estado do Rio de Janeiro - UERJ
Rua São Francisco Xavier 524 – 6028B – Rio de Janeiro – Brasil
fabiano@ime.uerj.br

Keywords: AHP, Inconsistent Matrices, Genetic Algorithms

Summary: *This work presents a proposition to solve the problem of inconsistency in Analytic Hierarchy Process (AHP) matrices using genetic algorithms. Decision matrices resulting from an application of AHP can be considered an effective method to structure and represent relevant information of a strategic problem. Inconsistency in the results is a real and frequent possibility. In this case, the results obtained would become ineffective considering the objectives of the model, which means no gains in decision making. The Genetic Algorithms are probabilistic search computer models which are based on the mechanics of natural selection and genetics, combining the concepts of selective adaptation and survival of the fittest. They are considered to be a powerful technique of stochastic optimization and, probably the most important evolutionary computer techniques. Its application to the AHP matrices case allows the detection of inconsistent matrices, while offers alternative solutions to the decision-maker.*

1. Introduction

Whenever decision making is mentioned, the idea of something imprecise or undefined comes to mind. The human being has been confronting this questioning in every moment of his existence, whether it means an individual or a collective decision. Anyway, the essence of decision problems may be synthesized into the fact that expected results cannot be obtained unless one takes an action.

Considering decision making as an inevitable process, the need of creating methods of making this task easier to accomplish becomes necessary. In this context, due to economical and social environment complexity and to the vertiginous fall of the technological innovation, it becomes clear that the progress depends more and more on adopting planning and management innovative procedures. An answer to those needs is the decision making methods.

These methods are tools which involve modern techniques of informational systems, artificial intelligence, statistical quantitative methods, cognitive and behavioral psychology, among others, and they aim to offer the users favorable conditions to choose, minimizing the chances of mistake in decision making. Among the decision support methods, we point out the Multicriteria Decision Support Methodologies (Roy, 1985), with their schools.

One of the most known and used multicriteria methods, based on Saaty's (1980) work, the Analytic Hierarchy Process – AHP is easily and successfully applied in specific cases of decision problems. AHP method is based on comparison matrices resulted from specialists' opinion when comparing criteria (alternatives), in pairs.

However, it may occur that the final matrix is inconsistent, even if it represents the specialists' opinion. The matrix consistency refers to necessary matrix properties such as transitivity and reciprocity. The inconsistency may occur for several reasons (Saaty, 1991). In such cases, the results obtained of each specialist would be ineffective to the model objectives.

The aim of this paper is to use stochastic research methods known as Genetic Algorithms to create alternative solutions. Considering the little consistency of a specialist matrix, our goal becomes the attempt to create n solutions (similar matrices), through an analysis of disturbs of the original matrix values.

2. Analytic Hierarchy Process (AHP)

As previously mentioned, the Analytic Hierarchy Process (AHP) developed by Saaty (1980) is probably the most used multicriteria methodology with several applications in different activities (Saaty, 1991). This method aims to select and choose alternatives, through a process that considers different evaluation criteria, and is based in three principles of analytical thinking: hierarchy construction, priority definition, logical consistency.

Hierarchy construction: problems, in AHP, are structured in hierarchic levels in order to search better comprehension and evaluation. Hierarchy construction is an elementary part of human rationalizing process. During this activity, people identify key elements to decision making, arranging them in affinity groups, which are organized in layers.

Priority definition: the adjustment of AHP priorities is based on people's ability of acknowledging the relationship between observed situations and objects, comparing them in pairs in light of an specific focus or criteria (pairwise judgments).

Logical consistency: in the method, it is possible to evaluate the resulting prioritization model according its consistency (Costa, 2006).

In the usage of AHP, it is necessary to define a global objective or main focus, identify the available alternatives (or choice possibilities) and select criteria and subcriteria (when necessary) to achieve the proposed objective. All theses elements must be hierarchically structured and the criteria selection must be complete, minimal and operational (Chankong and Haimes, 1983).

Considering the attributes E_1, E_2,...,E_n, which contribute to reach a determined objective, the methodology proposes to compare the relative importance of each pair of attributes related to a superior hierarchic level. So, specialists can contribute with an individual Judgment Matrix, like this $A = [a_{ij}]_{nxn}$, where a_{ij} represents the relative importance of the attribute E_i comparing to the attribute E_j, considering that $a_{ij} > 1$, if and if only E_i is more important than E_j, and that $a_{ij} = 1/ a_{ji}$ for each pair (i,j). Saaty (1991) proposed his own scale, named Fundamental Scale, to compare pairs of attributes.

After the application of the method, once we have all the attributes and alternatives hierarchically distributed, what interests us is that the basic properties of the symmetrical matrix – reciprocity and transitivity – are sustained, as well as the consistency of the final result , and having (n – 1) comparisons we are able to deduct the others.

3. Matrix Inconsistency

Although pairwise judgments are based in the professionals' experience and knowledge, inconsistencies may happen, especially when dealing with a great number of judgments. Even with the most advanced instruments, it is difficult to obtain consistency in practice, which makes it necessary to have a method capable to evaluate the importance of this precision in a specific problem. In our case, what is called inconsistency is a violation of proportionality, which can sometimes mean the violation of transitivity. According to Saaty (1991), the consistency of a reciprocal and positive matrix is equivalent that its maximal eigenvalue (Grabyll, 1983) equals the number of attributes involved in the model. This means that the closest the maximal eigenvalue ($\lambda_{máx}$) reaches the number of attributes of the model (n), the better will be the coherence of the judgments ($\lambda_{máx} = n$).

It is also possible to estimate a consistency deviation from the results of the division of ($\lambda_{máx}$ - n) by (n - 1). Some authors (Dias *et al*, 1996) prefer to use the module of the difference between the number of attributes involved in the model and the maximal eigenvalue, $|n - \lambda_{máx}|$. Anyway, according to Saaty (1991) the measure of the poor consistency will be possible to estimate when compared to the value ($\lambda_{máx}$ - n) / (n - 1), denominated Consistency Index (CI), with values picked from random judgments and its correspondents in reverse positions of a matrix of the same size, with reciprocal matrices randomly generated (Saaty, 1991). This measure is referred to as Consistency Reason, so CR = CI / RI, where IR is the Random (Consistency) Index of a reciprocal and positive matrix randomly generated and must vary according to each matrix order. The closest this reason gets to zero, the more consistent the matrix will be. If the reason equals less than 0,10, the consistency is considered satisfactory. A mathematical justification for considering this value satisfactory can be found in Vargas (1982).

However, it may occur that the final matrix, although representative of the specialists opinion, is little consistent. This may happen due to failures in individual matrices or even by the accumulation of errors. Precisely, we refer here to matrices which consistency reason equals to a value greater than 0,10.

A recent attempt of detecting and analyzing the inconsistency in matrices can be observed in Bramont (1996). The author presents ways to analyze the inconsistency of results obtained in applications of the AHP method, related to the usage of matrices in decision process.

4. Genetic Algorithms

The Genetic Algorithms (Holland, 1975) derive from the nature's evolution model. They are probabilistic research computational models that copy the natural evolution of the species, combining adaptation and survival concepts of the most capable individuals. They consist in groups of individuals and genetic operators that influence this population. They are computationally simple and powerful when it comes to optimal solutions.

When compared to traditional optimization models, the Genetic Algorithms (GA) present advantages as: robusty, natural parallelism, independency, simplicity and integration (Dornellas, 1997). An additional advantage is the opportunity to model optimization problems for which there is no explicit objective function, requiring a simulation model to evaluate the performance of the candidate solution. Stopping rules definition is a possible obstacle when applying the GAs. The ideal situation would be stopping the algorithm whenever the optimal solution is obtained, however the usual stopping rules are the maximum number of generations, the processing time limit and stopping when no substantial improvement happens after a few generations.

Although they are inherently random, the Genetic Algorithms utilize, in a very efficient way, the historical information existent in its code, in order to suggest new solutions, which improve the final result of the problem. Therefore, they are classified as emerging intelligence algorithms (Angeline, 1993). Holland (1975) proved that GAs cover the sample space in cubic order (n^3), which means, while the iterations grow in order n (which is computational interesting), the number of points covered is the sample space grows in n^3.

GA consists of a cycle of the following stages: creation of a population of potential decoded solutions; evaluation of this population; selection of the most capable individuals; creation of a new population through genetic manipulation. As in the Natural Evolution Theory, GAs considers solutions that are a string of bits, which decode the parameters for the problem. The size of the population affects the efficiency of the algorithm. Each iteration of the algorithm is called a generation and both terms will be considered the same for this paper.

Genetic Algorithms provide solutions according to an evaluation function – named Fitness Function – and to a series of operators (selection, reproduction, mutation and substitution). This function is a metric that diagnosis how adapted the individual is. The operator Selection chooses individuals to reproduce; in the Reproduction operator, there is the combination of two or more solutions to generate new ones; the Mutation operator is an alteration of bits intending to restore the population diversity; and the Substitution operator is responsible for replacing ancient solutions for the ones remaining from the reproduction and mutation processes (Gen & Cheng, 2000).

5. The Model

Considering a matrix with CR greater than 10% resulted from an Analytic Hierarchy Process application, the aim of the system turns to be the generation of n solutions with a CR lower or at least similar to 10%. If this is achieved, the result will be consistent matrices similar to the original one. To this application, it should be considered matrices with variable inconsistency values, as long as they allow solutions of reasonable proximity to the original.

Taking a certain matrix, the program initially analyses the matrix consistency. It calculates the maximal eigenvalue, the consistency index and reason. If this reason is over 10%, the matrix is considered inconsistent and the algorithm randomly generates a series of matrices with small disturbs in its values (avoiding the unitary diagonal) and sequentially generates an initial population. This population is submitted to the operators generating a new matrices population, which will have their evaluation function continually analyzed. This process is over when the program obtains matrices with CR lower than 10%.

To this study, the option was to decode disturbs around the elements of a matrix so the number of bits necessary to decode a matrix is given by the number of its elements multiplied by the number of bits necessary to make disturbs in its elements. That considerably reduces the number of resultant bits from decoding a matrix. The number of bits to be used to decode and the range of disturbs in the matrix elements can be parameterized. Each individual of a certain generation is recomposed in the matrix structure. Then, the matrix CR and the degree of non reciprocity are calculated.

In sequence, there is an analysis of the process steps, presenting in a proper way the operators as well as the beginning and end conditions, some comments and a simulation of the functioning. The language used was C++.

- Initialization: the initial population of n individuals is randomly generated. This population can be parameterized. The generations occur with matrices composed of small disturbs, which should evolve naturally. Then, they should be added to the original matrix and have their consistency degree constantly evaluated. These are quadratic n x n matrices, where n varies from 2 to 9. The limits imposed to the matrix dimensions (2 to 9), follow the orientations given by Saaty (1991). Notice that, similar to the biologic case, the evolution occurs only with diversity, so the greatest the variety of individuals in the initial population the easiest it will be to convert to the solution.

- Evaluation: the Fitness Function gives each individual the proper measure for the studied situation. This function would be represented like: Fitness Function = 1 – CR. The CR value varies from zero to one (0,1) so the function would never be negative (as required by the algorithm procedure) and the greater is the CR value the more apt the individual would be. However, it was noticed that because the CR is considerably low, it is not enough to assure the matrix ideal conditions. So, the function has suffered a restrictive alteration. This restriction requires that all values of the matrix that disrespect the condition of anti-symmetry and the CR value to be below 10 %. The new function is now Fitness Function = 1 – (CR

+ NR), and NR = |1 – (Tr + a_{ij} x a_{ji})|,where CR= Consistency Reason, NR = Non Reciprocity, Tr = Tolerance Rate, a_{ij} = elements of line i and column j. The Non Reciprocity (NR) works like a trigger from which the value is added to the function. To values of non reciprocity lower than the established limit, the Fitness Function is again evaluated as initially. This trigger condition allows to completely discard the influence of a tolerable non reciprocity in the Fitness Function, letting the algorithm evaluate the individuals in terms of its consistency reason up to a tolerable limit of non reciprocity. An output matrix is considered as solution as long as its Fitness Function, evaluated by the algorithm, is above the limit value established by the user.

- Selection: this operator selects individuals which are apt to reproduce themselves. As a mechanism of Selection, the Raking with Roulette Wheel method, developed by Aguiar and Costa (1997), was used. According to the method, each individual is associated to a number or ranking R_i, being $R_i = (Ni / N)^2$, where Ni is the number of individuals with Fitness value lower than the lowest individual i and N is the total number of individuals in the population. After the definition of these numbers, the traditional Roulette Wheel method is used.

- Reproduction: for the reproduction operator, it was used the Single Point and a 10% rate of reproduction. However, both in the case of the reproduction as well as of the rate, the values can be chosen by the system operator, once the model is entirely parameterized.

- Mutation: the mutation process is used to guarantee the diversity of the population. The question is to define the frequency it should occur. It was used a 10% probability or mutation rate. As in the reproduction case, this value can be altered at any moment.

- Stopping Condition: the algorithm stopping rule used requires the algorithm to stop searching when finding a solution or individual whose Fitness Function is lower than 10%. So, the number of solutions is limited only by the user. However, as the model allows the user intervention, the algorithm can be stopped at any moment or a time limit can be established to find a solution.

- System Environment Description: the system was developed in C++, using a standard library, allowing a high degree of portability. It was used the integrated development environment Bloodshed Dev-C++, version 4, Mingw compiler 2.95.2 -1 and GNU Debugger 4.18 (GDB).

- Input Pre-requisites: as requisites, the input matrix must be quadratic and diagonal. The system reads the matrix of text file, in the ascii format. The first line must contain the matrix dimension. The following lines must contain each of the matrix lines, with the elements separated by blanks.

- System Output: the system shows in the standard output the results of the processing.

To flex the process of analysis and convergence of results, the system keeps some of the most influent items parameterized. Each parameter has a default value internally predefined. Optionally, it is possible to specify the values through the command line, such as the number of individuals of a population, the minimal quantity of individuals which are apt to characterize a solution, the minimal value above which an individual is considered apt and the number of bits to decode the disturbs.

6. Example

Random matrices with unitary diagonals and variable sizes were generated to be used as tests in this model. However, to compose the model evolution documentation, we used referenced matrices already studied in other situations.

The chosen matrices were extracted from Saaty (1991) because this study is one of the most detailed and complete when it comes to matrix inconsistency. These matrices vary in size and degree of inconsistency.

The solutions obtained were quite satisfactory and respected the model conditions, considering the involved parameters. Notice that in some cases, the size of the population is small and the system rapidly converges. In Table 1, the characteristics of the matrices are showed, including initial and final Consistency Reasons from the example matrices.

Matrix	Size	Population	Initial CR	Generations	Final CR	Fitness
1	3	100	0.116884	1380	0.0737133	0.901138
2	6	100	0.238381	2262	0.0997367	0.900263
3	4	500	0.532033	291	0.0905882	0.909412
4	6	500	0.127419	117	0.0182739	0.981726
5	3	100	0.116884	492	0.0920225	0.907978
					0.0846125	0.915388
					0.0541502	0.903442

Table 1: Matrices Characteristics

In matrix 2, the solution found by the system is quite interesting, considering that the initial matrix CR is relatively high (notice this is the highest number of generations). Specifically in the case of matrix 3, the high value of the initial CR was an obstacle to the final result. It suggests that high initial CR values demand special treatment. Matrix 5 is, in fact, a variation of matrix 1, altering the stopping criteria of one of the three solutions. The solution matrices found by the system are plenty satisfactory and the CR values present small, non-meaningful variations. In matrices 3 and 4 cases, the size of the initial population was purposely raised, which would generally lead to more adapted solutions.

7. Conclusions

The main objective of this paper is to offer a possible solution to inconsistency in decision matrices. The approach involves Genetic Algorithms, which are introduced theoretically and practically (developed model). It is consensus, and a purpose of this study, that inconsistency in decision analysis must be treated in order to minimize the losses and make the decision process easier.

The importance of the Genetic Algorithms in this theme relates to the many advantages that this kind of computational method offers, among which the possibility of generating more than one solution to each matrix with an inconsistency problem, allowing the decision maker to have a greater number of options to replace the original matrix. On one hand, this subject may be seen as a complication, as in some circumstances the decision maker may not be interesting in choosing among various solutions close to the optimum. However, this becomes quite simple to resolve if the algorithm (model) is calibrated to offer only a solution for each problem.

There was a good performance of the algorithm model, requiring no user intervention and having no variations in the mutation and reproduction rates. Although these situations are very common in Genetic Algorithms, the system is still useful. It is worthy to notice the variation occurred in the reciprocity rate, according to the example, in order to keep solution similar to the original matrix. A necessary concern when using computational models is the time to process, a considerable factor in the problem solution evaluation. In this case, it was relatively short, considering the dimensions of the proposed problem, quite close to the expected time, which is an advantage of the model.

Attention to the fact that the models used as examples were all extracted of published cases. In most of these examples, it is satisfactory to respect the reciprocity condition, considering less important the transitivity condition. In fact, some authors consider that intransitivity among preferences may be considered as a natural phenomenon and not as a consequence of judgment errors.

References

Aguiar, E., Costa, J. F. S. (1997) “Uma Implementação de Pesquisa Estocástica utilizando Algoritmos Evolutivos”. *Cadernos do IME*, ISSN 1413-9022, vol 1, RJ, Brasil.

Angeline, P. J. (1993). *Evolutionary Algorithms and Emergent Intelligence*. Dissertation, The Ohio State University, USA.

Bramont, B.P.P. (1996) “Consideração de Incertezas nas Estimativas de Pesos dos Critérios em MCDM”. *Proceedings* of the XVI ENEGEP, SP, Brasil.

Chankong, Y., Haimes, Y. (1983) *Multiobjective Decision Making*. Amsterdam, Ed. North Holland.

Costa, H. G (2006) *Auxílio Multicritério à Decisão: Método AHP*. Latec/UFF - ABEPRO, RJ, Brasil.

Dias, L.M.C., Almeida, L.M.A.T., Clímaco, J.C.N. (1996) *Apoio Multicritério à Decisão*. Portugal, Universidade de Coimbra.

Dornellas, C.R.R. (1997) *Otimização dos Despachos Reativos Utilizando Algoritmos Genéticos*. M.Sc. thesis, COPPE – UFRJ, RJ, Brasil.

Gen, M., Cheng, R. (2000) *Genetic Algorithmṣ Engineering Optimization*. Wiley, NY, USA.

Graybill, F. A. (1983) *Matrices with Applications in Statistics*. Wadsworth Inc., California, USA.

Holland J. (1975) *Adaptation in Natural and Artificial Systems*. The Univ of Michigan Press, Michigan, USA.

Roy, B. (1985) *Methodologie Multicritere d’Aide á la Decision*. Paris, Econômica.

Saaty, T.L. (1980) *The Analytic Hierarchy Process*. New York: Mc Graw Hill, USA.

Saaty, T.L. (1991) *Método de Análise Hierarquica*. Makrom Books, 2Ed, SP, Brasil.

Vargas, L.G. (1982) “Reciprocal Matrices with Random Coefficients”. *Math. Model*, 3, 69-81, USA.