Sensitivity analysis in the Scenario Method: a multi-objective approach

Pablo J. Villacorta, Antonio D. Masegosa, Dagoberto Castellanos[†], Pavel Novoa and David A. Pelta
Models of Decision and Optimization Research Group
CITIC-UGR, Department of Computer Science and Artificial Intelligence
University of Granada, 18071 Granada, Spain
Email: {pjvi, admase, pavel, dpelta}@decsai.ugr.es, [†]dcastellanos@ugr.es

***Abstract*—Technology foresight deals with the necessity of anticipating the future to better adapt to new situations regarding innovations that directly affect business world. One widely spread methodology in technology foresight is Godet's Scenario Method, which includes a module (MICMAC) performing the so-called structural analysis. The goal of the structural analysis is to identify the most important variables in a system. To this end, it makes use of an influence matrix that describes the relations between the variables. This information is usually given by experts based on their own knowledge and experience. However, some of the information of the influence matrix may contain errors due to the subjective nature of the criteria and opinions of the experts. Here we propose a new analysis that follows a multi-objective approach and allows to measure the sensibility of the model versus possible errors at the input. The well-known NSGA-II algorithm has been used as a solver. The results are encouraging and deserve further investigation.**

***Keywords*-scenario method; technology foresight; Godet's method; multiobjective optimization;**

I. Introduction

Technology foresight analysis is a differentiating factor in innovation management and decision-making. The competitive advantages of organizations are achieved by accurately identifying the scenarios they must address. Organizations rely on systems that perform prospective analysis to be ahead of the changes in the environment.

One of the most frequently used methodologies to accomplish prospective analysis is the Scenario Method [1], [2], [3], [4] proposed by M. Godet, from the Laboratory for Investigation in Prospective Strategy and Organization (LIPSOR). It has been used in typical forecasting problems [5] but also in other kinds of contexts such as User Interface Design [6]. This methodology determines the possible futures by means of the definition of scenarios. A scenario can be considered a representation of a future state and the steps that lead to it. These scenarios are usually developed based on information from various experts which is obtained through opinion pools, panels, etc.

The Scenario Method carries out the analysis of the scenarios by means of a set of tools that provide support to accomplish the structural analysis, strategies of the actors, morphological analysis, expert methods and multicriteria decision. It covers from the analysis of the key variables to the analysis of the strategic actions that must be taken. Among the different stages, we have focused on the structural analysis since it is the first step of the complete methodology. To be precise, our aim is to measure the sensibility of the system versus errors in the data. On the one hand, we want to evaluate how the system is affected by small errors in the valuations given by the experts as input and, on the other hand, we want to find out critical valuations in which a small input error causes a great change at the output. This can be tackled as a multi-objective optimization problem where the goal is to find the group of variables that, with a minimum change, cause the maximum variation at the output of the model. The results will allow to determine which variables are of greater relevance for the process.

The contribution is structured as follows. Section II explains the structural analysis method (MICMAC) proposed by Godet. Section III reviews the concept of a multi-objective optimization problem and explains the suitability of this approach to study sensibility in the MICMAC method. All the specific components required to tackle our concrete problem are described in detail here. Section IV deals with the experiments conducted and section V with the results obtained. Finally section VI is devoted to conclusions and further work.

II. Godet's Structural analysis

Godet's structural analysis method, called MICMAC, defines a series of variables and a matrix describing the influence relations among them. Through the study of such relations, the method can uncover which variables are essential for the system. The method can be used standalone, as a decision support system, or as a part of a more complex scenario analysis.

First of all, a list of n variables is given by a group of experts. These are the variables that in principle could be of interest for the system being modeled. Then, a matrix with dimension n x n is defined in which the influence between any pair of variables are given on the basis of expertise knowledge. This matrix is known as the Matrix of Direct Influence (MDI). Each cell of the matrix states the influence of one variable over some other variable, measured in a 4-grade scale, as follows. MDI_{ij} represents the influence of variable i on variable j and its value can be

- 0 if variable i has no influence on variable j.

- 1 if variable i has a low influence on variable j.
- 2 if variable i has a medium influence on variable j.
- 3 if variable i has a strong influence on variable j.
- 4 if variable i has a *potential* relation with variable j.

A potential relation means that the influence is not clear but may become notorious under some conditions. For that reason Godet discards such relations during the structural analysis process. The main diagonal of the matrix is always 0 because a variable does not have influence on itself.

Some important measures can be computed from the MDI. The *direct influence* of a variable i over the rest is computed as the sum of all the values of row i of the *MDI*. In this sum, the potential influences are ignored and taken as 0. Similarly, the *indirect dependence* of a variable i of the rest is computed as the sum of all the values of column i. Therefore we have two different measures associated with every variable. With this information, an influence ranking and a dependence ranking are built by sorting the variables according to their influence and to their dependence, respectively. Both rankings serve as a first indicator of the importance of each variable in the system. These calculations are known as the *direct method*.

Other important measures that can be computed from the MICMAC are the *indirect influence* and the *indirect dependence* of the variables. They are calculated using the *indirect method*, which is an iterative process in two steps:

1) Initialization step. Let R_f and R_d be the influence and dependence rankings obtained with the direct method. Initialize M to be the original *MDI* matrix.
2) Iteration:
   - Do $M = M$ x MDI and compute the new influence and dependence rankings S_f and S_d with the resulting matrix, as explained above.
   - Compare S_f with R_f and S_d with R_d.
   - If both comparisons match, finalize. Otherwise, update the old rankings: let $R_f = S_f$ and let $R_d = S_d$ and go to step 2 again.

It is expected that both rankings converge in about 8 steps, thus the matrix used to compute the final rankings is *MDI*^{8}. Ranking convergence means that the rankings of two consecutive iterations are exactly the same. The final influence and dependence rankings are also important measures indicating the importance of every variable.

Notice that all these data allow some interesting plots to resume the information, such as 2D diagrams of influence-dependence, where a point in the space represents a variable as a pair (influence, dependence).

III. Sensitivity Analysis from a Multi-objective Approach

Since the values of the MDI matrix are provided by human experts based on their subjective opinion, they may contain errors. However there is no method that evaluates the sensitivity of the conclusions obtained by the MICMAC method when the input data contain errors. In the rest of this contribution we explain a novel approach that allows to measure the sensitivity of the variables and determine which are more affected by errors in the input data. This issue can be tackled as a multi-objective optimization problem as we will explain later. The next section introduces the basics of multi-objective optimization.

A. Multi-objective optimization

In a classical optimization problem with one single objective, we are interested in finding the values for a set of variables that minimize or maximize one objective function f, usually subject to constraints. A great variety of both analytical and heuristic techniques have been developed to solve this task and their performance has been successfully tested in many situations. However, a much more difficult problem arises when we want to optimize several functions $f_1, ..., f_n$ at the same time. If the objectives are not conflicting, then the problem is not really a multi-objective one since the optimal solution for one objective is also optimal for the rest. If, on the other hand, it is impossible to optimize all the functions at the same time because they run into conflict, the task becomes really difficult. In such conditions, a solution that is very good for one objective function may be very bad for the others. For that reason, all the objectives must be taken into account at the same time.

First of all, let us introduce the concept of *dominance* [7]. Let $f_1, ..., f_n | f_i : R^m \rightarrow R$ be a set of n objective functions that have to be minimized simultaneously. Let $X = (x_1, ..., x_m)$ and $Y = (y_1, ..., y_m)$ be two different feasible solutions satisfying the additional constraints of the problem. We say that X *dominates* Y (for minimization) iff

$$\forall i \in \{1, 2, ..., n\} : f_i(X) \leq f_i(Y) \wedge \\ \exists j \in \{1, 2, ..., n\} : f_j(X) < f_j(Y)$$

This definition states that solution X is better than solution Y if it is better or equal in all the objective functions and strictly better in at least one of them. Similarly, when solving a maximization problem, dominance is expressed with $\geq$.

The goal of multi-objective optimization is to find a *set* of solutions (not just one single solution) that is the closest to the so-called *Pareto-optimal front*. The Pareto-optimal front is the set of solutions of the feasible region that dominate all the others feasible solutions, and that are not dominated by any other feasible solution. Ideally, a multi-objective optimization process should return this set as an output. Since this is a very hard task, we search for good Pareto-fronts that are as close as possible from the optimal Pareto front.

B. Sensibility analysis as a multi-objective problem

Focusing again on our problem, we are interested in finding which cells of the influence matrix cause the maximum variaton at the output when affected by minimum changes. In other words, we search for a matrix with the *smallest* variation with respect to the original MDI matrix that causes the *maximum* variation at the output of the MICMAC indirect method. If both the variations in the influence matrix and the variations at the output can be numerically quantified, we can say we have two different conflicting *objective functions* that must be optimized at the same time. They are conflicting because the best solution for one of the objectives (for example a matrix with no variations in its cells) is the worst for the other one since it causes no changes at all in the influence and dependence rankings obtained. Thus we face a bi-objective optimization problem (with objective functions f_1, f_2) that is to be addressed by means of approximate heuristic techniques since the objective functions do not have a clear analytical expression but are the result of a computational procedure (i.e. the indirect method explained in section II).

The variables being optimized are the values of the new influence matrix M that is to be found in the search space of influence matrices with dimensions n x n. A scheme that summarizes the calculation of the values of the two objective functions is shown in Fig. 1. Since we assume that the variations in the input matrix come from small errors of human experts, the search space was restricted to the matrices whose cells differ from the original MDI matrix in no more than 1 unit (greater or smaller).

Objective function f_1 : distance between matrices. We want to *minimize* the variation in the MDI matrix or, in other words, we want to find another n x n influence matrix M that is the closest to the original MDI matrix. Different metrics can be employed to compute the distance between two matrices; the most suitable for our problem is the accumulated difference between the cells. It is a generalization of the *Hamming* distance that takes into account the magnitude of the change in every cell.

$$f_1(M) = d(M, MDI) \tag{1}$$

where the distance function is defined as

$$d(A,B) = \sum_i \sum_j |A_{ij} - B_{ij}|$$

Objective function f_2 : number of changes of the influence and dependence rankings of the indirect method with 8 iterations. We want to *maximize* the number of differences between the influence and dependence rankings obtained via the indirect method applied to the original *MDI* matrix and those obtained with the indirect method applied to the matrix being evaluated. In other words, we want to find another n x n influence matrix which leads to very

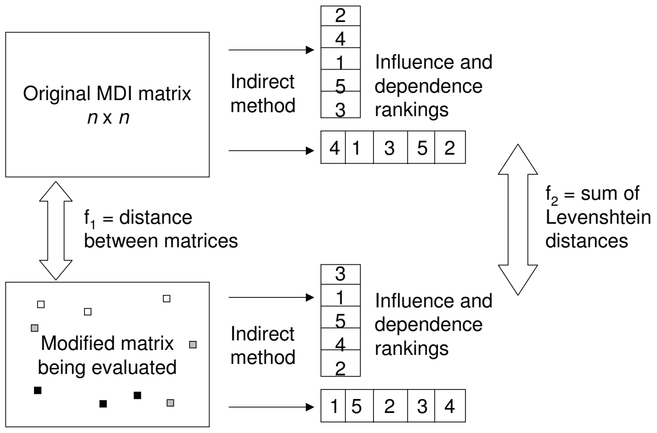
Original MDI matrix
n x n
Indirect method
2 4 1 5 3
Influence and dependence rankings
4 1 3 5 2
f1 = distance between matrices
f2 = sum of Levenshtein distances
Modified matrix being evaluated
Indirect method
3 1 5 4 2
Influence and dependence rankings
1 5 2 3 4

Figure 1. Computation of the objective functions f_1 and f_2

different influence and dependence rankings. The rankings corresponding to a matrix M are represented as two integer vectors v^M_{inf} and v^M_{dep}. Each vector contains a permutation of integers 1 to n in which each number identifies the variable in that position. If the content of $v^M_{inf}[i]$ is k, it means the i-th most important variable is k, according to the influence criterion. The same applies to the dependence criterion. Then, a measure of "similarity" between v^{MDI}_{inf} and v^M_{inf} is computed, and the same for the pair v^{MDI}_{dep} and v^M_{dep}. Finally, the value of the objective function is the sum of these two values:

$$f_2(M) = simil(v^{MDI}_{inf}, v^M_{inf}) + simil(v^{MDI}_{dep}, v^M_{dep}) \tag{2}$$

The computation of the similarity between two rankings, represented as two integer vectors, was done using a metric called the *Levenshtein distance* [8], [9]. It is widely employed in information theory to measure the similarity of two character strings but it is also useful in this case. It is a measure of the number of permutations required in one vector to obtain the other vector.

C. The NSGA-II algorithm

A number of multi-objective optimization algorithms have been proposed in the last decade. They are all able to find a set of non-dominated solutions in one single run. Although their performance was acceptable, they have all been criticized for their computational complexity, for their non-elitism approach and for the need to specify additional parameters. Deb et al. [10] proposed an improved version of their NSGA algorithm and called it NSGAII, which solved these drawbacks. This paper had an immediate impact and NSGAII was quickly applied to a great number of problems in engineering and other areas. It is currently one of the best performing multi-objective algorithms, and in addition the author offers a ready-to-use free implementation [1]. Further

[1]http://www.iitk.ac.in/kangal/codes.shtml

details about this algorithm can be found in [7].

IV. Experimental Framework

We chose a real problem with a real *MDI* matrix for our experiments. In order to have results that were comparable, we picked one of the examples that comes with the LIPSOR software tools. It is a prospective study about the determinants of the rural spaces in the time horizon of 2010. The example takes into account 50 variables; for informational purposes some of them are listed below.

1) Metropolization
2) International market organization
3) Food demand
4) Contribution of migration
5) Job market
6) Elderly people
7) Social politics
8) Activity diversification
9) Management of the environment
10) Ecological claims
11) Administrative land organisation
12) Political representation

A. NSGAII settings for sensibility analysis

Since there are 50 variables in total, the chromosome we are dealing with represents a 50 x 50 matrix so it is 2500 values length, and for the same reason, the influence and the dependence rankings have 50 elements, so the maximum of the sum of both Levenshtein distances is 100. Thus this is the maximum possible value of Objective 2.

Representation: a discrete representation scheme was used. Since we are interested in the results when small changes affect the MDI matrix, we want to explore matrices whose cells only differ in 1 unit (above or below) from the original matrix. This constraint was implemented by using a chromosome (representing a matrix) with values -1, 0 and 1 only. The fitness function was computed by previously adding the chromosome values to the MDI matrix, to obtain a new matrix M which is the one being evaluated (as explained in section III-B).

Population size and initialization: all the individuals were initialized to 0 in all positions, which means the population is formed by exact copies of the initial MDI matrix (because all cells present a variation of 0 units). This was done to assure all the individuals explored are very near to the original MDI matrix. At the first generations, the mutation is crucial since it is the only mechanism that introduces diversity in the population. The number of individuals of the population was set to 200. It has to be high because in our test case, the chromosomes are 2500 values length which is very large.

Parameters: the values for the crossover and mutation rates were the following:

- Crossover rate: 0.9.
- Crossover type: uniform. Preliminary experiments showed it performs better than one-point crossover.
- Mutation rate: 0.001 per variable. When mutation is to be applied, a new random value (-1, 0 or 1, each with the same probability) replaces the old value.

Additional constraints: the multi-objective problem itself does not have additional constraints. However, in a realistic sensibility analysis, the *MDI* matrix is not expected to contain more than 30 errors (out of 2500 cells) and thus, each individual of our algorithm is not allowed to have more than 30 cells different from 0. This constraint has to be checked after every crossover and mutation operation.

V. Results obtained

Five independent runs were made, and a Pareto-front of 200 individuals (i.e. as large as the complete population) was collected in every run, although it is a bit smaller as they contain repeated solutions. All the Pareto fronts were aggregated and sorted. After the aggregation, the dominated individuals were removed, as well as the repeated ones. The resulting Pareto-front has 23 individuals (Fig. 2).

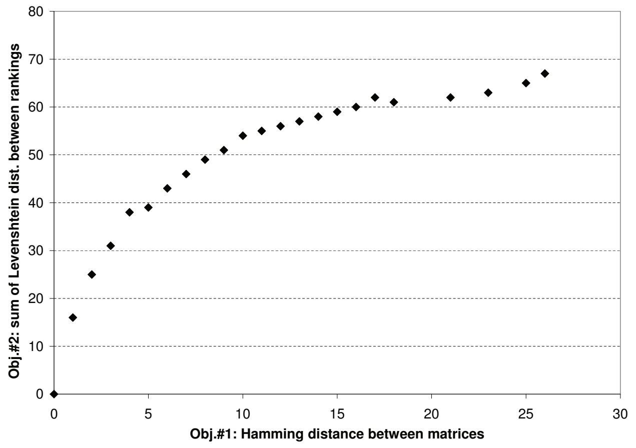

Figure 2. Pareto-fronts obtained in the executions of NSGAII algorithm

The ideal situation would be on the top left-most corner of the figure because Objective 1 (X axis) must be minimized while Objective 2 (Y axis) must be maximized. Note that, as we only allow a variation of 1 in a cell, then the generalized Hamming distance is indeed the traditional Hamming distance that measures the number of cells that are different. This is important because it can be seen in the figure that a variation of only 1 cell can cause a big change in the rankings (about 18 in the Levenshtein distance). We will later study which variables are affected by the changes but this situation can be explained as follows. When a variable that is in position, say, 25 in one ranking, and it is moved to position, say, 15, it causes a lot of variations: that variable changes but in addition, it makes variables in positions 15 to 24 move one position each (24 moves to 25, 23 moves to 24 and so on), and these changes increase quite a lot the Levenshtein distance.

Recall that the goal of our analysis is to determine which variables seem more affected by small variations in the MDI matrix. A way to discover this is to summarize the information of the Pareto-front and see if the individuals share common values in the same variables or they present a lot of variation between individuals. This is depicted in Fig. 3. Every bar has two parts, one on the left and one on the right, to show separately the contribution of the individuals with -1 and the individuals with +1 to the total number of variations in each position of the influence matrices represented by the individuals.

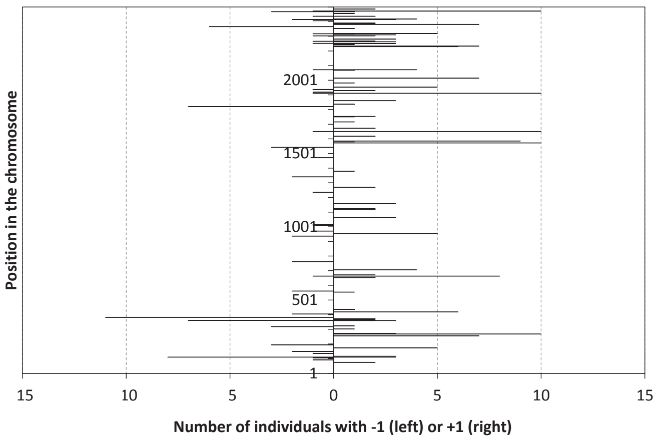

Figure 3. Number of individuals (without repetitions) of the Pareto-front containing a variation (+1 or -1) in each position of the individual

As can be seen, some cells are different from 0 in a great number (12 out of 23) of individuals of the Pareto-front, and this happens even in different independent runs. This is an important clue about the variables that are more responsive to variations. Another important conclusion can be drawn from Fig. 2 that is related to the former. Observe that there are solutions in the Pareto-front that achieve a Levenshtein distance of more than 30 (which in average is 15 in both rankings) by changing less than 4 cells. This is an important fact that deserves a thorought analysis. The figure does not inform of the *importance* of the changes in the rankings, i.e. if the positions that have changed were mostly on the top of the rankings, corresponding to more relevant variables, or on the bottom, corresponding to variables that are not really relevant and could be eliminated from the model if the prospective process is to be continued with more phases. For that reason, it is necessary to study what happens in such cases. We have used the MICMAC in order to test the solutions found by our method. Concretely, we have take the solution in which there is only one change in initial MDI matrix. The cell modified (from 0 to 1) corresponds to the influence of the variable 3 (food demand) on the variable 11 (administrative organisation of the territory). Evaluating this small change in the MDI matrix (1 cell out of 2500) the following variations in the original rankings were observed:

- Direct influence ranking:
  - Variable 3 goes from position 29th to 27th.
- Indirect influence ranking:
  - Variables 33 and 21 swap positions 11th and 12th.
  - Variables 17 and 17 swap positions 26th and 28th
  - Variable 3 goes from position 29th to 27th.
- Direct dependence ranking:
  - Variables 13 and 45 swap positions 23rd and 24th.
- Indirect dependence ranking:
  - Variable 11 goes from position 49th to 45th.

In this way, we see that a small change in a position of the matrix leads to six alterations of different importance in the original rankings, so this information is quite valuable to establish which relations are critical and therefore should receive greater attention. Although in this case the variables affected by a change in cell (3, 11) were not at the top of the rankings, this fact does not invalidate the foundations of our novel multi-objective approach.

VI. Conclusions and Further Work

In this contribution, a multi-objective approach for the evaluation of the impact of a small change on some variables of a system under study with technology foresight techniques has been presented. This will allow to determine which values are critical and thus which variables are of greater relevance for the process. Some insights have been given on the interpretation of the results obtained, which is an advantage of this method. The solutions of the Pareto-front can be traced and analysed in relation to the meaning of each variable in our concrete problem. The results of our approach are encouraging and deserve further investigation.

As future work, some enhancements such as introducing expert knowledge on the optimization process could be useful. An immediate improvement is to consider not only how many changes we see in the rankings but also which variables are affected by the rank changes, i.e. if the variables that change their positions were originally on the top of the ranking or not. This may be achieved by using a penalty factor when computing the Levenshtein distance of the fitness function for that objective.

Acknowledgment

This work has been partially funded by the project TSI-020513-2009-74 from the Spanish Ministry of Industry, Tourism and Commerce, the project TIN2008-01948 from the Spanish Ministry of Science and Innovation and P07-TIC-02970 from the Andalusian Government.

References

[1] M. Godet, “How to be rigorous with scenario planning,” *Prospective*, vol. 2, no. 1, pp. 5–9, 2000.

[2] P. Durance and M. Godet, “Scenario building: uses and abuses,” *Technological Forecasting and Social Science*, vol. 77, no. 9, pp. 1488 – 1492, 2010.

[3] M. Godet, *Scenarios and Strategic Management*. Butterworth-Heinemann, 1987.

[4] ——, *Creating Futures: Scenario Planning as Strategic Management Tool*. Economica, 2006.

[5] R. Bettencourt, “Strategic prospective for the implementation of employment policies in the Azores,” *Technological Forecasting and Social Change*, vol. 77, no. 9, pp. 1566 – 1574, 2010.

[6] Y.-C. Lee, Y. H. Chao, and S.-B. Lin, “Structural approach to design user interface,” *Computers in Industry*, vol. 61, no. 7, pp. 613 – 623, 2010.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, 2001.

[8] V. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics, Doklady*, vol. 10, no. 8, pp. 707 – 710, 1966.

[9] R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *Journal of the ACM*, vol. 21, no. 1, pp. 168 – 173, 1974.

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2000, vol. 1917.