A Novel Application of Evolutionary Computing in Process Systems Engineering

Jessica Andrea Carballido[1,2], Ignacio Ponzoni[1,2], and Nélida Beatriz Brignole[1,2]

[1] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC)
Departamento de Ciencias e Ingeniería de la Computación,
Universidad Nacional del Sur, Av. Alem 1253, 8000, Bahía Blanca, Argentina
[2] Planta Piloto de Ingeniería Química – CONICET,
Complejo CRIBABB, Camino La Carrindanga km.7 CC 717, Bahía Blanca, Argentina
{jac, ip, nbb}@cs.uns.edu.ar

Abstract. In this article we present a Multi-Objective Genetic Algorithm for Initialization (MOGAI) that finds a starting sensor configuration for Observability Analysis (OA), this study being a crucial stage in the design and revamp of process-plant instrumentation. The MOGAI is a binary-coded genetic algorithm with a three-objective fitness function based on cost, reliability and observability metrics. MOGAI's special features are: dynamic adaptive bit-flip mutation and guided generation of the initial population, both giving a special treatment to non-feasible individuals, and an adaptive genotypic convergence criterion to stop the algorithm. The algorithmic behavior was evaluated through the analysis of the mathematical model that represents an ammonia synthesis plant. Its efficacy was assessed by comparing the performance of the OA algorithm with and without MOGAI initialization. The genetic algorithm proved to be advantageous because it led to a significant reduction in the number of iterations required by the OA algorithm.

Keywords: Combinatorial Optimization Problem, PSE, Process-Plant Instrumentation Design, Multi-Objective Genetic Algorithm, Observability Analysis.

1 An Application in the Field of Process Systems Engineering

Process plants are networks of industrial items of equipment physically connected by streams. The instrumentation design problem is a challenging activity in the area of Process Systems Engineering (PSE). It consists in deciding on the most convenient amount, location and type of measuring devices to be incorporated into the industrial process. The objective is to achieve complete knowledge of the plant's operating conditions, while satisfying other goals such as sensor-cost minimization and maximum reliability. Due to the complexity of this task, the development of automatic decision-support tools for this purpose has become a challenge [1].

The computer-aided design of process-plant instrumentation is an iterative procedure that comprises several steps. In the first place, a steady-state mathematical model is built in order to represent plant behavior under stationary operating conditions. This

G.R. Raidl and J. Gottlieb (Eds.): EvoCOP 2005, LNCS 3448, pp. 12 – 22, 2005.

model is a set of algebraic equations that correspond to mass and energy balances, as well as relationships employed to estimate thermodynamic properties like densities, enthalpies, and equilibrium constants. A rigorous model usually involves not only linear functionalities, but also many bilinear and nonlinear equations. Apart from the model, an initial instrument configuration also has to be defined. This preliminary design classifies model variables into measured and unmeasured ones, the former being those whose values will be obtained directly from the sensors.

The next step is to carry out the Observability Analysis (OA), which consists in pinpointing the unmeasured variables that will be observable, i.e. those that can be calculated by means of model equations, regarding the measurements as constants. The OA Algorithm (OAA) used for this purpose [2] analyzes the structural relationships between model equations and unmeasured variables. This analysis is performed by permuting a sparse occurrence matrix built from information about both the model and the measurements in order to obtain a desirable pattern.

It is important to remark that some variables are critical for the industrial process under study because they represent vital information about it (i.e. temperature in a reactor), while others could be considered scarcely relevant. In principle, a careful OA should yield a classification where all the key unmeasured variables are observable. If, after an execution of the OAA, the result contains critical indeterminable variables, the configuration of sensors should be modified and the OA has to be repeated. In this way, the OA normally becomes an iterative procedure.

The last major step required to complete the entire design procedure is the classification of the measurements, also known as redundancy analysis [3]. This task should be carried out only after a satisfactory OA has been achieved.

This paper is focused on the search for an accurate automated OA initialization strategy. The flow diagram of the OA procedure with manual initialization is shown in Fig. 1, where the rectangular boxes represent automated tasks, while the others are associated with expert activities handed over to the decision maker (DM).

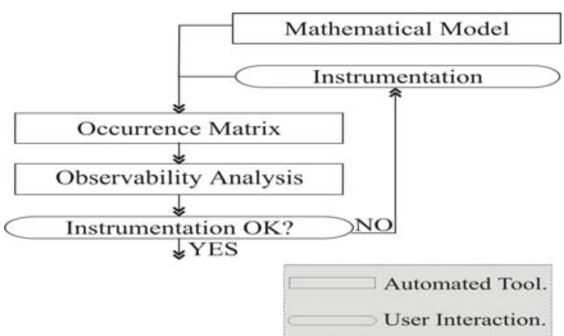

Fig. 1. Iterative process for the OA stage with manual initialization

Both the OA time efficiency and quality of the results depend on the starting point, and will therefore benefit from a careful choice of sensors. Notice that the number of iterations required in order to reach an acceptable result may vary significantly with the initialization. Since the OAA sweeps are very expensive as regards computing time, it is highly advantageous to have as few iterations as possible. This objective can be achieved by choosing an adequate initial instrument configuration. At present, there are no algorithms to make optimal selections in this sense. Therefore, plant engineers choose the sensors exclusively on the basis of their skill and experience. The purpose of this work is to develop an automated tool to tackle this problem, thus supporting them in the making of these complex decisions. The new scheme, shown in Fig. 2, represents the interaction between the DM and the instrumentation design package when the automated tool for initial configuration has been incorporated.

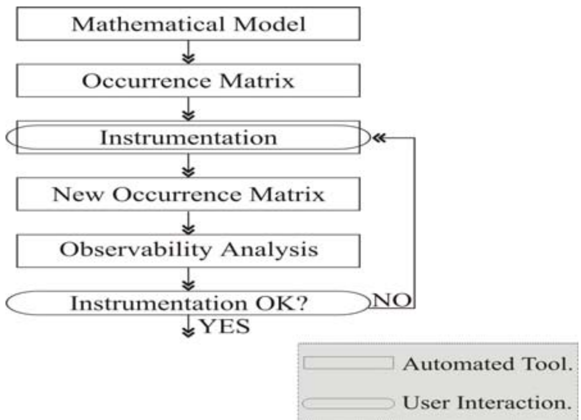

Fig. 2. Iterative process for the OA stage with automated initialization

2 The Sensor Choice as a Combinatorial Optimization Problem

The selection of the initial set of instruments can be classified as a combinatorial optimization problem involving several objectives expressed in different units and in mutual conflict. In particular, these features characterize the so-called multi-objective or multi-criterion optimization problems (MOPs), whose special characteristic is that they have no single solution. There is a set of valid solutions instead, and each one may be considered the solution of the problem. This holds since none of them outweighs or "dominates" the others in all the objective functions. The valid solutions are called non-dominated and form the Pareto front. The MOP is defined by Osyczka [4] as "the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria, which are usually in conflict with each other. Hence, the term 'optimize' means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."

For our particular problem, the conflicting objectives are associated with sensor reliability, purchase and installation costs of the network, and observability level provided by the resulting mathematical model. It is important to remark that the MOGAI is intended as a module for a specialized decision support system, where the DM plays a major role interacting at several points of the procedure as shown in Fig. 2. In this way, the definitive decisions that ultimately determine each configuration are always made by the DM. Then, a multi-objective approach is useful because it not only allows the DM to choose among several feasible alternative solutions, but also helps him weigh various criteria simultaneously. It is interesting to note that, in this case, the DM's expertise can never be totally replaced by an automated tool because there are many subtle, sometimes subjective, aspects impossible to capture with enough detail through mathematical formulations.

There is a wealth of literature about multi-objective optimization techniques [5, 6, 7, *passim*], ranging from the conventional approaches to the evolutionary ones. Traditional methods are very limited [5]. In general, as problem size grows, these strategies are too expensive to allow obtain results in polynomial times. Since Rosenberg [8] pointed out the potential of evolutionary algorithms for MOP solving interest of the evolutionary community in this area has grown enormously. This is justified on the grounds that most real-life problems are MOPs, and also because evolutionary algorithms have the inherent capability of finding the Pareto front in reasonable times [5]. Genetic algorithms (GAs) are particularly suitable for MOPs because they simultaneously deal with a set of possible solutions (population). Thus, several members of the Pareto optimal set can be found in a single run, instead of having to perform various runs, as is the case of the traditional mathematical programming methods. Moreover, in comparison with the typical optimization methods, GAs are less susceptible to shape or continuity, easily dealing with discontinuous or concave Pareto fronts.

3 Main Objective and Proposal

In this work we describe a new automated tool, whose purpose is to find a satisfactory initial sensor network configuration for process plants so that the number of iterations involved in the OA is reduced. In this case, a configuration is considered desirable when it is cheap, reliable and meaningful in the sense that it should provide as much plant information as possible. At the same time, short computing times were required, this being a standard demand for any initialization method. When there are several conflicting objectives, the notion of "optimum" means that we are really trying to find a good trade-off solution that fulfills all the targets as satisfactorily as possible.

Several aspects of our specific application led us to use a GA. First of all, it is not imperative for us to find an optimum, a solution near the Pareto front being good enough. Besides, we need various candidate solutions for the DM to make the final decision. Finally, the tool must be fast and efficacious for huge problem instances.

Founded on the reasons explained above, we decided to implement a multi-objective genetic algorithm based on an aggregative non Pareto method. The technique combines (or "aggregates") all the objectives into a single one, without incorporating the concept of Pareto optima directly. This approach was adopted mainly because it is efficient and works especially well for a small number of objectives [7].

4 The Genetic Algorithm

The input of the MOGAI for OA initialization is the occurrence matrix $\boldsymbol{O}$ built from the steady-state mathematical model of the plant under study. $\boldsymbol{O}$'s rows and columns respectively correspond to model equations and variables. In principle, every process variable, namely temperatures, pressures, flow rates and compositions, is associated to a sensor that could be chosen for its measurement. So, the GA also needs information about the cost and reliability of each potential instrument. In this work, the cost of measuring a variable was calculated as the price of the device plus its installation costs, while the reliability of a variable was considered inversely proportional to the instrument's average error reported by the manufacturer. This information was loaded in **N**-dimensional vectors, where **N** is the total number of model variables.

Representation of Potential Solutions to the Problem - Main Operators. The individuals were represented in the canonical (binary) form. Bit-flip mutation and one-point crossover operators were employed. Each genotype, represented here by the symbol i, should be interpreted as an entire sensor configuration, where a nonzero value on one of the bits means that the variable on that position should be measured. The string length is equal to the total amount of variables (**N**) in the model.

Parameter Control. An excellent review about parameter control was published by Eiben et al. [9]. They support the statement that any static set of parameters, i.e. one whose values remain fixed throughout a run, is in principle inappropriate. For this reason, we decided to explore parameter control techniques as an alternative. In particular, we obtained good results by using an adaptive mutation operator inspired in the one proposed by Bäck [10]. In our case, we defined an initial mutation probability equal to $1/\ell$, where ℓ is the length of the chromosome. This quantity is decreased during the evolution so as to increase exploitation as the algorithm evolves. When this parameter is applied, it is combined with an adaptive value based on the fitness of the individual to be mutated. The idea behind this operator is to give more chances of mutation to those individuals that are far from the optimum. As we shall see later, there is a utopian optimum for our fitness function, whose value is equal to the number of objectives.

Infeasible Individuals. An individual is not feasible when it contains a non-zero in a position that represents an unmeasurable variable, such as an enthalpy. We have given special treatment to infeasible individuals as follows. First, the initial population was generated with a restriction on the positions of the unmeasurable variables, which were always initialized with a zero. In addition, as new gene data could be introduced only through the mutation operator, those positions were regarded as "non-mutable". In this case, it was essential to implement this policy instead of applying penalties since the first test runs, where the generation of infeasible individuals was allowed, resulted in populations with too few valid individuals, at most 30%. Furthermore, these variables must be coded since they have to be present when the observability term of the fitness function is calculated, as will be discussed later.

Selection and Replacement. The selection method is based on the roulette wheel approach, which picks out the individuals that will constitute the parent pool according to the value of the objective function. The chosen individuals replace the old ones, building a new population that in turn undergoes crossover and mutation.

The Convergence Criterion. From a recent review about stopping criteria for GAs [11] it is clear that, in general terms, it is unadvisable to run a GA for a fixed number of generations. For this reason we decided to implement an adaptive termination condition based on the concept of schemata. A schema is a template that establishes similarities among chromosomes. It is represented through a string of symbols in {0. 1, #}, where # is a wildcard. For example, string 011001 is an instance of the schemata 01##0#. As stated by Radcliffe [12], when two parents are instances of the same schema, the offspring will also be an instance of that schema. In particular, if the schema carries good fitness to its instances, the whole population will tend to converge over the bits defined by that schema. Once convergence has been reached, all the offspring will be instances of that schema. Thus, the solution will also be an instance of that schema. For this reason, our criterion analyzes the genotypes until a high percentage becomes an instance of the same schema. For general information on genotypic termination criteria, see [13].

4.1 The Multi-objective Fitness Function

This algorithm aims at finding the individual that simultaneously exhibits the best trade-off performance with respect to the following three objective functions:

The Cost Term. Given a cost vector *cv* of length N, the total cost of an individual is the sum of the values of all the elements in *cv* that correspond to nonzero entries in *i*.

$$\boldsymbol{C}\,(\boldsymbol{i}) = \sum_{j=1}^{N} (\boldsymbol{cv}[j] * \boldsymbol{i}[j]) \ . \tag{1}$$

The Reliability Term. Given a reliability vector *rv,* and following the same line of reasoning, we have:

$$\boldsymbol{R}\,(\boldsymbol{i}) = \sum_{k=1}^{N} (\boldsymbol{rv}[k] * \boldsymbol{i}[k]) \ . \tag{2}$$

The Observability Term. In contrast with the other two objective functions, this one cannot be calculated in a straightforward way. Its estimation was based on the mathematical operation called Forward Triangularization (FT). Details on the FT procedure can be found in [2]. FT returns estimates on the number of unmeasured variables that can be directly calculated by solving individual equations from the system of algebraic equations, given the measurements defined in ***i***. In short, the value returned by the observability function is:

$$\boldsymbol{Ob}\,(\boldsymbol{i}) = \mathrm{FT}\,(\boldsymbol{i}). \tag{3}$$

The FT algorithm is the basic core of the OAA. The latter also includes other modules with more rigorous analysis tools, whose purpose is to refine the FT results at the expense of much higher computing times. Then, in view of its short run times, the FT constitutes an ideal criterion for an initialization algorithm.

The Merging Approach. The aggregating policy for the construction of the fitness function requires a criterion to reconcile the values of all the individual objectives, judiciously combining them so that none is undervalued. The standard procedure consists in normalizing each of them in the [0,1] range. Therefore, in this paper, the fitness function $\boldsymbol{F}$ was defined in terms of the three normalized objectives as follows:

$$\boldsymbol{F}\ (\boldsymbol{i}) = \boldsymbol{NR}(\boldsymbol{i}) + \boldsymbol{NOb}(\boldsymbol{i}) + 1 - \boldsymbol{NC}(\boldsymbol{i})\ . \tag{4}$$

Our algorithm tends to maximize $\boldsymbol{F}$, its values always lying between 0 and the total number of individual objectives. Equation 4 can be naturally expanded to meet this requirement for a greater number of objectives in the following way:

$$\boldsymbol{F}(\boldsymbol{i}) = \sum_{p=1}^{n} NOM_p + m - \sum_{q=1}^{m} NOm_q\ . \tag{5}$$

where n and m are the number of objectives to be maximized or minimized, respectively, $NOM_p \in [0, 1]$ is the p^{th} normalized objective to be maximized, $NOm_q \in [0, 1]$ is the q^{th} normalized objective to be minimized, and $\boldsymbol{F}(\boldsymbol{i}) \in [0, n+m]$.

The optimal (utopian) situation, i.e. $\boldsymbol{F}(\boldsymbol{i}) = n+m$, occurs when all the objectives to be maximized are equal to 1, while those to be minimized become 0. It should be noted that these features are remarkably advantageous. First of all, the expansion to consider additional objectives is straightforward. Besides, $\boldsymbol{F}$ moves within a closed bounded range of values, thus providing a clear threshold to be reached.

Number of Evaluations. The MOGAI evaluates $\boldsymbol{F}$ only when necessary. Whenever an individual remains unchanged from one generation to the other, its fitness value is not recalculated. Implementing this feature led to 10% savings in the number of evaluations, thus proportionally reducing the execution time of a complete GA run.

5 Experimentation

Brief Description of the Plant Under Analysis: The algorithmic performance was assessed by carrying out the instrumentation analysis of an industrial plant whose main features are described in Bike [14]. The plant produces 1500 ton/day of anhydrous liquid ammonia at 240 K and 450kPa with a minimum purity of 99.5%. The product is obtained by means of the Haber-Bosch process, which consists in a medium-pressure synthesis in a catalytic reactor followed by an absorption procedure that removes the ammonia with water. The liquid output from the absorber enters a distillation column that yields pure ammonia as top product. The plant also contains a sector with membranes, where hydrogen is recovered and then recycled to the feed. The rigorous mathematical model of this plant, used to build the occurrence matrix was generated using the ModGen package [15]. The resulting system contained 557 non-linear algebraic equations and 546 process variables.

The MOGAI Parameters: The population size was fixed in 100 individuals. Crossover probability was set at 0.7. The initial mutation probability was 0.0018 and, as explained above, it was forced to decrease as the algorithm evolved, its value being also combined with the fitness of the individual. The genotype length **N**, which amounted to the total number of process variables, was 546.

Some Industrial Results: Both the feasibility and convenience of using the MOGAI as an initialization tool for structural OAAs were evaluated through a detailed study of the ammonia plant. The most promising classifications obtained from a MOGAI run were analyzed. The results were compared in terms of sensor acquisition costs, reliability of the chosen instrumentation, and level of knowledge about the process obtainable both through direct measurements and estimations carried out from the model equations. With these guidelines, the most convenient initialization for the rigorous OA was selected. A complete OA process was executed next, and the results were compared against the configuration without automatic initialization suggested by Ponzoni et al. [16].

For the first stage, the three solutions whose features are summarized in Table 1 were selected. The letters M, O and I indicate the number of measured, observable and indeterminable variables, respectively. It can be observed that all the fitness values are satisfactory since they are close to 3, which is the upper bound for $\boldsymbol{F}$. In all cases, the reliability of the resulting configuration is greater than 99%. With respect to costs, B is significantly cheaper than A or C. However, in terms of observability, the results indicate that A and C are preferable since they have a lower number of indeterminable variables.

Table 1. Three MOGAI solutions

Config.	Fitness Value	Cost	Observability		
			M	O	I
A	2.538	$25,168	105	286	155
B	2.502	$12,642	92	275	179
C	2.512	$24,343	104	298	144

In order to complete the analysis, it is necessary to determine which indeterminable variables are critical, since their values should be known accurately in the final configuration. Table 2 shows the distribution of the critical variables and the incidence of measuring them in the final cost of the instrumentation. The expression C indicates the increment in the cost associated to the purchase and installation of the sensors that wouldmeasure all the indeterminable critical variables.

Table 2. Details on the critical variables for the solutions in Table 1

Config.	Critical Variables			Cost	
	M	O	I	C.	Final C.
A	7	10	12	$ 3,645	$28,813
B	6	10	13	$ 4,364	$17,006
C	7	10	12	$ 3,134	$27,477

From the number of indeterminable variables present in A and B (see Table 1) one could infer that the best configuration is C. However, from Table 2 it is clear that the number of indeterminable variables of interest is, in fact, similar. Hence, the most significant difference lies in the cost, thus favoring configuration B. If it is assumed that the final configuration should have no undeterminable critical variables, it becomes necessary to introduce in the analysis the costs derived from the addition of sensors to monitor all of them. The minimum cost increment associated to the incorporation of these measurements corresponds to configuration C. However, this is not enough to compensate the original difference in costs. Then, all in all, it can be concluded that B is the most convenient alternative.

Finally, in Table 3 the results obtained after carrying out the rigorous OA initialized with configuration B, are compared against the instrumentation reported in Ponzoni et al. [16] (configuration P). The OAA employed in those experiments was a GS-FLCN implementation [2] with manual initialization.

Table 3. Concluding Results

Config.	Cost	Observability		
		M	O	I
P	$ 14,772	52	257	237
B	$ 17,006	105	289	152

In terms of cost, configuration P seems to be more convenient. However, this choice leaves 9 undeterminable critical variables. If we added sensors at those points and considered the corresponding cost increments, the budget would raise to $ 17,922, thus becoming more expensive than B's. Furthermore, the use of the MOGAI also leads to better knowledge about the process, with a reduction over 40% in the total number of non-observable variables.

The use of the MOGAI as an initialization tool implies gains in both time and effort, also improving the reliability of the results by taking into account a higher number of interest factors. For this industrial case, the total amount of time required by the complete OA procedure was reduced in 83% thanks to the automatic initialization. The number of OA iterations diminished and therefore, there was a decrease in the effort the DM had to make for his analysis. More specifically, for the ammonia synthesis plant, the average run times of the MOGAI in a PC Pentium IV (2.8 GHz) amounted to approximately 150 seconds, while a complete iteration of the OA cycle normally takes more than an hour. This shows that the computational effort invested in making an automatic initialization is negligible in comparison with the order of magnitude of the times required by an OA iteration.

6 Conclusions

In this article we tackled the problem of selecting the best configuration of sensors to instrument an ammonia synthesis plant in order to assess the convenience of applying the MOGAI as an automated tool for initialization purposes. The objective function of the GA contemplates terms associated to cost, reliability and observability.

From the comparative analysis of the results achieved with and without automated initialization, it is possible to conclude that the use of the MOGAI makes the design methodology more efficient. Moreover, the automated initialization leads to results of higher quality by directing the search to the simultaneous fulfillment of several objectives.

Acknowledgments

The authors would like to express their acknowledgment to the "Agencia Nacional de Promoción Científica y Tecnológica" from Argentina, for their economic support given through Grant Nº11-12778. It was awarded to the research project entitled "Procesamiento paralelo distribuido aplicado a ingeniería de procesos" (ANPCYT Res Nº117/2003) as part of the “Programa de Modernización Tecnológica, Contrato de Préstamo BID 1201/OC-AR”.

References

1. Vazquez, G.E., Ferraro S.J., Carballido J.A., Ponzoni I, Sánchez M.C., Brignole N.B.: The Software Architecture of a Decision Support System for Process Plant Instrumentation. WSEAS Transactions on Computers, 4, 2, (2003), 1074-1079.
2. Ponzoni I., Sánchez M.C., Brignole N.B.: A New Structural Algorithm for Observability Classification. Ind. Eng. Chem. Res., 38, 8, (1999), 3027-3035.
3. Ferraro S.J., Ponzoni I, Sánchez M.C., Brignole N.B.: A Symbolic Derivation Approach for Redundancy Analysis. Ind. Eng. Chem. Res., 41, 23, (2002), 5692-5701.
4. Osyczka A.: Multicriterion Optimization in Engineering with FORTRAN Programs. Ellis Horwood Limited, (1984).
5. Toscano Pulido G.: Optimización Multiobjetivo Usando Un Micro Algoritmo Genético. Tesis de Maestría en Inteligencia Artificial, Universidad Veracruzana LANIA, (2001).
6. Fonseca C.M.: Multiobjective Genetic Algorithms with Application to Control Engineering Problems. PhD Thesis, Department of Automatic Control and Systems Engineering University of Sheeld, (1995).
7. Coello Coello C.A.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems, An International Journal, 1, 3, (1999), 269–308.

8. Rosenberg R.S.: Simulation of Genetic Populations with Biochemical Properties. PhD thesis, University of Michigan, Ann Harbor, Michigan, (1967).
9. Eiben A.E., Hinterding R., Michalewicz Z.: Parameter Control in Evolutionary Algorithms IEEE Transactions on Evolutionary Computation, 3, 2, (1999), 124-141.
10. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary Computation: Comments on the History and Current State, IEEE Transactions on Evolutionary Computation, 1, 1, (1997), 3-17.
11. Safe M., Carballido J., Ponzoni I., Brignole N.B.: On Stopping Criteria for Genetic Algorithms. In: Bazzan, A., Labidi, S. (eds.): Advances in Artificial Intelligence SBIA 2004 Lecture Notes in Artificial Intelligence Vol. 3171 Springer-Verlag, Berlin Heidelberg New York (2004) 405–413.
12. Radcliffe N.J.: Equivalence Class Analysis of Genetic Algorithms. Complex Systems 5, (1991).
13. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York, (1996).
14. Bike S.: Design of an Ammonia Synthesis Plant, CACHE Case Study. Department of Chemical Engineering, Carnegie Mellon University, (1985).
15. Vazquez G.E., Ponzoni I., Sánchez M.C., Brignole N.B.: ModGen: A Model Generator for Instrumentation Analysis. Advances in Engineering Software, 32, (2001), 37-48.
16. Ponzoni I., Brignole N.B., Bandoni J.A.: Estudio de Instrumentación para una Planta de Producción de Amoníaco empleando un Nuevo Algoritmo de Clasificación. AADECA´98, Argentina, 1, (1998) 59-64.