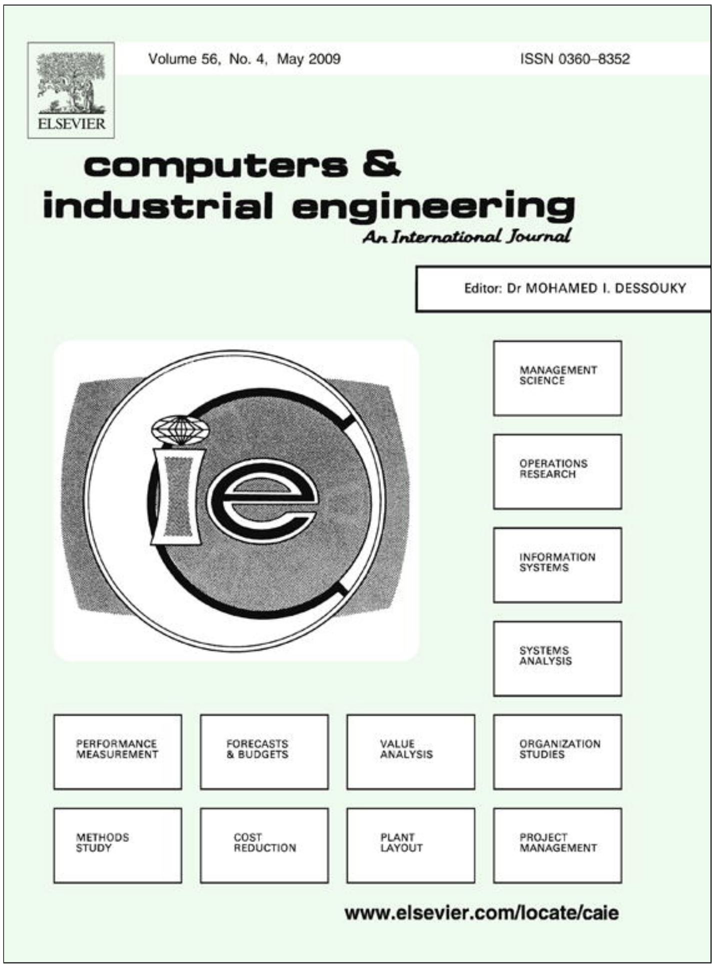
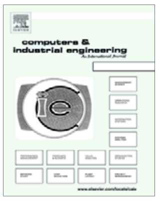

Volume 56, No. 4, May 2009
ISSN 0360-8352
ELSEVIER
computers & industrial engineering
An International Journal
Editor: Dr MOHAMED I. DESSOUKY
MANAGEMENT SCIENCE
OPERATIONS RESEARCH
INFORMATION SYSTEMS
SYSTEMS ANALYSIS
PERFORMANCE MEASUREMENT
FORECASTS & BUDGETS
VALUE ANALYSIS
ORGANIZATION STUDIES
METHODS STUDY
COST REDUCTION
PLANT LAYOUT
PROJECT MANAGEMENT
www.elsevier.com/locate/caie

Computers & Industrial Engineering 56 (2009) 1419–1428

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie



SID-GA: An evolutionary approach for improving observability and redundancy analysis in structural instrumentation design

Jessica A. Carballido [a,*], Ignacio Ponzoni [a,b], Nélida B. Brignole [a,b]

[a] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC), Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca, 8000 Buenos Aires, Argentina
[b] Planta Piloto de Ingeniería Química, CONICET, Bahía Blanca 8000, Argentina

ARTICLE INFO

Article history:
Received 11 December 2007
Received in revised form 28 April 2008
Accepted 3 September 2008
Available online 10 September 2008

Keywords:
Instrumentation design
Genetic algorithms
Combinatorial optimization problem
Decision support system

ABSTRACT

In this paper the core of a genetic algorithm designed to define a sensor network for instrumentation design (ID) is presented. The tool has been incorporated into a decision support system (DSS) that assists the engineer during the ID process. The algorithm satisfactorily deals with non-linear mathematical models, and considers four design objectives, namely observability, cost, reliability and redundancy, exhibiting properties that were either never addressed by existing techniques or partially dealt with in the literature. Its performance was tested by carrying out the ID of an ammonia synthesis industrial plant. Results were statistically analysed. A face validity study on the fitness function's soundness was also assessed by a chemical engineer with insight and expertise in this problem. The technique performed satisfactorily from the point of view of the expert in ID, and therefore it constitutes a significant upgrading for the DSS.

1. Introduction

Instrumentation design (ID) is a complex task that concerns with finding the optimal spatial arrangement of sensors in an industrial plant, with the purpose of obtaining a good knowledge of its operation conditions. This constitutes a crucial activity in the field of process engineering since a properly defined sensor network leads to improvements in the monitoring and safety of the critical industrial processes. In particular, our research group is working on the development of a decision support system (DSS) that aims at helping the engineer in this difficult task (Vazquez et al., 2003). The DSS is comprehended by two main stages, namely observability and redundancy analysis. These phases repeatedly assess the quality of a sensor network according to different criteria, starting from an initial sensor network that is refined and tuned during the whole process.

The specific problem of determining the initial sensor network that starts off the ID process is a crucial task, which has strong impact in the global performance of the DSS (Carballido, 2005). Traditionally, the selection of the initial set of instruments has been entirely carried out by an expert process engineer. The main drawback of this approach is that the results strongly depend on his knowledge. Furthermore, humans are naturally unprepared to deal with either huge amounts of information or many objectives concurrently, thus complicating the task of finding a configuration that represents a good balance between various factors in realistic problems.

In consequence, arose the need of counting with an automated tool that helps the process engineer with the definition of this starting sensor network. As this task falls into the category of *combinatorial multi-objective optimization problems*, where several aspects must be considered simultaneously, evolutionary computation constitutes a promising candidate to be used for its resolution. In particular, regarding the industrial engineering field, several problems have been effectively tackled with genetic algorithms (GAs) (Altiparmak, Mitsuo, Lin, & Turan, 2006; Dietz, Azzaro-Pantel, Pibouleau, & Domenech, 2007).

In this work the details of the implementation of a strategy called SID-GA (Genetic Algorithm for Structural Instrumentation Design) designed to find a good quality initial sensor network is presented. SID-GA constitutes the first evolutionary approach that follows criteria related to improve the two major analysis stages of the instrumentation design problem, observability and redundancy, so as to accomplish a good balance between network safety and costs. Moreover, to our knowledge, this contribution is original since it is also the first algorithm that finds a sensor network from the analysis of a non-linear mathematical model that represents the plant behavior.

The rest of the article is organized as follows. Section 2 contains a detailed introduction to the application. Next, in Section 3, a brief review about genetic algorithms applied to instrumentation design is presented. The fundamentals of our proposal are explained later.

[*] Corresponding author. Tel.: +542914595135; fax: +542914595136.
E-mail addresses: jac@cs.uns.edu.ar (J.A. Carballido), lidecc@cs.uns.edu.ar (I. Ponzoni).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.09.001

Nomenclature

B	bigraph corresponding to M
C	set of B columns
DL	dependency list
DM	direct method
DSS	decision support system
E	set of B edges
${}_{F}SN_{O\text{-}GA}$	SN obtained after a whole ID is started with a $SN_{O\text{-}GA}$
${}_{F}SN_{SID\text{-}GA}$	SN obtained after a whole ID is started with a $SN_{SID\text{-}GA}$
FT	forward triangularization algorithm
GA	genetic algorithm
GS-FLCN	global strategy first least-connected node algorithm
ID	instrumentation design
M	mathematical equation model
$\mathbf{M_M}$	maximum matching of B
OA	observability analysis
O-GA	genetic algorithm for observability analysis
R	set of B rows
RA	redundancy analysis
SID-GA	genetic algorithm for structural instrumentation design
SN	sensor network
$SN_{O\text{-}GA}$	SN yielded by the O-GA
$SN_{SID\text{-}GA}$	SN yielded by the SID-GA

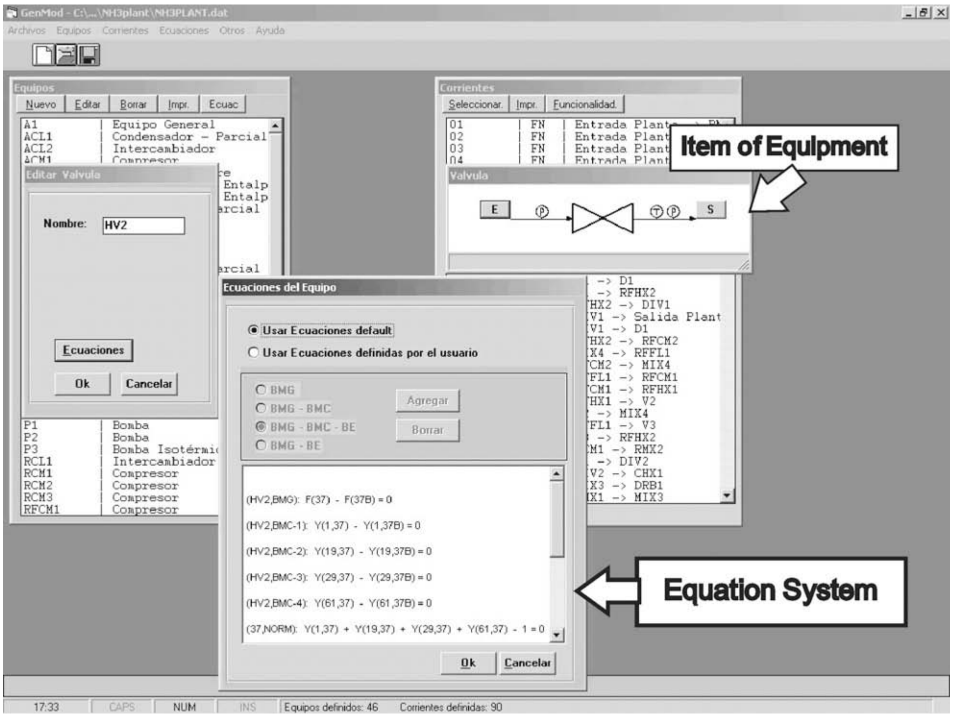
In Section 5 the experimentation and the main results are presented and analyzed. Finally, the main conclusions are summarized in Section 6.

2. The decision support system

The instrumentation design of an industrial plant basically consists in selecting the best sensor location, amount and type so as to attain plenty of information about the plant's functioning. One way of dealing with this problem is based on the analysis of a steady-state mathematical model that represents the behavior of the plant under stationary operating conditions. This is the approach that we have followed during the implementation of the DSS. Thus, the first module included in the DSS is the model generator, called ModGen (Vazquez, Ponzoni, Sanchez, & Brignole, 2001). In general terms, the process engineer specifies to the ModGen the types of items of equipment and streams that comprise the industrial plant under study. With this information, the ModGen generates a system of equations that represent the chemical properties and relationships between those items. Once the mathematical model has been generated, the variables represent the possible sites where sensors might be located. In Fig. 1, an example of the use of the ModGen is shown. As it can be seen, when a piece of equipment is introduced by the process engineer, the respective equations are generated by the ModGen. As more items are added and connected with the existing ones, new equations for those relationships and properties are included in the mathematical system. A more detailed explanation about this software can be found in Vazquez et al. (2001).

2.1. Classification of variables

ID techniques are oriented to discern about the information that will become available starting from a given sensor configuration by means of an analysis of the relationships between the model's variables contained in the non-linear equation system that represents the behavior of the plant. Therefore, once the mathematical model has been generated, the following task consists in defining an initial group of instruments, which splits the whole set of variables into two classes: measured variables, whose values can be obtained through a direct measurement; and unmeasured variables, containing the rest of the variables in the model. After the initial sensor network has been defined, two analysis stages interact in order to refine it.



GenMod - C:\...\NH3plant\NH3PLANT.dat

```
A1          | Equipo General
ACL1        | Condensador - Parcial
ACL2        | Intercambiador
ACM1        | Compresor
P1          | Bomba
P2          | Bomba
P3          | Bomba Isotérmic
RCL1        | Intercambiador
RCM1        | Compresor
RCM2        | Compresor
RCM3        | Compresor
RFCM1       | Compresor
```

Editar Valvula — Nombre: HV2

```
01    | FN  | Entrada Plant
02    | FN  | Entrada Plant
03    | FN  | Entrada Plant
04    | FN  | Entrada Plant
```

Ecuaciones del Equipo

● Usar Ecuaciones default
○ Usar Ecuaciones definidas por el usuario

○ BMG
○ BMG - BMC
● BMG - BMC - BE
○ BMG - BE

```
(HV2,BMG):  F(37) - F(37B) = 0
(HV2,BMC-1):  Y(1,37) - Y(1,37B) = 0
(HV2,BMC-2):  Y(19,37) - Y(19,37B) = 0
(HV2,BMC-3):  Y(29,37) - Y(29,37B) = 0
(HV2,BMC-4):  Y(61,37) - Y(61,37B) = 0
(37,NORM):  Y(1,37) + Y(19,37) + Y(29,37) + Y(61,37) - 1 = 0
```

17:33 | CAPS | NUM | INS | Equipos definidos: 46 | Corrientes definidas: 90

Fig. 1. Capture of a ModGen window, after the generation of the mathematical model that corresponds to a valve.

The first one, called observability analysis (OA) consists in sorting unmeasured variables into observable and unobservable, where an observable variable is one whose value can be calculated from the model's equations considering the measured variables as constant values. In other words, the OA looks forward to determining the amount of information about the process that can be achieved from a particular sensor configuration.

On the other side, the redundancy analysis (RA) stage deals with finding out which measurements are redundant, i.e., those variables that appear as measured but that can also be deduced from the rest of the measurements. In short, the RA aims at detecting those sensors which can be removed without reducing the degree of knowledge available about the behavior of the industrial process.

Various approaches are presented in the literature to carry out both stages (Crowe, 1989; Joris & Kalitventzeff, 1987; Kretsovalis & Mah, 1988a; Kretsovalis & Mah, 1988b; Meyer, Koehret, & Enjalbert, 1993). For our research work, a structural methodology was used since it is more robust and applicable than the other existing algorithms (Ponzoni, Sánchez, & Brignole, 1999). In this context, a methodology is structural when it is focused in the structure of the binary matrix underlying the mathematical model. In particular, two alternative methods were developed to accomplish the OA for the DSS, to be precise GS-FLCN (Ponzoni et al., 1999) and Direct Method (Ponzoni, Sánchez, & Brignole, 2004). The first one uses combinatorial searches to classify unmeasured variables, while the second one employs graph-decomposition algorithms to identify the set of observable variables. Concerning the RA, it can be carried out by applying the method presented by Ferraro, Ponzoni, Sánchez, and Brignole (2002), which proposes a practical procedure of symbolic derivation with chain rules to determine redundant measurements.

2.2. Impact of the initial sensor network quality on the analysis stages

As it was mentioned in the former section, the analysis stages start from an initial sensor network whose quality strongly impacts on the efficiency and quality of the resulting sensor network. In particular, for the OA, the algorithms classify the unmeasured variables through a procedure of incremental refinement. The engineer performs a detailed analysis of an intermediate solution yielded by the OA algorithm. Grounded on this study, he decides whether some of the measurements need to be added or removed. When this is the case, a new execution of the OA algorithm is carried out starting from the new configuration. Thus, the analysis is repeated, iterating this process until a satisfactory sensor network is achieved. A sensor network is satisfactory when it fulfills the process engineer's purposes. Therefore, the final instrument configuration yielded by the OA phase is a consequence of several executions of the OA algorithm together with the corresponding expert analysis. Each iteration is expensive in terms of time and effort demanded to the expert, as it can be estimated that in average it requires an hour of working time for a medium-size industrial plant (500 variables approximately). In this context, an appropriate initialization grants a strong reduction on the total amount of iterations required for the OA phase, since a properly chosen configuration redounds to less refinement work to reach a satisfactory classification.

On the other side, the RA detects redundant measurements by symbolically deriving the equations that are associated to measured variables, where observable variables are replaced by mathematical expressions as functions of the measurements. This process is accomplished by solving those equations symbolically and then substituting the observable variables into the redundant equations, canceling terms whenever possible (Ferraro et al., 2002). Nevertheless, this strategy losses effectiveness when the model contains numerous non-linear equations that cannot be symbolically solved. Moreover, as the problem size increases, the amount and complexity of the nesting, caused by the successive substitutions, becomes prohibitively expensive in terms of both memory storage and run times, thus making it practically impossible to obtain many final symbolic expressions. In view of these limitations arose the need of relying on an initial sensor network that tended to reduce the total amount of non-linear terms for the equations analyzed by the RA phase.

In Fig. 2, we show a parallel between the modules of the DSS and the manner in which they affect the variables of the mathematical model yielded by the ModGen. As it can be seen, the first variable classification occurs when the initialization module defines a sensor network, since a sensor network determines which

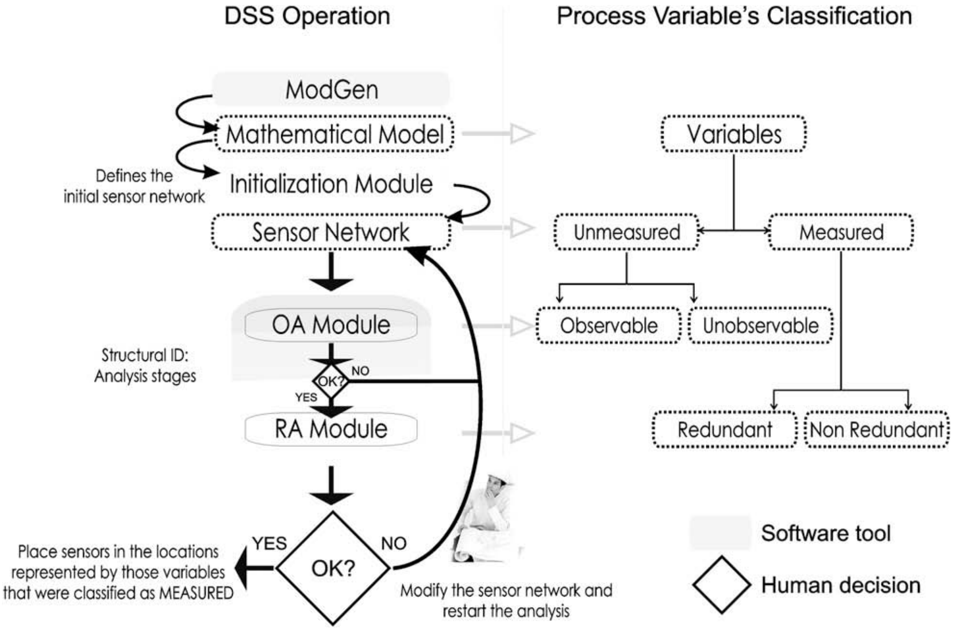

DSS Operation
Process Variable's Classification
ModGen
Mathematical Model
Defines the initial sensor network
Initialization Module
Sensor Network
OA Module
Structural ID: Analysis stages
OK?
NO
YES
RA Module
Place sensors in the locations represented by those variables that were classified as MEASURED
YES
OK?
NO
Modify the sensor network and restart the analysis
Variables
Unmeasured
Measured
Observable
Unobservable
Redundant
Non Redundant
Software tool
Human decision

Fig. 2. Parallel between main stages of the DSS, and the categorization of the model's variables.

variables will be named as measured. For example, if the sensor network establishes that an instrument should be located so as to measure the pressure of a given stream, then the variable that represents the pressure of that stream in the mathematical model is said to be measured since it can be assumed that its value will be known, i.e., it can be thought as a constant value. Later, with the execution of the OA stage, the variables that were not cataloged as measured are split into observable and unobservable. This means that the module decides which unmeasured variables can be calculated from the equation system, regarding the measured variables as constant values. As it was aforementioned, this is an iterative procedure since the process engineer decides whether he needs to change the sensor network after each run of the OA algorithm. Therefore, it may happen that he adds or removes an instrument of the sensor network, thus producing a new variable categorization into measured and unmeasured.

As soon as the OA yields a satisfactory result, the RA phase takes place. The execution of this module yields to the organization of measured variables as redundant or non-redundant, since it decides which instruments could be removed without changing the amount and quality of information obtained from the sensor network. Since a certain degree of redundancy is expected for safety reasons, the process engineer must decide whether the redundancy degree revealed by the sensor network is satisfactory. For a second time, this procedure may cause that the engineer does not feel the sensor network is good enough and then he may decide to add or remove some instruments, consequently changing the sensor network and restarting the whole analysis, hence resulting in a different variable classification.

As it can be perceived, the ID is not a trivial task as it comprises several hours of execution of the structural algorithms together with hours of human analysis. For this reason, it is very important that the initial sensor network, generated by the initialization module, is defined considering objectives that tend to minimize the amount of iterations of each analysis module and as a consequence of that, the human effort will also be highly reduced.

3. Evolutionary approaches for the definition of sensor networks

The selection of sensors has traditionally been carried out by the engineer in charge of the plant, who based his decisions on his own criteria, expertise and skill. However, as the process complexity grows, a human criterion becomes less reliable since a very careful analysis of huge amounts of information has to be accomplished. Moreover, several objectives and constraints need to be simultaneously taken into account. Hence, the necessity for the automation is clear giving us the idea of designing a tool to assist the engineer.

The multi-objective combinatorial nature of this problem, jointly with the complexity and size of the mathematical models that usually represent accurately industrial processes, gave us strong reasons to tackle the problem of defining an initial sensor network with evolutionary computation. In particular, it is well known that GAs are particularly suitable to solve multi-objective problems because they simultaneously deal with several possible solutions. In this sense, the first step consisted in making a review on the existing GAs for applications that were similar to our problem. In respect to the ID case, we found only two research groups tackling it with evolutionary computation.

Sen, Narasimhan, and Deb (1998) developed an evolutionary algorithm on the basis of graph theory for ID. It works on data for a linear model of the process and is implemented with a single-objective approach. However, it is used alternatively to fulfill cost, reliability and precision goals. Reliability is calculated as the sensor's malfunctioning probability, and precision corresponds to potential errors on the measurement procedure. They do not explain how they compute the sensor network's cost. Individuals are represented by graphs, defining crossover and mutation operators according to graph theory. Initial population is randomly generated, tournament is used to select the parent pool, and evolution always stops when 10 generations are reached. The problem tackled in the article contains 28 variables. As the objectives are dealt with independently, the results are reported for each one of them separately.

On the other hand, Carnero, Hernández, Sánchez, and Bandoni (2001) studied the problem of designing non-redundant sensor networks for non-linear systems by means of genetic algorithms, considering simultaneously two objectives related to the reliability and cost of the network. They represent process operation on steady state with a set of linear equations. Individuals are fixed-size binary strings, and the recombination operators are redefined in accordance with linear algebra. Roulette wheel is used as the selection method, and the number of generations is fixed in 300. The approach followed to deal with two objectives is the one presented by Srinivas and Deb (1995) named NSGA (non-dominated sorting genetic algorithm). The study case is the same as the one studied by Sen et al. (1998), i.e., a 28-variable linear model.

As to our requirements, these algorithms regrettably exhibit statements that limit their application to our problem instance. Firstly, both techniques base their studies on *linearized* models of the process. In our context, this would imply a dramatic simplification from the original model, which would unnecessarily reduce our methodology's scope, since the methods we use for OA and RA are applicable to non-linear models.

In addition, both articles present extremely basic versions of genetic algorithms. They use a fixed number of generations even though it has been proved this is not an efficacious convergence policy (Safe, Carballido, Ponzoni, & Brignole, 2004). They do not explain clearly the fundamentals, i.e., neither the foundations on the parameters they use, nor the reason why they do not implement adaptive parameters. This constitutes an important drawback as it has been stated in Eiben, Hinterding, and Michalewicz (1999). Moreover, the problem they chose to assess their techniques' performance is too small, the models consisting in 28 variables. Therefore, it would be overbold to try to employ these methodologies straightaway to big engineering problems, like our case whose model is huge thus aspiring to be realistic. Besides, small size problems may be tackled with any other optimization technique, and the use of a heuristic method is not well justified.

For this reason, in Carballido, Ponzoni, and Brignole (2005) a basic prototype for a GA is presented, which aims at finding a sensor network with cost, reliability and observability factors as simultaneous objectives considered by means of an aggregating approach. Individuals are binary strings, crossover and mutation are performed in the traditional way, roulette wheel selection is used and a genotypic convergence criterion stops the algorithm.

Nonetheless, the main drawback of this implementation consists in that it completely ignores the RA, which constitutes the second main stage of the ID process. Yet, it was a version that succeeded at improving the OA phase. As it was necessary to count with an algorithm that integrally deals with objectives related to both structural procedures, a new GA was developed. The SID-GA proposed in this article represents a successor of the former method (Carballido et al., 2005). The most striking amendment was the inclusion of a new objective in its fitness function so as to overcome the limitations above explained, i.e., to consider the RA phase. Moreover, the method was also improved by redefining the implementation of the term that directly impacts the OA. Then, SID-GA represents the first integrative approach that considers all main aspects of the instrumentation design task, more precisely, it takes into account the two main existing structural procedures for ID, the observability and the redundancy analysis.

4. SID-GA: A genetic algorithm for the initialization of instrumentation design

4.1. Chromosomal codification

The decision about the installation of sensors in a process plant to monitor its behavior can be translated into stating which variables in the mathematical model will to become constant values through measurements. Therefore, the most natural representation for the individuals is the binary codification (see Fig. 3). According to this representation, each chromosome defines a possible sensor network by assuming that every gene position represents a variable in the mathematical system. A 1-gene indicates that the variable is to be measured – i.e., an instrument should be placed in the plant's location associated to that variable thus becoming constant –, while a 0-gene denotes an unmeasured variable. Therefore, the chromosomal length is equal to the number of variables in the mathematical model.

4.2. Fitness function

The main purpose for the algorithm presented here was to design a tool oriented to enhance redundancy and observability analysis. Moreover, cost and reliability issues were also considered. According to the classification proposed by Coello Coello, Lamont, and Van Veldhuizen (2007), there are three main approaches that can be used to tackle problems with several objectives (see Fig. 4). In particular, our methodology is framed into the aggregation or *scalarization* approach. More specifically, we implemented the fitness function as a linear combination of the four objectives. At this point it is important to mention that, whereas nowadays the trend is to use Pareto dominancy techniques, there are certain problems, particularly *multi-objective* combinatorial optimization problems, in which aggregating functions can provide very good approximations of the Pareto optimal set, even outperforming Pareto-based approaches (Coello Coello et al., 2007, p. 70). Moreover, as stated in Deb (2001, p. 50), faced with more than two objectives as in our case, the aggregation method is the most convenient. Finally, it is important to mention that all the objectives were equally weighted. This design decision was taken since every process engineer had different ideas on how to weight each objective.

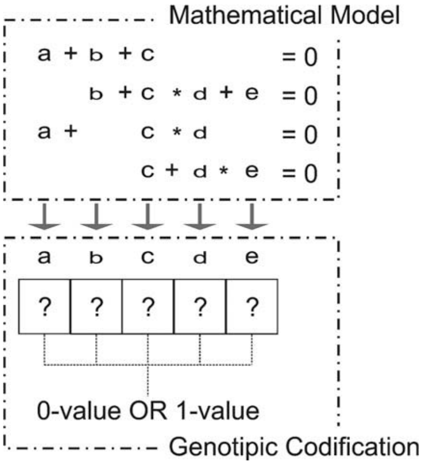


Fig. 3. Chromosomal codification used by SID-GA.

Then, following the *scalarization* approach, as soon as the four objectives are calculated in the manner that will be explained later, their values are normalized and combined into a single function, as shown in Eq. (1),

$$\boldsymbol{F}(\boldsymbol{i}) = \boldsymbol{NR}(\boldsymbol{i}) + \boldsymbol{NObs}(\boldsymbol{i}) + 2 - \boldsymbol{NC}(\boldsymbol{i}) - \boldsymbol{NRed}(\boldsymbol{i}) \tag{1}$$

where $\boldsymbol{NR}(\boldsymbol{i})$, $\boldsymbol{NObs}(\boldsymbol{i})$, $\boldsymbol{NC}(\boldsymbol{i})$ and $\boldsymbol{NRed}(\boldsymbol{i})$ are the normalized values corresponding to the reliability, observability, cost and redundancy terms, respectively. Conceptually, the reliability term gives an idea of the sensor network failure probability; the observability term gives an approximation of the amount of variables that can be obtained form the mathematical model considering the measured variables as constant values; the cost term represents the purchase and installation prices; and the redundancy term gives an estimation of the amount of linear terms in the mathematical model that will be used in the RA. Reliability and observability are maximized, and the other two objectives are minimized. The whole function is maximized. The shape of this fitness function gives us the possibility to have a range of fitness values varying from 0 to 4 to measure the goodness of a solution, having $\boldsymbol{F}(\boldsymbol{i}) = 4$ as the best situation. Moreover, it can be straightforwardly extended to Eq. (2), for the case when more objectives are pursued,

$$\boldsymbol{F}(\boldsymbol{i}) = \sum_{p=1}^{n} NOM_p + m - \sum_{q=1}^{m} NOm_q \tag{2}$$

where n and m are the amounts of objectives to be maximized and minimized, respectively, $\mathrm{NOM}_p \in [0\ldots1]$ is the pth normalized objective to be maximized, NOM_q is the qth objective to be minimized, and $F(i)$ ranks in the interval $[0\ldots n + m]$. The optimum

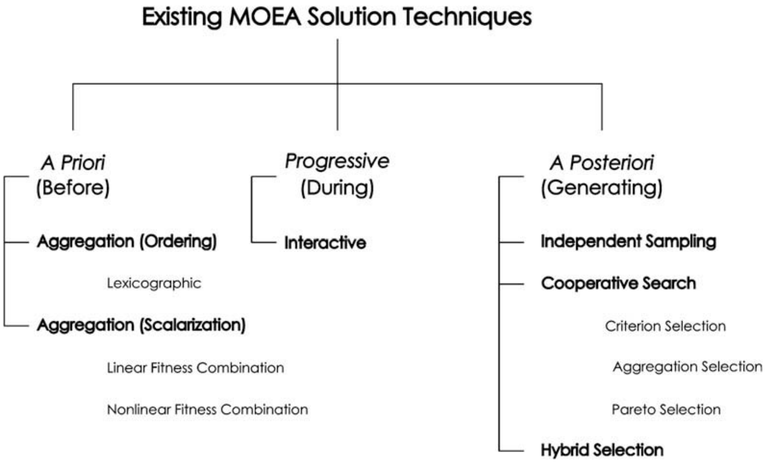


Fig. 4. Multi-objective evolutionary algorithms' taxonomy based on Fig. 1.18 presented by Coello Coello et al. (2007, p. 55).

situation arises when $\boldsymbol{F(i)}$ becomes equal to $n + m$, i.e., all the terms for the objectives to be maximized return a 1-value, and the ones to be minimized return a 0-value.

Below we shall give a detailed explanation of the mode in which each of these objectives is calculated. It is important to take into consideration that for the observability and redundancy objectives, it is not possible to present an explicit mathematical formulation. For this reason, the accomplishment of these objectives will be described in a pseudo-algorithmic manner.

4.2.1. The observability analysis term

The OA algorithms are rigorous and computationally expensive. Thus, it is not reasonable to use them for computing the term of a fitness function. Therefore, an approximation on the amount of observable variables should be obtained through some high-speed procedure. Moreover, as the GA constitutes an initialization tool for the classification of variables, a good estimation of the observability degree suffices.

As it was introduced in Section 2.1, two structural algorithms were developed by this research group to perform the OA. The first one, GS-FLCN, is a combinatorial routine that classifies unmeasured variables by means of matrix rearrangements (Ponzoni et al., 1999). The second approach, called Direct Method (DM), is founded on graph theory (Ponzoni et al., 2004) and is not at all combinatorial. In Carballido et al. (2005), the observability term was calculated using the procedure called forward triangularization (FT) that constitutes the first step in the GS-FLCN. The FT operation estimates the number of variables that can be directly calculated from a sensor configuration, returning a lower bound on the potential quantity of observable variables.

In this article, the term designed to look for a configuration that maximizes the observability degree was built obeying the same philosophy of the procedures that comprise the OA module implemented with the DM. The gist of this policy is that the GA analyzes a given configuration's observability degree based on a simplification of the DM, capturing its preliminary fast stages. More specifically, the observability term is focused on the two first steps of the DM: Bigraph Construction and Maximum Matching Finding. In Fig. 5, the computation of the OA term is schematized. The first step consists in mapping the mathematical equation model **M** into a bigraph **B(R, C, E)**. **B** is conformed by two sets of nodes **R** and **C**, which represent the matrix rows and columns, respectively. The edges included in the set **E** link nodes of **R** with nodes of **C**. There is an edge between $\mathbf{r_i}$ y $\mathbf{c_j}$ if and only if the ith row of **M** contains a non-zero value in the jth column of **M**. The second step explores **B** in order to find a maximum matching $\mathbf{M_M}$, which is a set of edges of **B** of maximum cardinality, such as no one pair of $\mathbf{M_M}$ has nodes in common. The cardinality of $\mathbf{M_M}$ constitutes an upper bound estimation of the maximum amount of variables that can become observable from a given sensor configuration (Ponzoni, 2001). This upper bound estimation used in SID-GA is more accurate than the lower bound estimation in its ancestor Carballido et al. (2005).

More details about the implementation of the routines for Bigraph Construction and Maximum Matching Finding used in the DM can be found in Ponzoni et al. (2004).

4.2.2. The redundancy analysis term

The OA algorithm employed in the DSS classifies the unmeasured variables by means of a decomposition procedure of the mathematical model into smaller subsystems. This system's decomposition is not unique, and different decompositions usually yield to subsystems with different degrees of non-linearity (Ponzoni, 2001).

In other hand, the RA algorithm classifies measured variables into redundant and non-redundant by means of a hard-computing procedure. The complexity of this task is strongly associated to the non-linearity level of the equations' subsystem that corresponds to the measured and observable variables (Ferraro et al., 2002). Therefore, the objective computed by the RA term aims at reducing the non-linearity degree of the mathematical system by preferring those individuals that represent a configuration that yields the least amount of non-linear terms.

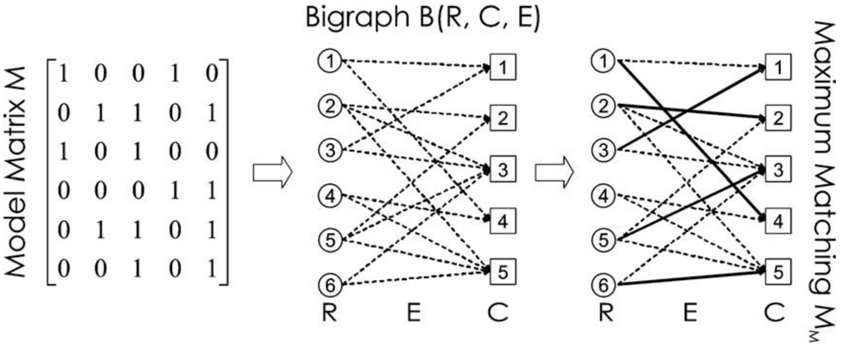
To calculate the RA term, it is necessary to count with information about the variables' dependencies. This information is obtained when the SID-GA begins by a procedure that defines, for every variable v in the model, all the variables that should be measured to make linear the terms where v appears. The output of this process is a dependency list (DL), which is later used as extra input every time the fitness function is calculated for a given individual.

For example, suppose we have the following mathematical system:

$$F(a,b)+c+F(a,d)=0$$
$$e+G(a,d)=0$$
$$F(a,d)+h=0$$

Assuming that F and G are non-linear functions that become linear if one of the parameters becomes a constant value, the DL would say that: ($a \rightarrow b, d$; $b \rightarrow a$; $c \rightarrow$ -; $d \rightarrow a$; $e \rightarrow$ -; $h \rightarrow$ -). This means that variable a appears in a non-linear term with variable b, and with another non-linear term with variable d. Then, the best situation in this scenario would be to place a sensor so as to convert variable a into a constant value since, if we do so, this simple academic example becomes linear.

Then, pursuing this idea, every potential sensor network is analyzed according to the following simplified pseudo-algorithm:



$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Fig. 5. OA term estimation computed for the fitness function of SIG-DA. The row nodes **R** and column nodes **C** are denoted with circles and squares, respectively. The edges **E** are drawn with dashed lines. The resulting maximum matching $\mathbf{M_M}$, depicted with solid lines, has cardinality 5, and it is conformed by the following edges: (1, 4), (2, 2), (3, 1), (5, 3) and (6, 5).

```
For every unmeasured variable $v_i$ in the individual
  For every variable $v_j$ ($j \neq i$) that depends on $v_i$ according to the DL
    If $v_j$ is measured
      then Add 0 (the optimal situation)
      else Add CONSTANT (the worst situation)
```

4.2.3. The cost and reliability terms

The terms oriented to find a sensor network with minimum cost and maximum reliability (minimum error) are calculated directly from two vectors that, respectively, store information about purchase prices and sensor errors, the latter ranging between 0 and 1. They are calculated as a linear function that simply adds every value of the corresponding vector stored in a position where there is a non-zero in the individual, as shown in Eqs. (3) and (4):

$$\boldsymbol{C}(\boldsymbol{i}) = \sum_{j=1}^{N} \boldsymbol{cv}[j]^{*}\boldsymbol{i}[j]. \tag{3}$$

$$\boldsymbol{R}(\boldsymbol{i}) = \sum_{k=1}^{n} (\boldsymbol{rv}[k]^{*}\boldsymbol{i}[k]). \tag{4}$$

where $\boldsymbol{cv}$ and $\boldsymbol{rv}$ are the cost and reliability vectors, respectively.

It is important to mention that in order to obtain realistic values, several manuals and magazines were consulted, and finally the information published on the magazine *Cole-Parmer Int.* was picked.

4.3. Standard algorithmic features

Crossover is performed in the traditional manner with one cut-point. This is particularly suitable for our problem instance since the performance of the algorithm depends on the order that variables have been arranged in the representation. It is more likely to keep together genes that are near each other (in the neighborhood) and it can never keep together genes from opposite ends of the string. This is advantageous because variables that are nearby in the codification have in general a stronger relationship in the equation system than those at opposite positions. Thus, with one-point crossover we are trying to keep together those variables that are more likely to form an assignment subset. As regards population, a fixed-size population was used. Finally, an elitist policy was adopted.

4.4. Novel algorithmic features

Convergence is controlled by an ad hoc-implemented genotypic approach, which follows the recommendations on stopping criteria suggested in Safe et al. (2004). This approach considers that no more generations are needed when it determines that the genotypic information in the population will not yield to improvements in further evolution. Mutation is implemented as a deterministic and adaptive operator (Bäck, Hammel, & Schwefel, 1997), based on the bit-flip approach. Its initial probability value is defined as 1/l, where l is the length of the chromosome (i.e., the total # of variables), and it is reduced as the evolution proceeds according to a numerical value defined at the beginning of the algorithm. Besides, its adaptive aspect arises when it is applied: the fitter the individual to be mutated, the lower the probability of mutation becomes.

4.5. Unfeasible individuals' treatment

A constraint for this problem is that some positions in the chromosome represent variables that stand for locations in the plant that cannot be physically measured. For this reason, the only allele allowed for those positions is a zero. This chromosomal feature is achieved by generating the individuals in the initial population with a 0-value in those positions, and by restricting the mutation operator from changing them.

It is essential to mention that we used 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 should not be erased from the chromosome since they simplify the calculation of both observability and redundancy terms.

4.6. The selection method

The implementation presented in Carballido et al. (2005) of the GA used roulette wheel selection. For the SID-GA version 2-tournament selection was implemented, being this method the most efficient and efficacious obtained after a comparative study of the algorithmic performance among roulette wheel, tournament and ranking selection.

5. Experimentation

The algorithmic performance was assessed for the ID of an industrial ammonia synthesis plant, and its behavior was analyzed by a fair comparison with the application its ancestor's to the same study case. From now on, for the sake of simplicity, the previous GA will be named O-GA as it only takes into account the Observability phase, while the new implementation (the SID-GA) also considers the redundancy stage, i.e., the SID-GA contemplates the whole structural analysis for instrumentation design. The main parameters we used were: population size = 100; crossover probability = 0.7 and initial mutation probability = 0.0018 (1/546, 546 being the amount of variables in the mathematical model).

5.1. Industrial example: An ammonia synthesis plant

The industrial plant chosen as a realistic example to assess the performance of this software component was designed by Bike (1985) to produce 1500 ton/day of ammonia at 240 K and 450 KPa. A minimum purity of 99.5% can be obtained with this process by means of the Haber–Bosch process, which consists in producing ammonia through synthesis at medium pressure, prior to its recovery by absorption with water. The hydrogen and nitrogen feed stocks are provided by a coal gasification facility, and also contain inerts, like argon and methane. The kinetics of the ammonia synthesis reaction over a double-promoted iron catalyst can be described on the basis of the following rate equation: $N_2 + 3H_2 \Longleftrightarrow 2NH_3$. The catalytic reactor's product is fed to a flash whose gas output stream is the feed of the absorber, while its liquid output stream feeds a distillation column that yields ammonia at specification conditions as its top product. There is also a recovery section, composed of membranes, whose output is recycled. A complete scheme of the plant is shown in Fig. 6. The mathematical model that represents the plant, obtained by ModGen, consists of 546 equations and 557 variables.

5.2. Performance evaluation

Experimental research was conducted in two stages. A hundred runs were executed for each algorithm: O-GA and SID-GA. The best four fitness configurations obtained by each implementation were selected. Later, an expert in ID was consulted to pick the sensor network from each group that constituted the best initialization, according to his skill and experience. In this way, by a combination of the automated tool and the expert's knowledge, two sensor

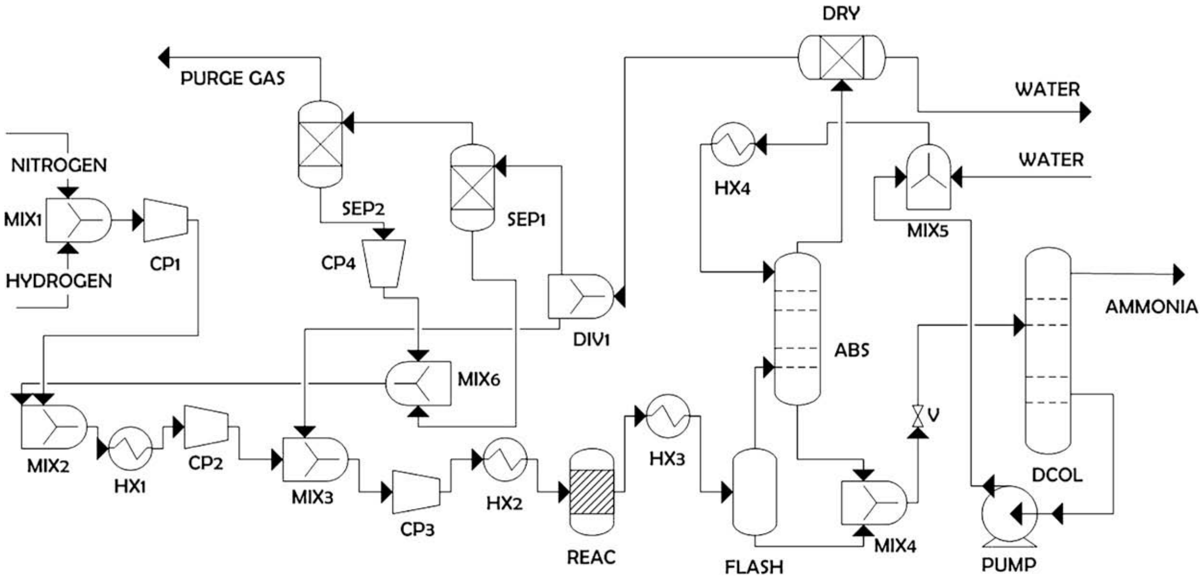


Fig. 6. Ammonia synthesis plant's flowsheet.

networks (*SN*) were selected. One yielded by the O-GA, and the other one was suggested by the SID-GA. We will name them $SN_{\text{O-GA}}$ and $SN_{\text{SID-GA}}$, respectively. In a 2nd phase, the engineer instrumented the process using $SN_{\text{O-GA}}$ completing the whole analysis, and later repeated the same procedure starting with $SN_{\text{SID-GA}}$.

In Table 1 the average results attained for 100 runs by both implementations in the first stage are summarized. The three top rows point out the values obtained for the terms corresponding to observability, cost and error, respectively. The value obtained for the RA term is not reported in this table since this feature was only considered by the SID-GA approach.

From the analysis of the reported values arises that O-GA performs better than SID-GA for the three objectives. This was an expected behavior since the incorporation of a new objective in SID-GA inevitably worsens the rest of the factors. However, it must be noticed that both levels of observability and error stay close to those in O-GA, being the cost the only term that exhibits a more significant relative variation. On the other hand, rows 4, 5 and 6 have to be evaluated together. In row 4 the average generation where the algorithm converged is reported, while in the last row the time in seconds required by the algorithm to reach that generation is stated. As it can be observed in row 6, there is a huge reduction in the time required by SID-GA for each generation, which derives from the change in the method applied to calculate the OA term. The reduction in execution time achieved by SID-GA, compared with O-GA, amounts to approximately 1/3 (row 6). However, SID-GA needs a higher number of generations to converge due to the inclusion of the new objective.

Table 1
First stage: O-GA and SID-GA avg. results

		O-GA	SID-GA
1	Observability degree	80%	76%
2	Cost (USD)	88,32	117,17
3	Error	0.095	0.104
4	Convergence Gen.	50	170
5	Total Exec. Time (s)	20.301	21.702
6	Time by Gen. (s)	0.406	0.128

For the second stage of the experimentation, the process engineer selected the best *SN* obtained by O-GA and SID-GA. Then, following the methodology previously described in Fig. 2, two complete ID were carried out using both *SN*s as initial sensor networks, assuming alternatively that the O-GA and SID-GA were the initialization modules. After these analyses were completed, two final sensor networks were obtained ($_{F}SN_{\text{O-GA}}$ and $_{F}SN_{\text{SID-GA}}$). The values corresponding to our four objectives were calculated for those resulting sensor networks (Table 2). In column 2, #*Non-linear* shows the amount of non-linear terms that appear in the equation system assuming as constant values those variables that are measured in the $_{F}SN$s. This value exhibits that $SN_{\text{SID-GA}}$ achieves a meaningful reduction in the total number of non-linear terms. Moreover, analyzing the sub-columns #*Total* and *Perc*., it can be appraised that the decrease represents a 12% of the proportional number of non-linear terms. This rebounded in an important benefit for the RA since a mathematical model with a smaller degree of non-linearity strongly reduced its computational effort. This confirms the gain with the incorporation of the fourth objective.

If the configurations are analyzed according to the observability degree, it can be perceived that the amounts of variables to be measured (*Meas*. in Table 2) are very similar, while the set of obser-

Table 2
Second stage: results after a complete ID was initialized with two *SN*s

1 Final *SN*s	2 Terms			3 Observability			4 Cost	5 Error
	# Non-linear	# Total	Perc.	Meas.	Obs.	%Obs.		
$_{F}SN_{\text{OGA}}$	239	673	35%	105	359	85%	$ 23,800	0.0965
$_{F}SN_{\text{SID-GA}}$	184	779	23%	102	311	76%	$ 27,200	0.0969

Table 3
Face validity stage

SN N°	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Fitness	2.82	2.80	2.78	2.77	2.75	2.73	2.71
1st Expert's ranking	1	2	4	5	3	7	6
2nd Expert's ranking	2	1	3	4	5	6	7

vable variables attained by means of the $_FSN_{OGA}$ is superior in a 9%. Finally, the difference in cost favors the configuration yielded by the O-GA, while the sensor error's level is approximately the same in both cases.

A global analysis of the results illustrated in Table 2, reveals that the incorporation of the RA objective diminishes the degree of non-linearity of the equation system. Thus, the main objective of this work has been fulfilled. However, as it was to be expected, this gain went to the detriment of the remaining objectives.

5.3. Face validity

Additionally to the results presented in the former section, a face validity process was effectuated where two experts were asked to rank between seven *SN*s, randomly chosen from the 100 suggested by the SID-GA in phase 1. The experts counted with the specification of the industrial process, without knowing anything about the fitness values associated to none of the *SN*s. The objective was to assess the fitness function's fairness with respect to the expert's criterion.

In Table 3 the sensor networks are ordered according to their fitness values. The numbers in the rows *a* and *b* illustrate the position in which the experts located each *SN* in their rankings. As it can be perceived, the arrangements established by the experts are very close to the one determined by the fitness function, existing variations only between those positions that correspond to vastly adjacent fitness values. This gives a subjective but significant idea that the fitness function designed for the SID-GA properly qualifies the relative goodness between the individuals, thus substantiating the use of the aggregating approach.

When the engineers were asked about how they applied their criteria to decide the rankings, it became evident as well that the different objectives were dissimilarly weighted as their decisions were made. Hence, as future work, it is planned to incorporate variable weights to the different terms so as to allow the engineer to define priorities between the goals as *a priori* defined parameters.

6. Conclusions

In this article a multi-objective GA called SID-GA, designed to enhance the quality and availability of process knowledge, is presented. The SID-GA constitutes the only existing implementation, at least from those reported in the literature, which initializes the structural instrumentation design procedures simultaneously considering several objectives. As well, it exhibits novel features related to parameters and termination criteria. In particular, the most important contribution of this original method is the incorporation of specific objectives to improve the two main ID stages: observability and redundancy analysis. Moreover, the SID-GA constitutes the first technique that attains a sensor network from *non-linear* mathematical models.

The performance of the algorithm was assessed by means of the instrumentation of an ammonia synthesis plant. The results were compared against those obtained with an ancestor technique, which was designed for the same purpose. This previous method succeeded in overcoming human's limitations, but that did not consider one of the most important stages of the ID procedure, the RA, as an objective.

The results achieved by the new algorithm revealed that the proposed method accomplished an important reduction in the non-linearity degree of the mathematical system related with the RA phase, thus committing with the main objective for which it had been designed. Likewise, satisfactory results were obtained with respect to the other objectives contemplated by the fitness function. Furthermore, it proved to be better than its ancestor in terms of run times.

Besides, the effectiveness of the fitness function in evaluating the quality of a solution was tested through a face validity process carried out with two experts. This experiment confirmed that the aggregating approach in the fitness function turns out to be suitable to integrate the different objectives for our problem instance.

As an interesting topic for future research, weights for the terms in the fitness function will to be included as fixed start-up parameters to allow the engineer to establish his own priorities. The setting of these weights will be defined by each engineer before an ID begins. Finally, an analysis with experts to define new factors to be considered will be performed as a detailed tailor-made extension of our method's scope.

Acknowledgments

Authors acknowledge the ANPCyT from Argentina, for Grant N° 1652, and SeCyT (UNS), for Grants PGI 24/N019, PGI 24/ZN15, PGI 24/ZN16.

References

Altiparmak, F., Mitsuo, G., Lin, L., & Turan, P. (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers & Industrial Engineering, 51*, 197–216.

Bäck, T., Hammel, U., & Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation, 1*(1), 3–17.

Bike, S. (1985). Design of an ammonia synthesis Plant. Dep. Chem. Engng, Carnegie-Mellon University.

Carballido, J. A. (2005). Diseño de Algoritmos Evolutivos para Localización Inicial Optima de Sensores en Plantas Industriales. Ph.D. thesis. Argentina, Bahía Blanca: Universidad Nacional del Sur.

Carballido, J. A., Ponzoni, I., & Brignole, N. B. (2005). A novel application of evolutionary computing in process systems engineering. In G. R. Raidl, & J. Gottlieb (Eds.), *Evolutionary computation in combinatorial optimization, Lecture notes in computer science* (Vol. 3448, pp. 12–22). Berlin Heidelberg: Springer-Verlag.

Carnero, M., Hernández, J., Sánchez, M., & Bandoni, A. (2001). An evolutionary approach for the design of nonredundant sensor networks. *Industrial & Engineering Chemical Research, 40*, 5578–5584.

Coello Coello, C. A., Lamont, G., & Van Veldhuizen, D. (2007). *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer Academic Pubs.

Crowe, C. M. (1989). Observability and redundancy of process data for steady state reconciliation. *Chemical Engineering Science, 44*, 2909–2917.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York, NY: John Wiley & Sons, Inc.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2007). Strategies for multiobjective genetic algorithm development application to optimal batch plant design in process systems engineering. *Computers & Industrial Engineering, 54,* 539–569.

Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation, 3*(2), 124–141.

Ferraro, S. J., Ponzoni, I., Sánchez, M. C., & Brignole, N. B. (2002). A symbolic derivation approach for redundancy analysis. *Industrial & Engineering Chemical Research, 41*(23), 5692–5701.

Joris, P., & Kalitventzeff, B. (1987). Process measurement analysis and validation. In *Proceedings of the XVIII conference on the use of computers in chemical engineering CEF'87* (pp. 41–46). Taormina: Italia.

Kretsovalis, A., & Mah, R. S. H. (1988a). Observability and redundancy classification in generalized process networks – I. Theorems. *Computers & Chemical Engineering, 12*, 671–687.

Kretsovalis, A., & Mah, R. S. H. (1988b). Observability and redundancy classification in generalized process networks – II. Algorithms. *Computers & Chemical Engineering, 12*, 689–703.

Meyer, M., Koehret, B., & Enjalbert, M. (1993). Data reconciliation on multicomponent network process. *Computers & Chemical Engineering, 17*, 807–817.

Ponzoni, I. (2001). Aplicación de teoría de grafos al desarrollo de algoritmos para clasificación de variables. Ph.D. thesis. Argentina, Bahía Blanca: Universidad Nacional del Sur.

Ponzoni, I., Sánchez, M. C., & Brignole, N. B. (1999). A new structural algorithm for observability classification. *Industrial & Engineering Chemical Research, 38*(8), 3027–3035.

Ponzoni, I., Sánchez, M. C., & Brignole, N. B. (2004). A direct method for structural observability analysis. *Industrial & Engineering Chemical Research, 43*(2), 577–588.

Safe, M. D., Carballido, J. A., Ponzoni, I., & Brignole, N. B. (2004). On stopping criteria for genetic algorithms. In A. Bazzan, & S. Labidi (Eds.), *Advances in artificial intelligence – SBIA 2004, Lecture notes in computer science* (*subserie lecture notes in artificial intelligence*) (Vol. 3171, pp. 405–413). Berlin Heidelberg: Springer-Verlag.

Sen, S., Narasimhan, S., & Deb, K. (1998). Sensor network design of linear processes using genetic algorithms. *Computers & Chemical Engineering, 22*, 385–390.

Srinivas, N., & Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.

Vazquez, G. E., Ferraro, S. J., Carballido, J. A., Ponzoni, I., Sánchez, M. C., & Brignole, N. B. (2003). The software architecture of a decision support system for process plant instrumentation. *WSEAS Transactions on Computers, 2*(4), 1074–1079.

Vazquez, G. E., Ponzoni, I., Sanchez, M. C., & Brignole, N. B. (2001). ModGen: A model generator for instrumentation analysis. *Advanced Engineering Software, 32*(1), 37–48.