Initial Sensor Network Design with a Multi-Objective Genetic Algorithm

JESSICA A. CARBALLIDO, IGNACIO PONZONI, NÉLIDA B. BRIGNOLE

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Av. Alem 1253 – 8000 - Bahía Blanca
ARGENTINA
Planta Piloto de Ingeniería Química (PLAPIQUI)
Universidad Nacional del Sur - CONICET
Complejo CRIBABB – Camino La Carrindanga km 7 – CC 717 - Bahía Blanca
ARGENTINA
jcarballido@plapiqui.edu.ar; ip@cs.uns.edu.ar; dybrigno@criba.edu.ar

Abstract. A Multi-Objective Genetic Algorithm (MOGA) application, which is based on the aggregating approach, is proposed in this article. Its aim is to find a consistent instrument configuration for industrial process plants that will constitute a convenient initial set of input data for structural Observability Analysis Algorithms (OAs). The better this configuration is, the faster the OAs will converge to a satisfactory solution. Algorithmic effectiveness was evaluated through the analysis of small academic case studies. The results obtained through our algorithm show excellent performance. Therefore, it can be stated that the prototype presented in this work is good enough to serve as a sound basis for the development of the definitive MOGA module, whose implementation will support large-size industrial plant models.

1 Introduction

A very important research issue in Process Systems Engineering is Instrument Network Design (IND) [1, 2, 3, 4, 5, 6] for industrial plants. Process plants are networks of industrial items of equipment connected by streams. IND's aim is to decide on the most convenient amount, location and type of measuring devices to be placed so as to get complete knowledge of the plant's operating conditions, while satisfying other goals such as sensor-cost minimization and maximum reliability. Plant engineers, who have plenty of knowledge about the process being modeled and the various goals of interest, typically carry out this job. Therefore, they will be regarded as the "end users" of the IND software.

One of the most widely used approaches to accomplish this task is based on the construction and analysis of the steady-state mathematical models that represent plant behavior under stationary operating conditions. Each model is a set of algebraic equations that correspond to mass and energy balances, including the relationships employed to estimate thermodynamic properties like densities, enthalpies, and equilibrium constants.

The first step of the analysis consists in defining an initial instrument configuration. This preliminary design classifies model variables into:

- Measured Variables: those ones whose values will be directly obtained from the instruments.
- Unmeasured Variables: those that remain unknown (indeterminable) unless their values are estimated through model equations.

Once this first classification has been defined, the next objective is to determine which unmeasured variables will be observable, i.e. which ones can be calculated from the measurements and the model equations. This crucial task is called Observability Analysis (OA) [7]. OA algorithms analyze the relationships between model equations and unmeasured variables, regarding the measurements as constants. So, the OA results depend on the instrument configuration chosen as initialization. Then, it is possible to improve the IND results by providing a sensible initialization for the OA algorithms.

After an OA sweep has been performed, indeterminable variables of interest may remain. When that is the case, the end user will have to modify the sensor configuration and start another OA run. This makes plant instrumentation design an iterative process, as shown in Fig. 1.

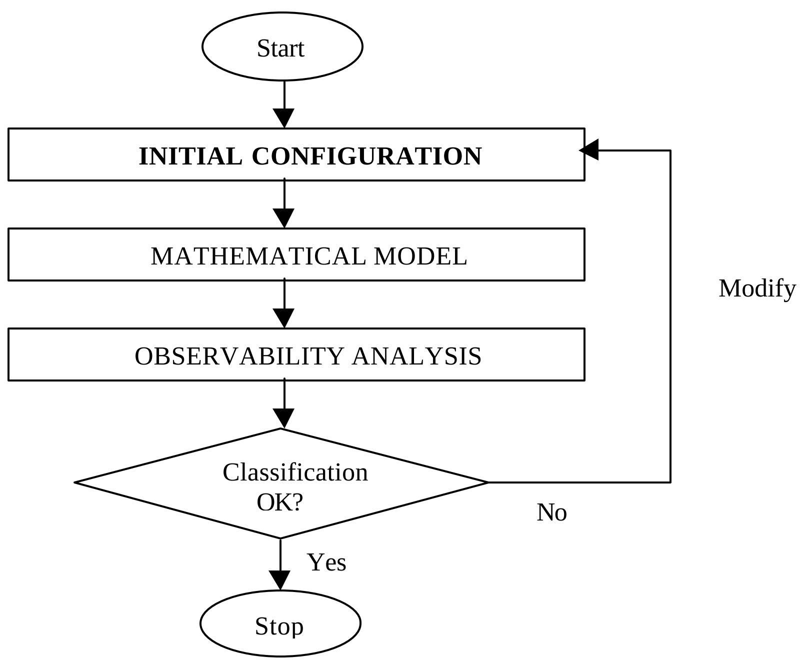

Fig. 1. Iterative process for instrumentation design

The number of iterations required in order to reach an acceptable configuration strongly depends on the initialization. Since every OA stage is very expensive regarding computing time, it is highly advantageous to have as few iterations as possible. Improving the initial instrument configuration can reduce the number of iterations. At present, there are no algorithms to determine initializations that are optimal in this sense and plant engineers choose the sensors on the basis of their skill and experience. However, it would be useful to have an automated tool to tackle this problem and support them when making those complex decisions.

The problem of finding a good initial instrument configuration for an industrial plant pursues several conflicting goals. In general, the problem can be formulated as the task of "placing sensors at suitable locations in order to achieve maximum information about the plant with the minimum quantity of instruments". But when deepening on problem characteristics, other objectives, such as cost and reliability, come up and the problem basically becomes a Multi-Objective Optimization Problem (MOP).

MOP Solving has been the target of diverse research areas (computer science, engineering, industry and chemistry, among others [8]). Many powerful deterministic and stochastic techniques for handling these optimization problems have resulted from engineering, computer science and other related disciplines. In particular, Evolutionary Algorithms (EAs) are a generic stochastic approach, which is commonly used.

As stated by Coello [9], EAs seem to be particularly suitable to solve MOPs because they simultaneously deal with several possible solutions. In the mid-1980s Schaffer did pioneering work in the field of EA implementations for MOPs. Since then, the application of this method has received growing interest from researchers [10, 11].

At present, there are many different implementations of Multi-Objective Evolutionary Algorithms. A simple classification of the existing methods [9] categorizes them into First and Second Generation Techniques. In turn, the former can be subdivided into Nonpareto- and Pareto-Based Approaches. The Vector-Evaluated Genetic Algorithms and the Aggregating Approaches are examples of Nonpareto Techniques [12].

Instead of simultaneously ploughing through all the individual objectives as the Pareto Techniques do, the Aggregating Approaches combine (aggregate) all the objectives into a single one. The aggregation may be done either by addition, multiplication or any other combination. These methods are characterized by their efficiency and are also easy to implement.

In this article we propose a Multi-Objective Genetic Algorithm (MOGA) application for finding an initial instrument configuration, which is based on the aggregating approach. The paper is organized as follows: in Section 1 the main objectives will be introduced, the methodology for solving them is presented in Section 2, study cases and results are reported next and finally, in Section 4, conclusions and future research directions are discussed.

2 Main Objective

The aim of this work is the design of an automated tool in order to find a satisfactory initial sensor network configuration for process plants that succeeds in reducing the number of iterations involved in the OA. In this case, a configuration is considered desirable when it is cheap, reliable and also gives us as much plant information as possible. When there are several objectives, the notion of "optimum" means that we are really trying to find a good trade-off solution between the targets. At the same time, since we are looking for an initialization method, short computing times are required.

3 Methodology

The first step of our research consisted in finding a suitable method that had the specific characteristics mentioned in Section 2. EAs were chosen because they have proved to be the most promising tool for a wide variety of applications due to their effectiveness and reduced computing time.

An initial prototype called AE-THA [13] was presented in the VIII CACIC (8^{th} Argentinian Congress on Computer Science). This implementation yielded satisfactory initial configurations in the context of several single-objective problems. The new approach presented in this paper is based on the same idea, but this time additional objectives have to be simultaneously satisfied.

3.1 The Genetic Algorithm

The input of a genetic algorithm (GA) for OA initialization is the occurrence matrix ***O*** built from the steady-state mathematical model of the plant under study. The rows and columns of this matrix correspond to model equations and variables, respectively. The GA also needs information about the cost and reliability associated to the instruments that would be required in order to measure each variable. The cost of measuring a variable is calculated as the price of the instrument plus its installation cost. The reliability of a variable is associated with that of the instrument that measures it. This information is presented in two N-dimensional vectors, where **N** is the total number of variables in the model.

3.2 The Representation

The individuals' genotypes are binary strings that represent possible instrument configurations as follows:

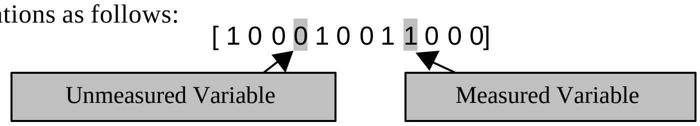

Fig. 2. Genotype Interpretation

The length of a binary string is equal to the total number of variables (**N**) in the mathematical model. The representation for the individuals is the classical one, as well as the crossover and mutation operators. In this paper, the symbol ***i*** will be used to represent either the individual or its genotype depending on the context.

3.3 The Fitness Function

The objective of the algorithm is to find the individual ***i*** that simultaneously exhibits the best performance with respect to cost, reliability and observability. Therefore, there is a trade-off that involves the following three objective functions:

$$\begin{aligned}&\text{Min } Cost\,(\boldsymbol{i}) = \text{Min } \boldsymbol{C}\,(\boldsymbol{i})\\&\text{Max } Reliability\,(\boldsymbol{i}) = \text{Max } \boldsymbol{R}\,(\boldsymbol{i})\\&\text{Max } Observability\,(\boldsymbol{i}) = \text{Max } \boldsymbol{Ob}\,(\boldsymbol{i})\end{aligned}$$

In the rest of this section, the procedure employed to estimate these functions will be explained in detail through a brief example. Let us consider the following values for the occurrence matrix ***O,*** an individual ***i*** and the sensor cost and reliability vectors ***cv*** and ***rv****:*

$$i = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \boldsymbol{O} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

cv = [6500 3020 16700 4000 7300 30500 29000 120900 38200]
rv = [0.5 0.48 0.67 0.49 0.55 0.72 0.68 0.97 0.76]

For this problem instance, these values have been generated at random. Nevertheless, it should be noted that, in practice, the process engineer usually defines all of them with the help of a model generator interface.

Given these input data, the objective functions are calculated individually and then merged into a single fitness function.

The Cost Term The total cost of an individual is the sum of the values of all the elements in ***cv*** that correspond to non-zero entries in ***i***.

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

Then, in the example:
cv = [6500 3020 16700 4000 7300 30500 29000 120900 38200]
C (***i***) = 16700+29000+120900 = 166600

The Reliability Term Following the same reasoning, we have:

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

For the example,

$\boldsymbol{rv}$ = [0.5 0.48 0.67 0.49 0.55 0.72 0.68 0.97 0.76]

$\boldsymbol{R}\,(\boldsymbol{i})$ = 0.67+0.68+0.97 = 2.32

The Observability Term In contrast with the other two objective functions, this one cannot be calculated in a direct way. Its estimation is based on the mathematical operation called Forward Triangularization (FT). The algorithm that implements the FT receives ***i*** and ***O***, and returns the number of variables whose values can be obtained by solving individual model equations with the measurements defined through ***i***. The procedure involves the following steps:

1 – Define a clone ***m*** of the individual ***i***.

2 – Using mask ***m***, look for the first row in ***O*** that contains a non-zero in only one position where ***m*** has a zero. In other words, find the first model equation that contains only one unmeasured variable for the set of instruments defined through ***i***.

3 – Indicate that position in the mask by setting the corresponding element to 1.

4 – Repeat Steps 2 and 3 until no changes are made to ***m*** in a complete overhauling.

5 – FT(***i***) = number of non-zeroes (***m***) - number of non-zeroes (***i***)

FT returns the number of non-zeroes that were added to ***m*** in the whole procedure. This quantity indicates how many unmeasured variables can be directly calculated from the system of algebraic equations given the measurements defined in ***i***.

In short, the value returned by the observability function is:

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

For the example:

$$\boldsymbol{m} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad \begin{array}{r} \\ \text{Mask} \\ \mathbf{O} = \\ \\ \\ \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

In the second row there is a non-zero in the 5th position, so the mask is rebuilt with this position set to 1. The next mask sweep is completed without finding any rows with the characteristics required in Step 2. So, the procedure ends and ***Ob***(***i***)=FT(***i***) = 1

The Merging Approach The Aggregating Approach for the construction of the fitness function requires a normalization criterion to compatibilize the $\boldsymbol{C}(\boldsymbol{i})$, $\boldsymbol{R}(\boldsymbol{i})$ and $\boldsymbol{Ob}(\boldsymbol{i})$ values. The standard procedure consists in normalizing all the objectives in the [0,1] range. Thus,

$$\boldsymbol{NC}(\boldsymbol{i}) = \boldsymbol{C}(\boldsymbol{i}) \,/\, (\mathbf{N} * \mathrm{MC}),\ \boldsymbol{C}(\boldsymbol{i}) \in [0..\mathrm{MC} * \mathbf{N}]\ . \tag{4}$$

$$\boldsymbol{NR}(\boldsymbol{i}) = \boldsymbol{R}(\boldsymbol{i}) \,/\, \mathrm{N},\ \boldsymbol{R}(\boldsymbol{i}) \in [0..\mathbf{N}]\ . \tag{5}$$

$$\boldsymbol{NOb}(\boldsymbol{i}) = \boldsymbol{Ob}(\boldsymbol{i}) \,/\, \mathbf{N},\ \boldsymbol{Ob}(\boldsymbol{i}) \in [0..\mathbf{N}]\ . \tag{6}$$

where:

- $\boldsymbol{NC}(\boldsymbol{i})$, $\boldsymbol{NR}(\boldsymbol{i})$ and $\boldsymbol{NOb}(\boldsymbol{i})$ are the three normalized objectives
- MC is the maximum cost of monitoring a variable, and
- **N** is the total number of variables.

Finally, for the three objectives considered in this paper, the fitness function $\boldsymbol{F}(\boldsymbol{i})$ can be defined as:

$$\boldsymbol{F}(\boldsymbol{i}) = \boldsymbol{NR}(\boldsymbol{i}) + \boldsymbol{NOb}(\boldsymbol{i}) + 1 - \boldsymbol{NC}(\boldsymbol{i})\ . \tag{7}$$

In the example, $\mathbf{N} = 9$; MC = 100000; $\boldsymbol{NC}(\boldsymbol{i}) = 0.1851$; $\boldsymbol{NR}(\boldsymbol{i}) = 0.257$; $\boldsymbol{NOb}(\boldsymbol{i}) = 0.11$ and $\boldsymbol{F}(\boldsymbol{i}) = 1.1819$.

Our algorithm aims at maximizing $\boldsymbol{F}(\boldsymbol{i})$ so that its values always lie between 0 and the total number of individual objectives. Equation 7 can be easily expanded to meet this requirement for a greater number of objectives as follows:

$$\boldsymbol{F}(\boldsymbol{i}) = \sum_{p=1}^{n} NOM_p + m - \sum_{q=1}^{m} NOm_q\ . \tag{8}$$

where:

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

The optimal situation, i.e. $\boldsymbol{F}(\boldsymbol{i})$ = n+m, occurs when all the objectives to be maximized are equal to 1 and those to be minimized become 0.

In this paper we have focused on a minimum number of contradicting objectives in order to build a simple initial prototype that contains the main features of the general problem. The natural follow-up of this work will surely imply the enlargement of the fitness function so as to include additional objectives that take into account other desirable characteristics such as a low degree of non-linearity in the resulting assigned equations and ease of solvability for a given variable.

4 Some experiments and results

To test the effectiveness of our approach for small problems, an auxiliary algorithm called ESA was also implemented. The ESA (Exhaustive Search Algorithm) performs a complete inspection along the search space. It looks for the individual that implies the best compromise among all the objectives using the same fitness function as the MOGA. The exhaustive examination carried out by the ESA is obviously unfeasible for big problems, but it is useful as a means of generating "exact" optimal solutions for academic examples, which can be considered "true" solutions when testing prototype performance.

Three case studies (*c10*, *c12* and *c14*) with the following features were developed:

- The population size is the 15% of the total search space.
- Matrices with 3% of non-zeroes are randomly generated.
- Crossover and mutation probabilities are 0.6 and 0.1, respectively.
- The number of generations is 500 in all cases.

The case studies *c10*, *c12* and *c14* represent problems with ten, twelve and fourteen variables, respectively. Fifty runs were executed for each one of the cases. Optimal results were found in **all cases,** the **hit ratios** reaching **100%**. Table 1 shows the amount of generations required by the algorithm in order to reach the optimum:

CASE STUDY	AV. GENERATION Nº
C10	34,2
C12	75,81
C14	88,88

Table 1. Average number of generations required for algorithmic convergence

It can be inferred from Table 1 that the total number of generations required to reach the optimum increases as matrix size grows.

4 Conclusions and future research directions

In this work we present a multi-objective genetic algorithm application for initial sensor network design. The results reveal that our implementation exhibits an excellent performance when applied to small problems. Three objectives were considered in the fitness function and a suitable generalization formula was proposed.

The next stage of our research will consist in testing algorithmic effectiveness with large-size realistic problems. However, there are several points to consider in order to guarantee trustworthy results for industrial models. The most crucial complex feature to be determined is the selection of adequate convergence criteria. Once the most convenient termination policy has been developed, it will be possible to test our algorithm for medium-size academic examples and huge industrial cases.

References

1. Vaclavek, V. and Loucka, M. “Selection of Measurements Necessary to Achieve Multicomponent Mass Balances in Chemical Plants”, *Chem. Engng. Sci.*, 31 (12), 1976, pp. 1199-1205.
2. Romagnoli, J.A. and Stephanopoulos, G. “On the Rectification of Measurement Errors for Complex Chemical Plants”, *Chem. Engng. Sci.*, 35, 1980, pp.1067-1081.
3. Joris, P. and Kalitventzeff, B. “Process Measurement Analysis and Validation”, *XVIII Congr. The Use of Comp. in Chem. Engng, CEF’87*, Italia, 1987, pp. 41-46.
4. Kretsovalis, A. and Mah, R.S.H. “Observability and Redundancy Classification in Generalized Process Networks-II. Algorithms”, *Comp. & Chem. Engng.* 12, 1988, pp. 689-703.
5. Madron, F. “Process Plant Performance. Measurement and Data Processing for Optimization and Retrofits”, *Ellis Horwood Ltd.: Chichester*, England, 1992.
6. Meyer, M. *et al*. “Data Reconciliation on Multicomponent Network Processes”, *Comp. & Chem. Engng.*, 17, 1993, pp. 807-817.
7. Ponzoni I., Sánchez M.C., Brignole N.B. “A New Structural Algorithm for Observability Classification”, *Industrial & Engineering Chemistry Research*, Vol. 38, No. 8, 1999, pp. 3027-3035.
8. Dias, Alexandre H. F. and de Vasconcelos Joao A., “Multiobjective Genetic Algorithms applied to solve Optimization Problems”, *IEEE Transactions on Magnetics*, Vol 28, No. 2, March 2002.
9. Coello Coello Carlos A., “A Comprehensive Survey of Evolutionary Based Multiobjective Optimization Techniques”, http//www.lania.mx/~ccoello.
10. Eckart Zitzler and Lothar Thiele, “Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study”, Parallel Problem Solving from Nature -PPSN V, Amsterdam, The Netherlands, September 1998, pp. 292-301.
11. Van Veldhuizen, David A. and Lamont Gary, B. “Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art”, *Evolutionary Computation* Vol. 8, Nº 2, 2000, pp. 125-147.
12. Schaffer,J.D., “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms”, *Proceedings of the First International Conference on Genetic Algorithms, and Their Applications*, 1985, pp. 93–100.
13. Carballido Jessica A. *et al*. “AE-THA: Un Algoritmo Evolutivo para la Localización Inicial Óptima de Sensores en Plantas Industriales”, *Congreso Argentino de Ciencias de la Computación, Buenos Aires, 2002.*