Expert Systems with Applications 39 (2012) 2643–2649

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Multi-objective evolutionary approaches for intelligent design of sensor networks in the petrochemical industry

Rocío L. Cecchini [a,*], Ignacio Ponzoni [a,b], Jessica A. Carballido [a]

[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, (8000) Bahía Blanca, Argentina
[b] Planta Piloto de Ingeniería Química CCT Bahía Blanca (ex – CRIBABB), UNS – CONICET, Cno la Carrindanga km 7, CC 717, Bahía Blanca, Argentina

ARTICLE INFO

Keywords:
Intelligent sensor network design
Evolutionary computation
Multi-objective combinatorial optimization

ABSTRACT

The design of optimal sensor networks for an industrial process is a complex problem that requires the resolution of several tasks with a high level of expertise. The first of these subproblems consists in selecting an initial sensor network as the starting point for the instrumentation design. This particular task constitutes a combinatorial optimization problem, where several goals are prosecuted by the designer. Therefore, the initialization procedure can be defined as a multi-objective optimization problem. In this paper, the use of multi-objective evolutionary approaches to assist experts in the design of an initial sensor network is proposed and analyzed. The aim is to contrast the advantages and limitations of Pareto and non-Pareto techniques in the context of this industrial application. The algorithms consider objectives related to cost, reliability and level of information associated with a sensor network. The techniques were evaluated by means of a comparative analysis for a strongly non-linear mathematical model that represents an ammonia synthesis plant. Results have been contrasted in terms of the set coverage and spacing metrics. As a final conclusion, the non-Pareto strategy converged closer to the Pareto front than the Pareto-based algorithms. In contrast, the Pareto-based algorithms achieved better relative distance among solutions than the non-Pareto method. In all cases, the use of evolutionary computation is useful for the expert to take the final decision on the preferred initial sensor network.

1. Instrumentation design as a decision making problem

The *instrumentation design* (ID) problem constitutes a challenging activity in the area of process systems engineering. It consists in deciding on the most convenient amount, location and type of measuring devices to be incorporated into a chemical industrial process. The aim is to attain complete information of the plant's operating conditions, while satisfying other goals such as sensor-cost minimization and maximum reliability. 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 (Singh, Gernaey, & Gani, 2009). Due to the complexity of this task, the development of automatic decision-support tools for this purpose has become a challenge (Carballido, Ponzoni, & Brignole, 2009; Vazquez et al., 2003).

The computer-aided design of process-plant's instrumentation can be modeled by means of an iterative procedure that comprises several steps (see Fig. 1). First, in order to represent the plant behavior under stationary operating conditions, a mathematical steady-state model is built. This model is a set of strongly non-linear algebraic equations that correspond to mass and energy balances, as well as relationships employed to estimate thermodynamic properties like densities, enthalpies, and equilibrium constants. Apart from the model, an initial instrument configuration has to be defined. This preliminary configuration classifies model variables into measured and unmeasured, 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 identifying the unmeasured variables that will be observable, i.e. those that can be calculated by means of model equations, assuming that the measurements are constant values. The OA algorithm (OAA) used for this purpose (Ponzoni, Sanchez, & Brignole, 1999) analyses the structural relationships between model equations and unmeasured variables. It is important to note that the employed OAA is based in combinatorial exploration techniques, which require considerable computing time. Moreover, the observability stage is iterative in nature, whereby it becomes imperative to have an initialization designed to reduce the number of executions of the OAA [4]. The last major step required to complete the entire design procedure is the classification of the measurements, also known as redundancy analysis (Ferraro, Ponzoni,

* Corresponding author.
E-mail addresses: rlc@cs.uns.edu.ar (R.L. Cecchini), ip@cs.uns.edu.ar (I. Ponzoni), jac@cs.uns.edu.ar (J.A. Carballido).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.08.119

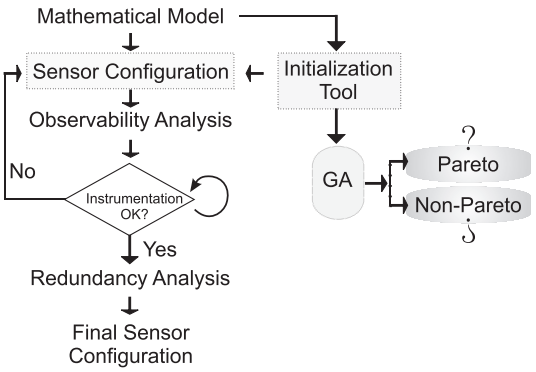

Fig. 1. Stages of ID with the potential initialization techniques.

Sanchez, & Brignole, 2002). This task should be carried out only after a satisfactory OA has been achieved.

1.1. Multi-objective optimization problem

As a result of the before mentioned, the efficiency and the quality of the results of the OAA strongly depend on the starting sensor configuration (Carballido et al., 2009). Notice that the number of iterations required in order to reach an acceptable result may vary significantly conditioned by the initialization. Since the OAA sweeps are very expensive as regards computation times, it is highly advantageous to have as few iterations as possible. This objective can be achieved by choosing an adequate initial instrument configuration (Asteasuain, Carballido, Vazquez, & Ponzoni, 2006).

As the initialization procedure of the ID concerns with finding the optimal spatial arrangement of sensors for an industrial plant, it belongs to the class of *spatial allocation problems* (Li, Curry, & Boyd, 2004; Matthews, Craw, Elder, Sibbald, & MacKenzie, 2000) and constitutes an NP-complete problem. It can also be naturally suited as a combinatorial *multi-objective optimization problem* (MOP), where several aspects must be considered simultaneously. Traditionally, this initialization was conducted by engineers based on their expertise, without the use of any intelligent tools. Recently, several initialization algorithms based on multi-objective evolutionary algorithms (MOEAs) were proposed (Carballido et al., 2009). Even though Carballido et al. (2009) successfully applied aggregative optimization functions over the problem aimed here, those aggregative approaches have not been compared with other well known MOEAs.

1.2. MOP solving with evolutionary algorithms

The different techniques designed to tackle MOPs range from the conventional to the evolutionary ones. Traditional methods are very limited because, in general, as the size of the problem grows, they become too expensive to achieve results in polynomial times (Pulido, 2001). Several authors have pointed out and showed the potential of evolutionary algorithms for MOP solving, and the interest of the evolutionary community in this area started to grow very quickly (Coello Coello, Lamont, & Van Veldhuizen, 2007a; Deb, 2004; Rosenberg, 1967; Schaffer, 1985a). This was justified on the grounds that most real-life problems are MOPs, and also because evolutionary algorithms have the inherent capability of finding multiple solutions close to the Pareto front in reasonable times (Casillas & Martınez-López, 2009: Coello Coello et al., 2007a; Deb, 2004; Guenounou, Belmehdi, & Dahhou, 2009; Pulido, 2001). In particular, evolutionary algorithms (EAs) are suitable for MOPs since they simultaneously deal with a set of possible solutions. Moreover, in comparison with the standard optimization methods, EAs are less susceptible to the shape or continuity of the Pareto fronts (Coello Coello, Lamont, & Van Veldhuizen, 2007b).

Multi-objective EAs can be classified into two main categories depending on whether the concept of Pareto optimality is considered or not (Coello Coello, 2006). In the first case, the population is ranked based on dominance features, and the fitness is assigned depending on the ranking of the individuals. In the case of non-Pareto techniques, the methods are implemented by combining or aggregating the different objectives in order to reach a total order over the individuals (Bäck, Fogel, & Michalewicz, 1997; Fonseca et al., 1993; Schaffer, 1985b).

Regarding the aggregation schemes, the most general approach involves the use of aggregative functions that combine all the objectives to derive a global criterion. In this way, a scalar objective is handled, and constraints are incorporated with associated thresholds and penalty functions. As it is well known, these methods usually have good performance when tackling combinatorial optimization problems with convex Pareto fronts (Coello Coello, 2006). However, even though these techniques are efficient, easy to implement and appropriate for handling few objectives, they present problems when the Pareto front is concave, and they are also very sensitive to the parameter's selection (Coello Coello et al., 2007b). Moreover, they generally have difficulties in handling mixed optimization problems where some objectives are minimized and others are maximized (Deb, 2001).

1.3. Main objectives and proposal

In this work the performance of Pareto and non-Pareto approaches implemented for the ID problem is compared (Fig. 1). In particular, the analysis involves the comparison among the aggregative approach presented in Carballido, Ponzoni, and Brignole (2005) and two state-of-the-art Pareto-based evolutionary algorithms: the Improved nondominated sorting genetic algorithm (NSGA-II) (Deb, Agrawal, Pratap, & Meyarivan, 2002) and the improved strength Pareto evolutionary algorithm (SPEA2) (Zitzler, Laumanns, & Thiele, 2001). These two Pareto strategies were implemented for the ID problem within the PISA (platform and programming language independent interface for search algorithm) framework (Bleuler, Laumanns, Thiele, & Zitzler, 2003). Besides helping to structure the presentation of different algorithms in a coherent way, PISA also reduces the implementation effort significantly (Laumanns & Laumanns, 2005).

The main contribution of this work is constituted by the analysis of different multi-objective evolutionary algorithms for decision support in real-world ID problems. It is important to remark that even though the adapted strategies in this work were thought to tackle the instrumentation task, they can also be applied to any problem belonging to the class of *spatial allocation problems*, thus extending their potential application range and the relevance of the results and conclusions.

The rest of the paper is organized as follows: in Section 2 the main common features of the EAs are explained in detail and the individual characteristics of each method are presented. In Section 3, the trialing framework, including the metrics and the case of study, is described; then the results are exposed. Finally, the main conclusions are discussed.

2. Multi-objective evolutionary algorithms for ID

The EAs presented in this article were custom-made to tackle the ID problem. Therefore, it was necessary to design the structure of the individuals, establish the crossover and mutation operators

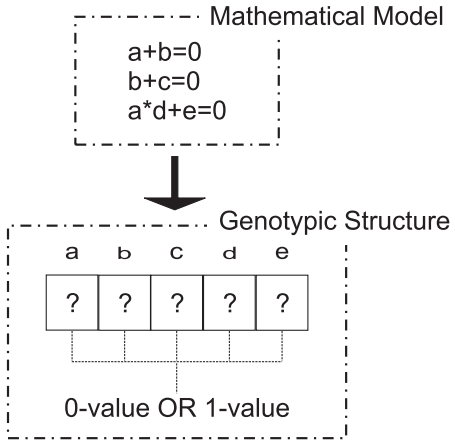

Fig. 2. Genotypic structure for a simplified mathematical model with five variables.

in accordance with this representation, and outline the manner in which the different objectives should be calculated.

The individuals, which represent feasible sensor configurations, are binary strings where a non-zero value in position i means that variable i should be measured, i.e. a sensor should be installed in the location of the plant represented by the ith variable of the mathematical model. Therefore, the length of these chromosomes equals the amount of variables in the mathematical model. For the example illustrated in Fig. 2, a feasible individual would be represented by the genotype [0, 1, 0, 0, 1], meaning that a sensor should be placed in the sites of the plant represented by variables b and e.

With this binary representation, traditional crossover and mutation operators can be directly applied. In particular, one-point crossover and independent bit mutation were implemented.

As regards the objectives, the proposal consists in looking for the sensor configuration with minimum cost and maximum reliability and observability. The estimation of the goals related to purchasing prices and sensor errors corresponding to an individual *ind* is formulated by Eqs. (1) and (2). The equations are intended to find inexpensive sensor networks with minimum average error degree and maximum knowledge about the plant's functioning.

$$Cost(ind) = \sum_{i=1}^{length} (prices[i] * genotype[i]) \quad (1)$$

$$Rel(ind) = 1 - \frac{\sum_{i=1}^{length} (errors[i] * genotype[i])}{\sum_{i=1}^{length} genotype[i]} \quad (2)$$

length is the amount of variables in the mathematical model; *prices* is a vector that holds the purchasing price of each instrument; *errors* is a vector that holds the variations introduced by each instrument; *genotype* is the binary string that represents the configuration as shown in Fig. 2.

With reference to the third objective, the observability degree is attained through a procedure that follows the same philosophy of the procedures that comprise the whole OA module. The main idea in relation to this term is that the observability objective of the EA is constructed based on a simplification of a rigorous method called GS-FLCN (Ponzoni et al., 1999), capturing the fast stages from it so as to obtain a bound on the number of observable variables. The GS-FLCN is a combinatorial procedure that classifies unmeasured variables in observable and unobservable by means of a structural rearrangement of the occurrence matrix (a binary matrix that has a non-zero value in the entry ij if variable i is present in equation j). This matrix corresponds to the highly nonlinear equation system that represents the plant behavior. The first routine carried out by the GS-FLCN strategy, called *forward triangularization* (FT), is performed to detect and remove all the rows containing only one nonzero element. This means that the equations relative to those rows contain only one unknown, then, they can be solved in a straightforward manner. After these equations are detected and solved, the search must be repeated until no more equations with a single unknown are left. In short, this procedure gives lower bound estimation of the number of observable variables. More details on this procedure can be found in Ponzoni et al. (1999).

The observability (*Obs*) term of the algorithms presented in this paper executes the *FT* procedure in order to evaluate the impact of each configuration, preferring those that yield the greatest amount of observable variables possible. For the example exhibited in Fig. 2, and assuming the vectors: *prices* = [2000, 1000, 3000, 2800, 1200], *errors* = [0.1, 0.12, 0.09, 0.11, 0.07] and *genotype* = [0, 1, 0, 0, 0], the three objectives would yield the following values:

$$Cost(ind) = 1000; \quad Rel(ind) = 1 - 0.12; \quad Obs(ind) = 2$$

Note that the value attained for the observability objective arises since, if you have variable b as a constant (that would happen if b was measured according to the configuration represented by *genotype* = [0, 1, 0, 0, 0]), you can obtain the values of variables a and c. In the following subsections some particular features of the three algorithms presented in this work will be introduced. It is important to consider that all of them preserve the aforementioned features.

2.1. Non-Pareto approach: aggregation

The main distinctive attribute of the aggregative approach is constituted by the manner in which all the objectives are gathered. The key goal is to attain a single representative value that will be later used during the selection process. Particularly, the estimation of the fitness function is carried out by combining the values of the three objectives with an approach inspired on the typical linear aggregating functions. It has to be taken into account that, as the different objectives take different orders of magnitude, they have to be appropriately scaled before they are integrated to form the composite objective function. Then, after the normalization of the objectives has been performed, the fitness function F is built as follows:

$$Maximise\ F(ind) = nObs(ind) + nRel(ind) + 1 - nCost(ind) \quad (3)$$

Note that, as the objectives are scaled to the range [0,1], the structure of Eq. (3) ensures that the values of F vary between 0 and 3 (where 3 stands for the number of objectives of the problem). Therefore, good sensor configurations will have values near 3. Moreover, the shape of Eq. (3) also gives the possibility to generalize it to any amount of objectives, as shown in Eq. (4).

$$Maximise\ F(ind) = \sum_{p=1}^{n} NOM_p + m - \sum_{q=1}^{m} NOm_q \quad (4)$$

n is the amount of objectives to be maximized; m is the amount of objectives to be minimized; $NOM_p \in [0,1]$ is the pth normalized objective to be maximized; $NOm_q \in [0,1]$ is the qth normalized objective to be minimized; $F(ind) \in [0, n+m]$.

The optimal (utopian) situation, i.e. $F(ind) = n + m$, 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, F moves within a closed bounded range of values, thus providing a clear threshold to be reached. In regard to the selection process, once all the individuals have been evaluated by means of the F function, they are chosen to form a mating pool according to the binary tournament selection process. This technique was used since it was the one that

yielded the best preliminary results after a comparison among roulette wheel, ranking and n-tournament selection methods was carried out.

2.2. Pareto-based approaches: NSGA-II and SPEA2

The NSGA-II and SPEA2 algorithms are two of the most studied and efficient Pareto-based MOEAs.

The NSGA-II algorithm works as follows:

1. Create a random population of size n, P_0, and sort it based on the non-domination concept. In this step, a rank equal to the non-domination level is assigned to each solution (1 for the first front, 2 for the second front, and so on), where minimization of rank is assumed.
2. Create a population of n offsprings, Q_0, using recombination, mutation and binary tournament selection operator based on crowding distances.
3. Execute the ith generation according the next three steps:
   (a) Construct a population $R_i = P_i \cup Q_i$ of size $2S$.
   (b) Sort the new population R_i based on the non-domination level. Since all previous and current population members are included in R_i, elitism is ensured. Solutions belonging to the best front, i.e. $Fr_1(R_i)$, are the best solutions in the combined population R_i.
   (c) Finally, if the size of $Fr_1(R_i)$ is smaller than S, all members of the set $Fr_1(R_i)$ are chosen for the new population P_{i+1}. The remaining members of the population P_{i+1} are chosen from subsequent nondominated fronts in the order of their ranking until no more sets can be accommodated. If $Fr_j(R_i)$ is the last front from which individuals can be accommodated in the population, but not all the members can enter in the population, then a decision needs to be made to choose a subset of individuals from Fr_j. In order to decide which members of this front will win a place in the new population, the NSGA-II uses a selection criterion based on a crowded-comparison operator that favors solutions located in less crowded regions. This crowded comparison is applied based on the objective space R^k.

The SPEA2 algorithm starts with an initial population P_0, of size S and an empty external population $\overline{P}_0$ with a maximum capacity of $\overline{S}$. During the ith generation the SPEA2 repeats four basic steps:

1. Calculate the nondominated set of P_i, i.e. $Fr_1(P_i)$, and copy it to $\overline{P}_i$ (i.e., $\overline{P}_i = \overline{P}_i \cup Fr_1(P_i)$).
2. Remove all dominated solutions of $\overline{P}_i$. If the number of nondominated external solutions exceeds $\overline{S}$, prune $\overline{P}_i$ selecting a representative by means of a clustering method. Using this method, group the individuals in $\overline{S}$ classes or clusters based on a crowded distance. Then, select the individual with the lowest distance to the others within each cluster as the representative of the cluster.
3. Calculate the fitness of each individual within $P_i \cup \overline{P}_i$. Then, select individuals from $P_i \cup \overline{P}_i$ using binary tournament selection until the mating pool is filled.
4. Finally, create a population of S offspring, P_{i+1}, applying problem-specific recombination and mutation.

As it was aforementioned, the Pareto-based strategies were built on the base of a platform called PISA (Bleuler et al., 2003). PISA is a text-based interface for search algorithms. It splits an optimization process into two modules. One module contains all the parts that are specific to the optimization problem (e.g., evaluation of solutions, problem representation, and variation of solutions) and it is called the *Variator*. The other module contains the parts of an optimization process which are independent of the optimization problem (mainly the selection process). This part is called the *Selector*. These two modules are implemented as separate programs which communicate through text files. In the selection module a candidate solution can be represented by an identifier and a set of objective values describing the quality of this individual. This identifier is the only information passed from the problem module to the selection module. In the *Variator*, the actual genotypes matching those identifiers are stored.

For the Pareto-based algorithms presented in this article, a *Variator* specific for the ID application has been designed and implemented, and then it was combined with the *Selectors* corresponding to the NSGA-II and SPEA2 optimization algorithms. For both cases, the optimization process advances through the following steps alternating between the *Selector* and the *Variator*. Steps 1 and 3 correspond to the specific properties and procedures that had to be designed and implemented for the ID problem instance and they will be later explained in more detail. Steps 2 and 4 depend on the selection procedure that is being used; in this case, NSGA-II or SPEA2.

Step 1: The *Variator* creates an initial collection of individuals and calculates the objective values. This initial set of individuals is an array of binary strings representing the possible configurations. The evaluation of the objectives is performed as it was explained at the beginning of this section. However, PISA demands that all the objectives should be minimized. Therefore, the *observability* (*Obs*) objective was transformed into *indeterminable* (*Indet*) with Eq. (5). Also, the reliability (*Rel*) objective was transformed into Error by attaining its complementary value (see Eq. (6)).

$$Indet(ind) = length - Obs(ind) - \sum_{i=1}^{length} genotype[i] \tag{5}$$

$$Error(ind) = 1 - Rel(ind) \tag{6}$$

Step 2: The selection module chooses a collection of parent individuals from the initial population which it thinks are promising.

Step 3: The *Variator* alters these individuals in order to get a new collection of offspring individuals. This variation is carried out with the recombination and mutation operators. These operators were implemented as it was explained above in this section.

Steps 2 and 3 are repeated until a fixed number of generations is reached. All the data exchange is established using text files. The synchronization of the modules is achieved by writing a state variable into a text file which both programs can read and update. The modules can be programmed in different programming languages or even run on different (distributed) platforms. For this work, C++ over a Windows platform was used.

3. Experimental framework and results

In order to compare the algorithms, two well known performance metrics have been used, namely *Set Coverage* and *Spacing* (Deb, 2001). The industrial case under study is an ammonia chemical plant whose mathematical model has 546 variables. Regarding the parameters, the algorithms were run with populations of 100 individuals. The crossover probability was set to 0.7 and the mutation probability was set to 0.1, with a bit flip probability set to 1/*length* (1/546 = 0.018). All the EAs ran until 100 generations were reached.

3.1. Performance metrics

There are two distinct goals in multi-objective optimization: (1) discover solutions as close to the Pareto-optimal solutions as possible, and (2) find solutions as diverse as possible in the non-dominated front (Deb, 2001). In some sense, these two objectives are orthogonal to each other. The first one requires a search *towards* the Pareto front, while the second one requires a search *along* it.

3.1.1. Set coverage metric

This measure, suggested by Zitzler (1999), can be used to establish which one of the two methods A and B better suits the goal (1). It calculates the proportion of solutions of B which are weakly dominated by solutions of A, as follows:

$$C(A,B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|} \tag{7}$$

The metric value $C(A,B) = 1$ means that all members of B are weakly dominated by A. On the other hand, if $C(A,B) = 0$ no member of B is weakly dominated by A. Since the domination operator is not symmetric, $C(A,B)$ is not necessarily equal to $1 - C(B,A)$. Thus, it is necessary to calculate both to understand how many solutions of A are covered by B and vice versa.

3.1.2. Spacing

Schott (1995) proposed a metric to analyze goal (2) in the context of a single algorithm, which is calculated with a relative distance measure between two consecutive solutions in the obtained non-dominated set, as follows:

$$\mathbf{S} = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \tag{8}$$

where: Q is the set of non-dominated solutions, $\bar{d} = \sum_{i=1}^{|Q|} d_i / |Q|$ and $d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^{M} |f_m^i - f_m^k|$.

The distance measure is the minimum value of the sum of other solutions in the non-dominated set. **S** measures the standard deviations of different d_i values. When the solutions are nearly uniformly spaced, **S** will be small. Thus, the smaller the *spacing* value is, the better the algorithm performs as regards goal (2).

3.2. Case of study

The performance of the algorithms was evaluated by carrying out the instrumentation analysis of a chemical industrial plant. The main features of this model are described by Bike (1985). Briefly, the plant produces 1500 ton/day of anhydrous liquid ammonia at 240 K and 450 kPa 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 rigorous mathematical model of this plant was generated by using the *ModGen* package (Vazquez, Ponzoni, Sanchez, & Brignole, 2000). The resulting system contained 557 non-linear algebraic equations and 546 process variables.

3.3. Results and analysis

The experiments consisted in a sequence of 25 runs per algorithm. *Spacing* and *Set Coverage* metrics were estimated for each run. Table 1 shows all the combinations (A,B) between the three EAs for which the *Set Coverage* metric was calculated, and Table 2 presents the results for the *Spacing* metric. AGG^{AP} stands for the non-Pareto *AGGregative APproach*.

As it can be observed, $C(NSGA-II, AGG^{AP})$ and $C(SPEA2, AGG^{AP})$ are 0 in all of the runs. This means that no member of any AGG^{AP}'s front is weakly dominated by any of the members of the fronts

Table 1
Results for the computation of the set coverage metric for 25 runs.

Run	Set coverage A / B	NSGA-II / SPEA2	NSGA-II / AGG^{AP}	SPEA2 / NSGA-II	SPEA2 / AGG^{AP}	AGG^{AP} / NSGA-II	AGG^{AP} / SPEA2
1		0.000	0.000	0.217	0.000	0.652	1.000
2		0.333	0.000	0.261	0.000	1.000	1.000
3		1.000	0.000	0.000	0.000	1.000	1.000
4		0.000	0.000	1.000	0.000	1.000	1.000
5		0.000	0.000	1.000	0.000	1.000	1.000
6		0.077	0.000	0.529	0.000	0.824	0.923
7		0.542	0.000	0.250	0.000	0.679	1.000
8		0.429	0.000	0.154	0.000	1.000	1.000
9		1.000	0.000	0.000	0.000	0.636	1.000
10		0.000	0.000	1.000	0.000	0.667	0.706
11		0.571	0.000	0.368	0.000	1.000	1.000
12		0.421	0.000	0.391	0.000	1.000	0.895
13		0.087	0.000	0.790	0.000	1.000	1.000
14		1.000	0.000	0.000	0.000	1.000	1.000
15		0.050	0.000	0.846	0.000	1.000	1.000
16		0.182	0.000	0.632	0.000	0.737	0.432
17		0.000	0.000	1.000	0.000	1.000	0.857
18		0.000	0.000	1.000	0.000	1.000	0.636
19		0.222	0.000	0.783	0.000	1.000	1.000
20		0.810	0.000	0.154	0.000	0.808	0.810
21		0.000	0.000	1.000	0.000	1.000	1.000
22		0.952	0.000	0.071	0.000	1.000	1.000
23		0.000	0.000	1.000	0.000	1.000	1.000
24		0.727	0.000	0.107	0.000	1.000	0.955
25		0.318	0.000	0.600	0.000	1.000	1.000
Avg.		0.349	0.000	0.526	0.000	0.920	0.929

Table 2
Results for the computation of the spacing metric for 25 runs.

Run	Spacing		
	AGG^{AP}	NSGA-II	SPEA2
1	0.040	0.030	0.063
2	0.035	0.023	0.035
3	0.031	0.013	0.051
4	0.060	0.039	0.020
5	0.043	0.039	0.036
6	0.118	0.027	0.031
7	0.033	0.030	0.035
8	0.027	0.021	0.023
9	0.043	0.027	0.023
10	0.018	0.031	0.026
11	0.144	0.045	0.057
12	0.144	0.023	0.031
13	0.192	0.027	0.021
14	0.018	0.017	0.018
15	0.057	0.034	0.045
16	0.013	0.042	0.015
17	0.029	0.032	0.028
18	0.000	0.053	0.021
19	0.041	0.034	0.038
20	0.026	0.027	0.033
21	0.030	0.024	0.066
22	0.057	0.032	0.041
23	0.143	0.056	0.022
24	0.136	0.018	0.022
25	0.050	0.028	0.030
Avg.	0.061	0.031	0.033

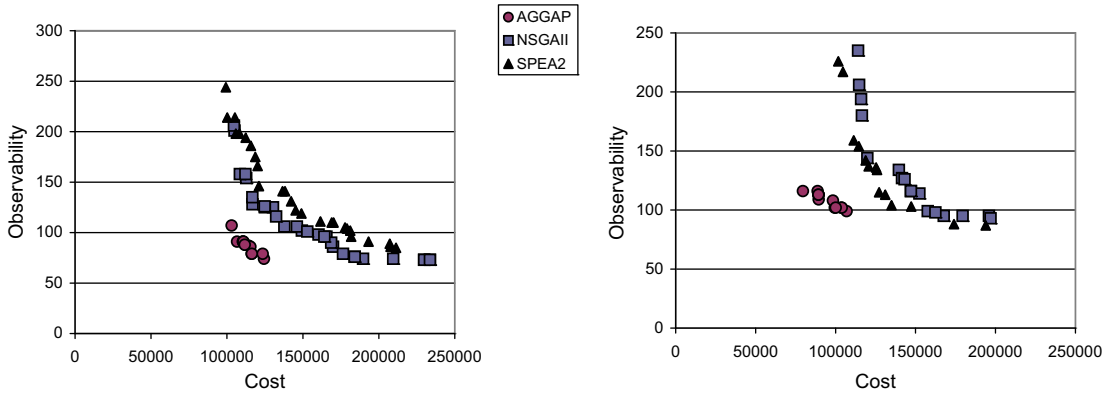

Fig. 3. Pareto fronts obtained by each method for two random runs.

obtained by the Pareto approaches, in any of the runs. As regards the comparison between both Pareto approaches, none of them clearly outperforms the other one. With respect to the *Spacing* metric, the aggregative approach shows less diversity along the fronts, while both Pareto-based search algorithms exhibit similar values.

These results give the idea that the AGG^{AP} yields a set of non-dominated individuals that are closer to the Pareto front than the individuals attained by the other two Pareto-based methods. However, the non-dominated solutions obtained by these last methods cover the fronts in a more uniform mode.

Fig. 3 shows the fronts obtained in two different runs where these suppositions are left of manifesto. Also, as it can be observed from the figure, the fronts are convex, which gives more grounds to the conclusion that AGG^{AP} performs better than its counterparts in terms of goal (1) of Section 3.1. It is important to remark that, for clarifying purposes, we have decided to present a two-dimensional front considering *cost* and *observability*, since *reliability* values were very similar throughout all the runs of all of the EAs.

3.4. Expert role as the decision maker

It is important to note that any multi-objective approach based on a global search procedure, as the evolutionary computing algorithms proposed here, will report a set of non-dominated solutions. Therefore, the last decision about which sensor network should be used as starting point of the instrumentation design relays on the expert. In general, this decision will be taken in terms of the experiences and preferences of the designer and, in some particular cases, alternative instrumentation design studies can be conducted from different starting points in order to yield a final decision.

4. Conclusions

In this article the problem of applying different multi-objective evolutionary algorithms in order to assist experts in the initialization of instrumentation design procedures was tackled. This problem is characterized as NP-complete, and it belongs to the class of spatial allocation problems. Non-Pareto and Pareto-based approaches were implemented, and an analysis of their performance on the instrumentation design of an ammonia synthesis plant was performed. The analysis of the results was focused on the quality of the final fronts as regards two metrics, namely set coverage and spacing.

The best performance concerning the closeness of the fronts to the actual Pareto front was exhibited by the aggregative approach. However, as regards the diversity of the solutions, the Pareto-based strategies attained superior results. As to the specific problem instance being tackled in this article, the goal is to find adequate configurations to start the instrumentation design process. Therefore, the aggregative approach constitutes the most convenient method to be used as the initialization tool for the instrumentation design of this chemical plant.

Nevertheless, beyond this particular case study, this paper illustrates how the use of multi-objective evolutionary algorithms can support the expert work by giving alternative high quality sensor networks.

Finally, as the instrumentation design problem was tackled by using the PISA standard, the *variator* implemented in this work can serve as a real-world benchmark problem by means of which the performance of specific multi-objective algorithms can be assessed. Then, the multi-objective evolutionary algorithms proposed in this work do not only represent a valuable tool for instrumentation design per se, but they can also be useful for the designers of new multi-objective optimization algorithms.

References

Asteasuain, F., Carballido, J. A., Vazquez, G. E., & Ponzoni, I. (2006). Using computational intelligence and parallelism to solve an industrial design problem. In J. S. Sichman, H. Coelho, & S. O. Rezende (Eds.), *IBERAMIA-SBIA. Lecture notes in computer science* (Vol. 4140, pp. 188–197). Springer.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (1997). *Handbook of evolutionary computation*. Bristol, UK: IOP Publishing Ltd.

Bike, S. (1985). *Design of an ammonia synthesis plant. CACHE case study*. Department of Chemical Engineering, Carnegie Mellon University.

Bleuler, S., Laumanns, M., Thiele, L., & Zitzler, E. (2003). PISA: A platform and programming language independent interface for search algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *EMO. Lecture notes in computer science* (Vol. 2632, pp. 494–508). Springer.

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.), *EvoCOP. Lecture notes in computer science* (Vol. 3448, pp. 12–22). Springer.

Carballido, J. A., Ponzoni, I., & Brignole, N. B. (2009). SID-GA: An evolutionary approach for improving observability and redundancy analysis in structural instrumentation design. *Computers & Industrial Engineering, 56*, 1419–1428.

Casillas, J., & Martínez-López, F. J. (2009). Mining uncertain data with multiobjective genetic fuzzy systems to be applied in consumer behaviour modelling. *Expert Systems and Applications, 36*, 1645–1659.

Coello Coello, C. A. (2006). Evolutionary multiobjective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine, 1*, 28–36.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007a). *Evolutionary algorithms for solving multi-objective problems. Genetic and evolutionary computation* (second ed.,). New York, NY: Springer-Verlag, Inc.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007b). Evolutionary algorithms for solving multi-objective problems. *2nd ed.* New York: Springer. ISBN 978-0-387-33254-3.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.

Deb, K. (2004). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*, 182–197.

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

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international conference on genetic algorithms* (pp. 416–423). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Guenounou, O., Belmehdi, A., & Dahhou, B. (2009). Multi-objective optimization of TSK fuzzy models. *Expert Systems and Applications, 36*, 7416–7423.

Laumanns, M., & Laumanns, N. (2005). Evolutionary multiobjective design in automotive development. *Applied Intelligence, 23*, 55–70.

Li, V. C.-W., Curry, G. L., & Boyd, E. A. (2004). Towards the real time solution of strike force asset allocation problems. *Computers & OR, 31*, 273–291.

Matthews, K. B., Craw, S., Elder, S., Sibbald, A. R., & MacKenzie, I. (2000). Applying genetic algorithms to multi-objective land use planning. In L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, & H.-G. Beyer (Eds.), *GECCO* (pp. 613–620). Morgan Kaufmann.

Ponzoni, I., Sanchez, M., & Brignole, N. (1999). A new structural algorithm for observability classification. *Industrial & Engineering Chemistry Research, 38*, 3027–3035.

Pulido, G. T. (2001). *Optimización Multiobjetivo Usando un Micro Algoritmo Genético*. Master's thesis Maestría en Inteligencia Artificial, Universidad Veracruzana Xalapa, Veracruz, México.

Rosenberg, R. S. (1967). *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Harbor, Michigan.

Schaffer, J. D. (1985a). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international conference on genetic algorithms* (pp. 93–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Schaffer, J. D. (1985b). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st international conference on genetic algorithms* (pp. 93–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Singh, R., Gernaey, K. V., & Gani, R. (2009). Model-based computer-aided framework for design of process monitoring and analysis systems. *Computers & Chemical Engineering, 33*, 22–42.

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

Vazquez, G. E., Ponzoni, I., Sanchez, M. C., & Brignole, N. B. (2000). ModGen: A model generator for instrumentation analysis. *Advances in Engineering Software, 32*, 37–48.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland, TIK-Schriftenreihe Nr. 30.

Zitzler, E., Laumanns, M., & Thiele, L., 2001. *SPEA2: Improving the strength Pareto evolutionary algorithm*. Technical report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.