Algorithms **2009**, *2*, 259-281; doi:10.3390/a2010259

OPEN ACCESS
algorithms
ISSN 1999-4893
www.mdpi.org/algorithms

Article

Design of Sensor Networks for Chemical Plants Based on Meta-Heuristics

Mercedes Carnero [1], José L. Hernández [1] and Mabel C. Sánchez [2,*]

[1] Dpto. de Ciencias Básicas, Facultad de Ingeniería, Universidad Nacional de Río Cuarto, Campus Universitario, (5800) Río Cuarto, Argentina
[2] Planta Piloto de Ingeniería Química (UNS-CONICET), Camino La Carrindanga Km 7, (8000) Bahía Blanca, Argentina
E-mails: mcarnero@ing.unrc.edu.ar (M.C.); jlh@ ing.unrc.edu.ar (J.L.H.).

* Author to whom correspondence should be addressed; E-mail: msanchez@plapiqui.edu.ar

Received: 3 November 2008; in revised form: 8 January 2009 / Accepted: 17 February 2009 / Published: 20 February 2009

Abstract: In this work the optimal design of sensor networks for chemical plants is addressed using stochastic optimization strategies. The problem consists in selecting the type, number and location of new sensors that provide the required quantity and quality of process information. Ad-hoc strategies based on Tabu Search, Scatter Search and Population Based Incremental Learning Algorithms are proposed. Regarding Tabu Search, the intensification and diversification capabilities of the technique are enhanced using Path Relinking. The strategies are applied for solving minimum cost design problems subject to quality constraints on variable estimates, and their performances are compared.

Keywords: Sensor location, Stochastic optimization, Tabu search, Scatter search, Population based incremental learning algorithms.

1. Introduction

A reliable and complete knowledge of current plant state is essential for plant monitoring, regulatory and supervisory control, real time optimization, planning and scheduling, etc. The quality and availability of variable estimates strongly depend on the sensor network (SN) installed in the process and the data reconciliation packages used to enhance the precision of estimates during plant operation.

The problem of selecting a set of variables to be measured, which is optimal with respect to some specified criteria and simultaneously provides the quantity and quality of information required from the process, is called the sensor network design problem (SNDP).

Different types of instrument arrangements arise depending on the number and location of the selected measurements. A minimum-number SN contains the smallest amount of instruments that allow the estimation of all unmeasured variables. If more sensors are used than the minimum required to satisfy the aforementioned condition, a so-called redundant SN is obtained. The quantity of sensors contained in both types of arrangements is known in advance to problem resolution. In practice it is necessary to satisfy constraints only on a subset of key measured or unmeasured variable estimates. In this case a general SN is designed without knowing in advance the cardinality of the optimal sensor set. As only a subset of variables are of real practical interest, the optimal selection of measurements for general SNs is a powerful tool for the design of large-scale plants.

Diverse criteria have been proposed for the optimal selection of sensor structures, such as the minimization of cost while ensuring quality constraints (precision, reliability, estimability) on variable estimates, the maximization of the minimum variable-estimate availability subject to cost constraints, the minimization of the global error of variable estimates for a fixed instrumentation-project budget, etc.

In SND the important decision to be made with regard to each stream variable is whether to measure it or not. To mathematically formulate these decisions, binary variables are employed which indicate the presence or absence of sensors. The problem is usually multimodal and involves many binary variables, therefore a huge combinatorial optimization problem subject to constraints should be solved.

At first meta-heuristics based on Genetic Algorithms (GAs) [1-2] were proposed to deal with the design of sensor structures. In this regard, Sen et al. [3] developed an evolutionary procedure devoted to the selection of flowmeters for nonredundant SNs. It optimizes single criteria such as cost, reliability or estimation accuracy based on concepts from graph theory. Carnero et al. [4] dealt with the optimal design of non-redundant structures for linear processes that ensure the observability of all unmeasured variables while optimizing single or multiple criteria. The solution procedure was developed using a GA whose operators were modified based on linear algebra concepts. Furthermore, Viswanath and Narasimhan [5] presented an evolutionary approach for the design of linear redundant SNs that maximize the reliability of variable estimates. In previous references, the quantity of sensors to be installed in the network is fixed before the run of the optimization procedure.

Regarding the design of general SNs, characterized by the fact that the optimal number of sensors is unknown in advance, Chao-An et al. [6] presented the design of maximum-availability networks subject to cost and precision constraints, but they solved the problem for a small size network using the classic GA. Gerkens and Heyen [7] presented two ways of parallelizing the GA, namely the global parallelization and the distributed GA, to reduce the solution time. They concluded that both techniques allow reducing the elapsed time but the second one is more efficient. Also Benqlilou et al. [8] applied GAs to solve the design and retrofit of reliable SNs. Their implementation was performed using the GA toolbox of MATLAB program. Then a hybrid procedure based on GAs (HGA) was developed by Carnero et al. [9] to minimize the instrumentation network cost subject to precision constraints on key variables. They used a structured population in the form of neighbourhoods and a local optimizer of the best current solutions, which provide a good balance between the algorithm capabilities of exploration and exploitation. Recently, Gerkens and Heyen [10] proposed a general approach for designing the cheapest SN able to detect and locate a set of specified faults. They applied the previously developed parallel procedure [7] to select sensors and their locations.

In recent years some applications of Tabu Search (TS) meta-heuristic [11] for the solution of chemical engineering problems have appeared [12 - 14]. The technique guides a local-search procedure that explores the solution space beyond local optimality. The local search uses an operation called move to define the neighbourhood of a given solution. It was reported that TS has a more flexible and effective search behaviour than other stochastic methods as consequence of the use of adaptive memory. This motivated the development of new strategies for the design and upgrade of SNs. Within the framework of TS, a Strategic Oscillation Technique around the feasibility boundary (SO-TS) was reported by Carnero et al. [15-16]. A comparative performance-analysis indicated that the strategy efficiently searches the solution space, significantly reducing the number of required calls to the evaluation function in comparison with HGA and the Classic TS (C-TS) [9].

There exist other population-based methodologies that have demonstrated a rewarding performance for solving hard optimization problems and constitute attractive alternatives to GAs. In this regard, Scatter Search (SS) [17-18] is founded on the premise that systematic designs and methods for creating new solutions afford significant benefits beyond those derived from recourse to randomization. Also Estimation of Distribution Algorithms (EDAs) [19-20] offer a novel evolutionary paradigm. They make use of a probabilistic model, learnt from the promising solutions, to guide the search process. Within the framework of EDAs approach, Population Based Incremental Learning Algorithms (PBIL) are devised, which introduce the concepts of competitive learning (typical in artificial neural networks) to direct the search [21]. Successful applications of SS and PBIL have been reported to solve complex combinatorial problems, such as the vehicle routing, knapsack and scheduling problems [22 - 24].

Furthermore Path Relinking (PR) has been proposed as a method to better explore the solution space of complex problems when a set of promising solutions is known [11]. It has been incorporated to algorithms based on TS to provide better intensification and diversification capabilities to the search. The technique is recognized as an effective tool to solve difficult combinatorial problems [25].

The rewarding performance of SS and PBIL algorithms for solving multimodal optimization problems with a huge number of binary or integer variables sustains their selection to address the SNDP and perform a comparative performance analysis among SO-TS, SS and PBIL. Furthermore the performance of the effective search strategy PR is analyzed within the framework of TS and compared also with SO-TS.

For this work new strategies to solve the sensor location problem are developed. They are based on the meta-heuristics SS, PBIL and PR within the framework of TS (PR-TS). The distinctive features of the algorithms are provided and comparative performance studies are conducted among SO-TS, PR-TS, SS and PBIL for solving instrumentation designs of industrial networks extracted from the literature.

The rest of the paper is organized as follows. In Section 2 the design problem is briefly introduced. Sections 3 to 5 present algorithms based on the meta-heuristics TS, SS and PBIL respectively. Application results are provided and discussed in Section 6 and, conclusions are addressed in Section 7.

2. Sensor Network Design and Upgrade Problem

Let us assume the operation of a process under steady state conditions can be represented by the following set of *m* non-linear algebraic equations **R**

$$\mathbf{R(z)} = \mathbf{R(x, u)} = \mathbf{0}, \tag{1}$$

where **z** stands for the n-dimensional vector of process variables, and **x** and **u** represent the vector of measured and unmeasured variables respectively. The problem of optimal selection of instruments during plant design or upgrade consists of determining the optimal partition of vector **z** in vectors **x** and **u**, and it is formulated as follows

$$\begin{array}{lll} Min & f(\mathbf{q}) & \\ s.t. & & \\ & g_j(\mathbf{q}) \le g_j^*(\mathbf{q}) & \forall\, j \in S \\ & q_i = 1 & \forall\, i \in I_0 \\ & \mathbf{q} \in \{0,1\}^{n-|I_0|} & \end{array} \tag{2}$$

where **q** is a *n*-dimensional vector of binary variables such that: $q_i = 1$ if variable *i* is measured, and $q_i = 0$ otherwise. For the sake of simplicity it is assumed that there is only one potential measuring device for each variable; if it is not the case the number of binary variables increases significantly. Furthermore *f*(**q**) represents a one-dimensional objective function, g_j(**q**) indicates the constraint imposed on the quality of the *j*-th key variable estimate, considering there exists only one constraint for each key variable, *S* is the set of key process variables, and I_0 stands for the initial set of instruments that is empty at the network design stage. No instruments' localization restrictions are imposed in this formulation but they may arise in practice.

Different performance criteria of the sensor structure, *f*(**q**), are used depending on the specific application. Frequently the life cycle instrumentation cost for the design or upgrade project leads the selection; nevertheless reliability measures are sometimes preferred for safety reasons. A wide variety of objective functions have been used: instrumentation cost, global error of variable estimates, system reliability, variable reliability and availability, the economic value of instrumentation projects, etc, [4], [9], [26].

Regarding the set of constraints, **g**(**q**), engineers not only require to know the value of key variables for economic, safety or environmental reasons (estimability constraints), but impose conditions about the precision, reliability or availability of variable estimates.

In general nonlinear discrete optimization problems arise. For large scale processes the dimension of the search-space for optimization models represented by equation (2) increases significantly. In consequence the design turns out to be a huge combinatorial optimization problem that may have many local optima. In these cases, it is really valuable that the solution procedure provides at least a good solution, if not the global optima, and that also can be run in parallel computers to reduce execution times.

3. Tabu Search Approach

In this section the technique SO-TS, which currently appears as the best approximate solution method for the SNDP, is revised [15-16]. Also C-TS meta-heuristic is modified to incorporate the effective strategy PR for search intensification and diversification. The objective is to analyze the behaviour of PR to address this particular design problem.

3.1 An Overview

Tabu Search is a meta-heuristic approach devoted to the solution of optimization problems. It uses a guided local-search procedure to explore the entire solution space in a way that makes it difficult to be entrapped in local optima and prevents solution cycling. The strategy incorporates adaptive memory that allows local searches being guided by the information collected previously, which is presented in the form of Tabu lists [11].

Given a current solution **q**, a neighbourhood of possible solutions, N(**q**), is defined at each iteration by modifying **q** through an operation called move, whose definition is highly problem-dependent. A modified neighbourhood N′(**q**) is obtained from N(**q**), as result of maintaining a selective history of the recent states encountered during the search. The neighbours in N′(**q**) are inspected to select the best one that becomes the starting point for the new iteration (**q**′), even if it is worse than **q**. Furthermore the best solution ever found (**q***) is saved. Aspiration criteria are applied to determine when the Tabu lists can be overridden to avoid missing good solutions.

Tabu Search makes use of short-term and long-term memory. The first one prevents solution cycling and being entrapped in local optima. It identifies the elements of N(**q**) excluded from N′(**q**) because they correspond to solutions recently visited. Short-term knowledge is provided by means of a Recency based Tabu List, where new solutions are incorporated at each iteration, and remained there as forbidden moves during the Tabu Tenure Period, *pt*.

The use of long-term memory allows the inclusion of solutions not ordinarily found in N(**q**), because it penalizes the elements of N′(**q**) associated to moves that have been done more often. These are contained in the Frequency based Tabu list that represents the long-term memory of the technique.

Within the framework of TS, other procedures are incorporated for search intensification and diversification such as SO and PR. The first one consists of a sequence of destructive and constructive phases. Given a feasible solution, the search is strategically driven to cross the feasibility boundary and to continue in the infeasible region (destructive phase) until certain depth is reached, then the search changes the direction towards the feasible region where it continues until the same depth (constructive phase). The process of repeatedly crossing the feasibility boundary from different directions originates an oscillatory behaviour. Standard TS mechanisms are applied to avoid going back over previous trajectories.

Path relinking consists in selecting two components of a reference set *R* which is made up of high quality solutions. They are called Initiating Solution (*IS*) and Guiding Solution (*GS*) respectively. A path is generated in the neighbourhood space from the *IS* to the *GS* selecting moves that introduce attributes contained in *GS*. The rationale behind the strategy is that high quality solutions share certain attributes, and their combinations produce other solutions that could be better than those contained in *R* and even better than the best current one. The strategy performance depends on the rules adopted to generate the reference set *R*, select *IS* and *GS* and identify the neighbour structure and the guiding attributes.

3.2 Special features of Tabu Search-based strategies for Sensor Network Design

3.2.1 Initial Solution

A solution is represented by a vector **q** whose *i*-th element is 1 or 0 depending on whether the process variable is measured or not. The procedure used to generate the initial population in the GA-based strategy developed by Carnero et al. [9] is run. Each member of the population represents a set of sensors that provides an estimate for all key variables. Then, the subset of individuals that satisfy the restrictions imposed on variable estimates is inspected to determine the best solution. This is selected as the TS initial solution $\mathbf{q_0}$.

3.2.2 Neighbourhood

Given a solution **q**, its neighbourhood N(**q**) comprises a set of new solutions, $\mathbf{q_N}$, that are at a Hamming distance of one with respect to **q**, that is

$$\mathrm{N}(\mathbf{q}) = \left\{\mathbf{q}_\mathrm{N} / \boldsymbol{q}_{Ni} \neq \boldsymbol{q}_i \text{ and } \boldsymbol{q}_{Nj} = \boldsymbol{q}_j \ \forall \boldsymbol{j} \neq \boldsymbol{i}\right\} \tag{3}$$

3.2.3 Tabu Lists

The Recency based Tabu list, **t**, is a vector of dimension *n*. A non-zero element of **t** indicates that this variable move is forbidden because it was modified recently. Furthermore its value is the number of remaining iterations until *pt* for this move is elapsed.

The Frequency based Tabu list is represented by a vector **h** of dimension *n*. The *i*-th component of **h** reports the number of moves of variable *i* used to generate the next solution during *ph* iterations. If the search process has become stuck in a specific area, it is necessary to direct the search to unvisited areas or regions visited less frequently. Consequently the evaluation function corresponding to the *i*-th allowable move is penalized in proportion to h_i. After *ph* iterations vector **h** is reset.

3.2.4 Evaluation Function

A member of the neighbourhood is evaluated using a function, *F* that takes into account constraint violations as follows

$$F = \begin{cases} f(\mathbf{q}) & \text{if } \mathbf{q} \text{ is feasible} \\ f_{\max}(1 + Q(\mathbf{q})) & \text{if } \mathbf{q} \text{ is infeasible} \end{cases} \tag{4}$$

where f_{max} represents an upper bound of *f*(**q**) for feasible individuals and *Q*(**q**) takes into account constraint violations as follows,

$$Q(\mathbf{q}) = \frac{1}{R_u} \sum_{r=1}^{R_u} \frac{g_r(\mathbf{q}) - g_r^*(\mathbf{q})}{g_r(\mathbf{q})} \tag{5}$$

where R_u is the number of unsatisfied constraints.

3.2.5 Aspiration and Termination Criterion

If the best neighbour is in a tabu area but has a better evaluation function value than **q*** then its tabu property is overridden.

Termination on convergence criterion has been implemented. If the improvement after *#MaxIter* iterations is no larger than a threshold, the search is stopped.

3.2.6 Strategic Oscillations within the framework of Tabu Search

For the sensor structure design problem the neighbourhood is built as follows. Given a feasible set of instruments **q**, the destructive phase consists in eliminating one measurement per iteration, therefore the amount of null elements in the members of N(**q**) increases. The search crosses the feasibility boundary and proceeds in the infeasible region until the evaluation function reaches the bound L_0. Then it turns around and the constructive phase is initiated by incorporating measurements. In this phase, the amount of zero elements in the members of N(**q**) lowers and the search returns to the feasible region. The constructive phase finishes when the number of measurements is greater than the bound L_1. The assumed bounds are the instrumentation cost if all variables are measured plus the cost of the most expensive measurement for L_0, and 80% of the length of **q** for L_1.

3.2.7 Path relinking within the framework of Tabu Search

After a predefined number *frecpr* of consecutive moves, the basic TS procedure switches to a PR phase that stops when the reference set *R* has a cardinality ≤ 1. Then, either stopping conditions are verified or the procedure is repeated to form a new *R*.

The procedure devoted to build the set *R* is essential to generate new high quality solutions. In this work, *R* is created during the TS phase and then improved during the PR phase. To build *R* the following steps, which aim at ensuring quality and diversity of solutions, are executed:

a) A set *P* is made up of each solution that, at some stage of the TS phase, improves the best current solution and becomes the best one

b) The first half of *R* is loaded with the best solution vectors **p** from set *P*

c) For each solution vector **p** that belongs to *P* but is not included in *R*, that is $\mathbf{p} \in \{P/R\}$, the Hamming distance, *d*, between **p** and *R* is computed

$$d(\mathbf{p}, R) = Min \quad \sum_{j}^{n} \left| q_j^i - p_j \right| \quad \forall\, i \in R \tag{6}$$

d) The second half of R is loaded with $\mathbf{p} \in \{P/R\}$ that maximizes the Hamming distance. The *IS* and *GS* are defined as the worst and best solutions in R, respectively. Regarding the rules to identify the neighbour structure and the guiding attributes, two types of moves are proposed:

a) Let us consider $\mathbf{q_i}$ and $\mathbf{q_g}$ represent the current *IS* and *GS*, respectively. Starting from $\mathbf{q_i}$, the idea is to generate neighbours that tend to $\mathbf{q_g}$ at last. To generate the first neighbour, $\mathbf{n_1}$, it is set equal to $\mathbf{q_{i.}}$ Then the elements of both vectors are compared from left to right until a difference is found. For this position, for example the k-th position, the element of $\mathbf{q_i}$ is replaced by the corresponding element of $\mathbf{q_g}$. The comparison finishes and the first neighbour is obtained. The same procedure is repeated between $\mathbf{n_1}$ and $\mathbf{q_g}$ to obtain $\mathbf{n_2}$ but in this case the comparison starts at the (k+1) position. The rest of the neighbours are obtained in the same way until the last neighbour has only one element that differs with respect to $\mathbf{q_g}$.

b) The rationale behind this move is to build the neighbourhood by changing the position of a measurement. Consequently the total number of measurements remains unchanged. At first, the arithmetic difference between vectors $\mathbf{q_g}$ and $\mathbf{q_i}$ is calculated for each element. Positive (+1), negative (-1) and zero differences are obtained. The positions of positive and negative differences are registered in position vectors **pp** and **np**. Then all combinations between each element of **pp** and all the elements of **np** form a set of interchanges between the corresponding elements of $\mathbf{q_i}$. Each interchange generates a neighbour.

3. 3 Tabu Search based procedures

3.3.1 Classic Tabu Search

```
Generate an initial solution $\mathbf{q_0}$
Set $\mathbf{q}^*=\mathbf{q}=\mathbf{q_0}$ and $F(\mathbf{q}^*) = F(\mathbf{q_0})$
for i =1 to #Max Iter do
    Generate neighbourhood $N(\mathbf{q})$
    Select $\mathbf{q}' \in N(\mathbf{q})$ with the lowest F value
    If $\mathbf{q}' \in N(\mathbf{q})$ satisfies the aspiration criterion $F(\mathbf{q}') < F(\mathbf{q}^*)$
        Set $\mathbf{q}^*=\mathbf{q}'$ and $F(\mathbf{q}^*) = F(\mathbf{q}')$
    else
        Select a new solution $\mathbf{q}' \in N(\mathbf{q})$ that minimizes $F(\mathbf{q}')$ and is non-tabu
    endif
    Set the reverse move for pt iterations and update h
    Set $\mathbf{q}=\mathbf{q}'$
endfor
return q*
```

3.3.2 Tabu Search with Strategic Oscillations

```
Select a feasible initial solution $\mathbf{q_0}$
Set $L_0 = \sum_i cost_i + max(cost)$
Set $L_1 = 0.8 * length(\mathbf{q_0})$
Set remove=1
Set $\mathbf{q}^*=\mathbf{q}=\mathbf{q_0}$
for i =1 to #MaxIter do
    if remove=1
            Generate neighbourhood $N_1(\mathbf{q})$ and evaluate
            Get best neighbour $\mathbf{q'}$
            If $F(\mathbf{q'}) > L_0$
                    remove = 0
            else
                    Update tabu list and frequency table
                    $\mathbf{q}=\mathbf{q'}$
            endif
    else
            Generate neighbourhood $N_2(\mathbf{q})$ and evaluate
            Get best neighbour $\mathbf{q'}$
            If sum($\mathbf{q}$) > $L_1$ and $\mathbf{q}$ is feasible
                    remove=1
            end
            $\mathbf{q}=\mathbf{q'}$
    end
    if $F(\mathbf{q})<F(\mathbf{q}^*)$
            $\mathbf{q}^*=\mathbf{q}$
    end
end
return $\mathbf{q}^*$
```

3.3.3 Tabu Search with Path Relinking

```
R, the reference set; $\mathbf{q}^*$, the current best solution
Select the initial solution $\mathbf{q_i}$ and the guiding solution, $\mathbf{q_g}$
Set $\mathbf{q}=\mathbf{q_i}$
while $\mathbf{q} \neq \mathbf{q_g}$
    Generate neighbourhood $N_1(\mathbf{q})$ (path 1)
    Generate neighbourhood $N_2(\mathbf{q})$ (path 2)
    Select a solution $\overline{\mathbf{q}} \in N_1(\mathbf{q}) \cup N_2(\mathbf{q})$ that minimize $F(\overline{\mathbf{q}})$ or satisfies the aspiration criterion
    Set $\mathbf{q} = \overline{\mathbf{q}}$
return $\mathbf{q}$
```

4. Scatter Search Approach

4.1 An Overview

Scatter Search is an evolutionary approach which is established on the principle that systematic methods for creating new solutions provide significant benefits beyond those obtained by random algorithms.

An algorithmic description of this meta-heuristic was presented by Glover in 1998 [27]. In this version, a starting set of solutions that guarantees a certain level of diversity is generated and heuristic procedures, devised for the problem under consideration, are applied to improve these solutions. Then a subset of the best vectors, in terms of quality and diversity, is selected as the Reference Set *Refset*. New solutions are generated by applying structured combinations of subsets of *Refset,* and the same heuristic procedures applied above are used to improve new solutions. A collection of the best improved solutions is added to *Refset*. The steps are repeated until *Refset* remains unchanged.

The well known algorithm template proposed by Glover [27] is made up of five methods devoted to perform the following tasks: Diversification Generation, Improvement of Solutions, Reference Set Generation and Upgrade, Subset Generation and Solution Combination. In the next subsection the ad-hoc methods developed to carry out these tasks for the optimal design of SNs are described.

4.2 Special features of the Scatter Search-based strategy for Sensor Network Design

4.2.1 Diversification Generation

To generate a collection of diverse trial solutions the procedure proposed in Carnero et al. [9] to initialize the GA population is run. Each solution guarantees that all key variables can be estimated. Nevertheless there exist feasible and unfeasible solutions in this set that provide the required diversity to enhance the exploratory capabilities of the technique.

The most diverse solutions with respect to the fitness function defined by equation (4) are incorporated in a set *U* of size *Usize* = 50.

4.2.2 Improvement of Solutions

Two procedures are devised to improve each solution contained in *U*.

a) Improvement Method 1

The neighbourhood of the solution is inspected to find a sensor structure with the same number of measurements but that has an *F* value lower than the current solution. If a better solution is not found, the current one remains in U.

b) Improvement Method 2

For large neighbourhoods the previous method is time consuming, therefore another technique is proposed to improve solutions obtained by combination.

At first building blocks or patterns that are shared by high quality solutions are identified by means of the false outputs of the XOR or Exclusive OR function. Then all neighbours constituted by these blocks are generated.

For example, let us consider that $\mathbf{q_1}$=[1 1 1 1 1 1 0] and $\mathbf{q_2}$= [0 1 1 1 1 0 1] are two solutions obtained by combination that share the block [- 1 1 1 1 - -]. Therefore, the neighbourhood inspected to improve the solutions is made up of the following vectors N($\mathbf{q_1}$, $\mathbf{q_2}$) = {[0 1 1 1 1 0 0], [0 1 1 1 1 0 1 0], [0 1 1 1 1 1 1],[1 1 1 1 1 0 0], [1 1 1 1 1 0 1], [1 1 1 1 1 1 1]}. If this procedure fails to incorporate new solutions to *Refset*, the first one is run.

4.2.3 Reference Set Generation and Upgrade

Twenty percent of the solutions contained in *U* are incorporated to *Refset*. The first half of *Refset* is loaded with the best solutions in *U*. The remaining elements of *Refset* are the most diverse solutions in *U*. These are determined in terms of the Hamming distance as it is explained in Section 3.2.7.

4.2.4 Subset Generation and Solution Combination

The simplest form of subset generation, which consists in generating all pairs of reference solutions, is selected. To combine a couple of solutions $\mathbf{q_1}$ and $\mathbf{q_2}$, a mechanism made up of the subsequent steps is implemented [11]:

a) Let consider each binary variable as a random discrete variable C_k and assign to it the following probability distribution

$$\begin{aligned} P(C_k = 1) &= \frac{F_1 q_1^k + F_2 q_2^k}{F_1 + F_2} \\ P(C_k = 0) &= 1 - P(C_k = 1) \end{aligned} \tag{7}$$

where q_1^k and q_2^k stand for the value of the *k*-th binary variable in $\mathbf{q_1}$ and $\mathbf{q_2}$ respectively.

b) Generate a random sample of size 1 for C_k. To obtain it, the P(C_k = 1) is first calculated using (7) for given values of q_1^k, q_2^k, F_1 and F_2. Then a random uniform number *r* is generated and compared with P(C_k = 1). If $r \le$ P(C_k = 1), the *k*-th element of the combined solution is 1, otherwise it is zero. These conditions are stated by equation (8)

$$q^k = \begin{cases} 1 \text{ if } r \le \mathrm{P}(C_k = 1) \\ 0 \text{ if } r > \mathrm{P}(C_k = 1) \end{cases} \tag{8}$$

If q_1^k and q_2^k have the same value for their *k*-th component, then it is reproduced in the combined solution. If the values are different, the application of the above procedure enhances the probability that good solutions transfer their components to the combined solution, given that the numerator of equation (7) is weighted by the binary value of the components.

4.3 Scatter Search Based Procedure

```
Set U = Ø
U= Diversification Generation method(Usize)
U= Improvement Method 1(U)
RefSet= Reference Set Generation and Upgrade method(U)
Refset=Sort(Refset)
NewSolution = TRUE
while (NewSolution)
   NewSolution = FALSE
   Generate subsets of RefSet, using the Subset Generation method, with at least
   one new solution.
   while (there is at least one subset without evaluation)
       Select the subset and label it as evaluated.
       Apply the Solution Combination method to the solutions in the subset.
       Apply Improvement Method 2 to each solution obtained by
        combination. Let q be the improved solution.
       If F(q) is better than F(q_worst) and q is not included in RefSet
              q_worst = q
              Refset=Sort(Refset)
              NewSolution = TRUE
       else
              q= Improvement Method 1(q)
              If F(q )< q_best solution in RefSet
                     q_best =q
                     NewSolution = TRUE
              endif
       endif
       endwhile
   endwhile
```

5. Population Based Incremental Learning Approach

5.1 An Overview

Estimation of Distribution Algorithms are Evolutionary Algorithms that work with a population of candidate solutions. At first an initial population is generated and their members are evaluated using the objective function. Those with better function values are selected to build a probabilistic model of the population, and a new set of points is sampled from the model. The process is iterated until a termination criterion is fulfilled.

Therefore EDAs' approach is based on the evolution of a probabilistic model of the solution space. The potential solutions included in the population are assumed as realizations of multivariate random variables, whose joint probability distribution can be estimated and updated. In this sense, a solution vector $\mathbf{q}$ ={ q_1, q_2, ..., q_n} can be considered as a sample of an n-dimensional vector $\mathbf{Q}$ = { Q_1, Q_2, ..., Q_n} where Q_i is a binary variable.

Several EDAs have been proposed with a variety of models and learning algorithms. EDAs can be broadly divided, according to the complexity of the probabilistic models used to capture the interdependencies between the variables, into univariate, bivariate or multivariate approaches [19-20].

An univariate EDA, called PBIL, was originally proposed by Baluja [21] who introduced the concept of competitive learning (typical in artificial neural networks) to guide the search. The technique explicitly maintains statistics about the search space and uses them to direct its exploration. The objective of the algorithm is to create a real valued probability vector which, when sampled, reveals high quality solution vectors with high probability.

The original PBIL methodology considers independent variables, thus the product of their marginal distributions constitutes the joint distribution of all variables. This is updated to take into account the structure of best current solutions. The use of PBIL with univariate probabilistic models has shown good results for solving complex problems [23-24].

In this work a parallel implementation of the algorithm that allows *NPBIL* instances being executed independently is applied. After the *NPBIL* populations evolve one iteration, their probability vectors are related by means of uniform crossover using a crossover probability *Pinteraction.*

5.2 Special features of PBIL-based strategies for Sensor Network Design

5.2.1 Initial Solution and Evaluation Function

The procedure devised to create the set *Refset* is also applied to generate the initial population for each instance, which is made up of *N* individuals. Furthermore the function defined by equation (4) is selected as the evaluation function. It should be noted that certain features of the proposed methodologies are maintained fixed in order to perform comparisons on the same basis.

5.2.2 Marginal Distribution Estimation

The initial population is used to estimate the marginal probability of the variables for the first iteration of the algorithm. Each random unidimensional variable Q_i (i =1:n) follows a Bernoulli Distribution. The maximum likelihood estimate of the expected value of Q_i is its sample mean. Therefore the vector of sample means for each instance is evaluated as follows

$$p_i = \frac{1}{N}\sum_{1}^{N} q_i \quad (i = 1{:}n) \tag{9}$$

5.2.3 Selection and Local Search

The probabilistic model of the solution space is used to generate another set of solutions by simulation. Then, Improvement Method 2 is applied to each member of the population. The best one in terms of the *F* values is selected in order to upgrade the probability vector.

5.2.4 Distribution Probability Upgrade

The probability vector is updated, position by position, using the learning rate *LR* as follows

$$p_i^u = p_i^c(1 - LR) + s_i LR \quad (i = 1{:}n) \tag{10}$$

where p_i^u is the updated probability, p_i^c stands for the current probability and s_i represents the *i*-th element of the best solution.

The value of *LR* is essential to the convergence of the algorithm. High values of *LR* introduce a bias towards specific solution structures avoiding the exploration of different regions of the search space; consequently they originate problems of premature convergence.

5.2. Mutation

To introduce diversity in the search, each element of the probability vector is involved in a mutation procedure, with probability *PMUTA*, as follows

$$p_i^u = p_i^c\,(1 - MS) + rand(0,1)\,MS \quad (i = 1{:}n) \tag{11}$$

where *MS* is the mutation amount [28].

5.3 PBIL Based Procedure

```
Initiate NPBIL probability vectors $\mathbf{p}_k$ ($k$ =1,…,NPBIL)
while (stopping criteria = .FALSE.)
  for $k$ = 1,…, NPBIL do
      Generate N individuals by simulation according to $\mathbf{p}_k$
      Evaluate the fitness function F for each member of the population
      Apply the Improvement Method 2 to each member of the population
      Select the best solution
      Upgrade $\mathbf{p}_k$ using the best solution and the learning rate LR
      Mutate $\mathbf{p}_k$ using a probability of mutation PMUTA and a quantity of mutation MS
  end for
  Set OffSpring= Ø
  for $k$ = 1,…, NPBIL step 2 do
      Select 2 individuals (parents) among all vectors $\mathbf{p}$
      if random<Pinteraction
         Use uniform crossover to calculate two children
         Add children to OffSpring
      else
         Add parents to OffSpring
      endif
  endfor
```

```
    for $k=1,\ldots, NPBIL$ do
        Set $\mathbf{p}_k=OffSpring(k)$
    endfor
endwhile
```

6. Analysis of Results

Previous methods are applied to the instrumentation design of process flowsheets frequently used as test cases in the literature. Problem 1 and 2 are solved for different case studies, and a comparative performance analysis is conducted. First both problems are described; then results are presented and discussed.

6. 1. Problem 1

The minimum cost SNDP that satisfies precision and estimability constraints for a set of key variable estimates is formulated as follows

$$\begin{aligned} &Min \quad \mathbf{c}^{\mathrm{T}}\mathbf{q} \\ &s.t. \\ &\qquad \hat{\sigma}_j(\mathbf{q}) \le \sigma_j^* \qquad \forall\ j \in S_\sigma \\ &\qquad E_k(\mathbf{q}) \ge 1 \qquad \forall\ k \in S_E \\ &\qquad \mathbf{q} \in \{0,1\}^n \end{aligned} \tag{12}$$

where $\mathbf{c}^{\mathrm{T}}$ is the acquisition cost vector; $\hat{\sigma}_j$ is the standard deviation of the j-th variable estimate after a data reconciliation procedure is applied and E_k stands for the degree of estimability of variable k. For this formulation E_k is set equal to one, consequently only a variable classification procedure run is needed to check its feasibility. Furthermore S_σ and S_E are the set of key process variables with requirements in precision and estimability respectively.

For Problem 1 five case studies are solved that are presented in order of increasing complexity.

a) Cases 1 to 4

The selection of flowmeters for an industrial steam metering network is performed. The process consists of 11 units interconnected by 28 streams. It is assumed that there is no restriction for the location of sensors on any stream, therefore the search space is made up of 2^{28} solutions. Data of cost and standard deviation for measurement errors are obtained elsewhere [1]. This flowsheet is frequently used to test new methodologies in the area of process monitoring.

Different case studies are run increasing the number of streams subject to estimability and precision constraints. In this sense, the set of key variables for Cases 1 and 2 is constituted by only three flowrates, but they are associated with distant streams for the second case. The number of key variables is increased to 6 and 10 for Cases 3 and 4 respectively. In Table 1 this information is summarized.

All cases were previously addressed using the HGA procedure. Furthermore Cases 1, 3 and 4 were presented in Carnero et al. [15-16] where it was reported that the technique SO-TS outperforms both HGA and C-TS.

Table 1. Case Studies' Constraints.

Case	Constraints
1	E ≥ 1 for stream 1 σ_2^*=0.025 σ_6^*=1.785
2	E ≥ 1 for stream 2 σ_{10}^*=1.048 σ_{28}^*=1.445
3	E ≥ 1 for streams 17 23 $\sigma_4^* = 2.199$ $\sigma_8^* = 3.281$ $\sigma_{21}^* = 1.754$ $\sigma_{25}^* = 1.709$
4	E ≥ 1 for streams 7 16 18 20 $\sigma_4^* = 2.199$ $\sigma_5^* = 1.065$ $\sigma_8^* = 3.281$ $\sigma_{12}^* = 1.345$ $\sigma_{27}^* = 1.415$ $\sigma_{28}^* = 1.445$
5	E ≥ 1 for streams 5 12 14 35 37 44 62 70 77 σ_{10}^*=1584.2, σ_{17}^*=1359.6, σ_{35}^*=200.7, σ_{39}^*=1580.6, σ_{56}^*=122.72, σ_{69}^*=1284.4
6	A_4^*= 0.9, A_8^*= 0.9 σ_3^*=0.7, σ_8^*=0.5
7	A_4^*= 0.8, A_8^*= 0.8, A_{10}^*= 0.8 σ_3^*=0.7, σ_8^*=0.15 σ_{12}^*=0.4

b) Case 5

The flowmeter network design for a simplified ethylene plant is conducted. The process involves 82 streams and 47 units. It is assumed that the standard deviation of flowmeters is 2.5% of the corresponding true flowrates [15]. The search space of this problem is made up of 2^{82} solutions, therefore it is large in comparison with other cases presented in the literature. In Table 1 the constraints of the optimization problem are included.

Case 5 was previously solved using the method SO-TS, which also showed better performance than C-TS and HGA to address this design problem.

6.2. Problem 2

The problem of minimizing the life cycle cost of an arrangement of sensors subject to precision and availability constraints on a set of key flowrate estimates is posed as follows

$$\begin{array}{lll} Min & LCC(\mathbf{q}) & \\ s.t. & & \\ & \hat{\sigma}_j(\mathbf{q}) \le \sigma_j^* & \forall\ j \in S_\sigma \\ & A_k(\mathbf{q}) > A_k^* & \forall\ k \in S_A \\ & \mathbf{q} \in \{0,1\}^n & \end{array} \tag{13}$$

where *LCC* indicates the life cycle cost of the sensor structure, A_k stands for the availability of the *k*-th variable estimate and, S_σ and S_A are the set of variables subject to precision and availability constraints on their estimates respectively. The formulations used for *LCC* and A_k in this work are the same presented by Carnero et al. [16]. Variable availabilities are calculated using the cutsets of the process graph and sensor failure probabilities. The evaluation of availability constraints incorporates additional complexity to Problem 2. Two case studies are presented for this formulation that differ in the number of problem constraints.

Cases 6 and 7

The flowmeter network design for a simplified hydrodealkylation plant, constituted by 8 nodes and 14 edges, is conducted by solving the optimization problem represented by equation (13). The search space is formed by 2^{14} solutions. Process flowsheet, purchase cost and precision data for the set of available sensors are extracted from the work by Viswanath and Narasimhan [5]. Problem constraints are included in Table 1.

Also Cases 6 and 7 were previously solved using the SO-TS procedure [16] which systematically outperforms C-TS and HGA.

6.3. Results

In this study strategies' performances are compared in terms of solution quality, computational effort and robustness. All procedures were developed using MATLAB program. Regarding the initial condition of the algorithms, the local-search procedures SO-TS and PR-TS are run using the same initial solution, which is the best solution of the initial population generated to run SS and PBIL based methods.

In Table 2 the parameter settings for each technique are provided. Table 3 presents the solutions obtained for each case study. Regarding the population-based algorithms, the minimum and average solution values and the coefficient of variation (CV), obtained by the execution of 20 runs, are reported.

Regarding the quality of solutions, the same results are obtained for both local-search procedures in general. Nevertheless for the largest problem instance, Case 5, SO-TS achieves a lower objective function value than PR-TS does. Furthermore the comparison between the population-based algorithms reveals the superiority of PBIL because if finds the reported minimum value for each test case for most of the runs. These solution values are obtained by both SO-TS and PBIL.

The number of calls to the evaluation function, reported in Table 4, is considered as a measure of the time spent by each meta-heuristics. The rationale behind this assumption is that the time consumed in function evaluations is approximately 90% of the total elapsed time by each procedure. Computational experiments support this assertion.

The minimum number of required calls to obtain the solution is reported for SO-TS and PR-TS. The average values of calls are presented for population-based algorithms.

It can be seen from Table 4 that the procedure PR-TS requires more function evaluation calls than SO-TS to obtain solutions of the same quality. Regarding the population-based strategies, PBIL achieves better solutions than SS at the expense of consuming more calls. It should be noted that the ratio between the number of calls required by PBIL and SS diminishes considerably for the test case of largest dimension.

Table 2. Parameters of the Implemented Strategies.

Parameter Value	SO-TS	PR-TS	SS	PBIL
pt	$\lfloor \sqrt{n} \rfloor$	$\lfloor \sqrt{n} \rfloor$	---	---
ph	23	60	---	---
Lo	$\sum_{i=1}^{n} cost(i) + max(cost)$	---	---	---
L_1	0.8**n*	---	---	---
|*R*|	---	10	---	---
frecpr	---	15	---	---
# Max Iter	300	200	---	100
|*Refset*|	---	---	12	---
NPBIL	---	---	---	8
N	---	---	---	12
Pinteraction	---	---	---	0.7
PMUTA	---	---	---	0.02
MS	---	---	---	0.05
LR	---	---	---	0.1

Table 3. Solutions of the Case Studies.

Case	SO-TS	PR-TS	SS (for 20 runs)			PBIL (for 20 runs)		
			Min	Mean	CV	Min	Mean	CV
1	533.6	533.6	533.6	587.9	11.28	533.6	533.6	0.00
2	894.9	894.9	894.9	1006.3	17.90	894.9	894.9	0.00
3	752.3	752.3	752.3	767.4	2.33	752.3	754.0	0.98
4	1178.0	1178.0	1178.0	1178.0	0.00	1178.0	1178.0	0.00
5	50,845	50,846	50,847	54,183	6.32	50,845	50,846	0.08
6	62,322	62,322	62,322	63,535	4.54	62,322	62,322	0.00
7	80,548	80,548	80,548	80,821	0.22	80,548	80,567	0.11

The procedures are also run using random initial solutions to analyze their robustness. Twenty runs are performed for Case Study 5, which is the largest problem instance. Statistics of the objective function values are reported in Table 5. This shows that the lowest objective function value is obtained by the SO-TS technique. Nevertheless the biases of the mean values, with respect to the minimum reported value, of the local-search procedures are considerably higher than those corresponding to population-based algorithms. Furthermore, PBIL is more robust than SS because it presents lower values for the bias and CV.

Table 4. Number of calls to the evaluation function.

Case	SO-TS	PR-TS	SS (AVG)	PBIL (AVG)
1	717	1,130	3,795	16,540
2	781	1,063	4,771	16,674
3	39	56	2,631	15,622
4	110	413	1,928	7,326
5	8,298	10,535	129,369	253,364
6	42	56	1,442	6,882
7	260	600	2,112	9,621

Table 5. Case Study 1 Solutions using Random Initialization.

	SO-TS	PR-TS	SS	PBIL
Min	50,846	54,974	50,847	51,124
Max	290,572	290,296	290,295	55,312
Mean	197,612	265,620	66,711	51,989
Deviation	116,792	73,913	52,803	1,375
CV	59.10%	27.83%	79.15%	2.64%

7. Conclusions

Meta-heuristic approaches for the design of SNs in chemical plants are presented and compared in this work. Within the framework of TS, an ad-hoc PR technique is implemented to provide better intensification and diversification capabilities to TS. Also methods based on SS and PBIL meta-heuristics are proposed, and a parallel procedure is implemented for PBIL. The performance of these strategies for solving test cases from the literature is analyzed.

In relation to local-search procedures, the incorporation of the PR technique within the framework of TS does not provide advantages for solving the SNDP with respect to the procedure SO-TS presented in a previous work.

Regarding population-based algorithms, PBIL presents a better overall performance. The mean value of the objective function and the coefficient of variation are lower than the corresponding ones to SS for all cases. Also the technique based on SS is more sensitive to the quality and diversity of the initial set of solutions than the design method based on the PBIL algorithm.

The comparison of SO-TS and PBIL strategies indicate that the first one strongly depends on the quality of the initial solution. If a good starting point is provided it produces a high quality solution with a low computational effort. In contrast, PBIL is more robust. It is capable of making replicas of the best solutions starting from lower quality initial points at the expense of an increment of the computational time. As PBIL algorithms can be naturally run in parallel, the total elapsed time can be reduced for a given number of calls. In this case, data transfer among parallel processors should only be performed for the crossover of probability vectors. Furthermore it should be noted that solution improvement methods based on local search enhance significantly the performance of population-based strategies.

Acknowledgements

The authors wish to thank the financial support of CONICET (National Research Council of Argentina), ANPCyT (National Agency for the Science and Technological Promotion), UNS (Universidad Nacional del Sur, Bahía Blanca, Argentina), UNRC (Universidad Nacional de Rio Cuarto), and Agencia Córdoba Ciencia (Ministerio de Ciencia y Tecnología de la Provincia de Córdoba).

References and Notes

1. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
2. Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: New York, NY, USA, 1989.
3. Sen, S.; Narasimhan, S.; Deb, K. Sensor network design of linear processes using genetics algorithms. *Comput. Chem. Eng.* **1998**, *22*, 385-390.
4. Carnero, M.; Hernández, J.; Sánchez, M.; Bandoni, A. An Evolutionary Approach for the Design of Non-Redundant Sensor Networks. *Ind. Eng. Chem. Res.* **2001**, *40*, 5578-5584.
5. Viswanath, A.; Narasimhan, S. Multi-objective Sensor Network Design Using Genetic Algorithms. In Proc. 4th IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries, Seoul, Korea, 2001; pp. 265-270.
6. Chao-An, L.; Chuei-Tin, K; Chin-Leng; Chen-Liang, C. Optimal Sensor Placement and Maintenance for Mass-Flow Networks. *Ind. Eng. Chem. Res.* **2003**, *42*, 4366-4375.
7. Gerkens, C.; Heyen, G. Use of Parallel Computers in Rational Design of Redundant Sensor Networks. *Comput. Chem. Eng.* **2005**, *29*, 1379-1387.
8. Benqlilou, C.; Graells, M.; Musulin, E.; Puigjaner, L. Design and Retrofit of Reliable Sensor Networks. *Ind. Eng. Chem. Res.* **2004,** *43,* 8026-8036.
9. Carnero, M.; Hernández, J.; Sánchez, M.; Bandoni J. On the Solution of the Instrumentation Selection Problem. *Ind. Eng. Chem. Res* **2005**, *44*(2), 358-367.
10. Gerkens, C.; Heyen, G. Sensor placement for fault detection and localization. *Comp. Aided Chem. Eng.* **2008**, *25*, 355-360.
11. Glover, F.; Laguna, M. *Tabu Search*; Kluwer Academic Publishers: Norwell, MA, USA, 1997.
12. Lin, B.; Miller, D.C. Solving Heat Exchanger Network Synthesis Problems with Tabu Search. *Comp. Chem. Eng.* **2004**, *28*, 1451-1464.
13. Lin, B.; Miller, D.C. Tabu Search Algorithm for Chemical Process Optimization. *Comp. Chem. Eng.* **2004**, *28*, 2287-2306.

14. Cavin, L.; Fischer, U.; Glover, F.; Hungerbuhler, K. Multiobjective Process Design in Multi-purpose Batch Plants using a Tabu Search Optimization Algorithm. *Comp. Chem. Eng.* **2004**, *28*, 459-478.
15. Carnero, M.; Hernández, J.; Sánchez, M. Optimal Sensor Network Design and Upgrade using Tabu Search. *Comp. Aided Chem. Eng.* **2005**, *20*, 1447-1452.
16. Carnero, M.; Hernández, J.; Sánchez, M. A Tabu search procedure for sensor structure optimization. *Trans. Built Environ.* **2005**, *80*, 35-44.
17. Laguna, M.; Hossell, K.P.; Martí, R. *Scatter Search: Methodology and Implementation in C*; Kluwer Academic Publishers: Norwell, MA, USA, 2002.
18. Martí, R.; Laguna, M.; Glover, F. Principles of Scatter Search. *Eur. J. Operational Res.* **2006**, *169*, 359–372.
19. Larranaga, P. and Lozano, J. A. *Estimation of Distribution Algorithms. A New tool for Evolutionary Computation*; Kluwer Academic Publishers: Boston/Dordrecht/London, 2001.
20. Shakya, S. *Probabilistic model building Genetic Algorithm (PMBGA): A survey Technical Report*; Computational Intelligence Group, The Robert Gordon University: Aberdeen, Scotland, UK, 2003.
21. Baluja, S. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* (Technical Report CMU-CS-94-163); Carnegie Mellon University; Pittsburgh, PA, USA, 1994.
22. Martí, R. Scatter Search – Wellsprings and Challenges. *Eur. J. Operational Res.* **2006**, *169*, 351-358.
23. Wan, S.; Qiu, D.: Vehicle Rourting Optimization Problem with Time Constraint using Advanced PBIL Algortihm. In Proc. IEEE Int. Conf. on Service Operations and Logistics, and Informatics 2008; pp. 1394-1398.
24. Pang, H.; Hu, K.; Hong, Z. Adaptive PBIL Algorithm and Its Application to Solve Scheduling Problems. In Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design 2006; pp. 784-789.
25. Ghamlouche, I.; Crainic, T.G.; Gendreau, M. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Ann. Operat. Res.* **2004**, *131*, 109-133.
26. Narasimhan, S; Jordache, C. *Data Reconciliation and Gross Error Detection: An Intelligent Use of Process Data*; Gulf Publishing Company: Houston, TX, USA, 2000.
27. Glover, F. A Template for Scatter Search and Path Relinking. In *Artificial Evolution, Lecture Notes in Computer Science*; Hao, J.K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D., Eds.; Springer: Berlin, Germany, 1998; Vol. 1363, pp. 13-54.
28. Chaves, J.; Domínguez, D.; Vega, M.; Gómez, J.; Sánchez, J. Parallelizing PBIL for Solving a Real-World Frequency Assignment Problem in GSM networks. In Proc. 16th Euromicro Conference on Parallel, Distributed an Network-Based Processing (IEEE Computer Society), 2008; pp. 391-398.

Notation

A_k	Availability of the *k*-th variable estimate
$\mathbf{c}$	Acquisition cost vector
d	Hamming distance
E_k	Degree of estimability of variable *k*
f	Objective function value
F	Evaluation function
$\left\lfloor \sqrt{n} \right\rfloor$	nearest integer of $\sqrt{n}$ towards minus infinity
frecpr	PR frequency
$\mathbf{g}$	Vector of constraints
GS	Guide solution for PR
$\mathbf{h}$	Frequency based Tabu list
I_0	Initial set of instruments
IS	Initial solution for PR
Lo	Lower bound for TS
L_1	Upper bound for TS
LCC	life cycle cost of the sensor structure
LR	Learning rate
MS	Mutation amount
N	Number of process variables
N	Neighbourhood of possible solutions
N	Number of individuals for each instance of PBIL
NPBIL	Number of instances of PBIL executed in parallel
# Max Iter	Maximum Number of Iterations for TS
pt	Tabu Tenure Period
Pinteraction	Crossover probability for PBIL
PMUTA	Mutation probability for PBIL
$\mathbf{q}$	Solution vector
Q	Penalty function
$\mathbf{Q}$	Random *n*-dimensional vector
$\lvert R \rvert$	Cardinality of the Reference Set for PR
$\lvert Refset \rvert$	Cardinality of the Reference Set for SS
R_u	Number of unsatisfied constraints
$\mathbf{R}$	Process model equations
S	Set of key variables
S_σ	Set of key variables subject to precision constraints
S_σ	Set of key variables subject to estimability constraints
S_A	Set of key subject to availability constraints
$\mathbf{t}$	Recency based Tabu list
$\mathbf{u}$	Vector of unmeasured variables

$\mathbf{x}$	Vector of measured variables
$\mathbf{z}$	Vector of process variables
$\hat{\sigma}_j$	Standard deviation of the *j*-th variable estimate

Acronyms

CV	Coefficient of Variation
EDA	Estimation of Distribution Algorithm
GA	Genetic Algorithm
HGA	Hybrid Genetic Algorithm
PBIL	Population Based Incremental Learning Algorithm
PR	Path Relinking
SN	Sensor Network
SNDP	Sensor Network Design Problem
SO	Strategic Oscillation
SS	Scatter Search
TS	Tabu Search