- ORIGINAL ARTICLE -

Tuning a hybrid SA based algorithm applied to Optimal Sensor Network Design

Ajustes de un algoritmo híbrido basado en SA aplicado al diseño óptimo de redes de sensores

José Hernández[1], Carolina Salto[2,3], Gabriela Minetti[2], Mercedes Carnero[1], Carlos Bermudez[2], and Mabel Sanchez[4]

[1]*Grupo de Optimización - Facultad de Ingeniería – Universidad Nacional de Río Cuarto, Rio Cuarto, Argentina*
jlh, mcarnero@ ing.unrc.edu.ar
[2]*LISI - Facultad de Ingeniería, Universidad Nacional de La Pampa, General Pico, Argentina*
{saltoc, minettig}@ing.unlpam.edu.ar
[3]*CONICET, Argentina*
[4]*Departamento de Ingeniería Química, Universidad Nacional del Sur (UNS) and Planta Piloto de Ingeniería Química - PLAPIQUI (UNS-CONICET), (8000) Bahía Blanca, Argentina*
msanchez@plapiqui.uns.edu.ar

Abstract

Sensor network design problem (SNDP) in process plants includes the determination of which process variables should be measured to achieve a required degree of knowledge about the plant. We propose to solve the SNDP problem in plants of increasing size and complexity using a hybrid algorithm based on Simulated Annealing (HSA) as main metaheuristic and Tabu Search embedded with Strategic Oscillation (SOTS) as a subordinate metaheuristic. We studied the tuning of control parameters in order to improve the HSA performance. Experimental results indicate that a high-quality solution in reasonable computational times can be found by HSA effectively. Moreover, HSA shows good features solving SNDP compared with proposals from the literature.

Keywords: Cooling Schedule, Optimization, Sensor networks, Simulated Annealing

Resumen

El problema de diseño de una red de sensores en plantas de proceso (*Sensor Network Design Problem*, SNDP) consiste en determinar las variables de proceso que deben ser medidas, a fin de alcanzar el grado de conocimiento requerido de dicha planta. Proponemos resolver el problema SNDP en plantas de tamaño y complejidad creciente utilizando un algoritmo híbrido basado en Recocido Simulado (Hybrid Simulated Annealing, HSA) como metaheurística principal y Búsqueda Tabu con Oscilación Estratégica como metaheurística subordinada. Investigamos los ajustes de los parámetros de control para obtener el mejor desempeño del HSA. Los resultados experimentales indican que el HSA puede efectivamente encontrar una solución de buena calidad en tiempos de cómputo razonable. Mas aún, HSA muestra buenas características en la solución de SNDP en comparación con algoritmos propuestos en la literatura.

Palabras claves: Esquemas de enfriamiento, Optimization, Recocido Simulado, Redes de Sensores

1 Introduction

The monitoring in a chemical plant is absolutely crucial and it has considerable impacts on aspects such as economic, safety or control. Bearing in mind that the process state at any time must be warranted. For that purpose the information is collected by sensors distributed throughout the plant, responsible for measuring and transmitting the values of magnitudes such as temperature, humidity, pressure, etc. The set of devices used in the measurement is called the sensor network (SN, Sensor Network). The SN design is systematically made by formulating an optimization problem called Sensor Network Design Problem (SNDP), which is a discrete optimization problem. In real work scenarios, the number of variables involved is quite large and the SNDP formulation can be more or less complex depending on the performance criteria and the restrictions set imposed on it.

SNDP was formulated by Bagajewicz [1], and tackled by Nguyen and Bagajewicz [2] using a new tree search method that exploits certain cost properties of the different nodes in the tree to efficiently prune non optimal nodes using a breadth-first/level traversal tree search method to obtain the global optimum. Other approaches modeled the problem combining the integer and non-linear programming and solved it by means of depth-first or breadth-first tree searches. The main disadvantage of all these methods is that they are

highly time consuming. Consequently, to achieve an efficient optimization, it is important to have a tool that allows solving the problem for different formulations of different complexity and size.

In this sense, metaheuristic methods appear as attractive general-purpose optimization methodologies that allow to solve a wide range of formulations, including those where it is required optimize more than one performance criterion at a time, i.e. multi-objective problems. The literature presents genetic algorithms [3], swarm intelligence [4], among others population-based metaheuristics. Furthermore, hybridized metaheuristics have been reported to solve this problem, Carnero et al. [5] propose an algorithm called PBIL_SOTS, which combines estimation of distribution algorithm with a tabu search improved by using a strategic oscillation.

In contrast with the population-based metaheuristics previously mentioned, the Simulated Annealing (SA) [6] may be classified into the trajectory-based group and it has proved to be an efficient method to solve many hard combinatorial optimization problems [7]. In [8], we present a hybrid algorithm based on Simulated Annealing (HSA) as main metaheuristic and Tabu Search embedded with a Strategic Oscillation (SOTS), as a subordinate metaheuristic, to solve the SNDP. Its adaptation to the resolution of a particular problem implies, among other aspects, to make an adjustment of its parameters that can obtain the best performance of the proposed technique. Therefore, the focus has been on the tuning of the algorithmic control parameters to reach an equilibrium between the solution quality and time consumption. The current work is an extension of our previous article [8], where the objective is reformulated into three research questions:

- *RQ1: Can the HSA find high quality solutions for the SNDP? (Efficacy)*
- *RQ2: Which combination of the control parameter tuning allows to reach the best performance? (Efficiency)*
- *RQ3: Does HSA improve the techniques reported in the state-of-the-art to solve this problem? (Competitive with the state-of-the-art)*

In addition, from the analysis of the previous results we extend our work by proposing a new research question:

- *RQ4: Which combination of the control parameter adjustments impacts over the number of evaluations? (Efficiency)*

To answer the previous questions about the HSA behaviour, we address more extensively the SNDP and the HSA algorithm description. In this way, we provide a more thorough insight into both problem and algorithm descriptions. Moreover, a comparison with other methods and numerical test results is provided for several case studies, and the effectiveness of the proposed method is analyzed. The analysis of the results is enlarged by including other metrics, that allows us to improve the comprehension of the relation between the solution quality and the HSA computational effort.

The rest of this article is organized as follows. In Section 2 the SNDP is described. Sections 3 and 4 introduce and explain the approach proposed in this work. Section 5 refers to the experimental analysis and the methodology used. Sections 6 and 7 present the analysis of results and the comparison with the literature. Finally, the main conclusions and future lines of research are drawn in Section 8.

2 Sensor Network Design Problem

The SNDP is summarized as a problem of finding the minimum cost network that satisfies precision and estimability constraints. Formally, a SNDP solution has to satisfy these constraints for a set of key variable estimates, as stated by Eq (1), where $\mathbf{q}$ is an n-dimensional vector of binary variables such that $q_i = 1$ if variable i is measured, and $q_i = 0$ otherwise, $\mathbf{c}^T$ is the cost vector; $\hat{\sigma}_k$ is the estimate standard deviation of the k-th variable contained in S_σ after a data reconciliation procedure is applied [9], and E_l stands for the degree of estimability of the l-th variable included in S_E. Furthermore, S_σ and S_E are the set of key process variables with requirements in precision and ability to be estimated, respectively.

$$\min \mathbf{c}^T \mathbf{q} \tag{1}$$

subject to:

$$\begin{aligned} \hat{\sigma}_k(\mathbf{q}) \leq \sigma_k^*(\mathbf{q}) \qquad & \forall\, k \in S_\sigma \\ E_l \geq 1 \qquad & \forall\, l \in S_E \\ \mathbf{q} \in \{0,1\}^n & \end{aligned}$$

In this formulation, it is assumed that a linearized algebraic model represents plant operation, measurements are subject to noncorrelated random errors, there is only one potential measuring device for each variable, and there are no restrictions for the localization of instruments. Regarding the definition of degree of estimability, let us first denote $\mathbf{A}(p)$ as the set of all possible combinations of p measurements, and call $A_j(p)$, the j-th element (combination) of this set. The l-th variable (measured or not) has a degree of ability for estimation E_l, if it remains estimable after the elimination of any combination $A_j(E_l 1) \in \mathbf{A}(E_l 1)$ and it becomes unobservable when at least one set, $A_j(E_l) \in \mathbf{A}(E_l)$, is eliminated [10]. If E_l is set equal to one, the feasibility of the constraint can be checked by executing a variable classification procedure, which can be accomplished by matrix projection, QR decomposition, or matrix co-optation [11, 12].

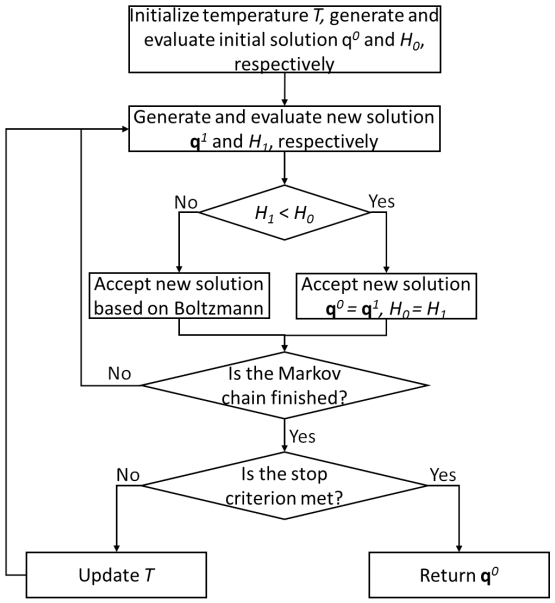

Figure 1: Simulated Annealing flowchart.

3 Simulated Annealing Algorithm

Simulated Annealing is a well-studied trajectory-based metaheuristic used to address discrete and, to a lesser extent, continuous optimization problems. SA is based on the analogy to the annealing process of metal and glass, which assumes a low energy configuration when cooled with an appropriate cooling schedule. The SA algorithm simulates the energy changes in a system subjected to a cooling process until it converges to an equilibrium state (steady frozen state), where the physical material states correspond to problem solutions, the energy of a state to cost of a solution, and the temperature to a control parameter.

At the beginning (with a high temperature), SA accepts solutions with high cost values under a certain probability (Boltzmann criterion) in order to explore the search space and to escape from local optima. During the annealing process this probability decreases according to the cooling temperature, intensifying the search and reducing the exploration in order to exploit a restricted area of a search space. Fig. 1 shows the flowchart of this algorithm.

Simulated Annealing evolves by a sequence of transitions between states and this sequence is generated by transition probabilities. Consequently, SA can be mathematically modeled by Markov chains. Each chain is generated by a transition probability, which is computed involving the current temperature.

Most of the search components of SA are fixed in function of the problem to be solved. Consequently, the search space, cost (evaluation) function, perturbation operator, and local search are directly related to the problem. The main search components, which are variable during the process, are the initial temperature, the temperature through their annealing schedules (see Subsection 3.2), and the Markov chain length [7] (explained in Section 3.3).

Algorithm 1 Pseudocode for setting T_0

```
1: function initTemp (T_s)
2: T_0 = T_s;
3: while acceptability rate is not reached do
4:     update T_0 ;
5:     generate(q^1);
6:     H_1=evaluate(q^1);
7:     for i=0 to test do
8:         q^2 = perturbation_Operator(q^1);
9:         H_2=evaluate(q^2);
10:        if (H_2 < H_1) or (exp^((H_2-H_1)/T) > random(0,1)) then
11:            q^1=q^2;
12:            H_1 = H_1;
13:        end if
14:    end for
15: end while
16: return T_0
```

3.1 Initial Temperature

One of the most important issues in SA is the choice of the right initial temperature, which must not be excessively high to conduct a random search for a period of time but high enough to allow moves to almost neighborhood state. The classical and intuitive method consists of computing a temperature such that the acceptance ratio is approximately equal to a given value χ_0 [6]. Given a T_s seed temperature, the initial temperature is computed by the procedure shown in Algorithm 1. The output, T_0, is determined such that, when applying the Boltzmann criterion, worse solutions are accepted with a high probability value. To achieve this, the algorithm starts from a T_s that is increased until the acceptance ratio is reached.

3.2 Cooling Process

The scheme to control the annealing or cooling process is also crucial, so that the system gradually cools from a higher temperature, ultimately freezing to a global minimum state. Many attempts have been made to derive or suggest good schedules [7], being the most known cooling processes: the proportional [6], exponential [6], and logarithmic [13] schemes.

Proportional Scheme. In this case, the temperature is updated using the Eq (2), where α is a constant close to, but smaller than, 1 and calculated as the Eq (3) shown. This scheme is the most popular cooling function, since the temperature decay is neither too slow nor too fast allowing to achieve an equilibrium between exploitation and exploration.

$$T_{k+1} = \alpha \times T_k \tag{2}$$

$$\alpha = \frac{k}{k+1} \tag{3}$$

Exponential Scheme. The exponential cooling scheme produces the temperature decay by applying the Eq (4), where the constant $\alpha^k < 1$ is calculated as

presented in the Eq (5). This schedule quickly cools the temperature reducing the required time and iterations to converge to a good solution. In large and complex problems, this becomes in a disadvantage, given that the equilibrium between the exploitation and exploration is broken.

$$T_{k+1} = T_k \times \alpha^k \tag{4}$$

$$\alpha^k = \frac{e^k}{e^{k+1}} \tag{5}$$

Logarithmic Scheme. This cooling scheme modifies the temperature, as shown in the Eq (6). The chain converges to a global and minimal energy value, where the constant C is computed as the Eq (7) indicates. This schedule is too slow to be applied in practice but has the property of the convergence proof to a global optimum [14].

$$T_{k+1} = C \times T_k \tag{6}$$

$$C = \frac{ln(k)}{ln(k+1)} \tag{7}$$

Random Scheme. This last variant is a random schedule [15], which combines the three previous cooling schemes in only one schedule process. In each iteration, one of these schemes are randomly selected in order to reduce the temperature. In this way, we try to enhance the SA by aggregating the advantages of these three schemes and mitigating their disadvantages.

3.3 Markov Chain Length

The Markov chain length (MCL) is the number of required transitions (moves) to reach the equilibrium state at each temperature. This number can be either static or adaptive. At the first case, the number of movements is calculated before the search starts. The static approach, named MCL*s*, assumes that each temperature T_k is held constant for a sufficient and fixed number of iterations. In this work, each T_k is held constant for 30 iterations, a common number used in the scientific community.

For the adaptive way, the Markov chain length depends on the characteristics of the search. For instance, Cardoso et al. [16] consider that the equilibrium state is not be necessarily attained at each level of the temperature. Consequently, the cooling schedule is applied as soon as an improved candidate (neighbor) solution is generated. In this way, the computational effort can be drastically reduced without compromising the solution quality. This approach is referred as MCLa1. Another adaptive approach is propose by Ali et al. [17], named as MCLa2, which uses both the worst and the best solutions found in the Markov chain (inner loop) to compute the next MCL. This strategy allows the possibility of increasing the number of function evaluations at a given temperature if the difference between the worst and the best solutions increases, but the whole of the increased Markov chain will not be required if an improved solution is found.

4 Hybrid Simulated Annealing Algorithm for SNDP

In Hernández et al. [18], we propose an adapted and hybridized SA algorithm to solve the SNDP in chemical plants. SA works as main metaheuristic with a subordinated ad hoc local search, inspired in tabu search with strategic oscillation technique, SOTS, giving rise to the Hybrid Simulated Annealing algorithm. This hybridization is applied in two levels: in the first one to generate an initial solution, and in the second level to improve the solution during the annealing process.

The perturbation scheme of the current solution is carried out through a certain swapping number of measured variables to unmeasured ones and vice versa in order to generate a candidate solution $\mathbf{q}^2$ from $\mathbf{q}^0$. This swap mutation is applied over each variable with a certain probability (called Pswap) [19]. Furthermore, the temperature is updated using the geometric criterion.

The first issue of algorithm design is related to the representation of a solution. In equation 3 it is assumed that each variable can be measured with only one type of instrument. In this case, a solution to the SNDP problem is an n-dimensional vector $\mathbf{q}$= $\{q_1, q_2, \ldots, q_n\}$ of binary variables, where $q_i = 1$ if variable i is measured, and $q_i = 0$ otherwise, as explained in Section 2. In order to evaluate a solution, the Eq (8) is used as an objective function, where $\sum_{i=1}^{n} c_i$ is the cost of the SN when all variables are measured (upper bound of the SNDP objective function). When restrictions are not met, the Eq (9) computes $S(\mathbf{q})$, where *rno* and *rnp* represent the number of observability and precision constraints not met for a solution vector $\mathbf{q}$.

$$H = \begin{cases} \sum_{i=1}^{n} c_i q_i & \text{if } \mathbf{q} \text{ is feasible} \\ \sum_{i=1}^{n} c_i (1 + S(\mathbf{q})) & \text{if } \mathbf{q} \text{ is infeasible} \end{cases} \tag{8}$$

$$S(\mathbf{q}) = \frac{rno}{S_E} + \frac{1}{rnp} \sum_{i=1}^{r} np \frac{\hat{\sigma}_i - \sigma_i^*}{\hat{\sigma}_i} \tag{9}$$

The pseudo-code of the HSA algorithm proposed to solve the SNDP optimization problem is shown in Algorithm 2. HSA begins with the initialization of the temperature (line 2). After that, HSA generates an initial solution $\mathbf{q}^0$ applying a specific-heuristic proposed in [20] as the first hybridization level (line 3), which is then evaluated (line 4). Once the initialization process ends, an iterative process starts (lines 5 to 24). As a first step in the iteration, the second level of hybridization is carried out in order to intensify the search into

Algorithm 2 HSA Algorithm to solve the SNDP

```
1: k = 0;
2: T=initTemp(T_s);
3: generate(q^0) by means of the first hybridization level;
4: H_0 = evaluate(q^0);
5: repeat
6:     repeat
7:         k = k + 1;
8:         if random(0,1) < Pso then      ▷ Pso: probability to apply SOTS
9:             q^1 = SOTS_LS(q^0);
10:            H_1=evaluate(q^1);
11:            if (H_1 < H_0) or (exp^((H_1−H_0)/T) > random(0,1)) then
12:                q^0=q^1;
13:                H_0 = H_1;
14:            end if
15:        end if
16:        generate q^2 = Pswap(q^0);
17:        H_2 = evaluate(S_2);
18:        if (H_2 < H_0) or (exp^((H_2−H_0)/T) > random(0,1)) then
19:            q^0 = q^1;
20:            H_0 = H_2;
21:        end if
22:    until (k mod MCL) == 0
23:    update(T);
24: until stop criterion is met
25: return S_0;
```

the current region of the solution space. In this way a solution, $\mathbf{q}^1$, is obtained by applying the SOTS local search to $\mathbf{q}^0$ (line 9), and then the Boltzmann criterion is applied to accept $\mathbf{q}^1$ (lines 11-14). In the next step, the Pswap perturbation operator is used to obtain a neighbor, $\mathbf{q}^2$, from $\mathbf{q}^0$ (line 16), in order to explore another areas of the search space. If $\mathbf{q}^2$ is worse than $\mathbf{q}^0$, $\mathbf{q}^2$ can be accepted under the Boltzmann probability (line 18, second condition). In this way, at high temperatures (T) the exploration of the search space is strengthened. In contrast, at low temperatures the algorithm only exploits a promising region of the solution space, intensifying the search. In order to update T, a cooling schedule is used (line 22) and it is applied after a certain number of iterations (k) given by the Markov Chain Length (*MCL*) (line 23). Finally, SA ends the search when the stop criterion is met (1250 iterations of the mean loop).

The second design issue involves the main search components that are variable during the process. In order to study the impact of different initial temperatures in the performance of the HSA, we use T_s values belonging to $\{1,900\}$ from small to large seeds. In this way, we test very dissimilar seeds allowing a different number of HSA's main loop iterations. When the cooling scheme is studied, we propose four different HSA approaches, as introduced earlier: $\boldsymbol{HSA}_{Prop}$ that adopts the proportional annealing schedule, $\boldsymbol{HSA}_{Exp}$ uses the exponential cooling scheme, $\boldsymbol{HSA}_{Log}$ employs the logarithmic schedule, and $\boldsymbol{HSA}_{Rand}$ that applies the random scheme. Finally, we also consider three different ways to compute the Markov chain length, as explained in Section 3.3. By combining all the aforementioned approaches, a total of 24 HSA's variants are obtained to solve the SNDP.

5 Experimental Design

In order to evaluate HSA performance, a test set of 5 design problems were considered, which includes processes of different complexity and size and whose operation can be represented by both linear and nonlinear models. The first one involves 11 units and 28 streams. Case study 2 is a continuous stirred tank reactor (CSTR) [21], which is composed of 13 variables (total flow rates, compositions, and temperatures) and 5 mass and energy balances. The mineral flotation problem [22], MFP, is selected as case study 3. Its model is bilinear and composed of 24 variables (8 flow rates and 16 compositions) related by total and component mass balances. For case studies 2 and 3, the model is linearized around the nominal operation point of the process. Finally, case studies 4 and 5 correspond to large-scale process flow sheets, and variables are related by total mass balances (case study 4, 19 units and 52 streams; case study 5, 47 units and 82 streams). The interested readers can gain access to the file containing information about the case studies from `https://www.ing.unrc.edu.ar/archivos/sndp_cases.doc`. The standard deviation of flow meters is 2.5%, 1%, 2%, 2%, and 2% of the corresponding true flow rates for case studies 1-5, respectively. The complexity of the set of constraints imposed on all case studies can be found in [5].

The computational environment used in this work to carry out the experimentation consists of computers with Processor Intel Core i5 CPU 4440 @ 3.10 GHz, 4GB RAM, using MatLab R2011b. Because of the stochastic nature of the algorithms, 30 independent runs of each design problem were performed to gather meaningful experimental data and statistical confidence metrics were applied to validate the results and conclusions. As a result, a total of 3600 executions (24 HSA variants $\times$ 5 design problems $\times$ 30 runs) were carried out. Before performing the statistical tests, we first checked whether the data followed a normal distribution by applying the Shapiro–Wilks test. Where the data was distributed normally, we later applied an ANOVA test. Otherwise, we used the Kruskal–Wallis test. This statistical study allows us to assess whether or not there were meaningful differences between the compared algorithms with a confidence level of 99%.

6 Analysis of HSA Results

In this section, we summarize and analyze the results of using the HSA's variants proposed in this work on the all SNDP case studies in order to answer the *RQs* formulated in the Section 1. First, we study the solution quality of the 24 HSA's variants: HSAProp, HSAExp, HSALog, and HSARand with T_s values belonging to $\{1,900\}$ and the three ways to compute the Markov chain length. The tables 1 and 2 present

Table 1: Best and median cost values found by these 24 variants for the all cases and its percentage of hits

Case	Algorithms		$T_s = 1$			$T_s = 900$			KW
Study	*MCL*	*Cooling Scheme*	*Min.*	*Median*	*% hits*	*Min.*	*Median*	*% hits*	
1	All	All	**1106.46**	1106.46	**100**	**1106.46**	1106.46	**100**	=
2	All	All	**735**	735	**100**	**735**	735	**100**	=
3	All	All	**2928**	2928	**100**	**2928**	2928	**100**	=
4	All	All	**1154.34**	1154.34	**100**	**1154.34**	1154.34	**100**	=

Table 2: Best and median cost values found by these 24 variants for the Case Study 5

Algorithms		$T_s = 1$		$T_s = 900$		KW
MCL	*Cooling Scheme*	*Min.*	*Median*	*Min.*	*Median*	
MCLs	HSAProp	50845.37	54974.18	**50845.16**	**50845.16**	$\neq$
	HSAExp	**50845.16**	54974.18	**50845.16**	54974.18	=
	HSALog	50846.39	54974.18	**50845.16**	50846.18	$\neq$
	HSARand	50845.37	52909.78	50845.37	54974.18	$\neq$
MCLa1	HSAProp	**50845.16**	54973.16	**50845.16**	**50845.16**	$\neq$
	HSAExp	**50845.16**	52909.67	**50845.16**	52909.16	=
	HSALog	**50845.16**	50845.67	**50845.16**	**50845.16**	$\neq$
	HSARand	**50845.16**	**50845.16**	**50845.16**	**50845.16**	=
MCLa2	HSAProp	**50845.16**	52909.16	**50845.16**	**50845.16**	=
	HSAExp	**50845.16**	**50845.16**	**50845.16**	**50845.16**	=
	HSALog	**50845.16**	**50845.16**	**50845.16**	**50845.16**	$\neq$
	HSARand	**50845.16**	**50845.16**	**50845.16**	**50845.16**	$\neq$

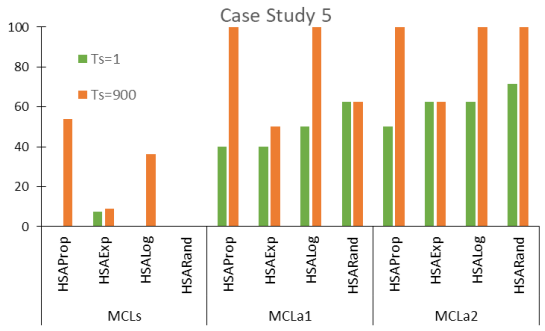

Figure 2: Hits of each algorithmic variant for case study 5.

the best and median cost values found by these 24 variants for all case studies, besides the percentage of hits (%hits) is included in Table 1 for cases 1-4 and in Fig. 2 for the fifth case. In these tables the best known cost values are boldfaced. Finally, we analyze the HSA's variants performance considering the average number of evaluations to find the best solution and the average execution total time in the figures 3 and 4, respectively.

6.1 Solution Quality

From the analysis of the result quality, an important separation of the case studies is observed. For the first four cases (see Table 1), the all algorithmic variants find the best known solution in each execution. However when the case study 5 is solved (see Table 2 and Fig. 2), different behaviors between the proposed algorithms is detected. These differences are statistically corroborated using the Kruskal-Wallis (KW) test with a confidence level, $\alpha = 0.01$.

The results for the fifth case study deserves a detailed analysis. If the T_s parameter is considered, the highest percentages of hits are reached for $T_s = 900$, and the optimal solution is found by 11 of 12 HsA's variants. This selection for T_s is statistically supported by the median values because they are equal to the optimum in 8 of 12 opportunities against 4 times when $T_s = 1$. Analyzing the three MCL options, a significant improvement in the solution quality is observed if the adaptive variants are applied. In other words, only the application of MCLa1 or MCLa2 in HSA warranties to find the optimum. Finally, the results are assessed considering the cooling scheme. None direct relation between the kind of cooling process and the result quality is observed, but the behavior of these schedules is highly dependent of the combination of the T_s value and MCL option. For example, when the parametric configuration $T_s = 1$ and MCLs is applied the best results are found by HSAExp, but if $T_s = 900$ the best option is HSAProp.

The results of the different HSA's variants indicates that they find the optimal solution for the all study cases. As a consequence, the *RQ1* related with the efficacy of the proposals is answered affirmatively .

6.2 Performance Analysis

The performance is analysed considering two factors: the number of evaluations to find the best solution and

Figure 3: Number of evaluations to find the best solution for each algorithmic variant including all case studies.

the total execution time. Attending the first performance factor for the case studies 1, 4, and 5, HSA with $T_s = 900$ requires less number of evaluations to find its best solutions against $T_s = 1$, as Fig. 3 shows. This observation is reversed for the third case, while for the second one no differences are detected, as the KW tests proved.

Regarding the total execution time values (see Fig. 4), the HSA's variants using MCLa2 spend a significant greater time than the other two MCL options. Being MCLs, the way to compute the Markov Chain length that minimizes the runtime for the all case studies. When the T_s values and schedule schemes are assessed together, HSA presents similar execution times

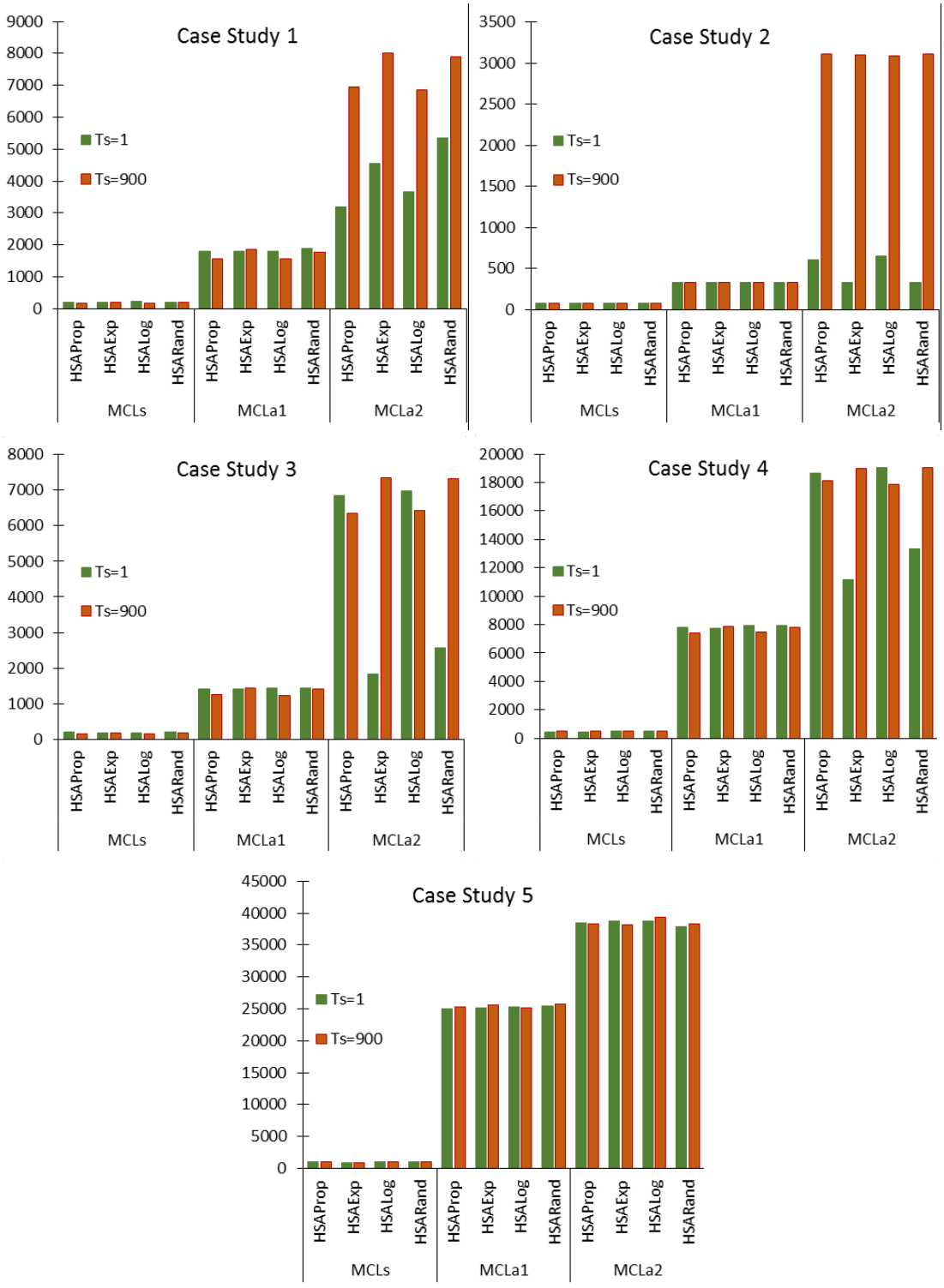

Figure 4: Total time (in seconds) spent by each algorithmic variant for all case studies.

for MCLs and MCLa1 in all study cases. Our conclusions are supported by the KW tests performed.

6.3 Discussion

The HSA's variants that implement MCLs maximize the performance, obtaining the best known solution for the first four case studies in every run. For the case study 5, the MCLs application allows to reach a relatively high percentage of hits with $T_s = 900$ and the proportional cooling scheme (HSAProp). Furthermore, a 100% of hits is achieved by the most time consuming HSA's variants (MCLa1 and MCLa2). However, the

Table 3: Comparison of the best HSA variant and PBIL_SOTS, considering best solution statistics for each case study.

Case study	Best solution		Mean best solution		KW
	PBIL_SOTS	Best HSA variant	PBIL_SOTS	Best HSA variant	
1	1106.50	1106.50	1106.50±0.00	1106.46±0.00	=
2	735.00	735.00	735.00±0.00	735.00±0.00	=
3	2928.00	2928.00	2929.20±0.69	2928.00±0.00	=
4	1154.34	1154.34	1154.34±0.00	1154.34±0.00	=
5	50845.16	50845.16	50886.63±41.29	50845.16±0.00	$\neq$

MCLa1 application is significantly less expensive that the MCLa2 one.

Considering the first four study cases, the HSA's variants that implement MCLs always obtain the best solution with the maximum performance, showing statistically similar behaviors (KW test with $\alpha = 0.01$). But if the complexity of the case to solve grows (study case 5), a trade-off between quality and time must be achieved. In this sense, the answer to *RQ2* and *RQ4* is summarized as follows: the best algorithmic approaches to solve SNDP are HSAProp and HSALog with $T_s = 900$ and the application of MCLa1 to calculate the Markov chain length.

7 Comparison of HSA Variants against a Literature Approach

In this section, we compare the behavior of the best algorithmic variants of HSA versus other well-known algorithm found in the literature for solving these SNDP case studies, in order to answer the *RQ3*. In this sense PBIL_SOTS, introduced by Carnero et al. [5], has recently reported results for these cases.

To compare this state-of-the-art algorithm versus the best HSA's variant, we use the averages and standard deviations for the five case studies, as shown in the Table 3. Furthermore, a KW test is carried out to corroborate the similarities or differences between them. In general, we can observe that the PBIL_SOTS behaves similarly to the HSA in the least complex case studies. However, statistically different behaviors are observed for the fifth case study. In this sense, two advantages in favor of HSA are observed: *i)* the average best solution is equal to the optimum, and *ii)* the optimum is found in each run because the standard deviation is equal to zero. In this way, we can say that the answer to the *RQ3* is affirmative.

8 Conclusions

This paper examines the appropriateness of using a hybrid SA with the SOTS method to solve the SNDP. The focus is on the analysis of the found solution quality and the performance of each HSA's variant, which arise from the combination of the algorithmic control parameter tuning. As a consequence, 24 algorithmic variants are introduced by considering two initial temperature values ($T_s \in \{1, 900\}$), four cooling schemes (proportional, exponential, logarithmic, and random), and tree ways to calculate the Markov Chain Length (one static and two adaptive). We empirically assess the effectiveness of the HSA's variants in terms of solution quality, execution time, and number of evaluations to answer the research questions formulated at the beginning of this work. This assessment is carried out by using five SNDP case studies of a growing complexity, which represent different degrees of difficulty.

The results we had obtained answer our four research questions regarding efficacy, efficiency, and competitive with a state-of-the-art algorithm. We conclude that the HSA's variants, which compute statically the Markov chain length (MCLs), always obtain the best solution with the minimum effort for the case studies 1, 2, 3, and 4. Instead, for the most complex case (the fifth one), a trade-off between quality and time is achieved when HSA sets the initial temperature to a high value ($T_s = 900$), uses an adaptive MCL (MCLa1), and applies the proportional cooling scheme. Finally, we concluded that HSA algorithm is a competitive algorithm in solving realistic SNDP cases.

A challenging extension of this work will be to tackle other SNDP formulations. The idea behind this future work is to improve the SA main heuristic by introducing different specific local search mechanisms.

Competing interests

The authors have declared that no competing interests exist.

Funding

The authors acknowledge the support of Universidad Nacional de La Pampa, Universidad Nacional de Río Cuarto, and the Incentive Program from MINCyT. The second author is also funded by CONICET.

Authors' contribution

All authors conceived the idea, wrote the program, conducted the experiments, analyzed the results, and finally, wrote and revised the manuscript. Moreover, all authors read and approved the final manuscript.

References

[1] M. Bagajewicz, "Design and retrofit of sensor networks in process plants," *AIChE Journal*, vol. 43, no. 9, pp. 2300–2306, 1997.

[2] D. Nguyen and M. Bagajewicz, "New efficient breadth-first/level traversal tree search method for the design and upgrade of sensor networks," *AIChE Journal*, vol. 57, no. 5, pp. 1302–1309, 2011.

[3] C. Gerkens and G. Heyen, "Use of parallel computers in rational design of redundant sensor networks," *Computers and Chemical Engineering*, vol. 29, no. 6 SPEC. ISS., pp. 1379–1387, 2005.

[4] Y.-J. He and Z.-F. Ma, "Optimal design of linear sensor networks for process plants: A multi-objective ant colony optimization approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 135, pp. 37–47, 2014.

[5] M. Carnero, J. L. Hernández, and M. Sánchez, "Optimal sensor location in chemical plants using the estimation of distribution algorithms," *Industrial & Engineering Chemistry Research*, vol. 57, no. 36, pp. 12149–12164, 2018.

[6] S. Kirkpatrick, C. G. Jr, and M. Vecchi, "Optimization by simulated annealing," *Science*, no. 220, pp. 671–680, 1983.

[7] E. Talbi, *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

[8] J. Hernandez, C. Salto, G. Minetti, M. Carnero, C. Bermudez, and M. Sanchez, "Optimal instrumentation: Adjustment and hybridization of a simulated annealing based technique," in *XXV Congreso Argentino de Ciencias de la Computación (CACIC 2019)*, pp. –, Oct. 2019.

[9] M. Bagajewicz and M. Sánchez, "Reallocation and upgrade of instrumentation in process plants," *Computers Chemical Engineering*, vol. 24, no. 8, pp. 1945 – 1959, 2000.

[10] M. Bagajewicz and M. Sánchez, "Design and upgrade of nonredundant and redundant linear sensor networks," *AIChE Journal*, vol. 45, no. 9, pp. 1927–1938, 1999.

[11] J. Romagnoli and M. Sanchez, *Data Processing and Reconciliation for Chemical Process Operations*. Academic Press: San Diego, CA, 2000.

[12] S. Narasimhan and C. Jordache, *Data Reconciliation and Gross Error Detection*. Gulf Publishing Company: Houston, TX, 2000.

[13] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, 1988.

[14] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[15] C. Bermudez, C. Salto, and G. Minetti, "Solving the multi-period water distribution network design problem with a hybrid simulated anealling," in *Computer Science – CACIC 2018* (P. Pesado and C. Aciti, eds.), (Cham), pp. 3–16, Springer International Publishing, 2019.

[16] M. F. Cardoso, R. L. Salcedo, and S. F. de Azevedo, "Nonequilibrium simulated annealing: A faster approach to combinatorial minimization," *Ind. Eng. Chem. Res*, vol. 33, pp. 1908–1918, 1994.

[17] M. Ali, A. Törn, and S. Viitanen, "A direct search variant of the simulated annealing algorithm for optimization involving continuous variables," *Computers & Operations Research*, vol. 29, no. 1, pp. 87 – 102, 2002.

[18] J. Hernandez, C. Salto, G. Minetti, M. Carnero, and M. C. Sanchez, "Hybrid simulated annealing for optimal cost instrumentation in chemical plants," *Chemical Engineering Transactions*, vol. 74, pp. 709–714, May 2019.

[19] K.-L. Du and M. N. S. Swamy, *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*. Birkhäuser Basel, 1st ed., 2016.

[20] M. Carnero, J. Hernández, and M. Sánchez, "A new metaheuristic based approach for the design of sensor networks," *Computers Chemical Engineering*, vol. 55, pp. 83 – 96, 2013.

[21] M. Bhushan and R. Rengaswamy, "Design of sensor network based on the signed directed graph of the process for efficient fault diagnosis," *Industrial & Engineering Chemistry Research*, vol. 39, no. 4, pp. 999–1019, 2000.

[22] H. Smith and N. Ichiyen, "Computer adjustment of metallurgical balances," *Computer Applications and Process Control. (CIM) Bulletin*, vol. 66, pp. 97–100, 1973.

Citation: J. Hernandez, C. Salto, G. Minetti, M. Carnero, C. Bermudez, and M. Sanchez.
Tuning a hybrid SA based algorithm applied to Optimal Sensor Network Design. Journal of Computer Science & Technology, vol. 20, no. 1, pp. 23-32, 2020.
DOI: 10.24215/16666038.20.e03.
Received: February 17, 2020 **Accepted:**April 17, 2020