Optimal Instrumentation: Adjustment and Hybridization of a Simulated Annealing Based Technique

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

[1] Grupo de Optimización - Facultad de Ingeniería – Universidad Nacional de Río Cuarto, Rio Cuarto, Argentina
{jlh, mcarnero}@ ing.unrc.edu.ar
[2] Laboratorio de Investigación en Sistemas Inteligentes (LISI) - Facultad de Ingeniería - Universidad Nacional de La Pampa– General Pico, Argentina - [3] CONICET
{saltoc, minettig}@ ing.unlpam.edu.ar
[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. This work presents an analysis on the behavior of a particular metaheuristic technique inspired by Simulated Annealing, SA, for the resolution of a problem of sensor network design in process plants, SNDP. This design problem is formulated as a combinatorial optimization problem and includes the selection and determination of the number of process variables that must be measured to achieve the specific state of knowledge about the plant. Different parametric configurations for the proposed heuristic are analyzed and discussed, as well as their performance in plants of increasing size and complexity.

Keywords: sensor network, combinatorial optimization, metaheuristics.

1 Introduction

In a chemical plant, having a knowledge of the process state at any time is absolutely crucial with impact on aspects such as economic, safety or control. In other words, the monitoring of the plant must be guaranteed. 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 general, the number of variables that are involved in these problems for a real work scenario is quite large and the formulation can be more or less complex depending on the performance criteria and the restrictions set imposed on it.

Therefore, to use an efficient optimization tool that allows to solve the problem for different complexity and size formulations is an important topic.

There are exact methods that can find an exact solution to the problem by providing a global optimum. For real plant problems with a high number of variables, these strategies need huge amounts of resources and cannot offer a solution in acceptable times. Heuristics, on the other hand, can provide good solutions, from an industrial point of view, they take less time to compute and, for the last decades, they have been paid special attention to solve difficult engineering problems.

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 too time consuming. In order to mitigate this disadvantage, different metaheuristic methods were proposed to solve mono or multi-objective problem instances, such as 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] proposed the PBIL_SOTS technique, which combines estimation of distribution algorithm with a tabu search improved by using an oscillation strategy.

The Simulated Annealing (SA), is a simple, general purpose Monte-Carlo method, which was developed for combinatorial optimization [6]. In contrast with the population-based metaheuristics aforementioned, it 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]. However, heuristics are, in most cases of general purpose. Its adaptation for 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.

The goal of this work is to solve the Optimal SNDP applying a SA hybridized with heuristic methods, focusing in the adjustment (or tuning) of the algorithmic control parameters to reach an equilibrium between the solution quality and time consuming. A comparison with other methods and numerical results of tests on several instances are given, and the effectiveness of the proposed method is analyzed.

The rest of this article is organized as follows. In Section 2 the SNDP is described. Section 3 introduces and explains the approach proposed in this work. Section 4 refers to the experimental analysis and the methodology used. Finally, the main conclusions and future lines of research are drawn in Section 5.

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 **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, 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.

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{q} \\ \text{s.t.} \quad & \\ & \hat{\sigma}_k(\mathbf{q}) \le \sigma_k^*(\mathbf{q}) \qquad \forall\, k \in S_\sigma \\ & E_l(\mathbf{q}) \ge 1 \qquad \forall\, l \in S_E \\ & \mathbf{q} \in \{0,1\}^n \end{aligned} \tag{1}$$

In this formulation, measurements are subject to non-correlated random errors; there is only one potential measuring device for each variable, and there are no restrictions for the localization of instruments. The feasibility of the constraints can be checked by executing a variable classification and data reconciliation procedure.

3 Hybrid Simulated Annealing Algorithm

In this section, a description of SA and its variants is introduced. After that, the Hybrid Simulated Annealing, HSA, to optimize the cost instrumentation in chemical plants is explained and how the SA variants are adapted the HSA.

3.1 Simulated Annealing Algorithm

Simulated annealing is a well-studied trajectory-based metaheuristic used to address discrete and, to a lesser extent, continuous optimization problems. 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 in order to explore the search space and to escape from local optima. During the annealing process this probability decreases according to temperature cooling; intensifying the search and reducing the exploration in order to exploit a restricted area of a search space.

Simulated annealing evolves by a sequence of transitions between states and these transitions are generated by transition probabilities. Consequently, SA can be mathematically modeled by Markov chains, where a sequence of chains is generated by a transition probability, which is calculated 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, and the Markov chain length [7].

Therefore, one of the most important issues in SA is the choice of the right initial temperature, which must not be too high to conduct a random search for a period of time but high enough to allow moves to almost all neighborhood states. The classical and intuitive method consists in computing a temperature such that the acceptance ratio is approximately equal to a given value [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. T_0 achieve this, the algorithm starts from a T_s that is increased until the aforementioned acceptance is reached.

The scheme to control the annealing or cooling process is also crucial, so that the system cools gradually from a higher temperature, ultimately freezing to a global minimum state. Many attempts have been made to derive or suggest good schedules [7]. The most known cooling process in the literature are proportional, exponential [6], and logarithmic schemes [8]. Furthermore, a random schedule is considered [9].

In the proportional one, the temperature is updated using Eq. (2), where α is a constant close to, but smaller than, 1 and calculated as Eq. (3) shows. This scheme is the most popular cooling function, since the temperature decay is not too slow neither too fast allowing to achieve an equilibrium between exploitation and exploration.

$$T_{k+1} = \alpha T_k \quad \textbf{(2)} \qquad \alpha = k/(k+1) \quad \textbf{(3)}$$

The exponential cooling scheme produces the temperature decay by applying Eq.(4), where the constant $\alpha^k<1$ is calculated as presented in Eq. (5). This schedule quickly cools the temperature reducing the required time and iterations to converge to a good solution. In big and complex problems, this becomes in a disadvantage, given that the equilibrium between the exploitation and exploration is broken.

$$T_{k+1} = \alpha^k T_k \quad \textbf{(4)} \qquad \alpha = e^k/e^{k+1} \quad \textbf{(5)}$$

Algorithm 1: Pseudocode of algorithm for setting initial temperature T_0

```
Function init_temp(Ts)
initialize  T0 = Ts;
while (acceptability rate is not reached)
   Increment T0;
   generate a solution q1 ;
   evaluate q1 in H1
   for i=0 to test
      generate a new solution q2from q1 applying Swap mutation under Pswap;
      evaluate q2 in H2;
      Apply Boltzmann criterion and count solutions that was
      accepted;
   end
end
return T0;
```

The logarithmic schedule modifies the temperature, as shown in Eq. (6), the chain converges to a global and minimal energy value, where the constant C is computed as 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 [10].

$$T_{k+1} = CT_k \quad \textbf{(6)} \qquad C = \log(k)/\log(k+1) \quad \textbf{(7)}$$

Finally, the random schedule combines the three previous cooling schemes in only one schedule process. At each iteration, this scheme randomly selects one of these schemes in order to reduce the temperature. In this way, the advantages of these three schemes are aggregated and their disadvantages mitigated.

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, it is calculated before the search starts. For instance, a given proportion of the neighborhood of the current solution is explored. Another static approach assumes that each temperature T_k is held constant for a sufficient and fixed number of iterations.

For the adaptive way, the Markov chain length depends on the characteristics of the search. For instance, [11] consider that to reach the equilibrium state at each temperature is not necessary, i.e., the cooling schedule is applied as soon as an improving candidate (neighbor) solution is generated. In this way, the computational effort can be drastically reduced without compromising the solution quality. Ali et al. [12] propose another adaptive approach, which uses both the worst and the best solutions found in the Markov chain (inner loop) to compute the next MCL.

3.2 Hybrid Simulated Annealing Algorithm for Optimal Cost Instrumentation in Chemical Plants

In Hernández [13], an adapted and hybridized SA algorithm to solve the SNDP in chemical plants was proposed. SA works as main heuristic with a subordinated ad-hoc local search, inspired in tabu search with strategic oscillation technique, SOTS, giving rise to the Hybrid Simulated Annealing (HSA_SOTS) algorithm. The hybridization in HSA_SOTS 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*). Furthermore, the temperature is updated using the geometric criterion [14].

In this work, we study an important algorithm design issue that involves the main search components and are variable during the process i.e, the temperature and the MCL. In order to study the impact of different initial temperatures in the performance of the HSA_SOTS, we use *Ts* values belonging to {1,900} from small to large seeds. In this way, we test very dissimilar seeds allowing a different number of HSA_SOTS's main loop iterations. When the cooling scheme is studied, we propose four different HSA approaches: HSAProp that adopts the proportional annealing schedule, HSAExp uses the exponential cooling scheme, HSALog employs the

logarithmic schedule, and HSARand that applies a random cooling scheme. Finally, we consider three different ways to compute the MCL. At the first, MCL is statically calculated and each *t* is held constant for 30 iterations (number commonly used in the literature), named MCLs. The other two ways to implement the adaptive techniques proposed by Cardoso et al. [11] and Ali et al. [12] identified as MCLa1 and MCLa2, respectively.

4 Experimental Design

In order to evaluate HSA_SOTS performance, a test set of 5 design problems were considered which comprise processes of different complexity and size and whose operation can be represented by both linear and non-linear models. The sizes of the considered instances range from 28 to 82 decision variables. 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 complexity of the set of constraints imposed on all case studies can be found in [5]. The stop condition of the HAS_SOTS's variants is to reach 1250 iterations of the mean loop.

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 instance 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 7200 executions were carried out.

4.1 Analysis of Results

In this section, we summarize and analyze the results of using the HSA_SOTS's variants proposed in this work on the all problem instances. First, we analyze the behavior of the variants HSAProp, HSAExp, HSALog, and HSARand with T_s values belonging to {1,900} and the three ways to compute the Markov chain length. Table 1 presents the best and median cost values found by these 24 variants for all instances, besides the percentage of hits (%hits). In this table the best results are boldfacing. Figure 1 shows the average execution total time for each algorithmic variant and case study.

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

Consequently, the results for the fifth case study deserves a detailed analysis. If the *Ts* parameter is considered, the highest percentages of hits are reached for *Ts*=900, and the optimal solution is found by 11 of 12 proposed approaches. This selection for T_s is statistically supported by the median values because they are equal to the optimal in 8 of 12 opportunities against 4 for 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_SOTS warrantees 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.

From the computational effort point of view, the HSA_SOTS's variants that implement MCLs minimize this effort, 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) is used. Furthermore, a 100% of hits is achieved by the most expensive HSA_SOTS variants (MCLa1 and MCLa2), computationally speaking. However, the MCLa1 application is significantly less expensive that the MCLa2 one.

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

Case Study	Algorithms		T_s=1			T_s =900		
	MCL	*Cooling Scheme*	*Min.*	*Median*	*% hits*	*Min.*	*Median*	*% hits*
1	All	All	**1106.46**	1106.46	**100.00**	**1106.46**	1106.46	**100.00**
2	All	All	**735.00**	735.00	**100.00**	**735.00**	735.00	**100.00**
3	All	All	**2928.00**	2928.00	**100.00**	**2928.00**	2928.00	**100.00**
4	All	All	**1154.34**	1154.34	**100.00**	**1154.34**	1154.34	**100.00**
5	MCLs	HSAProp	50845.37	54974.18	**0,00**	**50845.16**	**50845.16**	**53.84**
		HSAExp	**50845.16**	54974.18	**7.6**	**50845.16**	54974.18	**9.09**
		HSALog	50846.39	54974.18	**0,00**	**50845.16**	50846.18	**36.36**
		HSARand	50845.37	52909.78	**0,00**	50845.37	54974.18	**0,00**
	MCLa1	HSAProp	**50845.16**	54973.16	**40,00**	**50845.16**	**50845.16**	**100,00**
		HSAExp	**50845.16**	52909.67	**40,00**	**50845.16**	52909.16	**50,00**
		HSALog	**50845.16**	50845.67	**50,00**	**50845.16**	**50845.16**	**100,00**
		HSARand	**50845.16**	**50845.16**	**62.5**	**50845.16**	**50845.16**	**62.5**
	MCLa2	HSAProp	**50845.16**	52909.16	**50,00**	**50845.16**	**50845.16**	**100,00**
		HSAExp	**50845.16**	**50845.16**	**62.5**	**50845.16**	**50845.16**	**62.5**
		HSALog	**50845.16**	**50845.16**	**62.5**	**50845.16**	**50845.16**	**100,00**
		HSARand	**50845.16**	**50845.16**	**71.4**	**50845.16**	**50845.16**	**100,00**

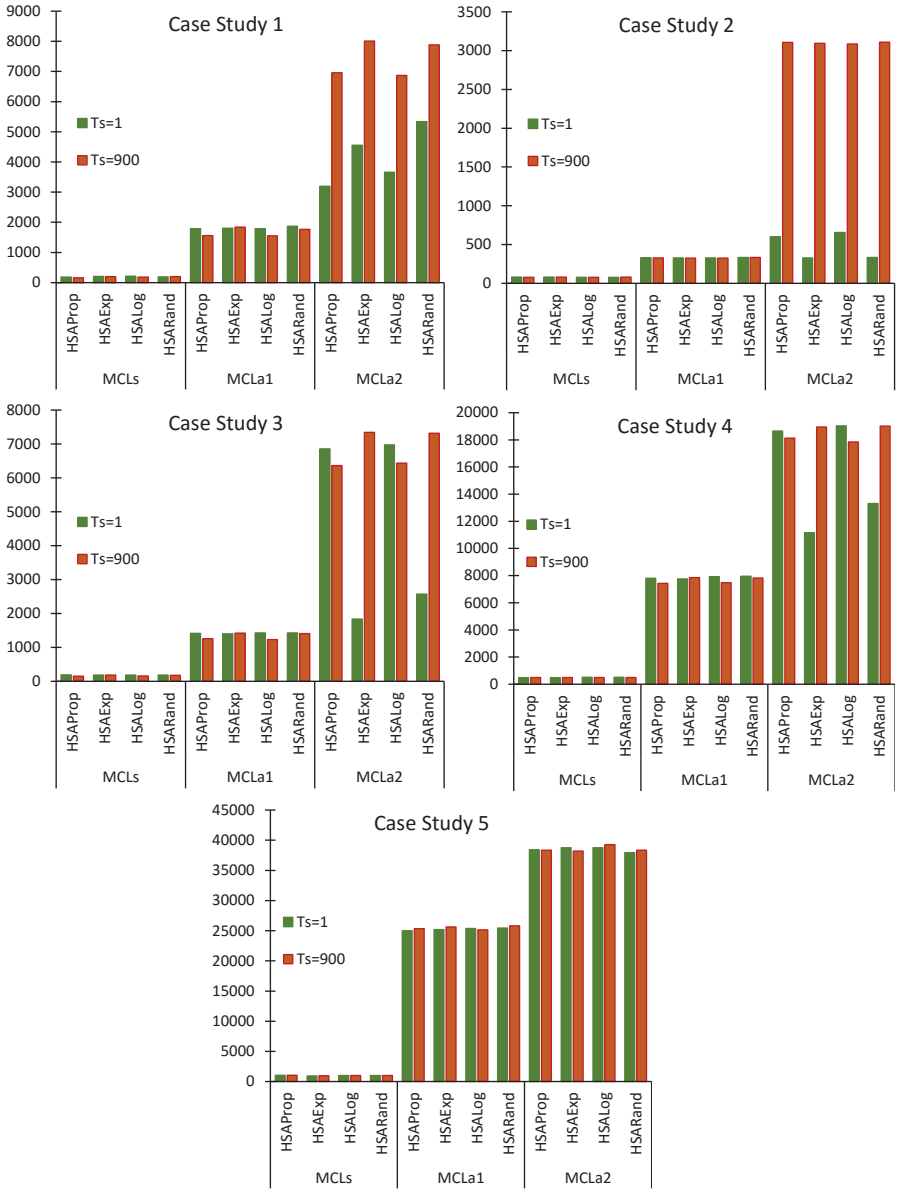

Fig. 1. Total time (in seconds) spent by each algorithmic variant for all case studies.

Summarizing, for the case studies 1, 2, 3, and 4, the HSA_SOTS's variants that implement MCLs always obtain the best solution with the minimum effort, showing statistically similar behaviors (KW test with α=0.01). But if the complexity of the case to solve grows (case study 5), a trade-off between quality and time must be

Table 2. Comparison of the best HSA_SOTS' variant and PBIL_SOTS, considering best solution statistics for each case study.

Case study	Best solution		Mean best solution±*SE*		KW
	PBIL-SOTS	Best HSA_SOTS' variant	PBIL-SOTS	Best HSA_SOTS' 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	≠

achieved. In this sense, the best algorithmic approaches are HSAProp and HSALog with T_s=900 and the application of MCLa1 to calculate the Markov chain length.

4.2 Comparison of HSA_SOTS Variants and the Literature Approaches

In this section, we compare the behavior of the best algorithmic variants of HSA_SOTS versus other well-known algorithm found in the literature for solving these SNDP case studies. 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_SOTS's variant, in Table 2 the best solution achieved by each of them for the 5 case studies, and their respective mean best solution and standard deviation are shown. Furthermore, a KW test is carried out to corroborate the similarities or differences between them. In general, we can observe that the algorithm from literature behaves similarly to HSA_SOTS in the least complex case studies, while statistically different behaviors are observed for the fifth case study. In this sense, two advantages in favor of HSA_SOTS are observed due to the average best solution is equal to the best known one, and the standard deviation indicates that it is found in every execution.

5 Conclusions

This work presents a SA hybridized with heuristic methods (HSA_SOTS), which is focused on the algorithmic control parameter tuning to solve the Optimal SNDP. As a consequence, 12 algorithmic variants arisen for each initial temperature considered ($Ts \in \{1,900\}$). These variants apply four different cooling schemes and tree ways to calculate the Markov Chain Length. Five SNDP case studies of a growing complexity were used to test and analyze our proposals.

From this analysis arises that the HSA_SOTS's variants, which implement 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 (5), a trade-off between quality and time is achieved when HSA_SOTS uses a Ts=900, MCLa1, and applies the proportional

cooling scheme. Furthermore, these proposals are competitive with the literature approaches.

In future works, other SNDP formulations will be tackled improving the SA main heuristic by introducing different specific local search mechanisms.

6 References

1. Bagajewicz M. Design and Retrofit of Sensors Networks in Process Plants. AIChE Journal, 3,2300–2306 (1997).
2. Nguyen D. Q., Bagajewicz M.; New Efficient Breath-first/level Traversal Methodology for the Design and Upgrade of Sensor Networks, AIChE Journal, 57(5), 1302-1309 (2011)
3. Gerkens C., Heyen G., Use of Parallel Computers in Rational Design of Redundant Sensor Networks, Comput. Chem. Eng, 29, 1379-1387 (2005).
4. He Y.J., Ma Z.F, Optimal Design of Linear Sensor Networks for Process Plants: A Multi-Objective Ant Colony Optimization Approach, Chemometrics and Intelligent Laboratory Systems, 135, 37-47 (2014).
5. Carnero M., Hernández J.L., Sánchez M., Optimal Sensor Location in Chemical Plants using the Estimation of Distribution Algorithms, Industrial & Engineering Chemistry Research, vol. 57(36), 12149–12164 (2018).
6. Kirkpatrick S., Gelatt C., Vecchi M.P., Optimization by Simulated Annealing, Science,220(4598),671-680 (1983).
7. Talbi E. G., 2009, Metaheuristics: From Design to Implementation, Wiley.
8. Hajek B., “Cooling Schedules for Optimal Annealing," Mathematics of Operations Research, vol. 13, no. 2, pp. 311-329 (1988).
9. Bermudez C., Salto C., Minetti G. Solving the Multi-Period Water Distribution Network Design Problem with a Hybrid Simulated Anealling. In: Pesado P., Aciti C. (eds) Computer Science – CACIC 2018. CACIC 2018. Communications in Computer and Information Science, vol 995. Springer, Cham (2019).
10. Geman S., D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 6, pp. 721-741, (1984).
11. Cardoso M. F., Salcedo R. L., de Azevedo S. F., Nonequilibrium Simulated Annealing: A Faster Approach to Combinatorial Minimization, Ind. Eng. Chem. Res, vol. 33, pp. 1908-1918, (1994).
12. Ali M., Trn A., Viitanen S., “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).
13. Hernandez J., Salto C., Minetti G., Carnero M., Sanchez M.C., Hybrid Simulated Annealing for Optimal Cost Instrumentation in Chemical Plants, Chemical Engineering Transactions, 74, 709-714 (2019).
14. Du Ke-Lin, Swamy M. N. S. (Ed), Search and Optimization by Metaheuristics, Techniques and Algorithms Inspired by Nature, Springer International Publishing, Switzerland (2016).