IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 3, September 2022, pp. 916~922
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i3.pp916-922

A new approach to solve the of maximum constraint satisfaction problem

Mohammed El Alaoui, Mohamed Ettaouil
Laboratory for Modelling and Scientific Computing, Sidi Mohamed ben Abdellah University, Fez, Morocco

Article Info

Article history:

Received Oct 28. 2021
Revised Apr 1, 2022
Accepted Apr 25, 2022

Keywords:

Constraint satisfaction problem
Hopfield neural networks
Simulated annealing algorithm

ABSTRACT

The premature convergence of the simulated annealing algorithm, to solve many complex problems of artificial intelligence, refers to a failure mode where the process stops at a stable point that does not represent to an overall solution. Accelerating the speed of convergence and avoiding local solutions is the concern of this work. To overcome this weakness in order to improve the performance of the solution, a new hybrid approach is proposed. The new approach is able to take into consideration the state of the system during convergence via the use of Hopfield neural networks. To implement the proposed approach, the problem of maximum constraint satisfaction is modeled as a quadratic programming. This problem is solved via the use of the new approach. The approach is compared with other methods to show the effectiveness of the proposed approach.

Corresponding Author:

Mohammed El Alaoui
Laboratory for Modelling and Scientific Computing, Sidi Mohamed ben Abdellah University
Fez, Morocco
Email: md.elalaoui@gmail.com

1. INTRODUCTION

Constraint programming is closely related to constraint satisfaction theory, which offers a simple formal scheme for representing and solving combinatorial problems of artificial intelligence [1]. Among the tasks solved by constraint programming: checking electronic circuits, calendar planning, schedule planning, as well as many combinatorial tasks [2]–[4]. Constraint programming is a programming paradigm in which relationships between variables are specified in the form of constraints. Formally, the maximum constraint satisfaction problem (Max-CSP) is defined by a set of variables which are linked by a set of constraints following a domain of definition for each variable. Max-CSP solution is an instantiation to satisfy the maximum of constraints [5].

The concept introduction of Max-CSP leads to extensive research for choosing an appropriate resolution method. In addition, exact methods require very high computational time due to the size and complexity of the problem. Whereas approximate methods are necessary for the mission to find an instantiation for the maximum constraint satisfaction problem.

The simulated annealing algorithm, in recent years, has been used to solve real problems especially optimization problems [6]. The peculiarity of the simulated annealing algorithm lies in its flexibility to adapt with any optimization problem [7]. This makes the simulated annealing algorithm more efficient, faster and easier to program to solve many optimization problems [8].

In view of the attention given to the simulated annealing approach to solve many optimization problems, this work adopted this approach as a method of solving Max-CSP. The simulated annealing method is able to avoid local minima to find the optimal solution. The optimal choice of a set of parameters like the cooling model, the initial temperature and the final temperature is essential to ensure good convergence. The limitation of the simulated annealing method lies in the inability of the method to take into account the behavior of the problem during convergence. In contrast, Hopfields neural network has proven its ability in the field of machine learning.

In this article, we propose a hybrid approach for solving maximum constraint satisfaction problems. The idea is to improve the simulated annealing algorithm in order to build a powerful system that can be adapted with any type of quadratic problem. To achieve this goal, we adopt the Hopfield network which is capable of taking the system state upon convergence to the simulated annealing algorithm to avoid the local solution. In order to perfect the proposed approach, three cooling models are used for the simulated annealing algorithm.

This work is structured in five sections. The section 2 presents a quadratic model for the maximum constraint satisfaction problem. The section 3 describes the hybrid approach which combines the neural network and simulated annealing. The section 4 allows to implement the proposed approach to solve the Max-CSP. The section 5 gives a conclusion and proposes an alternative avenue of research on another field of application.

2. MODELLING OF MAX-CSP

The constraint satisfaction problem can be defined as a network of variables that are related to each other. In this network, the assignment of a value to a variable with the satisfaction of all the constraints between each pair of variables is necessary to find a solution for the constraint satisfaction problem [9]. In some cases, satisfaction of all constraints is impossible given the complexity and size of the problem [10]. To deal with this problem, reducing the number of violated constraints is necessary as a partial solution [11]. This paradigm is known in the literature by the maximum constraint satisfaction problem. The Max-CSP consists in assigning a value to a variable for the entire network with the maximum constraint satisfaction [12]. More formally, the maximum constraint satisfaction problem is a form of model that is represented by a set of variables and a set of constraints. The aim of this work is to study the binary constraints of Max-CSP. The maximum constraint satisfaction problem is defined by a tuple $P = < X, D, C, f >$ such that:

$X = \{x_1, x_2, \cdots, x_n\}$: Set of n variables
$D = \{D(x_1), D(x_2), \cdots, D(x_n)\}$: Set of domains
$C = \{C_1, C_2, \cdots, C_m\}$: Set of m constraints
f: objective function

The basic idea for solving the maximum constraint satisfaction problem is based on assigning a value to a variable with minimization of number of violated constraints. In this context, a quadratic model under linear constraints is proposed. The modeling phase requires the declaration of the following mathematical notations: x_i is a decision variable, d_i is the size of decision variable x_i, v_r is the value assigned to the decision variable x_i, and N is the sum of the size of all variables.

The decision variable is defined by (1):

$$x_{ir} = \begin{cases} 1 & if \ x_i = v_r \\ 0 & otherwise \end{cases} \tag{1}$$

A unique value is selected for each decision variable. This expression is defined by (2):

$$\sum_{r=1}^{d_i} x_{ir} = 1 \qquad \forall i \in \{1,2, \dots, n\} \tag{2}$$

A relation R_{ij} between the variable x_i and the variable x_j makes it possible to define a binary constraint C_{ij}. In this modelization, a matrix q of dimension N is built starting from the checking of each constraint between the two variables x_i and x_j. An element of matrix q of row i (i.e. variable x_i) and column j (i.e. variable x_j) is defined:

$$q_{irjs} = \begin{cases} 1 & if \ (v_r, v_s) \notin R_{ij} \\ 0 & if \ (v_r, v_s) \in R_{ij} \end{cases} \setminus \ i \in \{1, \dots, n\}, j \in \{1, \dots, n\} \tag{3}$$

The constraint C_{ij} is expressed:

$$S_{ij} = \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} q_{irjs} x_{ir} x_{js} \tag{4}$$

The objective function is defined:

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} q_{irjs} x_{ir} x_{js} \tag{5}$$

The matrix form of the objective function $f\ (x)$ is expressed

$$f(x) = x^T Q x \tag{6}$$

The Max-CSP problem is modeled as a new quadratic programming, which constitutes an objective function subjected to a linear constraint.

$$(QP) = \begin{cases} Min\ f(x) = \frac{1}{2} x^T Q x \\ under\ constraint \\ \quad Ax = b \\ \quad x \in \{0,1\}^N \end{cases} \tag{7}$$

The matrix Q is a symmetric matrix of dimension $N \times N$ that represents the relationship between the decision variables. The matrix A is of dimension $N \times n$, which represents the linear constraint. The vector q is of dimension N. To solve this proposed model of Max-CSP, a new approach is proposed to solve it.

3. THE PROPOSED MODEL SOLVED BY NEW APPROACH

This section gives a representation of a hybrid approach that combines the simulated annealing method and the Hopfield neural network to solve the maximum constraint satisfaction problems. In the subsection of the simulated annealing approach, different cooling models are represented. Then Hopfield's neural network is represented as an adaptive approach to solving any quadratic problem. The last subsection presents a new hybrid approach that combines simulated annealing and the Hopfield neural network. First, a detailed description of the simulated annealing approach is shown in the next subsection.

3.1. Simulated annealing

The simulated annealing method mimics the physical phenomenon of crystallization [13]–[16]. Crystallization is an operation that allows a substance to transition from a liquid phase to a solid phase. This process was used in the simulated annealing method to solve an optimization problem. The operation of the simulated annealing method is related to a set comprises the initial temperature, final temperature and cooling model. Controlled cooling participates to ensure good convergence. The following notations are used for the simulated annealing method: x is a possible solution, $E(x)$ is an energy function, T_{max} ihe maximum temperature, T_{min} ihe minimum temperature, F_{trans} is a random transformation function, F_{gener} function generating a new state, and P is the transition probability defined by the following expression:

$$P(\Delta\mathrm{E}, \mathrm{T}) = \begin{cases} 1 & \Delta\mathrm{E} \le 0 \\ e^{\frac{-\Delta\mathrm{E}}{T_i}} & \Delta\mathrm{E} > 0 \end{cases}$$

The application of the simulated annealing method requires the use of a good cooling model to reduce the temperature of the energy function [17]–[19]. Controlled cooling makes it possible to switch from a high energy level to a low energy level. In the simulated annealing algorithm, choosing a good cooling model is important for better convergence. The following subsection describes the cooling models used in this work.

Simulated annealing algorithm pseudocode

```
Simulated-annealing(T_max, T_min, x_0)
i := 1
T_0 := T_max
x_best := x_0
        While T_0 > T_min do
        x_best := F_gener(x_{i-1})
        ΔE := E(x_best) − E(x_{i-1})
        If   P(ΔE, T_i) ≥ random(0,1) then
                x_i := x_best
        Else
                x_i := x_{i-1}
        End if
        If E(x_i) < E(x_best) then
                x_best := x_i
        End if
        i := i + 1
        T_{i+1} := F_tans (T_i)
Return x_best
```

3.1.1. Geometric model

The geometric cooling model is inspired by an arithmetic-geometric sequence in which each term makes it possible to deduce the next by multiplication by a constant factor [20]. This model is defined by the relation $T_k = \alpha T_{k-1} + \beta$. The factor α is selected from the interval [0.1]. The relation is an arithmetic sequence when $\alpha = 1$ and is a geometric sequence when $\beta = 0$. Therefore, the parameter α must be different from 0 and 1.

3.1.2. Logarithmic model

The logarithmic model was first proposed by Geman and Geman by the following formula : $T_{k+1} = c \times (log(k+1))^{-1}$ [21]. The logarithmic model proposes a relationship between the initial temperature and the final temperature. The temperature decreases in two phases: the first phase marks a rapid change in temperature in only a few first iterations. The second phase is characterized by a very slow change in temperature. Therefore, the convergence of this model is very slow and this requires considerable computation time.

3.1.3. Logarithmic model

The Lundy-Mees model is temperature cooling technique described by the following formula $T_{k+1} = c \times (1 + \beta\, T_k)^{-1}$ [22]. The parameter β is defined by the following relation: $\beta = \left(T_0 - T_f\right) \times \left(M.\, T_0.\, T_f\right)^{-1}$. The parameter T_0 represents the initial temperature, the parameter T_f represents the final temperature, and the parameter M is the number of iterations.

3.2. Continuous hopfield network

Physicist John Hopfield proposed the Hopfield model in 1982, it was a major breakthrough in the field of neural networks [23]. The Hopfield model not only allows to function as associative memory to help object recognition in image processing domain but also it is able to solve a lot of optimization problem such as the problem of installing a surveillance camera, the traveling salesman and the problems of maximum satisfaction of constraints [24], [25]. Due to the great use of this model, it has become the center of attraction for many researchers. Hopfields neural network is a fully connected network [26]. More formally, it is represented by a symmetrical matrix to guarantee the stability of this network. The Hopfield neural network is composed of n interconnected neurons [27]. The dynamics of the Hopfield neural network is described by the following differential (8):

$$\frac{du}{dt} = -\frac{u}{\tau} + Tv + i^b \tag{8}$$

The vector $v = (v_i)$ is the input vector of neurons and $u = (u_i)$ is the output vector of neurons with $1 \le i \le n$ and $u_i \in \{0,1\}$. The weight matrix is given by $T = \left(T_{i,j}\right)$ and i^b is the neuron bias. The hyperbolic function is used to calculate the output of each neuron. Neuron output is expressed:

$$v_i = -\frac{1}{2}\left(1 + \tanh\left(\frac{u_i}{u_0}\right)\right), \quad u_0 > 0 \tag{9}$$

where u_0 is a parameter used to control the gain of the activation function. Hopfield proved that the symmetry of the zero-diagonal matrix T is a sufficient condition for the existence of the Lyapunov function [28]. Therefore, the existence of the equilibrium point is guaranteed [29]. Continuous Hopfield networks are capable of solving combinatorial problems that have an energy function taking the following form:

$$\begin{cases} Min\, f(x) = \frac{1}{2}x^T Q x \\ under\ constraint \\ \quad Ax = b \\ \quad x \in \{0,1\}^N \end{cases} \tag{10}$$

3.3. Proposed hybrid approach

A hybrid algorithm consists of combining two or more different algorithms in order to arrive at an optimal solution. One of the objectives achieved in this work is to propose a quadratic model for the problem of maximum constraint satisfaction and to solve this model via a robust hybrid algorithm. This hybrid algorithm is a combination of two different approaches: the Hopfield neural network and the simulated annealing algorithm. Hopfield's neural network methodology has been widely used in optimization problems since their arrival. In this work, the Hopfield network was adopted to improve the convergence of the simulated annealing algorithm. This section presents a hybrid algorithm that can solve different problems of maximum constraint satisfaction.

Hybrid-algorithm

$i \coloneqq 1$
$h \coloneqq 0.5$
$T_0 \coloneqq T_{max}$
$x_{best} \coloneqq x_0$
While $T_0 > T_{min}$ do
$x_{best} := F_{gener}(x_{i-1})$
$\Delta E := E(x_{best}) - E(x_{i-1})$
If $E(x_i) < E(x_{i-1})$ then
$x_i \coloneqq x_{i-1} + hf(x_{i-1})$
End if
If $P(\Delta E, T_i) \geq random(0,1)$ then
$x_i := x_{best}$
Else
$x_i \coloneqq x_{i-1}$
End if
$i \coloneqq i + 1$
$T_{i+1} \coloneqq F_{tans}(T_i)$
Return x_{best}

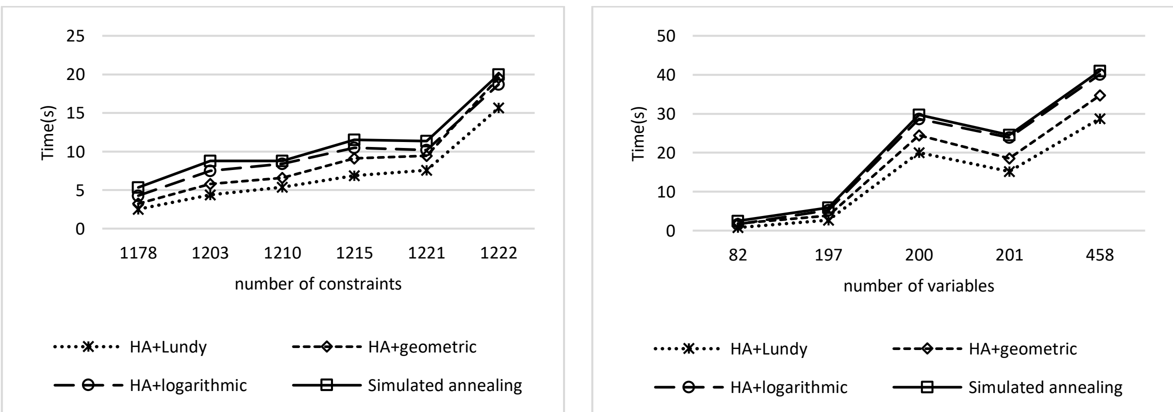
4. RESULTS AND DISCUSSION

The proposed approach that combines simulated annealing and the Hopfield neural network is used as a solver for the maximum constraint satisfaction problem. To assess the effectiveness of the proposed approach, a series of instances that represent real problems is used in this work. In this section, the basic simulated annealing algorithm is used to solve the maximum constraint satisfaction problem. In addition, the proposed approach is also used to carry out the research process to ensure good convergence.

This section presents the different instances (scens, CNF) used to evaluate the performance of the proposed approach. Software and hardware prerequisites are required to implement the proposed approach. Instances are run on a 3.0 GHs processor desktop and 4 GB RAM. The proposed algorithm is programmed through the use of Java object-oriented programming language. Given the stochastic nature of the proposed approach (the complexity of the algorithm and the structure of the test instance), the experiment was carried out 30 times. When implementing the proposed approach, a number of parameters can help with good convergence. These parameters are determined through preliminary experiment. The preliminary experiment made it possible to set the value of α and c at 0.99 and 3.5 respectively.

4.1. Experiments with scens instance

The scens instance represents that are used to compare the proposed approach with other methods. Figure 1(a) shows the average execution time for instance scenes with a fixed number of variables that is equal to 100 and a number of constraints varies between 1,178 and 1,222. Figure 1(b) shows the same instance scens but relatively large with a number of variables ranging from 82 to 458 and with a number of constraints varying from 382 to 5,286. The hybrid approach that combines the Hopfield neural network and simulated annealing with the Lundy cooling model (HA+ Lundy) has proven to be the most robust solver in terms of quality and runtime.



(a) (b)

Figure 1. Instance scens (a) number of variable is fixed at 100 and (b) number of variables beetwen 82 and 458

4.2. Experiments with CNF instance

In this experiment, the conjunctive normal form (CNF) instance is used to evaluate the performance of the proposed approach. The first step in this experiment is to extract the data from extensible markup language (XML) file. The second step is to represent the relationships that lie between the variables in a decision function that evaluate the maximum constraint satisfaction problem. Figure 2(a) shows the average execution time of the CNF instance with a number of variables fixed at 40. Figure 2(b) shows the average execution time of the CNF instance with a number of variables fixed at 80.

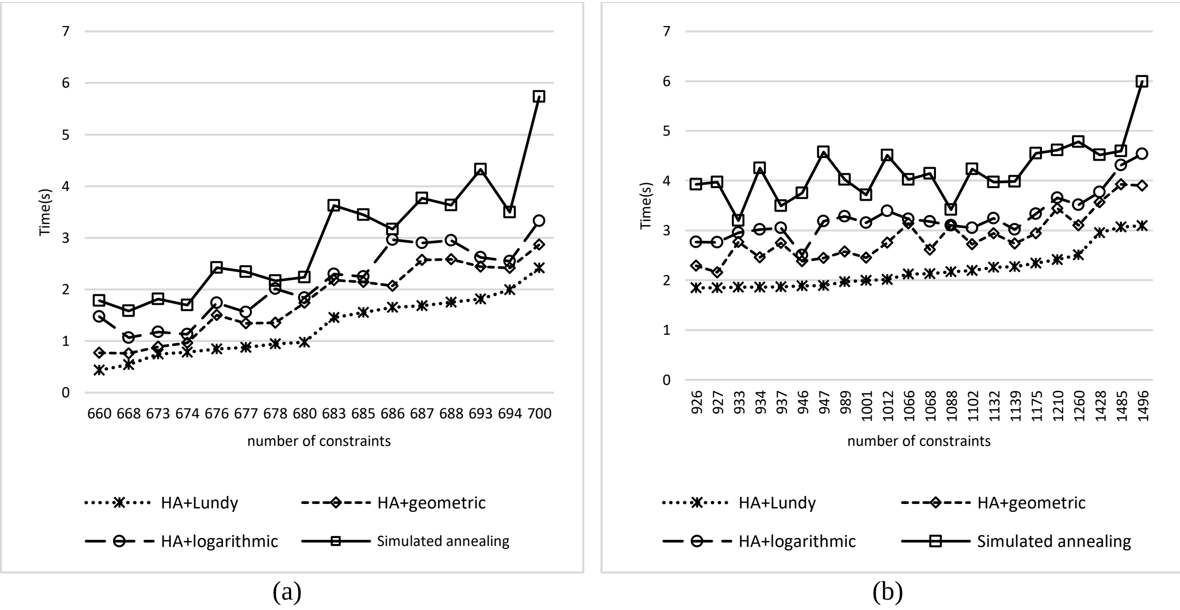


Figure 2. CNF instance (a) number of variable is fixed at 40 and (b) number of variable is fixed at 80

5. CONCLUSION

Hopfield's neural network was used in this work to improve the simulated annealing algorithm. This makes it possible to build a new hybrid approach. Hopfield's neural network is a robust algorithm that takes into account the previous information to improve the direction of the algorithm towards a better solution. The simulated annealing is fed by Hopfield's neural network during the research process. The proposed approach gives better results comparing with other conventional methods. The proposed approach has made it possible to solve the scens instance of variable number between 82 and 458 in a better execution time. And also allows to solve the CNF instance of variable number is set to 40 and to 80 in a better execution time better compared to other approaches. Future research should attempt to model and solve the quadratic model associated with the query optimization problem in databases.

REFERENCES

[1] K. Hirayama and M. Yokoo, “Distributed partial constraint satisfaction problem,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1330, 1997, pp. 222–236, doi: 10.1007/BFb0017442.

[2] V. Kumar, “Algorithms for constraint satisfaction problems: a survey,” *AI Magazine*, vol. 13, no. 1, pp. 32–44, 1992

[3] G. Dozier, J. Bowen, and A. A. Homaifar, “Solving constraint satisfaction problems using hybrid evolutionary search,” *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 1, pp. 23–33, Apr. 1998, doi: 10.1109/4235.728211.

[4] D. Steurer, “Fast SDP algorithms for constraint satisfaction problems,” in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2010, pp. 684–697, doi: 10.1137/1.9781611973075.56.

[5] M. C. Cooper, “High-order consistency in valued constraint satisfaction,” *Constraints*, vol. 10, no. 3, pp. 283–305, Jul. 2005, doi: 10.1007/s10601-005-2240-3.

[6] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, “A survey on new generation metaheuristic algorithms,” *Computers and Industrial Engineering*, vol. 137, Nov. 2019, doi: 10.1016/j.cie.2019.106040.

[7] I. Boussaïd, J. Lepagnot, and P. Siarry, “A survey on optimization metaheuristics,” *Information Sciences*, vol. 237, pp. 82–117, Jul. 2013, doi: 10.1016/j.ins.2013.02.041.

[8] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019, doi: 10.1007/s10462-017-9605-z.

[9] M. Charikar, K. Makarychev, and Y. Makarychev, “Near-optimal algorithms for maximum constraint satisfaction problems,” in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 62–68.

[10] M. Ettaouil and C. Loqman, "Constraint satisfaction problems solved by semidefinite relaxations," *WSEAS Transactions on Computers*, vol. 7, no. 7, pp. 951–961, 2008

[11] M. Ettaouil, C. Loqman, K. Haddouch, and Y. Hami, "Maximal constraint satisfaction problems solved by continuous hopfield networks," *WSEAS Transactions on Computers*, vol. 12, no. 2, pp. 29–40, 2013

[12] J. Larrosa and P. Meseguer, "Optimization-based heuristics for maximal constraint satisfaction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 976, 1995, pp. 103–120, doi: 10.1007/3-540-60299-2_7.

[13] A. Franzin and T. Stützle, "Revisiting simulated annealing: a component-based analysis," *Computers and Operations Research*, vol. 104, pp. 191–206, Apr. 2019, doi: 10.1016/j.cor.2018.12.015.

[14] D. Abramson, M. Krishnamoorthy, and H. Dang, "Simulated annealing cooling schedules for the school timetabling problem," *Asia-Pacific Journal of Operational Research*, vol. 16, no. 1, pp. 1–22, 1999

[15] B. Ceranic, C. Fryer, and R. W. Baines, "An application of simulated annealing to the optimum design of reinforced concrete retaining structures," *Computers and Structures*, vol. 79, no. 17, pp. 1569–1581, Jul. 2001, doi: 10.1016/S0045-7949(01)00037-2.

[16] M. F. Cardoso, R. L. Salcedo, and S. F. de Azevedo, "Nonequilibrium simulated annealing: a faster approach to combinatorial minimization," *Industrial and Engineering Chemistry Research*, vol. 33, no. 8, pp. 1908–1918, Aug. 1994, doi: 10.1021/ie00032a005.

[17] A. K. Peprah, S. K. Appiah, and S. K. Amponsah, "An optimal cooling schedule using a simulated annealing based approach," *Applied Mathematics*, vol. 8, no. 8, pp. 1195–1210, 2017, doi: 10.4236/am.2017.88090.

[18] S. Parthasarathy and C. Rajendran, "A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs-a case study," *Production Planning and Control*, vol. 8, no. 5, pp. 475–483, Jan. 1997, doi: 10.1080/095372897235055.

[19] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, Feb. 1993, doi: 10.1214/ss/1177011077.

[20] D. Henderson, S. H. Jacobson, and A. W. Johnson, "The theory and practice of simulated annealing," in *Handbook of Metaheuristics*, Boston: Kluwer Academic Publishers, 2006, pp. 287–319, doi: 10.1007/0-306-48056-5_10.

[21] D. Long, J. Viovy, and A. Ajdari, "My IOPscience," *Journal of Physics. Condensed Matter*, vol. 8, 1996

[22] M. Lundy and A. Mees, "Convergence of an annealing algorithm," *Mathematical Programming*, vol. 34, no. 1, pp. 111–124, Jan. 1986, doi: 10.1007/BF01582166.

[23] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, Jul. 1985, doi: 10.1007/BF00339943.

[24] A. Maurer, M. Hersch, and A. G. Billard, "Extended hopfield network for sequence learning: application to gesture recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3696, 2005, pp. 493–498, doi: 10.1007/11550822_77.

[25] K. Cheng, J. Lin, and C. Mao, "The application of competitive Hopfield neural network to medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 15, no. 4, pp. 560–567, Aug. 1996, doi: 10.1109/42.511759.

[26] M. El Alaoui, K. El Moutaouakil, and M. Ettaouil, "A multi-step method to calculate the equilibrium point of the continuous hopfield networks: application to the max-stable problem," *WSEAS Transactions on Systems and Control*, vol. 12, pp. 418–425, 2017

[27] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proceedings of the National Academy of Sciences*, Apr. 1982, vol. 79, no. 8, pp. 2554–2558, doi: 10.1073/pnas.79.8.2554.

[28] I. Goldhirsch, P.-L. Sulem, and S. A. Orszag, "Stability and Lyapunov stability of dynamical systems: a differential approach and a numerical method," *Physica D: Nonlinear Phenomena*, vol. 27, no. 3, pp. 311–337, Aug. 1987, doi: 10.1016/0167-2789(87)90034-0.

[29] P. M. Talaván and J. Yáñez, "A continuous Hopfield network equilibrium points algorithm," *Computers and Operations Research*, vol. 32, no. 8, pp. 2179–2196, Aug. 2005, doi: 10.1016/j.cor.2004.02.008.

BIOGRAPHIES OF AUTHORS

Mohammed El Alaoui is a Doctorate Status in applied computer sciences and mathematics from the Laboratory of Modeling and Scientific computing at the Faculty of Sciences and Technology of Fez, Morocco. He is a member of Artificial intelligence for engineering sciences group in the Laboratory of mathematical modeling, Operational Research and computer sciences. He works on Neural Network, Artificial Intelligence, classification problems, Database, constraint satisfaction problem and machine learning and Data warehouse. He can be contacted at email: md.elalaoui@gmail.com.

Mohamed Ettaouil is a Doctorate Status in Operational Research and Optimization, FST University Sidi Mohamed Ben Abdellah USMBA, Fez. Ph.D. in Computer Science, University of Paris 13, Galilee Institute, Paris France. He is a professor at the Faculty of Science and technology of Fez FST, and he was responsible for research team in modelization and pattern recognition, operational research and global optimization methods. He was the Director of Unit Formation and Research UFR: Scientific computing and computer science, Engineering Sciences. He is also a responsible for research team in Artificial Neural Networks and Learning, modelization and engineering sciences, FST Fez. He is an expert in the fields of the modelization and optimization, engineering sciences. He can be contacted at email: mohamedettaouil@yahoo.fr.