Neural network to solve fuzzy constraint satisfaction problems

Bouhouch Adil[1,2], **Er-Rafyg Aicha**[2], **Ez-Zahout Abderrahmane**[2]
[1]Department of Educational Technologies, Faculty of Educational Sciences, Mohammed V University, Rabat, Morocco
[2]Intelligent Processing Systems Team Computer Science Laboratory, Computer Science Department, Faculty of Sciences, Mohammed V University, Rabat, Morocco

Article Info

Article history:

Received Feb 2, 2023
Revised May 1, 2023
Accepted May 7, 2023

Keywords:

Continuous Hopfield neural network
Fuzzy constraint
Meta-heuristic
Quadratic problem

ABSTRACT

It has been proven that solving the constraint satisfaction problem (CSP) is an No Polynomial hard combinatorial optimization problem. This holds true even in cases where the constraints are fuzzy, known as fuzzy constraint satisfaction problems (FCSP). Therefore, the continuous Hopfield neural network model can be utilized to resolve it. The original algorithm was developed by Talaavan in 2005. Many practical problems can be represented as a FCSP. In this paper, we expand on a neural network technique that was initially developed for solving CSP and adapt it to tackle problems that involve at least one fuzzy constraint. To validate the enhanced effectiveness and rapid convergence of our proposed approach, a series of numerical experiments are carried out. The results of these experiments demonstrate the superior performance of the new method. Additionally, the experiments confirm its fast convergence. Specifically, our study focuses on binary instances with ordinary constraints to test the proposed resolution model. The results confirm that both the proposed approaches and the original continuous Hopfield neural network approach exhibit similar performance and robustness in solving ordinary constraint satisfaction problems.

Corresponding Author:

Bouhouch Adil
Department of Educational Technologies, Faculty of Educational Sciences, Mohammed V University
Rabat, Morocco
Email: a.bouhouch@um5r.ac.ma

1. INTRODUCTION

The constraint satisfaction problem (CSP) is a central part of artificial intelligence [1]. Many problems can be represented and solved as CSPs. In recent years, this concept has been extended to incorporate soft constraints, leading to the emergence of valued constraint satisfaction problems (VCSPs). VCSPs associate costs with sub-assignments of variables within the scope of constraints, and the goal is to find an optimal value for the aggregate function of all the valued constraints. This generalization has resulted in various extensions, including valued CSP [2], weighted CSP [3], and fuzzy CSP [4], [5].

In this paper, our focus lies on fuzzy CSP. In a fuzzy constraint satisfaction problem (FCSP), each constraint represents preferences through fuzzy sets rather than strict categorical satisfaction [6]. As a result, acceptability becomes a gradual notion, allowing for probable solutions. It is well-known that solving CSP is an No polynomial complete problem, meaning its values and status are unknown and the problem needs to be solved in polynomial time. Consequently, various approaches have been developed to tackle FCSPs [3], [4], [7], [8]. In this work, we introduce a generalization of our CSP model resolution [9], [10] to effectively solve FCSPs. The last model concerned is based on the Hopfield network [11]–[13], which has shown great efficiency in solving combinatorial problems. This fact has encouraged us to adopt and enhance this network for the resolution of fuzzy CSP problems. The Hopfield neural network was proposed by Hopfield and Tank [14] with both discrete and continuous modes, and it has been widely used in various applications, including identification [15], pattern recognition, and optimization [16]. The continuous Hopfield neural network (CHN) has also demonstrated its ability to solve hard optimization problems [17], [18]. Our main objective in this paper is to adapt the weights and settings of the continuous Hopfield neural network to be able to solve FCSP. Additionally, we study the capability of this model with fuzzy constraints to solve binary CSP as well. The structure of this paper is organized as follows: In the next section, we introduce some approaches for solving the fuzzy constraint satisfaction problem. In section 3, we adapt a general method proposed in [19] to solve quadratic problems (QP). The last section is devoted to the experimental results.

2. FUZZY CONSTRAINT SATISFACTION PROBLEMS

In practice a wide number of artificial intelligence problems and many other areas can be redefine as a fuzzy constraint satisfaction problem. The goal is to find a variables assignment which satisfied all constraints. In some case not all the given constraints need to be satisfied. In particular, how much the satisfaction intensity reached by a given assignment. In this paper we will consider binary FCSPs only. The typical way to solve a FCSP to extending a partial solution to build the complete one.

2.1. Fuzzy CSP formulation

In the context of this study, a fuzzy constraint is defined as the assignment of a grade to each tuple in the Cartesian product of variables involved in the constraint, which falls within the interval $[0, 1]$. The FCSP can be formulate by extending the definition of CSP as a quadruplet sets $(X; D; C; \mu_C)$ where:

- $Y = \{Y_1, ..., Y_N\}$ is a set of N variables.
- $D = \{D_1, ..., D_N\}$ where each D_i is a set of d_i possible values for X_i.
- $C = \{C_1, ..., C_M\}$ is a set of constraints.
- $\mu_C = \{\mu_{C_1}, ..., \mu_{C_M}\}$ is fuzzy valuation structure.

Each constraint C_i refers to an ordered subset of Y. We call This subset of variables the scope of C_i and is denoted by $var(C_i)$. The arity of constraint is the number of variables in its scope. The structure μ_C the brings together the different evaluation functions of each constraint, we consider the constraint c_i with a cardinality k (arity), and we designate by $(v_1, v_2..., v_k)$ the projection of domains values of a given assignment A on $var(c_i)$, in this case the function μ_{c_i} associates the tuple $(v_1, v_2..., v_k)$ with it degree of satisfaction regard to the constraint c_i. Thus, this fuzzy valued describe how well this tuple satisfies the associated constraint. When $\mu_{c_i} = 1$ we say that c_i is fully satisfies, while for the other extreme side 0 we say that c_i is fully violated. Then goal of resolving is to maximize the combination of membership. To calculate the cost of a FCSP solution there are many objective function, for example in [7] the cost is defined by the average of the satisfaction of the individual by a solution v: $\mu_C(v) = \frac{1}{N}\sum_{i=1}^{N} \mu_{C_i}(v)$.

Ruttkay [20] suggest to favourite the maximisation of the last membership function; therefore to achieve a perfect solution, it is necessary to ensure that all individual constraints are satisfied optimally. This means that each constraint should be fulfilled to the highest possible degree, maximizing the overall satisfaction level across all constraints: $Max(\mu_C(v)) = \{max\ \mu_{C_i}(v), for\, all\, C_i\}$. Medini and Bouamama [8] propose to maximise the worst membership function as: $Max(\mu_C(v)) = max\{min\{\mu_{C_i}(v)\,, for\, i = 1\, to\, M\}\}$.

3. FCSPS SOLVED BY CHN

To resolve a binary FSCP by CHN we need first to reformulate it as a quadratic problem (QP), this can be easily done by the association a binary variable x_{ik} to each variable y_i of FCSP model, with k is in the domain of y_i, such that:

$$x_{ik} = \begin{cases} 1, & \text{if value } k \text{ is assigned to variable } i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

for two variable i and j with r and s it assigned value respectively we put:

$$Q_{ijrs} = \begin{cases} 1 - \mu_{C_{ij}(r,s)} & \text{if}(r,s) \in C_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

let y_i and y_j two variables,then for constraints these associate binary fuzzy C_{ij}, we introduce a state function defined as:

$$S_{ij}(x) = \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} x_{ir} x_{js}\, Q_{ijrs} \tag{3}$$

we note that $x = \{x_{ik}\ , i \in [1..N], k \in d_i\}$ is the solution vector.

To be more generalize we can reformulate also unary constraint. So, for a variable i a constraint C_i will be reformulate as (4).

$$q_{ir} = \begin{cases} 1 - \mu_{C_i(r)} & \text{if}\, r \in C_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We defined Q and q in way to be equal 1 if the concerning constraint is fully violated. Considering all equations defined in (3) and (4), thus we can consider the objective function as a quadratic problem (QP):

$$f(x) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} x_{ir} x_{js}\, Q_{ijrs} + \sum_{i=1}^{N} \sum_{r=1}^{d_i} q_{ir}$$

valid solutions must conform to stringent constraints that can be mathematically expressed as a system of linear equations: $\sum_{r=1}^{d_i} x_{ir} = 1$, with $i = 1..N$ By puting matrix A with an $N \times M$ dimensions and the vector b with M dimension which is fully initialised to 1 This formulation can also be reformulate as $Ax = b$. So, the model is given as follows:

$$(QP) \qquad \begin{cases} Min & f(x) = x^T Q x + q^T x \\ with & Ax = b \\ & x \in \{0,1\}^n \end{cases}$$

now we can use any optimization approach to solve the last problem. The combinatorial nature of this problem favours the use of meta-heuristic approaches rather than execte approches based. We recently developed an approach based on Hopfield neural network which is ideal for this kind of NP-hard problem. To apply this recurrent neutral network to minimise the last objectif function, the objective is to construct an energy function in such a way that the local minima of the energy function of a Hopfield recurrent neural network (RNN) align with the viable solutions of the problem.

3.1. Hopfield neural network

In [14] is the first work which introduced Hopfield neural network, in this paper Hopfield and Tank, originally aimed to tackle combinatorial optimization problems. Since its inception, the Hopfield neural network has undergone extensive research, enhancements, and application in various domains, including optimization, pattern identification, and pattern recognition. Notably, the Hopfield neural network has shown promising capabilities in providing acceptable solutions to challenging optimization problems. In this paper, we aim to extend the applicability to solve quadratic programming (QP) by continuous Hopfield network (CHN) in the context of fuzzy constraint satisfaction problems (CSP). The Hopfield model comprises a set of interconnected neurons, typically represented by n units. The dynamics of the continuous Hopfield network (CHN) are governed by (5).

$$\frac{dy}{dt} = -\frac{x}{\tau} + T\, x + i^b \tag{5}$$

where x represents the vector of neuron inputs; y represents the vector of outputs, which belongs to $\mathbb{R}^n$; and T is denotes the weight matrix between each pair of nodes.

The network inputs are defined each one by an activation function $x = g(y)$, where $g(y)$ is a function that bounds the values between 0 and 1. The specific form of the activation function $g(y)$ is given by $g(y) = \frac{1}{2}(1 + \tanh(\frac{y}{u_0}))$. The energy function of the continuous Hopfield network (CHN) is defined as (6).

$$E(x) = x^t\, T\, x + {(i^b)}^t\, x \tag{6}$$

We construct the CHN energy function as in [19], [12], and we rewrite it to be able to solve the binary FCSPs quadratic formulation in the previous section (QP). The CHN energy function used in this work is based on the construction described in [19] and [12]. To adapt it for solving binary fuzzy constraint satisfaction problems (FCSPs), we modify the energy function to formulate it as a quadratic problem (QP).

$$E(x) = \frac{\alpha}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{r=1}^{d_i}\sum_{s=1}^{d_j} x_{ir}x_{js}\,Q_{ijrs} + \alpha\sum_{i=1}^{N}\sum_{r=1}^{d_i} q_{ir} +$$
$$\frac{1}{2}\phi\sum_{i=1}^{N}\sum_{r=1}^{d_i}\sum_{s=1}^{d_i} x_{ir}x_{is} + \beta\sum_{i=1}^{N}\sum_{r=1}^{d_i} x_{ir} + \gamma\sum_{i=1}^{N}\sum_{r=1}^{d_i} x_{ir}(1-x_{ir}) \tag{7}$$

The two first terms penalize constraints violation based on there satisfaction degrees, the second one ensure that each variable is assigned a single value. the third term aggregate all linear constraint $Ax = b$, and the last terms satisfies the integrity propriety to enforce neurone to be closed at 0 or 1 state. To calculate the the network weight link it's enough to match in (6) and (7):

$$\begin{cases} T_{irjs} = -\alpha(1-\delta_{ij})Q_{irjs} - \delta_{ij}\phi + 2\delta_{ij}\delta_{rs}\gamma \\ i^b_{ir} = -\alpha q_{ir} - \beta - \gamma \end{cases}$$

with $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{othwise } i \neq j \end{cases}$ is the Kroneckern symbol. In [19] to obtain an equilibrium point for the CHN, authors use an adapted algorithm based step by step update approach introduced by Talavaán and Yáñez in [21].

This approach significantly increases the convergence speed of the neural network. The main idea is to iteratively move in the direction $\frac{dy}{dt}$ until the network reaches the minimum energy state. To ensure stability, both Haddouch *et al.* [19] and Talavan and Yanez [22] utilize the hyperplane method. This method involves dividing the Hamming hypercube, which represents all possible solutions, by a hyperplane that contains only the feasible solutions. By using the hyperplane method, the CHN is guided towards stable and optimal solutions while ensuring convergence towards the global energy minimum. This technique helps to enhance the performance and effectiveness of the neural network in solving complex optimization problems. To ensure convergence, we impose two conditions by studying the derivative of the energie function E:

$$\frac{\delta E(x)}{\delta x_{ir}} = E_{ir}(x) = \frac{\alpha}{2}\sum_{j=1}^{N}\sum_{s=1}^{d_j} x_{js}\,Q_{irjs} + \alpha q_{ir} +$$
$$\frac{1}{2}\phi\sum_{s=1}^{d_i} x_{is} + \beta + \gamma(1-x_{ir}) \tag{8}$$

- The first one is imposed to prevent the assignment of two different values to one variable, denoted as $\exists i, j \in D(X_k)$ with $i \neq j$, and their corresponding neurons satisfying $x_{ki} = x_{kj} = 1$, it is necessary to ensure that the energy function, denoted as $E(x)$, satisfies the following condition: $E_{ir}(x) \geq \alpha\, q_{min} + 2\phi + \beta - \gamma \geq \varepsilon$.
- In order to ensure that all variables are assigned values and avoid the case where $x_{ir} = 0$ for all $r \in 1, \ldots, d_i$, an additional condition must be imposed on the energy function $E(x)$. This condition can be expressed as $E_{ir}(x) \leq \alpha\, d + \beta + \gamma \leq -\varepsilon$ with:

$$d = Max\left\{ \sum_{j=1}^{N}\sum_{s=1}^{d_j} Q_{irjs} + \sum_{k=1}^{d_i} q_{ik} \;/\; i \in [1, N] \;and\; r \in \{1, ..., d_i\} \right\}$$

We founded that the optimal value of ε is 10^{-5} and for parameter settings there are obtained by finding a solution that satisfies the given inequality [19]:

$$\begin{cases} \phi \geq 0, \quad \alpha > 0 \\ -\phi + 2\gamma \geq 0 \\ \alpha\, q_{min} + 2\phi + \beta - \gamma = \varepsilon \\ \alpha\, d + \beta + \gamma = -\varepsilon \end{cases}$$

4. COMPUTATIONAL EXPERIMENTS

In our knowledge there are no benchmark created for FCSP. Thus we will use on the one hand a random geneated FCSP, and on the other hand we study the performance of our solver on typical CSP instances [13], [23]. The first choice is aimed at demonstrating the practical relevance of our approach, highlighting its potential benefits in real-world applications. We strive to showcase how our approach can address practical challenges effectively. The second choice involves evaluating the performance of our approach across diverse problem types, including random, academic, and real-world problems. To achieve this, we utilize the benchmark datasets provided by Cril University [24]. These datasets encompass a wide range of problem instances, allowing us to assess the effectiveness and versatility of our approach across different problem domains. The last study is summarize in Table 1 with V represent the number variables in the specific instance; C is constraints number of a given instance; ratio mean is the average of the optimal value founded of a run series on each instance; and CPU(s) is the average execution time of multiple runs on this specific problem instance.

Table 1. Performance of CHNFCSP and CHNCSP over typical instances problems

Name of instance	V	C	FCHN		CHN		Means of Iteration
			Ratio means	CPU(s)	Ratio means	CPU(s)	
driverlogw-01c	71	217	3	0.0754	3	0.0754	237.74
composed-25-10-20-5	105	620	55.7	2.9317	55.7	2.9341	276.24
dsjc-125-1-5	125	736	78.72	0.7957	78.72	0.7957	331.12
frb30-15-5-mgd	30	210	40.98	1.0677	40.98	1.068	309.09
frb40-19-3-mgd	40	308	52.13	0.9303	52.13	0.9313	931.34
geom-30a-5	30	81	7.5	0.0559	7.5	0.0559	273.26
geom-30a-6	30	81	5.74	0.0521	5.74	0.0521	343.92
geom-40-2	40	78	27.17	0.0115	27.17	0.0115	125.33
geom-40-6	40	78	3.63	0.0933	3.63	0.0933	320.44
le-450-5a-5	450	5714	856.76	13.6679	856.76	13.6718	353.96
myciel-5g-5	47	236	16.66	0.4731	16.66	0.4738	304.78
myciel-5g-6	47	236	13.96	0.1035	13.96	0.1347	328.06
queens-5-5-5	25	160	18.98	0.027	18.98	0.0274	138.34
qwh-15-106-1	225	2324	66.76	1.797	66.76	1.8004	283.96
qwh-15-106-4	225	2324	59.21	4.06	59.21	4.0638	245.62
qwh-15-106-6	225	2324	67.76	2.3029	67.76	2.3099	351.14
qwh-20-166-0	400	5092	93.3	16.7906	93.3	16.7941	599.8
qwh-20-166-3	400	5092	87.89	2.881	87.89	2.882	433.58
qwh-20-166-6	400	5092	85.1	8.1842	85.1	8.191	407.44
queens-10	10	45	11.68	0.001	11.68	0.01	196.76
queens-20	20	190	43.23	0.0278	43.23	0.0278	316.9
queens-30	30	435	85.11	1.9348	85.11	1.935	411.34
rand-23-23-253-131-48021	23	253	45.8	1.3432	45.8	1.3456	261.47
rand-23-23-253-131-49021	23	253	40.87	1.5358	40.87	1.5353	298.32
rand-23-23-253-131-50021	23	253	33.04	0.699	33.04	0.6995	409.08

In our analysis, we have observed that there is no significant difference between continuous Hopfield neural network for fuzzy constraint satisfaction problems (CHNFCSP) and continuous Hopfield neural network for constraint satisfaction problems (CHNCSP) when it comes to solving normal CSPs. Both approaches exhibit comparable performance and solution quality. To generate random instances for experimentation, we consider an FCSP problem as a triple $(n, d, \bar{t})$, where n represents the number of variables as well as the domain size for each variable. The value of n is randomly selected from the interval 10 to 20. The parameters d and $\bar{t}$ denote the network density and tightness, respectively, and can take values from the set $0.1, 0.3, 0.5, 0.7, 0.9$. Thus, we can generate a total of 25 classes $(d, \bar{t})$ by considering all possible combinations of d and $\bar{t}$. For each of these $(d, \bar{t})$ classes, we randomly generate 20 instances, resulting in a total of 500 randomly generated CSPs for our evaluation. The truth value of each tuple admitted by a constraint is assigned a random value within the interval [0, 1]. The random instance generation process allows us to cover a diverse range of problem characteristics and evaluate the performance of our approach across various scenarios.

The performance evaluation of our approach compared to a genetic algorithm-based evolutionary algorithm [25] on random instances is presented in Table 2. The results are displayed in a matrix format, where each cell represents a particular combination of density and average tightness values. Within each cell, the average best truth value (mftv) discovered and the average CPU execution time for both algorithms are provided. Despite this approach and very old but remains the only one that has explicitly adapted the genetic algorithm for the case of Fuzzy constraints. Then we carry out other comparison with recent evolutionary approaches adapted to resolve of ordinary CSPs without fuzzy constraints. In order to study the capacity of this approach to solve the binary CSP, this last can be considerate as a particular case of FCSP. So we choose two algorithms elaborated specially to solve CSP [26], [27]. In their work, Ortiz *et al.* [26] employed a variable length genetic algorithm to address the challenge of encoding more intricate rules by utilizing a larger number of features and heuristic actions. The genetic algorithm's chromosome structure comprised ten values, where nine of them represented landscape features and the tenth indicated the chosen low-level heuristic for variable ordering. The objective was to find the chromosome in the genetic algorithm that closely matched the current solution state of the constraint satisfaction problem (CSP), and subsequently apply the corresponding low-level heuristic to guide the search process.

The difference between particle swarm optimization (PSO) and standard mother tee PSO (MPSO) [27] lies in the implementation of operators. In the discrete mother tree optimization (DMTO)-CSP MPSO approach, similar to feeder recommended pools, authors utilize recommendation pools for exploration and exploitation through these operators. They introduce a local recommended pool (LRP) and global RP (GRP) that contain variable assignments from the local best and global best, respectively, excluding the current particle. The assignments minimizing conflicts are selected. To ensure a fair comparison, we didn't rely solely on the original settings provided by the authors. Instead, we empirically determined them by varying the values and selecting the ones yielding the best performance. For the genetic algorithm (GA), we used a population size of 200, a mutation rate of 5%, and a crossover rate of 72%. As for MPSO, we set $\varphi_1 = \varphi_2 = 1$ and a fixed population size of 100. We conducted 500 runs for each method on randomly generated instances using the RB model [23]. This model, a variant of the standard Model B, is capable of generating challenging problem instances that are hard to solve. Figure 1 show the comparative results, in terms of violated Constraint Number as we can noted our approach performs closely to GA and can be improved by introducing heuristics like the chosen GA while generating the starting point like the number of constraint in which a given variable is implicated.

Table 2. Performance of GA and CHN over randoms instances problems

Algorithm	d	t=0.1		t=0.3		t=0.5		t=0,7		t=0.9	
		mftv	time	mftv	time	mftv	time	mftv	time	mftv	time
FCHN	0.1	0.9180	0.0521	0.8900	0.0520	0.8480	0.0495	0.7190	0.0420	0.0960	0.0056
GA		0.9180	0.0734	0.8880	0.0519	0.8470	0.0495	0.7130	0.0416	0.0950	0.0055
FCHN	0.3	0.6750	0.0383	0.5830	0.0341	0.3740	0.0218	0.0320	0.0019	0.0000	0.0000
GA		0.6700	0.0536	0.5780	0.0338	0.3630	0.0212	0.0310	0.0018	0.0000	0.0000
FCHN	0.5	0.4790	0.0272	0.3660	0.0214	0.0620	0.0036	0.0000	0.0000	0.0000	0.0000
GA		0.4790	0.0383	0.3610	0.0211	0.0590	0.0034	0.0000	0.0000	0.0000	0.0000
FCHN	0.7	0.3500	0.0199	0.1540	0.3004	0.0060	0.0004	0.0000	0.0000	0.0000	0.0000
GA		0.3460	0.0277	0.1480	0.3126	0.0050	0.0003	0.0000	0.0000	0.0000	0.0000
FCHN	0.9	0.2750	0.0156	0.1250	0.3701	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
GA		0.2720	0.0217	0.1230	0.3761	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

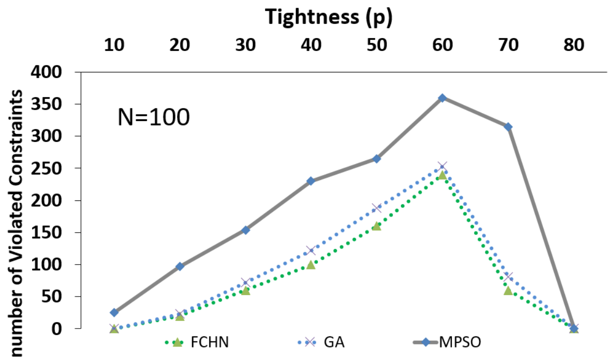

Figure 1. Performance of GA, PSO, and CHN-MNC random gendered instances with N=100

5. CONCLUSION

In this paper, we have addressed the challenges associated with fuzzy constraint satisfaction problems (FCSPs), which find numerous applications in various domains and are extensively studied in operations research. We have proposed a novel approach for solving binary fuzzy constraint satisfaction problems, comprising several key steps. First, we introduced a new model that formulates the FCSP as a 0-1 quadratic program subject to linear constraints. Next, we employed the continuous Hopfield network (CHN) to tackle this problem effectively. To evaluate the performance of our new approach, we conducted experiments comparing it to our previously developed method for solving CSPs using CHN. We ran these approaches on typical CSP instances with conventional constraints, as well as on a collection of randomly generated instances with varying levels of tightness. The results of our experiments demonstrate the effectiveness of our new approach for solving FCSPs. It outperformed the previous method on various problem instances, showcasing its improved performance and suitability for handling fuzzy constraints. This study contributes to the advancement of solving FCSPs and provides valuable insights for researchers and practitioners working on constraint satisfaction problems.

REFERENCES

[1] W. Song, Z. Cao, J. Zhang, C. Xu, and A. Lim, "Learning variable ordering heuristics for solving constraint satisfaction problems," *Engineering Applications of Artificial Intelligence*, vol. 109, p. 104603, 2022, doi: 10.1016/j.engappai.2021.104603.

[2] G. Matl and S. Živný, "Using a min-cut generalisation to go beyond boolean surjective VCSPs," *Algorithmica*, vol. 82, no. 12, pp. 3492–3520, 2020, doi: 10.1007/s00453-020-00735-1.

[3] M. Bidar and M. Mouhoub, "Solving weighted constraint satisfaction problems using a new self-adaptive discrete firefly algorithm," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 2198–2205, doi: 10.1109/SMC.2019.8914624.

[4] M. C. Vanegas, I. Bloch, and J. Inglada, "Fuzzy constraint satisfaction problem for model-based image interpretation," *Fuzzy Sets and Systems*, vol. 286, pp. 1–29, 2016, doi: 10.1016/j.fss.2014.10.025.

[5] Z. Ayadi, W. Boulila, I. R. Farah, A. Leborgne, and P. Gancarski, "Resolution methods for constraint satisfaction problem in remote sensing field: A survey of static and dynamic algorithms," *Ecological Informatics*, p. 101607, 2022, doi: 10.1016/j.ecoinf.2022.101607.

[6] S.-S. Lin, S.-L. Shen, A. Zhou, and Y.-S. Xu, "Risk assessment and management of excavation system based on fuzzy set theory and machine learning methods," *Automation in Construction*, vol. 122, p. 103490, 2021, doi: 10.1016/j.autcon.2020.103490.

[7] H. Fargier, "Selecting preferred solutions in fuzzy constraint satisfaction problems," in *Proceedings of the 1st Europ. Congress on Fuzzy and Intelligent Technologies (EUFIT'93)*, 1993, pp. 1128–1134.

[8] G. Medini and S. Bouamama, "Application of genetic algorithms to distributed optimization problems under fuzzy constraints," *Procedia Computer Science*, vol. 159, pp. 1258–1266, 2019, doi: 10.1016/j.procs.2019.09.295.

[9] A. Bouhouch, H. Bennis, L. Chakir, and A. El Qadi, "Neural network and local search to solve binary CSP," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 3, pp. 1319–1330, 2018, doi: 10.11591/ijeecs.v10.i3.pp1319-1330.

[10] M. Ettaouil, C. Loqman, and K. Haddouch, "Job shop scheduling problem solved by the continuous hopfield networks," *Journal of Advanced Research in Computer Science*, vol. 2, no. 1, pp. 31–47, 2010.

[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] M. Ettaouil, C. Loqman, Y. Hami, and K. Haddouch, "Task assignment problem solved by continuous hopfield network," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 206–212, 2012.

[13] F. Boussemart, C. Lecoutre, G. Audemard, and C. Piette, "XCSP3-core: A format for representing constraint satisfaction/optimization problems," *arXiv preprint arXiv:2009.00514*, 2020.

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

[15] M. Schleyer *et al.*, "Identification of dopaminergic neurons that can both establish associative memory and acutely terminate its behavioral expression," *Journal of Neuroscience*, vol. 40, no. 31, pp. 5990–6006, 2020, doi: 10.1523/JNEUROSCI.0290-20.2020.

[16] I. Shapoval and P. Calafiura, "Quantum associative memory in hep track pattern recognition," in *EPJ Web of Conferences*, 2019, vol. 214, p. 01012, doi: 10.1051/epjconf/201921401012.

[17] K. Lenin, "Power loss reduction by arctic wolf optimization algorithm," *The International Journal of Informatics and Communication Technology*, vol. 8, no. 3, pp. 111–116, 2019, doi: 10.11591/ijict.v8i3.pp111-116.

[18] Y. K. Qawqzeh *et al.*, "Applying the big bang-big crunch metaheuristic to large-sized operational problems," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2484–2502, 2020, doi: 10.11591/ijece.v10i3.pp2484-2502.

[19] K. Haddouch, M. Ettaouil, and C. Loqman, "Continuous hopfield network and quadratic programming for solving the binary constraint satisfaction problems." *Journal of Theoretical & Applied Information Technology*, vol. 56, no. 3, pp. 362–372, 2013.

[20] Z. Ruttkay, "Fuzzy constraint satisfaction," in *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, 1994, pp. 1263–1268, doi: doi: 10.1109/FUZZY.1994.343640.

[21] P. M. Talaván and J. Yáñez, "A continuous hopfield network equilibrium points algorithm," *Computers & operations research*, vol. 32, no. 8, pp. 2179–2196, 2005, doi: 10.1016/j.cor.2004.02.008.

[22] P. M. Talavan and J. Yanez, "The generalized quadratic knapsack problem. a neuronal network approach," *Neural networks*, vol. 19, no. 4, pp. 416–428, 2006, doi: 10.1016/j.neunet.2005.10.008.

[23] K. Xu and W. Li, "Exact phase transitions in random constraint satisfaction problems," *Journal of Artificial Intelligence Research*, 2000, doi: 10.1613/jair.696.

[24] G. Audemard, F. Boussemart, C. Lecoutre, C. Piette, and O. Roussel, “Xcsp 3 and its ecosystem,” *Constraints*, vol. 25, pp. 47–69, 2020, doi: 10.1007/s10601-019-09307-9.

[25] J. Bowen and G. Dozier, “Solving randomly generated fuzzy constraint networks using evolutionary/systematic hill-climbing,” in *Proceedings of IEEE 5th International Fuzzy Systems*, 1996, pp. 226–231, doi: 10.1109/FUZZY.1996.551746.

[26] J. C. Ortiz-Bayliss, H. Terashima-Marín, and S. E. Conant-Pablos, “Combine and conquer: an evolutionary hyper-heuristic approach for solving constraint satisfaction problems,” *Artificial Intelligence Review*, vol. 46, no. 3, pp. 327–349, 2016, doi: 10.1007/s10462-016-9466-x.

[27] W. Korani and M. Mouhoub, “Discrete mother tree optimization and swarm intelligence for constraint satisfaction problems,” in *ICAART (3)*, 2022, pp. 234–241, doi: 10.5220/0010803100003116.

BIOGRAPHIES OF AUTHORS

Bouhouch Adil is a computer sciences Professor at the faculty of sciences of education, and he is a member of Intelligent Processing Systems Team Computer Science Laboratory, Computer Science Department, Faculty of Sciences, Mohammed V University in Rabat. He was was awarded with the Ph.D. from the faculty of sciences in Meknas. Until now he participated to many international conferences, and published many papers. He can be contacted at email: bouhouch.adil@gmail.com.

Er-Rafyg Aicha is currently working toward the Ph.D. degree in computer science, speciality artificial intelligence. She is a member of Intelligent Processing Systems Team with the Computer Science Laboratory (Laboratoire de Recherche en Informatique) of Computer Science Department, Faculty of Sciences, Mohammed V University, Rabat, Morocco. She can be contacted at email: rafyqaicha@gmail.com.

Ez-Zahout Abderrahmane is currently an Associate Professor at: Department of Computer Sciences Faculty of Sciences, Mohammed V University. He graduated Ph.D. in Computer Sciences from ENSIAS College of IT and he is an active member of Intelligent Processing Systems and Security (IPSS) Team. He can be contacted at email: ezzahout@gmail.com.