Journal of Theoretical and Applied Information Technology
20th October 2013. Vol. 56 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CONTINUOUS HOPFIELD NETWORK AND QUADRATIC PROGRAMMING FOR SOLVING THE BINARY CONSTRAINT SATISFACTION PROBLEMS

[1] KHALID HADDOUCH, [1*]MOHAMED ETTAOUIL and [2]CHAKIR LOQMAN
[1]UFR: Scientific Computing and Computer sciences, Engineering sciences, Faculty of Science and Technology, University Sidi Mohamed Ben Abdellah Box 2202, Fez, Morocco
[2]Department of Computer Engineering, Moulay Ismail University, High School of technology, B. P. 3103, 50000, Toulal, Meknes, Morocco
E-mail: [1]haddouchk@yahoo.fr , [1*] mohamedettaouil@yahoo.fr , [2]chakirfst@yahoo.fr

ABSTRACT

Many important computational problems may be formulated as constraint satisfaction problems (CSP). In this paper, we propose a new approach to solve the binary CSP problems using the continuous Hopfield networks (CHN). This approach is divided into three steps: the first concerns reducing the size of the CSP problems using arc consistency technique AC3. The second step involves modeling the filtered constraint satisfaction problems as 0-1 quadratic programming subject to linear constraints. The last step concerns applying the continuous Hopfield networks to solve the obtained 0-1 optimization model. Therefore, the mapping procedure and an appropriate parameter setting procedure about CSP problems are given in detail. Finally, some computational experiments solving the CSP problems are shown.

Keywords: *Constraint Satisfaction Problems, Filtering Algorithms, Quadratic 0-1 Programming, Continuous Hopfield Networks, Energy Function.*

1. INTRODUCTION

A large number of problems in artificial intelligence and other areas of computer science can be viewed as special cases of the constraint satisfaction problems. Some examples include machine vision, belief maintenance, scheduling, temporal reasoning, graph problems, aircraft conflict, etc.

A constraint satisfaction problem is defined by a set of variables, a finite and discrete domain for each variable and a set of constraints. Each constraint restricts the combination of values that a set of variables may take simultaneously. Solving the CSP problem requires assigning a value for each variable from each domain in such a way that all constraints are satisfied. The task is to find one solution or all solutions. However, the CSP problems are NP-complete problems requiring a combination of heuristics and combinatorial search methods in order to be solved in a reasonable time[12].

A number of different approaches have been developed for solving the constraint satisfaction problems [10], [18], [19], [21]. Some of them use backtracking to directly search for possible solutions [5], others use consistency techniques to simplify the original problems [3], [4], [24], and some are a combination of these two techniques [2], [5], [13]. Moreover, we proposed different approaches to solve the constraint satisfaction problems. The first one consists of modeling a constraint satisfaction problem as 0-1 quadratic knapsack problem subject to quadratic constraint [6], [7]. The second one is a new model of the binary CSP problem as 0-1 quadratic programming which consists in minimizing the quadratic function subject to linear constraints[8]. The later model will be used in order to determine a generalized energy function for the binary constraint satisfaction problem. This step is the most important one for solving any binary CSP problem using continuous Hopfield networks.

This neural network was introduced by Hopfield and Tank [13], [14] and it has been extensively studied, developed and has found many applications in many areas, such as pattern recognition, model identification, and optimization [11], [28]. Moreover, this neural network is applied to solve many problems such as traveling salesman problems [27], graph coloring problems [17], allocation problems [16], etc. It is important to note

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

that all these problems can be reformulated as constraint satisfaction problems [15], [16], [21]. In this paper, we propose a general approach for solving the binary constraint satisfaction problems using the continuous Hopfield networks, which will be able to solve all problems that can be modeled in terms of constraint satisfaction problems.

This paper is organized as follows: In section 2, the arc-consistency procedure used in this approach is studied and the model of the binary CSP problems is formulated as a quadratic assignment problem with linear constraints. In section 3, a generalized energy function associated with the CSP problems is defined and a direct parameter setting procedure is computed. The proposed algorithm is described in the next section. Implementation details of the proposed approach, complexity analysis, and experimental results are presented in the last section.

2. REDUCING THE SIZE OF THE CSP PROBLEMS AND MODELING THEM

In general, a constraint satisfaction problem forms a class of models representing problems that have common properties, a set of variables and a set of constraints [21]. The variables should be instantiated from a discrete domain while making sure the constraints are satisfied. This study of CSP problems has become focused on binary constraint satisfaction problems i.e., the CSP with constraints of arity less than or equal to 2, and each constraint C_{ij} between variables y_i and y_j is defined by its binary relation R_{ij}. Formally speaking, the binary constraint satisfaction problems are defined by triple sets (Y,D, C) where:

- $Y = \{y_1,.....,y_N\}$ is a set of N variables,
- $D = \{D(y_1),.....,D(y_N)\}$ where each $D(y_i)$ is a set of d_i possible values for y_i,
- $C = \{C_{ij}\}$ is a set of m constraints which restricts the values that the variables y_i and y_j can simultaneously take.

The solution of constraint satisfaction problems is obtained by assigning each variable a value from its domain satisfying all constraints [10]. In order to use the continuous Hopfield networks for solving the constraint satisfaction problems, we opted for using the mathematical model which consists in modeling the CSP problems as a 0-1 quadratic programming. This model can be used to define the generalized energy function of the CHN. However, before modeling the CSP problems, we can use the arc consistency technique in order to reduce the size of the CSP problems and, thus, reduce the continuous Hopfield network architectures.

2.1 Consistency techniques

In general, a binary constraint satisfaction problem is represented as an undirected graph with the vertices corresponding to the variables and the edges corresponding to the constraints. It is known that, the binary CSP problems have only two kinds of constraints: unary constraints and binary constraints. The simplest degree of consistency that can be enforced on a CSP problem is node consistency which concerns only the unary constraints [10]. However, a stronger degree of consistency is arc consistency. The latter concerns the binary constraints in CSP problem and guarantees that each value admits at least a support in each constraint. Many algorithms have been proposed to establish arc consistency such as AC1, AC2,...,AC2000 and AC3.1[3], [4], [23], [24].

Definition 1

Let C_{ij} a binary constraint between two variables y_i and y_j, with two domains $D(y_i)$ and $D(y_j)$.

We call C_{ij} arc consistent if

- *$\forall a \in D(y_i)$, $\exists b \in D(y_j)$ such that (a, b) satisfies the constraint C_{ij} ,*
- *$\forall b \in D(y_j)$, $\exists a \in D(y_i)$ such that (a, b) satisfies the constraint C_{ij}} .*

We call a CSP problem arc consistent if all its binary constraints are arc consistent.

The description of the arc consistency AC3 technique which was used is the following [19]: Initially, all arcs (C_{ij},y_i) are put in a set named K. Then, each arc is revised in turn, and when the revision is effective (at least one value has been removed), the set K has to be updated. This revision removes values of $D(y_i)$ that have become inconsistent with respect to constraint C_{ij} [19]. The algorithm is stopped when the set K becomes empty. Therefore, the property of consistency is forced for CSP problem.

In this paper, the arc consistency algorithm plays an important role for solving the constraint satisfaction problems [10]. This consistency technique does not remove all inconsistent values from the variables'domains, but it can still eliminate many "obvious" inconsistencies and, thus, simplify

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the model of the CSP problems. In this context, there are three important roles of this filtering:

- Prove that the CSP problems do not admit a solution (if one domain after filtering is empty)[10].
- Reduce the size of the modeled CSP problems since for each value from variables'domains is associated with the binary variable. Therefore, if any inconsistent value from the variables'domains is deleted, the represented binary variable must be eliminated from the proposed model and, thus reduce the size of this model as much as possible.
- Reduce the Hopfield neural network architectures. When we reduce the size of this model, the number of neurons in the architecture of the continuous Hopfield networks is reduced because each binary variable is associated with one neuron in the continuous Hopfield networks.

2.2 Modeling The Constraint Satisfaction Problems

The constraint satisfaction problems have been recognized as an efficient model for solving many combinatorial and complex problems. These problems are formulated as 0-1 quadratic knapsack problem subject to quadratic constraint [6],[7]. Another model of the CSP problem as 0-1 quadratic programming, which consists of minimizing a quadratic function subject to linear constraints, has been proposed [8].

In the following, we want to present the proposed model of the binary constraint satisfaction problems. For each variable y_i of the CSP problem, we introduce d_i binary variables x_{ik} such that:

$$x_{ik} = \begin{cases} 1 & if\ y_i = v_k \\ 0 & Otherwise \end{cases} \qquad v_k \in D(y_i) \qquad (1)$$

Where $k \in \{1,.....,d_i\}, i \in \{1,.....,N\}$

This matrix is easily converted to a n-vector:

$$x = \left(x_{11} \quad \cdots \quad x_{1d_1} \quad \cdots \quad \cdots \quad x_{N1} \quad \cdots \quad x_{Nd_N} \right)^T$$

with $n = \sum_{i=1}^{N} d_i$ and $d_i = |D(y_i)|$

Based on this notion of encoding values for each variable, we can define the objective function and the constraints of our model.

2.2.2 Objective function

Each constraint C_{ij} between two variables y_i and y_j is defined by its relation R_{ij} (R_{ij} is a subset of the cartesian product $D(y_i) \times D(y_j)$, specifying the compatible values between y_i and y_j). Note that, if there is no constraint between variables y_i and y_j, $R_{ij}(v_r, v_s)$ holds for any pair (v_r, v_s) of $D(y_i) \times D(y_j)$. For each couple (v_r, v_s) such that $(v_r, v_s) \notin R_{ij}$, we generate a constraint:

$$x_{ir} x_{js} = 0 \qquad (2)$$

For each relation R_{ij}, the constraints (2) can be aggregated in a single constraint:

$$\sum_{r=1}^{d_i} \sum_{s=1}^{d_j} q_{irjs} x_{ir} x_{js} = 0 \qquad (3)$$

Where $q_{irjs} = \begin{cases} 1\ if\ (v_r, v_s) \notin R_{ij} \\ 0 \quad Otherwise \end{cases}$

and $i \in \{1,.....,N\}, j \in \{1,.....,N\}, i < j$

These constraints (3) can be aggregated in a single constraint:

$$f(x) = \sum_{i=1}^{N} \sum_{j=1 i<j}^{N} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} q_{irjs} x_{ir} x_{js} = 0 \qquad (4)$$

The later constraint can be writing in the following form:

$$f(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{d_i} \sum_{s=1}^{d_j} q_{irjs} x_{ir} x_{js} = 0 \qquad (5)$$

Where:

If $i<j$ then $q_{irjs} = \begin{cases} 1\ if\ (v_r, v_s) \notin R_{ij} \\ 0 \quad Otherwise \end{cases}$

If $i=j$ then $q_{irjs} = 0 \quad \forall r \in \{1,...,d_i\}$ and $\forall s \in \{1,...,d_j\}$,

If $i > j$ then $q_{irjs} = q_{jsir}$,

If there is no constraint between variables y_i and y_j then $q_{irjs} = 0 \quad \forall r \in \{1,...,d_i\}$ and $\forall s \in \{1,...,d_j\}$.

Finally, we can define the objective function $f(x)$ in the following way:

$$f(x) = \frac{1}{2} x^T Q x$$

Where the matrix Q is the $n \times n$ symmetric matrix and q_{irjs} are the elements of this matrix.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2.3 Constraints

Each variable y_i must take a unique value from its domain $D(y_i)$, then the linear constraints of the CSP problem are defined bellow:

$$\sum_{k=1}^{d_i} x_{ik} = 1 \qquad \forall i \in \{1,....,N\} \tag{6}$$

The later constraints can be written in the following matrix form:

$$Ax = b$$

Finally, we obtain the following 0-1 quadratic program (QP) with a quadratic function subject to linear constraints:

$$(QP) \quad \begin{cases} Min & f(x) = \frac{1}{2} x^T Q x \\ Subject\ to & \\ & Ax = b \\ & x \in \{0,1\}^n \end{cases}$$

Where Q is an $n \times n$ symmetric matrix, A is an $N \times n$ matrix and b is an N vector.

The following theorem establishes the links between the binary CSP problems and an optimization model QP.

Theorem 1 [8]

Let V(QP) an optimal value of the 0-1 quadratic programming (QP). The value V(QP) is equal to zero if and only if the binary constraint satisfaction problem has a solution.

In this work, our objective is to solve the constraint satisfaction problems using the continuous Hopfield networks. Then, in this case, the most important step consists of representing or mapping the constraint satisfaction problems in the form of the energy function associated with the continuous Hopfield networks. According to the proposed model, which consists of modeling the constraint satisfaction problems into a quadratic programming QP problem, we define the associated energy function and the parameters setting. Therefore, the continuous Hopfield networks can be used to solve the binary constraint satisfaction problems.

3. CONSTRAINT SATISFACTION PROBLEMS SOLVED BY CONTINUOUS HOPFIELD NETWORKS

As can be noticed, after filtering the constraint satisfaction problems via the arc consistency algorithm AC3, we can model it (filtered CSP problems) into 0-1 quadratic programming with a quadratic function subject to linear constraints. Based on this model, we present a general method for solving the CSP problems using the continuous Hopfield networks.

In the beginning of the 1980s, Hopfield published two scientific papers, which attracted a lot of interest. This was the starting point of the new area of neural networks, which continues today. In the same context, Hopfield and Tank presented the energy function approach in order to solve several optimization problems [17]. Their results encouraged a number of researchers to apply this network for solving different problems [1], [17], [27].

The continuous Hopfield neural network is a generalization of the discrete case. Afterwards, many researchers implemented continuous Hopfield network to solve the optimization problem, especially in mathematical programming problems. Then, the continuous Hopfield network is a major artificial neural network for solving optimization problems [27].

As we know, the CHN is a fully connected neural network i.e. the n neurons of Hopfield network are fully connected. Let W_{ij} be the strength of the connection from neuron i to neuron j. Each neuron i has an offset bias. The current state and the output of the neuron i are respectively represented by u_i and x_i [26]. The dynamics of the CHN is described by the differential equation:

$$\frac{du}{dt} = -\frac{u}{\tau} + Wx + i^b \tag{7}$$

Where u, x and i^b will be the vectors of neuron states, outputs and biases. In the continuous Hopfield networks, the relationship between the internal state u_i of a neuron i and its output level x_i is determined by an activation function, which is bounded below by 0 and above by 1. Commonly, this activation function is given by:

$$x_i = g(u_i) = \frac{1}{2}(1 + \tanh(\frac{u_i}{u_0})) \quad u_0 > 0 \tag{8}$$

It is a hyperbolic tangent, where u_0 is a parameter used to control the gain (or slope) of the

Journal of Theoretical and Applied Information Technology
20th October 2013. Vol. 56 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

activation function. The proof of stability of such continuous Hopfield networks relies upon the fact that $E(x)$ is a Lyapunov function, provided that the inverse function of $g^{\cdot}(u)$ (the first derivative of the activation function), exists. The existence of this equilibrium points for the CHN is guaranteed if a Lyapunov function exists. Hopfield proposed an energy function of the continuous Hopfield network that, if matrix W is symmetric [14], then the following Lyapunov function exists:

$$E(x) = -\frac{1}{2}x^T Wx - (i^b)^T x + \frac{1}{\tau}\sum_{i=1}^{n}\int_0^{x_i} g^{-1}(z)dz \qquad (9)$$

Where the function $g^{-1}(z)$ is a monotone increasing function. The idea is that the networks Lyapunov function, when $\tau \dagger \infty$, is associated with the cost function to be minimized in the combinatorial problem. In this way, the CHN output can be used to represent a solution of the combinatorial problem. Then, for solving any combinatorial problem via the continuous Hopfield networks, we will write this problem in the following form:

$$E(x) = -\frac{1}{2}x^T Wx - (i^b)^T x \qquad (10)$$

Typically, in the *CHN*, the energy function is made equivalent to the objective function which is to be minimized, while each of the constraints of the optimization problem are included in the energy function as penalty terms.

3.1 Generalized Energy Function For The Constraint Satisfaction Problems

In order to solve the constraint satisfaction problems using the continuous Hopfield networks, we define the generalized energy function for the CSP problems basing on the proposed model. Recall that, the constraint satisfaction problems are modeled as 0-1 quadratic programming with *n* variables and *N* linear constraints [8].

$$(QP)\quad \begin{cases} Min & f(x) = \frac{1}{2}x^T Qx \\ Subject\ to & \\ & Ax = b \\ & x \in \{0,1\}^n \end{cases}$$

The generalized energy function allows representing mathematical programming problems with quadratic objective function and linear constraints. This energy function includes the objective function $f(x)$ and it penalizes the linear constraints $Ax = b$ with a quadratic terms and a linear terms. Then, the generalized energy function must also be defined by [27]:

$$E(x) = E^O(x) + E^C(x) \qquad \forall\ x \in [0,1]^n \qquad (11)$$

Where:

- $E^O(x)$ is directly associated with the objective function of the constraint satisfaction problems,
- $E^C(x)$ is a quadratic function that penalizes the violated constraints of the CSP problems.

The following generalized energy function is proposed:

$$E(x) = \frac{\alpha}{2}x^T Qx + \frac{1}{2}(Ax)^T \Phi(Ax) + x^T diag(\gamma)(1-x) + \beta^T Ax \qquad (12)$$

Where $\alpha \in R^+$, $\beta \in R^N$, $\gamma \in R^n$, Φ is an $N \times N$ symmetric matrix and $diag(\gamma)$ denotes the diagonal matrix constructed from the vector γ.

To define the energy function for the CSP problems, the following considerations need to be taken into account so that the mathematical expression of energy function (12) is simplified. Only the main diagonal terms of the quadratic matrix parameter Φ are considered:

$$\Phi_{kj} = \begin{cases} 0 & if\ j \neq k \\ \phi & if\ j = k \end{cases}$$

Where ϕ is a scalar, All linear constraints are equally weighted, where β is the associated parameter, The parameter penalizing the non-extreme values of x_{ir} is γ. We notice that the CSP problems have only one family of linear constraints:

$$e_i(x) = \sum_{k=1}^{d_i} x_{ik} = 1 \quad \forall i \in \{1,...,N\} \qquad (13)$$

Consequently, the following generalized energy function for the CSP problems is proposed:

$$E(x) = \alpha f(x) + \frac{1}{2}\phi\sum_{i=1}^{N}(e_i(x))^2 + \beta\sum_{i=1}^{N}e_i(x) + \gamma\sum_{i=1}^{N}\sum_{r=1}^{d_i}x_{ir}(1-x_{ir}) \qquad (14)$$

which in algebraic form is:

$$E(x) = \frac{\alpha}{2}\sum_{i,j=1}^{N}\sum_{r=1}^{d_i}\sum_{s=1}^{d_j}q_{irjs}x_{ir}x_{js} + \frac{1}{2}\phi\sum_{i=1}^{N}\sum_{r=1}^{d_i}\sum_{s=1}^{d_i}x_{ir}x_{is}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$+\beta\sum_{i=1}^{N}\sum_{r=1}^{d_i}x_{ir}+\gamma\sum_{i=1}^{N}\sum_{r=1}^{d_i}x_{ir}(1-x_{ir}) \quad (15)$$

In our case, the weights W_{irjs} of the continuous Hopfield networks are in function of the coefficients corresponding to the quadratic terms $x_{ir}x_{js}$, and the external inputs i_{ir}^{b} (thresholds) are in function of the coefficients corresponding to the linear terms x_{ir} in the chosen energy function. Then to determine the weights and thresholds, we use the assimilation between equation (10) and the algebraic form of the generalized energy function. Finally, the weights and thresholds of the connections between the n neurons are:

$$\begin{cases} W_{irjs} = -\alpha(1-\delta_{ij})q_{irjs}-\delta_{ij}\phi+2\delta_{ij}\delta_{rs}\gamma \\ i_{ir}^{b} = -\beta-\gamma \end{cases} \quad (16)$$

Where δ_{ij} is the Kroenecker delta.

$$\delta_{ij}=\begin{cases} 1 & if\ i=j \\ 0 & if\ i\neq j \end{cases}$$

In this way, the quadratic programming has been presented as an energy function of continuous Hopfield networks. To solve an instance of the CSP problem, the parameter setting procedure is used. This procedure assigns the particular values for all parameters of the network, so that any equilibrium points are associated with a valid affectation for all variables when all constraints are satisfied.

3.2 Parameters Setting Of The Continuous Hopfield Networks

The weights and thresholds of the continuous Hopfield network depend on the parameters α, ϕ, β and γ. To solve the CSP problems via the CHN, an appropriate setting of these parameters is needed. In this subsection, our objective is to determine these parameters. The dynamics of the CHN must ensure that any invalid solution $x \notin H_F$ cannot be a stable point. Any constraint which is imposed on the dynamics of the CHN must be translated into a set of constraints on its parameter values, this is the kernel of the parameter setting. A feasible solution is guaranteed by the CHN from a stability analysis of the Hamming hypercube corners set:

$$H_C=\{x\in H : x\in\{0,1\}^n\}$$

Where $H=\{x\in[0,1]^n\}$ is the Hamming hypercube and $H_F=\{x\in H_C : Ax=b\}$ is a set of feasible solutions.

The parameter setting procedure is based on the partial derivatives of the generalized energy function:

$$\frac{\partial E(x)}{\partial x_{ir}}=E_{ir}(x)$$

Where:

$$E_{ir}(x)=\alpha\sum_{j=1}^{N}\sum_{s=1}^{d_j}q_{irjs}x_{js}+\phi\sum_{s=1}^{d_i}x_{is}+\beta+\gamma(1-2x_{ir}) \quad (17)$$

A point $x\in H$ will be an equilibrium point for the CHN if and only if the two following relations are satisfied [27]:

$$\begin{cases} E_{ir}(x)\geq 0 & if\ \ x_{ir}=0 \\ E_{ir}(x)\leq 0 & if\ \ x_{ir}=1 \end{cases} \quad (18)$$

Where $\forall i\in\{1,...,N\}$ and $\forall r\in\{1,...,d_i\}$

This procedure uses the hyperplane method [27], so that the Hamming hypercube H is divided by a hyperplane containing all feasible solutions. To avoid the stability of any no feasible and corner solution $x\in H_C-H_F$, the following instability conditions are imposed:

$$\begin{cases} E_{ir}(x)\leq -\varepsilon & if\ \ x_{ir}=0 \\ E_{ir}(x)\geq \varepsilon & if\ \ x_{ir}=1 \end{cases} \quad (19)$$

Based on this hyperplane method and the associated half-spaces, the complementary corners set of the feasible solutions for the CSP problem is partitioned and a set of analytical equations of the CHN parameter is proposed. The hyperplane method is briefly explained below; however, for simplicity reasons the following parameters constrains are first assigned:

- To minimize the objective function, we impose the following constraint: $\alpha>0$,
- On the other hand, to penalize the non-feasibility of the family of linear constraints $e_i(x)$, it is natural to impose the following constraint: $\phi\geq 0$,
- In order to guarantee the instability of the interior points $x\in H-H_C$, some initial conditions are imposed on some parameters:

$$W_{irir}=-\phi+2\gamma\geq 0 \quad (20)$$

Where $i\in\{1,...,N\}, r\in\{1,...,d_i\}$.

Given the linear constraints of the QP problem:

$$e_i(x)=1 \qquad \forall i\in\{1,...,N\} \quad (21)$$

The partition of H_C-H_F is defined as:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$H_C - H_F = H_{1,1} \bigcup H_{1,2}$$

This two partition are defined in the folowing way:

- $H_{1,1} = \{\exists i : e_i(x) > 1\} \bigcap \{e_0(x) \geq N\}$

where $e_0(x) = \sum_{i=1}^{N} e_i(x)$. In this case, one variable y_i has been assigned two different values v_r, v_s in $D(y_i)$ so that $x_{ir} = x_{is} = 1$. Consequently, the following instability condition is imposed:

$$E_{ir}(x) \geq 2\phi + \beta - \gamma \geq \varepsilon \qquad (22)$$

- $H_{1,2} = \{\exists i : e_i(x) < 1\} \bigcap \{e_0(x) < N\}$

In this case, one variable y_i has not been assigned any value $v_r \in D(y_i)$, such that $x_{ir} = 0 \quad \forall r \in \{1,...,d_i\}$. Therefore, the following instability condition is imposed:

$$E_{ir}(x) \leq \alpha d + \beta + \gamma \leq -\varepsilon \qquad (23)$$

With $d = Max\left\{\sum_{j=1}^{N}\sum_{s=1}^{d_j} q_{irjs}\right\}$

Then, we can determine the parameters setting by resolving the following system:

$$\begin{cases} \alpha > 0 \quad , \phi \geq 0 & \\ -\phi + 2\gamma \geq 0 & (24.a) \\ 2\phi + \beta - \gamma = \varepsilon & (24.b) \\ \alpha d + \beta + \gamma = -\varepsilon & (24.c) \end{cases}$$

The inequation (24.a) guaranteed the satisfaction of the integrity constraints ($x_{ir} \in \{0,1\}$), but the equations (24.b) and (24.c) guaranteed the satisfaction of the linear constraints. These parameters setting are determined by fixing α , ε and computing the rest of parameters γ, β and ϕ: $\phi = d\alpha + 2\varepsilon$, $\gamma = \phi / 2$ $\beta = \varepsilon - 3\gamma$ Then, the weights and thresholds of continuous Hopfield networks (system 16) can be computed using these parameters setting. Finally, we obtain an equilibrium point for the *CHN* using the algorithm depicted in [26], so compute the solution of constraint satisfaction problem.

4. DESCRIPTION OF THE NEW SOLVER OF CSP PROBLEMS

The new solver was developed to find a solution of the CSP problem via the continuous Hopfield networks. The diagram of this algorhitm is described by the following steps (Figure 1):

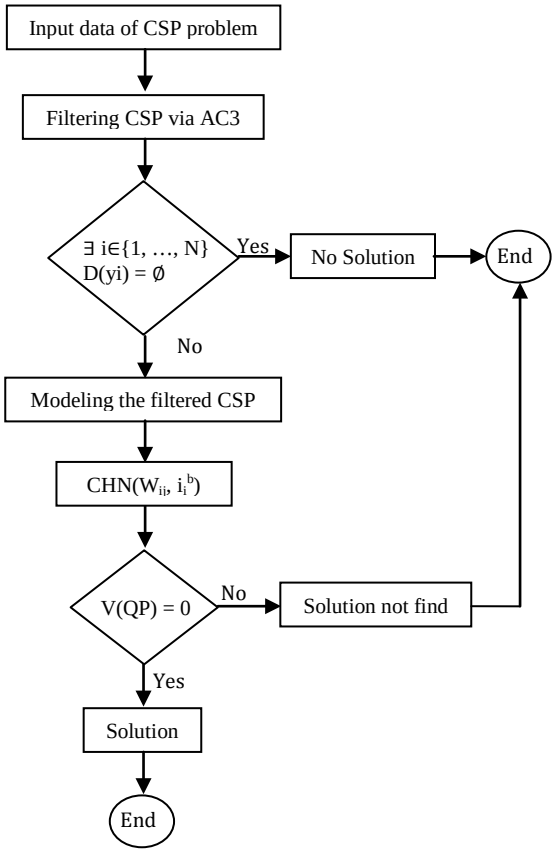

Figure 1: Diagram of the proposed algorithm

The important steps of this algorithm are the following:

- The first step is to read the data of CSP problem, this data is presented in XML file (database). This reading requires developing techniques to extract data (domains, variables, relationships and constraints).
- The main objective of the second step is to filter the CSP problem. There are two important roles of this filtering:
  - Reducing the size of the CSP problem,
  - Showing that the CSP problem does not admit a solution (if one domain after the filter is empty).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This filtering is realized by the arc consistency technique AC3.

- The third step concerns modeling the filtered CSP problem as 0-1 quadratic programming (determining the matrix Q, the matrix A and the vector b).
- The last step concerns using the CHN to obtain a solution of this model, so that if an optimal value of the 0-1 quadratic programming is equal to 0 (V(QP) = 0), the solution of CSP problem is found; else, the CSP problem has no solution (theorem 1).

5. COMPUTATIONAL EXPERIMENTS

In order to show the practical interest of the proposed approach, the experiments and the complexity analysis are studied in this section.

5.1 Complexity analysis

In this subsection we evaluate and determine the complexity of arc consistency algorithm (AC3) and the continuous Hopfield network algorithm. Given a constraint satisfaction problem with N variables, m constraints and d_{max} the size of the largest domain. The complexity of the AC3 algorithm is on the order of $O(md_{max}^3)$ average time complexity [20]. Simulating the continuous Hopfield network algorithm for large instances of constraint satisfaction problems constitutes real challenges in terms of the memory space that must be allocated for the weight matrix W and thresholds vector i^b, which is the highest-cost data structure. In general, the number of memory bytes required and the time complexity of simulating the continuous Hopfield networks are studied in detail in the following work [25].

5.2 Numerical results

The experiment aims at solving problems of different natures (random, academic and real-world problems). These series represent a large spectrum of instances (benchmarks). These experiments are executed in personal computer with a 2.79 GHz processor and 512 MB RAM. The performance has been measured in terms of the CPU time per second.

$$x_{ik} = 0.999 + \frac{d_i + 1 - k}{d_i} 10^{-5} U$$

Where $i = 1,...,N, k = 1,...,d_i$ and U is a random uniform variable in the interval $[-0.5, 0.5]$. Recall that, N is the number of variables. Based on a series of the experiments, α and ε are fixed by the following values:

$$\alpha = \frac{1}{N},\ \varepsilon = 10^{-4}.$$

The rest of parameters setting γ, β and ϕ are computed:

$$\phi = d\alpha + 2\varepsilon,\ \gamma = \phi / 2 \text{ and } \beta = \varepsilon - 3\gamma.$$

In our experimentation, some instances to be solved were chosen from the library benchmarks[21], are used to test this software.

A statistical study was performed in order to study the quality of our approach. This study is based on the calculation operator's performance:

- Ratio mode: the most repetitive (mode) optimal value obtained by CHN in number of run,
- Ratio mean: the average of the optimal value in a number of run,
- Ratio minimum : the smaller optimal value (better results obtained by our solver),
- Mean of CSP time: the average time consumed to obtain the solution in a number of run.

In this context, for each instance, if the minimum ratio is equal to 0, then the proposed approach has found the solution. Otherwise, the proposed approach has not found a solution.

The simulation results are presented in Table (1). Compared with other CSP solvers, the time resolution obtained by our approach is better than others, it does not exceed a few seconds in most cases.

We also compared our results with those given by two different solvers Abscon [29] And Concrete [30]. These results show that the continuous Hopfield network is an effective tool for solving constraint satisfaction problems. In general, our approach is largely effective; it can obtain a solution in a minimum time for most instances (Figure 1).

6. CONCLUSION

In this paper, we have proposed a new approach for solving binary constraint satisfaction problems. The interesting steps of this approach are: applying the algorithm *AC*3 to reduce the size of the *CSP* problems, modeling the filtered problem as a 0-1 quadratic programming subject to linear constraints and using the continuous Hopfield networks to solve this problem. It is also interesting to note that this method can be used with non-binary *CSP*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problems after converting the latter into binary *CSP* problems [22]. The experimental results show that our method can find a good optimal solution in a short time. Several directions can be investigated to try to improve this method such as reducing the architecture of Hopfield neural networks [9]. Other studies are in progress to apply this approach to many real-world problems such as aircraft conflict, timetabling and recognition.

TABLE 1: Computational Result Of The Typical CSP Instances

Name of instance	Number of execution	Ratio minimum	Ratio mean	Ratio mode	Mean of iterations	Mean of CPU Times
hanoi-3	200	0	0	0	89.8	0.004s
hanoi-4	200	0	0	0	97.91	0.039s
hanoi-5	200	0	0	0	108.04	2.217s
rand-25-25-300-147-53021	100	0	5.31	4	298.24	1.919s
rand-2-40-16-250-350-21	100	0	8.64	5	308.64	1.085s
rand-2-50-23-587-230-24	100	0	3.51	2	336.7	4.132s
frb35-17-4-mgd	100	0	6.03	4	288.02	0.909s
frb35-17-2-mgd	100	0	8.75	5	269.09	0.858s
frb50-23-5-mgd	100	0	13	11	360.14	4.304s
frb50-23-3-mgd	100	0	5.75	4	355.57	4.298s
geom-30a-6	200	0	8.46	9	6	0.035s
geo50-20-d4-75-46	100	0	5.49	5	327.14	2.920s
qcp-10-67-0	50	0	8.56	6	88	0.094s
qcp-20-187-3	25	0	13.88	8	116.72	1.446s
qwh-10-57-2	50	0	9.28	2	66.3	0.040s
qwh-15-106-7	25	0	9.88	3	76.88	0.246s
qwh-25-235-1	25	0	3.56	2	94.4	2.777s
queens-5-5-5	100	0	2.23	2	66.33	0.010s
langford-2-4	200	0	1.83	2	68.01	0.001s
myciel-5g-6	100	0	3.34	1	128.16	0.081s
driverlogw-08cc-sat	25	1	6.08	4	326.16	6.487s
composed-25-10-20-5	50	0	14.84	5	244.12	2.147s
geom-40-5	200	0	7.92	3	137.92	0.041s
geom-40-6	200	0	5.92	2	164.48	0.071s
qwh-20-166-5	25	0	9.04	4	79.96	0.904s
qwh-20-166-9	25	0	2.68	1	78.16	0.869s
rand-23-23-253-131-49021	100	0	12.43	5	286.51	0.965s
driverlogw-09-sat	25	2	15.8	12	360.44	21.575s
driverlogw-01c-sat	100	0	5.65	3	87	0.022s
queens-12	200	0	11.06	8	151.24	0.033s

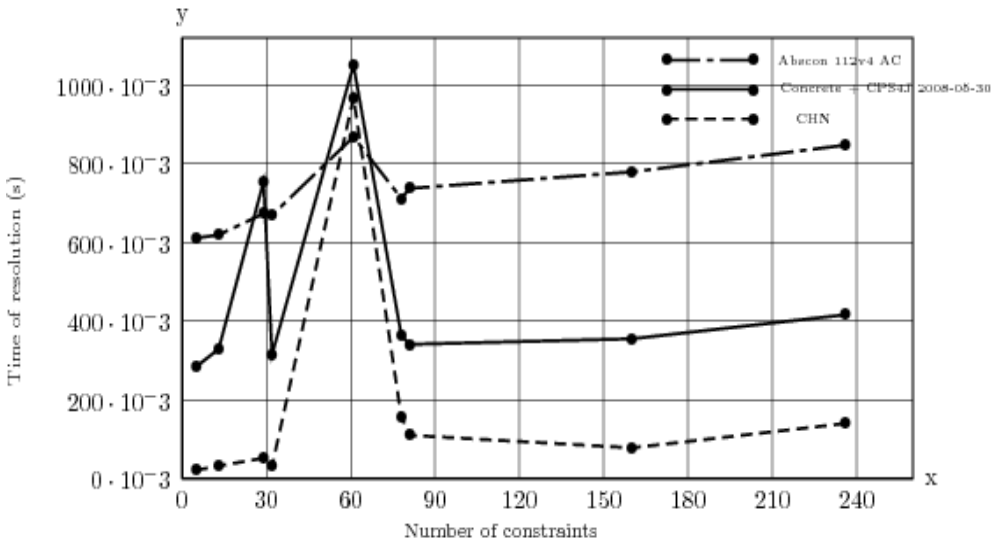

Figure 1: The Graph Of Comparison Between The Proposed Approach And The Other Solvers.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] A. AKOUM, B. DAYA and P. CHAUVET "Two neural networks for license number plates recognition", *Journal of Theoretical and Applied Information Technology*, Vol. 12, No. 1, 2010, pp. 25-32.

[2] F. Bacchus and A. Grove, "On the Forward Checking algorithm. In Principles and Practice of Constraint Programming, " *Springer-Verlag, New York*, Vol. 976, 1995, pp. 292–309.

[3] C. Bessière and J-C. Régin, "Refining the basic constraint propagation algorithm," *In proceedings of the Seventeeth International Joint conference on Artificial Intelligence (IJCAI')*, 2001, pp. 309–315.

[4] C. Bessière, "Arc-consistency and arc-consistency again," *Artificial Intelligence*, Vol. 65, No. 1, 1994, pp. 179–190.

[5] R. Dechter and D. Frost, "Backtracking algorithms for constraint satisfaction problems," *Constraints, International Journal*, 1998.

[6] M. Ettaouill, "A 0-1 quadraric knapsack problem for modelizing and solving the constraint satisfaction problem," *in Lecture Notes in Artificial Intelligence N°1323, Springer Verlag, Berlin*, 1997, pp. 61–72.

[7] M. Ettaouill and C. Loqman, "Constraint Satisfaction Problems Solved by Semidefinite Relaxations," *WSEAS TRANSACTIONS on COMPUTERS*, Vol. 7, No. 7, 2008, pp. 951–961.

[8] M. Ettaouill and C. Loqman, "A New Optimization Model for Solving the Constraint Satisfaction Problem," *Journal of Advanced Research in Computer Science*, Vol. 1, No. 1, 2009, pp. 13–31.

[9] M. Ettaouill and Y. Ghanou, "Neural architectures optimization and genetic algorithms," *WSEAS TRANSACTIONS on COMPUTERS*, Vol. 8, No. 3, 2009, pp. 526–537.

[10] M. Ettaouil, K. Haddouch and C. Loqman, "Quadratic programming and genetic algorithms for solving a Constraint Satisfaction Problems (*CSP*)", *JOURNAL OF COMPUTING*, Vol. 4, No. 4, APRIL 2012, pp. 43-52.

[11] A. Ghosh and all., "Object background classification using Hopfield type neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 6, No. 5, 1992, pp. 989-1008.

[12] R. M. Haralick and G. L. Elliott, "Increasing tree search efficiency for constraint satisfaction problems," *Artificial Intelligence*, Vol. 14, 1980, pp. 263–313.

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 79, 1982, pp. 2554–2558.

[14] J. J. Hopfield and D.W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, Vol. 52, 1985, pp. 1–25.

[15] S. Hosseini, K. Zamanifar, "Using a Conflict-Based Methed to Solve Distributed Constraints Satisfaction Problems," *Journal of Theoretical and Applied Information Technology*, Vol. 5, No. 4, April 2009, pp. 446-463.

[16] A. Idrissi, "Some methods to treat capacity allocation problems", *Journal of Theoretical and Applied Information Technology*, Vol. 37 No.2, 31March 2012, pp. 141-158.

[17] D. Kaznachey, A. Jagota and S. Das, "Neural network-based heuristic algorithms for hypergraph coloring problems with applications," *Journal of Parallel and Distributed Computing*, Vol. 63, No. 9, 2003, pp. 786-800.

[18] V. Kumar, "Algorithms for Constraint Satisfaction Problems: A Survey," AI Magazine, Vol. 13, No. 1, 1992, pp. 32–44.

[19] C. Lecoutre, F. Boussemart and F. Hemery, "Revision ordering heuristics for the Constraint Satisfaction Problem," *1st International Workshop on Constraint Propagation and Implementation (CPAI'04)*, 2004, pp. 9–43.

[20] C. Lecoutre, "Benchmarks in XCSP 2.1 XML representation of CSP/WCSP/QCSP instances," Available at http://www.cril.univ-artois.fr/lecoutre/research/benchmarks/benchmarks.html, 2008.

[21] C. Lecoutre, "Constraint Networks: Techniques and Algorithms," *ISTE/Wiley*, 2009.

[22] N. Mamoulis and K. Stergiou, "Solving non-binary CSPs using the Hidden Variables Encoding," *In Proceedings of CP' (2001).*

[23] AK. Marckworth, "Consistency in networks of relations," *Artificial Intelligence*, Vol. 8, 1977, pp. 99–118.

[24] R. Mohr and TC. Henderson, "Arc and path consistency revisited," *Artificial Intelligence*, Vol. 28, 1986, pp. 225–233.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[25] G. Serpen, "Managing spatio-temporal complexity in Hopfield neural network simulations for large-scale static optimization," *Mathematics and Computers in Simulation*, Vol. 64, 2004, pp. 279–293.

[26] P.M. Talaván and J. Yáñez, "A continuous Hopfield network equilibrium points algorithm," *Computers and Operations Research*, Vol. 32, 2005, pp. 2179–2196.

[27] P.M. Talaván and J. Yáñez, "The quadratic assignment problem. A neuronal network approach," *Neural Networks*, Vol. 19, No. 4, 2006, pp. 416-428.

[28] H. Shouval and all., "An all-optical Hopfield network: theory and experiment. A neuronal network approach," *International Journal of Neural Systems*, Vol. 1, No. 4, 1991, pp. 355–360.

[29] "CSP 2008 Competition: benchmarks results per solver Result page for solver Abscon 112v4 AC, " *Available at http://www.cril.univ-artois.fr/CPAI08/results/solver.php?idev=15&idsolver=363, 2008*.

[30] "CSP 2008 Competition: benchmarks results per solver Result page for solver Concrete + CPS4J 2008-05-30", *Available at http://www.cril.univ-artois.fr/CPAI08/results/solver.php?idev=15&idsolver=305, 2008.*