Improved Optimal Competitive Hopfield Network for the Maximum Stable Set Problem

Mohamed Ettaouil [a] * Chakir Loqman [a] å and Karim Elmoutaouakil [a] •
UFR: Scientific Computing and Computer sciences, Engineering sciences
[a] Faculty of Science and Technology. University Sidi Mohamed Ben Abdellah, Box 2202, Fez, Morocco

***Abstract*—A large number of problems in artificial intelligence and other areas of computer science can be viewed as special cases of the Maximum Stable Set Problem (MSSP). In this paper, we propose a new approach to solve the MSSP problem using the continuous Hopfield network (CHN). The proposed method is divided into two steps: the first one involves modeling the MSSP problem as a 0-1 quadratic programming, and solving this model via the CHN which rapidly gives a local minimum. The second step concerns improving the initial solution by adding a linear constraint to the first model; then, we use the CHN to solve the obtained model. We prove that this approach is able to determine a good solution of the MSSP problem. To test the theoretical results, some computational experiments solving the MSSP problem are shown.**

Keywords- maximum stable set problem; quadratic 0-1 programming; continuous Hopfield network

I. INTRODUCTION

The Maximum Stable Set Problem (MSSP) consists of finding a stable set in graph G of maximum cardinality $\alpha(G)$. Aside from its theoretical interest, the MSSP problem arises in applications in information retrieval, experimental design, signal transmission, and computer vision [1]. The stable set problem is NP-hard in the strong sense, and hard even to approximate[11]. The MSSP problem can be solved using polynomial time algorithms for special classes of graphs such as perfect graphs and t-perfect graphs, circle graphs and their complements, claw-free graphs, and graphs with long odd cycles [13]. But, the existence of a polynomial time algorithm for arbitrary graphs seems unlikely.

Different approaches have been discussed in the literature to solve the maximum stable set problem exactly. An implicit enumeration technique of Carrahan's and Pardalos's [3], computational results for different stable set linear programming relaxations have been reported by Gruber and Rendl [10], an effective evolution of the tabu search approach is presented in the original work of Friden, Hertz and de Werra [8]. The MSSP problem can be solved via the Continuous Hopfield Network (CHN).

The CHN was proposed by Hopfield and Tank [12] to solve combinatorial problems; some authors have treated the Quadratic Knapsack Problem (QKP) through this neuronal approach [2],[9],[16]. Within these papers, the feasibility of the equilibrium points of the CHN cannot, for the general case, be assured; moreover, the solutions obtained are, often, not good enough. To avoid this problem, a general methodology was proposed to solve the Generalized Quadratic Knapsack Problem $(GQKP)$ [14].

Since the differential equation, which characterizes the dynamics of the CHN, is analytically hard to solve, many researchers used to make use of the famous Euler method. However, this latter proved to be highly sensitive with respect to initial conditions, and it requires a lot of CPU time for medium or greater size CHN instances. That is why a robust algorithm was proposed to calculate an equilibrium point [15]. After these ameliorations, the CHN was used to solve the Traveling Salesmen Problem [14], Constraint Satisfaction Problem [7] and the Placement of the Electronic Circuits Problem [5].

In this work, we propose a new approach to solve the MSSP problem. The proposed method is divided into two steps: the first one involves modeling the MSSP problem as a 0-1 quadratic programming, and solving this model via the continuous Hopfield network, we call the obtained stable set the initial solution of the MSSP problem. The second step concerns improving the initial solution by adding a linear constraint to the first model; then, we use the CHN to solve the obtained model. In this approach, we prove that the integrated constraint plays a central role for ameliorating the initial solution.

This paper is organized as follows: In section 2, we present an introduction of the continuous Hopfield network. The maximum stable set problem is modelized as a 0-1 quadratic program in the section 3. In section 4, we use the CHN to calculate an initial stable set in a short time. Section 5 is devoted to improve the initial solution. Implementation details of the proposed approach and experimental results are presented in the last section.

II. THE CONTINUOUS HOPFIELD NETWORK (CHN)

In the beginning of the 1980s, Hopfield published two scientific papers, which attracted much interest. This was the starting point of the new era of neural networks, which continues today. Hopfield showed that models of physical systems could be used to solve computational problems. Moreover, Hopfield and Tank [12] presented the energy function approach in order to solve several optimization problems including the traveling salesman problem (TSP), analog to digital conversion, signal processing problems and linear programming problems. Their results encouraged a number of researchers to apply this network to different problems such as object recognition, graph recognition,graph coloring problems, economic dispatch problems and constraint satisfaction problems.

The CHN of size n is a fully connected neural network with n continuous valued units. Let $T_{i,j}$ be the strength of the connection from neuron j to neuron i. Each neuron i has an offset bias of i_i^b.

The dynamics of the CHN is described by the differential equation:

$$\frac{du}{dt} = -\frac{u}{\tau} + Tx + i^b \quad (1)$$

where u, x and i_b will be the vectors of neuron states, outputs and biases.

The output function $x_i = g(u_i)$ is a hyperbolic tangent, which is bounded below by 0 and above by 1.

$$g(u_i) = \frac{1}{2}(1 + \tanh(\frac{u_i}{u_0})) \qquad where \qquad u_0 > 0$$

where u_0 is a parameter used to control the gain (or slope) of the activation function.

If, for an input vector u_0, a point u^e exists such that $u(t) = u^e \quad \forall t \ge t_e$, for some $t_e \ge 0$, this point is called an equilibrium point of the system defined by the differential equation (1). Such an equilibrium point will also be called the limit point of the CHN. The existence of equilibrium points for the CHN is guaranteed if a Lyapunov or an energy function exists. Hopfield showed that, if matrix T is symmetric, then the following Lyapunov function exists [12]:

$$E(x) = -\frac{1}{2} x^t T x - (i^b)^t x + \frac{1}{\tau} \sum \int g^{-1}(v) dv$$

The CHN will solve those combinatorial problems, which can be expressed as the constrained minimization of:

$$E(x) = -\frac{1}{2} x^t T x - (i^b)^t x \quad (2)$$

which has its extremes at the corners of the n-dimensional hypercube $[0,1]^n$. The idea is that the networks Lyapunov function, when $\tau \to \infty$, is associated with the cost function which will be minimized in the combinatorial problem. In this way, the CHN output can be used to represent a solution of the combinatorial problem. This process has been termed mapping the problem onto the Hopfield network and is described in the following way for the quadratic assignment problem.

Given the combinatorial optimization problem with n variables and m linear constraints

$$(P1) \quad \begin{cases} Min & \frac{1}{2} x^t Q x + q^t x \\ Subject\ to & \\ & Ax = b \\ & x_i \in \{0,1\} \qquad i = 1,...,n \end{cases}$$

To solve the quadratic programming $(P1)$, the following sets are needed:

- H is a set of the Hamming hypercube :

$$H \equiv \{x \in [0,1]^n\}$$

- H_C is a set of the Hamming hypercube corners :

$$H_C \equiv \{x \in H : x_i \in \{0,1\},\ i = 1,...,n\}$$

- H_F is a set of feasible solutions :

$$H_F \equiv \{x \in H_C : Ax = b\}$$

Remark 1

Given an instance (n, m, Q, q, A, b) of the problem $(P1)$, some conditions must be imposed on the mapped problem so that its equilibrium points can be associated with local minimums of that optimization problem, with m is the number of constraints.

An energy function must also be defined by :

$$E(x) = E^0(x) + E^R(x) \qquad \forall\ x \in H \quad (3)$$

Where:

- $E^0(x)$ is directly proportional to the objective function of the problem.
- $E^R(x)$ is a quadratic function that not only penalizes the violated constraints of the problem, but also guarantees the feasibility of the solution obtained by the CHN. This function must be constant $\forall x \in H_F$ and an appropriate selection of this function is crucial for correct mapping.

The following generalized energy function was previously proposed [14]:

$$E(x) = \alpha(\frac{1}{2} x^t Q x + q^t x) + \frac{1}{2}(Ax)^t \phi (Ax) + x^t diag(\gamma)(1 - x) + \beta^t Ax$$

with the parameters $\alpha \in IR$, $\gamma \in IR^n$, $\beta \in IR^m$ and the $m \times m$ matrix parameter ϕ. This energy function was introduced to overcome the problem observed with the energy functions used by other authors, including Aiyer [9] and Brandt et al. [2].

In this work, our objective is to solve the maximum stable set problem (MSSP) using the continuous Hopfield network. In this case, the most important step consists of representing or mapping the MSSP problem in the form of an energy function associated with the continuous Hopfield network. In the next section, we will present a modelization of the MSSP problem as a quadratic 0-1 programming. According to this model, this step of mapping becomes easy and more general.

III. Formulation of the Maximum Stable Set Problem

Given an undirected graph $G = (V, E)$ with $V = \{v_1, v_2 ..., v_n\}$. A stable set of a graph G is a set of nodes S with the property that the nodes of S are pairwise non adjacent. The maximum stable set problem (MSSP) consists to find a stable set of maximum size.

To solve the MSSP problem via the CHN, it must be expressed as a linear assignment problem with a quadratic constraint.

Let $S \subset V$ be a stable set of nodes. For each node v_i of the graph G, we introduce the binary variables x_i such that:

$$x_i = \begin{cases} 1 & if \quad v_i \in S \\ 0 & Otherwise \end{cases}$$

• Two adjacent nodes v_i and v_j cannot be in the set S:

$$(v_i, v_j) \in E \Rightarrow x_i x_j = 0 \quad (4)$$

The constraints (4) can be aggregated in a single one:

$$h(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} x_i x_j = 0 \quad (5)$$

With $$b_{ij} = \begin{cases} 1 & if \quad (v_i, v_j) \in E \\ 0 & Otherwise \end{cases}$$

• The objective function of the mathematical programming model is:

$$f(x) = -\sum_{i=1}^{n} x_i$$

Consequently, the MSSP problem can be expressed in the following algebraic form:

$$(QP) \begin{cases} Min & f(x) = -\sum_{i=1}^{n} x_i \\ Subject\ to & \\ & h(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} x_i x_j = 0 \\ & x \in \{0,1\}^n \end{cases}$$

Example 1

We consider a sample graph with eight nodes and nine edges, which contains tree stable sets:

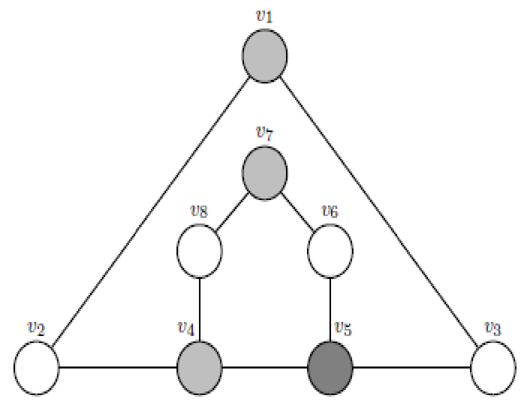

Figure 1: Example of a graph containing tree stable sets

The maximum stable set problem associated with this graph can be modelized as the following quadratic programming (QP):

$$(QP) \begin{cases} Min & f(x) = -\sum_{i=1}^{8} x_i \\ Subject\ to & \\ & h(x) = x^T B x = 0 \\ & x \in \{0,1\}^8 \end{cases}$$

$$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$and \quad x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)^T$$

In this section, the MSSP problem was modelized as a quadratic 0-1 programming which consists in minimizing the linear function subject to quadratic constraints (QP). To solve this model, many different methods are tried and tested such as interior point, semidefinite relaxations [6] and lagrangian relaxations[17]. In this paper, we introduce the continuous Hopfield network for solving the QP problem.

IV. A CLASSICAL CONTINUOUS HOPFIELD NETWORK FOR THE MSSP PROBLEM

The main purpose of this section is to apply the CHN in order to solve the maximum stable set problem (MSSP). First, we formulate the energy function associated with the MSSP problem. Then, we select a convenient parameters setting of this function.

To solve the maximum stable problem via CHN, we formulate the energy function:

$$E^0(x) = -\alpha\sum_{i=1}^{n} x_i + \frac{1}{2}\phi\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}x_i x_j + \gamma\sum_{i=1}^{n} x_i(1-x_i) \qquad (6)$$

Basing on a simple comparison between the equation (2) and the equation (6), we determine the weights and thresholds as follows:

$$\begin{cases} T_{i,j} & = -\phi b_{ij} + 2\delta_{ij}\gamma \\ i_i^b & = \alpha - \gamma \end{cases}$$

With $\delta_{ij} = \begin{cases} 1 & if\ i = j \\ 0 & if\ i \neq j \end{cases}$ is the Kronecker symbol.

The parameters ϕ, γ and α must be selected such that the equilibrium points of the CHN, associated with the MSSP, are feasible.

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

$$\frac{\partial E^0(x)}{\partial x_i} = E_i(x) = -\alpha + \phi\sum_{j=1}^{n} b_{ij}x_j + \gamma(1-2x_i)$$

The parameters-setting are determined by the hyperplane method [15]. This method involves dividing the Hamming hypercube H by a hyperplane containing all feasible solutions so that the evolution of CHN verifies two properties: any solution not belonging to this hyperplane will be pushed on to it and, second, any infeasible solution belonging to the hyperplane is ejected from it.

Before presenting this method, some conditions are imposed to simplify the determination of these parameters-setting:

$$\phi > 0,\ \gamma \geq 0$$

- To minimize the objective function, we impose the following constraint:

$$\alpha > 0$$

- The following constraint is necessary to avoid the stability in the interior points $x \in H - H_C$:

$$T_{i,i} = 2\gamma \geq 0$$

Since the problem MSSP has only one constraint, we have:

$$H_C - H_F = \{x \in H_C / h(x) > 0\}$$

Let $x \in H_C - H_F$, in this case, two adjacent nodes v_i and v_j are in the stable set S, then $x_i = x_j = 1$ and therefore the value x_i will decrease if $E_i^0(x) \geq \varepsilon$ where $\varepsilon > 0$.

The following constraint is obtained:

$$-\alpha + \phi - \gamma \geq \varepsilon$$

Joining all of these parametric constraints yields the following:

$$\begin{cases} \alpha > 0,\ \phi > 0,\ \gamma \geq 0 \\ -\alpha + \phi - \gamma \geq \varepsilon \end{cases}$$

A feasible solution could be the following:

$$\begin{cases} \alpha > 0,\ \phi > 0,\ \gamma \geq 0 \\ -\alpha + \phi - \gamma = \varepsilon \end{cases}$$

Finally, the weights and thresholds of the *CHN* can be calculated using these parameters setting. Then, the continuous Hopfield network converges rapidly to a local minimum which is sometimes not good. In order to avoid this shortcoming, we propose a new approach basing on the integration of a linear constraint in the first model (QP) which ensures the improvement of the initial solution.

V. AMELIORATION OF THE CONTINUOUS HOPFIELD NETWORK FOR THE MSSP PROBLEM

Since an analog neural network converges rapidly to a local minimum, we can run the network many times starting from different initial conditions within a short period of time; eventually, we may find a good solution. Unfortunately, this cannot be guaranteed; consequently, the solutions obtained are sometimes not good. To improve this solution, we integrate a linear and simple constraint to the initial quadratic programming problem (QP):

$$\sum_{i=1}^{n} x_i > v_0 \qquad (7)$$

where v_0 is the value of the QP problem given by the CHN in the first step.

Then, we obtain the following problem :

$$(QP_1)\begin{cases} Min \qquad f(x) = -\sum_{i=1}^{n} x_i \\ Subject\ to \\ \qquad h(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}x_i x_j = 0 \\ \qquad \sum_{i=1}^{n} x_i > v_0 \\ \qquad x \in \{0,1\}^n \end{cases}$$

In order to map the QP_1 problem onto the continuous Hopfield network, the inequality (7) must be transformed into an equation through the slack variable $x_{n+1} \in [0,1]$. In this way, the QP_1 problem is stated as:

$$(QP_1)\begin{cases} Min & f(x) = -\sum_{i=1}^{n} x_i \\ Subject\ \ to & \\ & h(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} x_i x_j = 0 \\ & -\sum_{i=1}^{n} x_i + (n-v_0)x_{n+1} = -v_0 \\ & x_i \in \{0,1\}, \quad i = 1,..., n \\ & x_{n+1} \in [0,1] \end{cases}$$

To solve the QP_1 problem via the CHN, we use the energy function defined as follows:

$$E^1(x) = -\alpha f(x) + \frac{1}{2}\phi h(x) + \frac{\xi}{2}(-\sum_{i=1}^{n} x_i + (n-v_0)x_{n+1})^2 - \rho(\sum_{i=1}^{n} x_i - (n-v_0)x_{n+1}) + \gamma\sum_{i=1}^{n} x_i(1-x_i) \quad (8)$$

By a simple comparison between the equation (2) and the equation (8), we obtain:

$$\begin{cases} T_{i,j} & = -\phi b_{ij} - \xi + 2\delta_{i,j}\gamma & i,j = 1,...,n \\ T_{i,n+1} & = T_{n+1,i} = \xi\,(n-v_0) & i = 1,...,n \\ T_{n+1,n+1} & = -\xi\,(n-v_0)^2 & \\ i_i^b & = \alpha + \rho - \gamma & i = 1,...,n \\ i_{n+1}^b & = -\rho(n-v_0) & \end{cases}$$

The selection of the parameters ϕ, ξ, γ, α and ρ must give a stable set S_1 which contains more nodes than S_0. But for simplicity reasons, the following parameter constraints are first assigned:

$$\phi > 0,\ \ \gamma \geq 0,\ \ \rho \geq 0,\ \ \xi \geq 0$$

The parameter-setting procedure, which ensures the feasibility of the solution, is based on the partial derivatives of the generalized energy function:

$$\forall i \in \{1,..., n\}$$

$$\begin{cases} \dfrac{E^1(x)}{\partial x_i} = -\alpha + \phi\sum_{j=1}^{n} b_{ij} x_j + \xi(\sum_{i=1}^{n} x_i - (n-v_0)x_{n+1}) - \rho + \gamma(1-2x_i) \\ \dfrac{E^1(x)}{\partial x_{n+1}} = \xi(n-v_0)(-\sum_{i=1}^{n} x_i + (n-v_0)x_{n+1}) + \rho(n-v_0) \end{cases}$$

• To minimize the objective function, we impose the following constraint:

$$\alpha > 0$$

• The following constraint is necessary to avoid the stability of the interior points:

$$T_{i,i} = -\xi + 2\gamma \geq 0 \qquad i = 1,..., n$$

Since the QP_1 problem has two constraints, The partition of $H_C - H_F$ is defined as : $H_C - H_F = W_{0,0} \bigcup W_{1,0} \bigcup W_{1,1}$

• $W_{0,0} = \{h(x) > 0\}$, in this case, two adjacent nodes v_i and v_j are in the stable set S, then $x_i = x_j = 1$ and $b_{ij} = 1$. Therefore, the value x_i will decrease if $\dfrac{E^1(x)}{\partial x_i} \geq \varepsilon$ where $\varepsilon > 0$. Then, The following constraint is obtained:

$$-\alpha + \phi + 2\xi - \xi(n-v_0) - \rho - \gamma \geq \varepsilon$$

• $W_{1,0} = \{-\sum_{i=1}^{n} x_i + (n-v_0)x_{n+1} > -v_0\} \cap \{-\sum_{i=1}^{n} x_i \leq -v_0\}$.
In this case, the slack variable x_{n+1} must be decreased so that the partial derivative $\dfrac{E^1(x)}{\partial x_{n+1}} > 0$. Then, the following constraint is obtained:

$$-\xi v_0 + \rho \geq 0$$

• $W_{1,1} = \{-\sum_{i=1}^{n} x_i + (n-v_0)x_{n+1} < -v_0\}$. In this case, the slack variable x_{n+1} must be decreased so that the partial derivative $\dfrac{E^1(x)}{\partial x_{n+1}} < 0$. Then, The following constraint is obtained:

$$-\xi v_0 + \rho \leq 0$$

Joining all of these parametric constraints yields the following:

$$\begin{cases} \alpha > 0,\ \ \phi > 0,\ \ \gamma \geq 0 \\ -\xi + 2\gamma \geq 0 \\ -\alpha + \phi + 2\xi - \xi(n-v_0) - \rho - \gamma \geq \varepsilon \\ -\xi v_0 + \rho \geq 0 \\ -\xi v_0 + \rho \leq 0 \end{cases}$$

These parameters setting are determined by fixing α, ε, γ and computing the rest of the parameters ξ, ρ and ϕ:

- $\xi = 2\gamma$,
- $\rho = \xi v_0$,
- $\phi = \gamma(2n-3) + \varepsilon + \alpha$.

The following theorem prove that the integration of the linear constraint ensures the improvement of the solution given by CHN in the first step.

Theorem 1

Let S_0 *and* S_1 be, respectively, the stable set obtained by solving the problem QP and QP_1 using the CHN. We have $|S_0| < |S_1|$

Where $|S_i|$ presents the the number of elements in the set S_i.

Proof

If S_0 presents the stable set associated with the solution x^0 of the QP problem, then $|S_0| = \sum_{i=1}^{n} x_i^0$.

If S_1 presents the stable set associated with the solution x^1 of the QP_1 problem, then $|S_1| = \sum_{i=1}^{n} x_i^1$.

Since x^1 satisfied the constraints of the QP_1 problem, we have

$$|S_1| = \sum_{i=1}^{n} x_i^1 > |S_0|$$

Thus

$$|S_1| > |S_0|$$

Remark 2

- If the initial solution is a global one, then the set of feasible solutions of the QP_1 problem is empty; in this case, the algorithm which calculates the equilibrium point converges based on the number of iterations. This convergence is characterized by the repetition of the unstable point [15]. Consequently, we keep the initial solution because it is a global one.

- We can repeat this process several times in order to solve the max stable set problem. But, we want, only, to improve the solution of the maximum stable set problem in a short time.

VI. Computational Experiments

In order to show the practical interest of the approach proposed in this paper, we have worked on a series of experimentations to solve the max stable set problem. Most of the graphs are taken from the 2nd DIMACS Challenge [4]. These graphs were contributed as test problems for solving the maximum clique problem. For these graphs, we took the complement of these graphs and applied our maximum stable set approach. the results are supplied in table 1. The CPU time was recorded using an IBM compatible PC (Pentium IV, 1.82 GHz and 512 MB of RAM) running through *Java language*.

The initial states are randomly generated:

$$x_i = 0.999 + \frac{n+1-i}{n} 10^{-5} w$$

Where $i = 1,\ldots,n$ and w is a random uniform variable in the interval $[-0.5, 0.5]$.

Recall that, n is the number of the nodes.

-In the first step:

We choose the parameters $\alpha = 1.0250$, $\varepsilon = 10^{-6}$ and $\gamma = 0.7$; the parameter ϕ was computed from the equation $\phi = \alpha + \gamma + \varepsilon$.

-In the second step:

We choose the parameters $\alpha = 10.3329$, $\varepsilon = 10^{-4}$ and $\gamma = 4.34001$; the parameters ξ, ϕ and ϕ were computed from the equations $\xi = 2\gamma$, $\rho = \xi v_0$ and $\phi = \gamma(2n-3) + \varepsilon + \alpha$

TABLE I. NUMERICAL RESULTS

Graph	V	$\|E\|$	$\alpha(G)$	CHN1		CHN2	
				$\alpha_1(G)$	time(s)	$\alpha_2(G)$	time (s)
brock200-2	200	10024	12	8	0.312	10	0.423
brock200-3	200	7852	15	10	0.297	11	0.401
brock400-4	400	20035	33	18	1.250	25	0.378
C.125.9	125	787	34	32	0.046	34	0.098
C.250.9	250	3141	44	34	0.047	36	0.108
c-fat200-1	200	18336	12	12	0.281	-	0.473
c-fat200-2	200	16665	24	24	0.265	-	0.413
c-fat200-5	200	11427	58	58	0.125	-	0.272
DSJC125.1	125	736	34	30	0.125	34	0.212
DSJC125.5	125	3891	10	9	0.087	10	0.193
DSJC125.9	125	6961	4	4	0.031	-	0.072
mann-a9	45	72	16	12	0.0 31	15	0.063
mann-a27	378	702	126	117	0.656	122	0.732
gen200-p0.9-44	200	1990	44	31	0.031	33	0.043
gen400-p0.9-65	400	7980	65	41	0.641	59	0.802
hamming6-2	64	192	32	32	0.047	-	0.063
hamming6-4	64	1312	4	3	0.015	4	0.027
johnson8-2-4	28	186	4	4	0.031	-	0.045
johnson8-4-4	70	560	14	9	0.047	11	0.063
johnson16-2-4	120	1620	8	8	0.125	-	0.198
keller4	171	5100	11	6	0.188	9	0.256
p-hat300-1	300	33917	8	6	0.688	8	0.931
p-hat300-2	300	22922	25	22	0.688	24	0.734
p-hat300-3	300	11460	36	31	0.578	34	0.712
san200-0.7-2	200	5970	18	12	0.078	16	0.056
san200-0.9-3	200	1990	44	31	0.015	40	0.021

Legend of the Table I

- - : the repetition of the unstable point.
- $\alpha_i(G)$: the size of the stable set obtained by CHN in the step i.

The result in the theorem 1 is reinforced with the experimental results. In fact, the columns five and seven of the table 1 show that $\alpha_1(G) < \alpha_2(G)$ when $\alpha_1(G) \neq \alpha(G)$.

As it is shown in the table 1, our method gives the maximum stable set in the first step for some instances graphs, for example $\alpha_1(c-fat) = \alpha(c-fat)$. Finally, the best results are obtained by this approach. From theoretical point of view, our approach is very powerful, as it can solve the large MSSP problems in a short time.

VII. CONCLUSION

In this paper, we have proposed a new approach to solve the MSSP problem using the continuous Hopfield network (CHN). The proposed method is divided in two steps: the first one involves modeling the MSSP problem as a 0-1 quadratic programming, and solving this model via the CHN which gives, rapidly, a local minimum. The second step concerns improving the initial solution by adding a linear constraint to the first model; then, we use the CHN to solve the obtained model. We have proved that the proposed approach is able to determine a good solution of the MSSP problem. To test the theoretical results, some computational experiments solving the MSSP problem were presented. A future direction of this research is to apply this approach to improve the solution of the Graph Coloring Problem [14] given by the CHN; moreover, this method can be generalized to improve the solution of the problems which have a quadratic objective function. In the upcoming project, we will suggest a new reduction to reduce the time used by CHN to calculate a solution for an optimization problem. The new reduction and the proposed method will be combined to improve the CHN solution in a short time.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks for the referee for his/her helpful suggestions.

REFERENCES

[1] E. Balas and C. S. Yu, *Finding a maximum clique in arbitrary graph*. SIAM Journal on computing, 15(4), 1986, pp. 1054-1068.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] R.D. Brandt, Y. Wang, A.J. Laub, and S.K. Mitra, *Alternative networks for solving the travelling salesman problem and the list-matching problem*. vol. II, proceedings of the International Conference on Neural Networks, San Diego, USA, 1988, pp. 333-340.

[3] R. Carrahan and P. M. Pardalos, *An exact algorithm for the maximum clique problem*. Operations Research Letters, 9, 1990, pp. 375-382.

[4] DIMACS (1992). *The Second DIMACS Implementation Challenge*. ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/.

[5] M. Ettaouil, K. Elmoutaouakil and Y. Ghanou, *The Continuous Hopfield Networks (CHN) for the Placement of the Electronic Circuits Problem*. Journal of Wseas Transactions on Computers, 12(8), 2009, pp. 1865-1874.

[6] M. Ettaouil, C. Loqman, *Constraint Satisfaction Problems Solved by Semidefinite Relaxations*. Journal of Wseas Transactions on Computers, 7(7), 2008, pp. 951-961.

[7] M. Ettaouill and C. Loqman, *A New Optimization Model for Solving the Constraint Satisfaction Problem*, Journal of Advanced Research in Computer Science, 1(1), 2009, pp. 13-31.

[8] C. Friden, A. Hertz, and D. de Werra, *Stabulus: a technique for finding stable sets in large graphs with tabusearch*. Computing, 42, 1989, pp. 35-45.

[9] A.H. Gee and S.V.B. Aiyer, and R.W. Prager, *An analytical framework for optimizing neural networks*. Neural Networks, 6, 1993, pp. 79-97.

[10] G. Gruber and F. Rendl, *Computational experience with stable set relaxations*. SIAM J Opt, 13, 2003, pp. 1014-1028.

[11] J. Håstad, Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math., 182, 1999, pp. 105-142.

[12] J.J. Hopfield, D.W. Tank, *Neural computation of decisions in optimisation problems*. Biological Cybernetics, 52, 1985, pp. 1-25.

[13]] C. Mannino and A. Sassano, *An exact algorithm for the maximum cardinality stable set problem*. J. Computational Optimization and Applications, 3, 1994, pp. 243-258.

[14] P.M. Talaván, J. Yàñez, *The generalized quadratic knapsack problem*.A neuronal network approach, Neural Networks 19, 2006, pp. 416-428.

[15] P.M. Talaván and J. Yàñez,*A continuous Hpfield network equilibrium points algorithm*. computers and operations research, 32, 2005, pp. 2179-2196.

[16] K. Tatsumi, Y. Yagi, and T. Tanino, *Improved projection Hopfield network for the quadratic assignment problem*. SICE 2002 proceedings of the 41 st SICE annual conferencen, 4, 2002, pp.2295-2300.

[17] B. Thiongane, A. Najih and G. Plateau, *An Adapted Step Size Algorithm for a 0-1 Biknapsack Lagrangean Dual*. Annals of Operations Research, 139(1), 2005, pp. 353-373.