An improved competitive Hopfield network with inhibitive competitive activation mechanism for maximum clique problem

Gang Yang, Nan Yang, Junyan Yi, Zheng Tang

A R T I C L E   I N F O

Article history:
Received 19 June 2012
Received in revised form 3 October 2012
Accepted 9 November 2012

Keywords:
Competitive Hopfield network
Maximum clique problem
Inhibitive competitive mechanism
Community detection

A B S T R A C T

In this paper, we analyze the formula of weights definition in the discrete competitive Hopfield network (DCHOM) and point out its flaw when using it to solve some special instances of maximum clique problem (MCP). Based on the analysis, we propose an improved competitive Hopfield network algorithm (ICHN). In ICHN, we introduce a flexible weight definition method which excites the competitive dynamics, and we also present an initial values setting strategy which efficiently increases the probability of finding optimal solutions. Furthermore, an inhibitive competitive activation mechanism is introduced to form a new input updating rule which reduces significantly the number of neurons with an intermediate level of activations. Our algorithm effectively overcomes the flaw of the DCHOM, and exhibits powerful solving ability for the MCP. Experiments on the benchmark problems and practical applications verify the validity of our algorithm.

1. Introduction

Maximum clique problem (MCP) is a representative problem of combinatorial optimal problem and it is also a well-studied NP-Hard problem [13,4]. This problem is computationally intractable even when it is used to approximate with certain absolute performance bounds. MCP has many practical applications in diverse fields such as computer vision, information retrieval, cluster analysis, fault tolerance. Moreover many important problems such as maximum independent set, minimum vertex covering, quadratic zero-one problem can be easily reduced to MCP [20]. Hence, it is becoming more and more important to find the optimal and near-optimal solutions to MCP in practical applications and theory researches [19]. In 1992, a maximum Hopfield network was successfully proposed by Takefuji et al. [20] and Lee et al. [14] to handle a class of NP-complete optimization problems, including MCP, which was often hard to solve by neural networks. Then researchers Funabiki et al. summarized several basic algorithms that can solve the MCP effectively [17]. In 2003, Galán et al. modeled competitive Hopfield-type neural networks such as maximum neural network to solve MCP, and illustrated maximum neural network using a completely synchronous model that cannot guarantee energy decrease and sometimes generates inaccurate results and oscillatory behaviors in the convergence process [9]. In 2005, Wang et al. proposed a discrete competitive Hopfield neural network which contains stochastic dynamics to escape from local minima [22]. They applied their improved competitive Hopfield neural network in some other practical problems soon after, and proposed further improvements [21]. These newly proposed methods validate the superior performance of competitive mechanism in neural networks substantially through solving some well-known benchmark problems. Subsequently, Yi et al. proposed a shrinking chaotic maximum neural network, which also applied the flexibility of competitive property of maximum neural network and chaotic dynamics sufficiently, to solve MCP [24]. Most of the conventional neural networks fundamentally utilize gradient descent dynamics, while Chen and Aihara proposed a transient chaotic neural network (TCNN) [7,8] based on the continuous Hopfield neural network (CHNN) for combinational optimization problems [6]. TCNN is a better algorithm than the other algorithms only with gradient descent dynamics, which was analyzed and improved to solve MCP in 2009 [23]. But it is difficult to balance the chaotic dynamics and gradient descent dynamics to converge to a stable equilibrium point corresponding with an acceptably near-optimal solution [11]. Compared with chaotic dynamics, competitive mechanism displays more effective intrinsic characteristics to help neural network escape from local minima. In 2008, a neuro-GA approach using a maximum neural network, along with the chaotic mutation capability of genetic algorithms, was proposed to solve the reduced maximum clique
problem [1]. Sequently, the neuro-GA algorithm was used to solve maximum fuzzy clique problem [3]. Moreover, evolutional algorithm is an important kind of powerful method to solve combinatorial optimization problems. Recently, a reactive evolutionary algorithm with guided mutation (EA/G), named R-EVO, is proposed as a typical evolutional method to solve MCP [5].

In this paper, based on the analysis of the weights definition in discrete competitive Hopfield network (DCHOM), an intrinsic flaw is pointed out when we are using it to solve some special instances of the MCP. Furthermore, we propose an improved competitive Hopfield network algorithm (ICHN), and introduce a flexible weight definition method to excite the competitive dynamics of the ICHN. For increasing the successful probability of finding optimal solutions, an initial values setting strategy is introduced into the ICHN, and an inhibitive competitive activation mechanism is embedded into neuron updating functions to reduce significantly the number of neurons with intermediate level of activations. The ICHN effectively overcomes the flaw of original maximum neural network, and exhibits powerful solving ability for MCP. Simulation results show that the ICHN has superior ability for the MCP through solving well-known benchmark problems and practical problems (MCP in community detection) within a reasonable number of iterations.

2. Maximum neuron model for MCP

Let $G = (V, E)$ be an arbitrary undirected graph, where $V = \{1, \ldots, n\}$ is the vertex set of $G$ and $E \subseteq V \times V$ is the edge set of $G$. $A = (a_{ij})_{n \times n}$ is the adjacency matrix of $G$, where $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ if $(i, j) \notin E$. The complement graph of $G = (V, E)$ is the graph $\bar{G} = (V, \bar{E})$, where $\bar{E} = \{(i, j) | i, j \in V, i \neq j\}$. Given a subset $S \subseteq V$, we call $G[S] = (S, E \cap S \times S)$ the subgraph induced by $S$. A graph $G = (V, E)$ is complete if all its vertices are pairwise adjacent, that is, $\forall i, j \in V, (i, j) \in E$. A clique $C$ is a subset of $V$ such that $G[C]$ is complete. The MCP requires a clique that has the maximum cardinality.

Based on the previous researchers’ works, [20,14] formulate the MCP problem as a global minimization of the function

$$E = \sum_{x=0}^{n} \sum_{y=0}^{n} \sum_{i=1}^{2} t_{xy}v_{xi}v_{yi}$$

subject to the constraints $\sum_{i=1}^{2} v_{xi} = 1$ for $x = 0, 1, \ldots, n$, where $v_{xl}$ is $0$ or $1$, and $v_{01} = 1$ determinately. The weight matrix is defined by

$$t_{01} = t_{10} = \frac{1}{4} \left( \sum_{i=1}^{2} a_{ij} - 1 \right)$$

$$t_{ij} = 0, \quad t_{ij} = \frac{1}{4a_{ij}} \quad \forall i \neq j, \quad i, j = 1, \ldots, n.$$  

In this formulation, the first step in solving the MCP is to construct a graph $G_{01}$ by adding a vertex $0$ to $G$ (the complement graph of $G$). And $t_{ij}$ represents the weight of an edge between vertices $i, j$. Fig. 1 shows a simple graph $G$ with 6 vertices and 10 edges and the graph $G_{01}$. The maximum clique of $G$ contains vertices #2, #3, #5 and #6. The second step is to minimize the summation of the weights whose vertices belong to the same partition set, where $V_{1}$ is partitioned into $V^{+} = \{x/v_{x1} = 1, v_{x2} = 0\}$ corresponding to the vertices in the clique and $V^{-} = \{x/v_{x1} = 0, v_{x2} = 1\}$ corresponding to the vertices not included in the clique.

Galan et al. presented that maximum neural network using completely synchronous model cannot guarantee energy decrease and sometimes generates inaccurate results and oscillatory behaviors in the convergence process. However, they just illustrated the oscillatory behavior of parallel maximum neural network for MCP, the critical factor of weight setting was not revealed in detail. Moreover, due to lacking the influence of neurons’ delicate adjustment in their discrete model, the initial values become a very important factor to affect the solution quality and convergence speed. Thus in our algorithm we consider these inferior situations generally and propose effective methods to overcome these disadvantages.

3. The proposed algorithm for MCP

In the competitive network model for the MCP, as the added vertex #0 connects with all the other vertices, it directly influences the whole network competitive process. In the convergence process, $v_{01}$ is always set as 1, which drives those vertices connected with vertex #0 to get saturation to 1 tentatively and gradually. And $v_{02}$ drives other vertices to saturate to 0 gradually. Similarly, the competition between $v_{x1}$, $v_{x2}$, ..., and $v_{xm}(x = 1, \ldots, n)$ produces powerful searching ability for the neural network to solve some special combinatorial optimization problems such as N-Queen problem and channel assignment problem, especially when the competitive group $v_{m}(i = 1, \ldots, m)$ has adaptive numbers of vertices.

The weight definition of vertex #0 is a crucial factor determining the competitive intensity of neural networks. However, for a large number of MCP instances, the original weight definition function in Eq. (2) cannot guarantee that neural networks obtain powerful competitive activation to search optimal solutions regrettably. Hence either in completely synchronous model or asynchronous model, the competitive Hopfield networks using Eq. (2) are easy to get convergence to local minima. Here we give a simple example to show the flaw of weight definition Eq. (2) specifically in competitive neural networks for the MCP. Let us consider two simple graphs in Fig. 2. The weight matrices gotten...
from their relative graphs \(G_M\) are shown as follows:

\[
T_a = (T^a_0) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

\[
T_b = (T^b_0) = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

From the weight matrices, we can see that the weight values of vertex \#0 are all equal to zero. That means the vertex \#0 cannot influence the other vertices convergence process completely, whether the vertices are \(V^+\) or \(V^-\). Lacking the competitive effect produced by the vertex \#0, competitive Hopfield networks cannot form globally effective neurons’ competition. Ultimately, those models degenerate to be normal Hopfield networks [12], which is the worst situation specially using \(E(1)\) to set weight values. Furthermore, in some complex MCP graphs that the density of complement graph \(G_M\) is great, \(E(1)\) usually produces too large weight values to constrain all of the \(V^+\) vertices to saturate to 1, which is an inaccurate result obviously. So in those special MCP graphs, \(E(1)\) is not suitable for setting vertex \#0 weight values.

In order to overcome the flaw of weight definition, we introduce a flexible weight definition method which influences the competitive dynamics greatly, as shown as follows:

\[
t_0 = t_0 = \frac{1}{4} \left( \sum_{j=1}^{n} w_{ij} - p \right)
\]

\[
t_i = t_i = \frac{1}{4} \left( \sum_{j=1}^{n} w_{ij} \right) \quad \forall i \neq j, \quad i, j = 1, \ldots, n.
\]

where parameter \(p\) is relative with the density of graph \(G_M\). In our method, the special situation in which all weight values of vertex \#0 are equal to zero can be avoided by tuning parameter \(p\).

Similarly, when the density of graph \(G_M\) is large, the competitive power of vertex \#0 can be lessened by setting parameter \(p\) as a large value. The flexible weight definition method acting as a competition intensity tuning strategy overcomes the flaw of the original methods ultimately, and greatly increases the probability of converging to the optimal solutions.

Due to output vertices only having two states \((0/1)\) and lacking middle states in the convergence process in discrete Hopfield networks, the initial values become very important to influence the final solution quality. For MCP, there is a simple heuristic rule in general graphs: a vertex degree is larger, then it will have higher probability to be a node of maximum clique. In order to increase the solution quality and convergence speed, in our algorithm we introduce an initial value setting strategy as follows:

if Degree\(_x\) > AvgDegree\(_G\) then
  \(v_x(0) = 1, v_y(0) = 0, \) when random(0, 1) \(\geq 0.2\)
  \(v_x(0) = 0, v_y(0) = 1, \) when random(0, 1) \(< 0.2\)
else
  \(v_x(0) = 0, v_y(0) = 1, \) when random(0, 1) \(\geq 0.2\)
  \(v_x(0) = 1, v_y(0) = 0, \) when random(0, 1) \(< 0.2\)
end if

In our algorithm the initial value setting strategy drives those vertices with large degrees to obtain high probability to be set to 1 initially. As there is a random function in our strategy to decide the initial values, our algorithm obtains stochastic characteristic in a way. The mutual effects of heuristic rule and stochastic characteristic make the network neurons easily positioned in the domain of the optimal solution. So they greatly improve the solution quality of competitive neural networks.

In our algorithm we introduce an inhibitive competitive activation mechanism (ICAM) to accelerate or decelerate neurons competitive intensity, then drive the network energy increase or decrease temporarily. Therefore ICAM can help the network escape from local minima or accelerate the convergence speed. The new updating rule of neuron inputs is modified as follows:

\[
u_{ix}(k) = -2 \sum_{j=0}^{n} t_{xy}v_{yj}(k) - g(u_{ix}(k-1) - u_{yj}(k-1))
\]

\[
g(x) = \alpha(p)x e^{-p|x|}
\]

\[
\alpha(k+1) = \alpha(k)(1 - \delta)
\]

where \(u_{ix}(k)\) denotes the internal state of neuron \(x\) in group \(V^+\); \(g(x)\) is a nonlinear function forming the inhibitive competitive activation. \(u_{ix}(k) - u_{yj}(k)\) means that a neuron makes competition with its neighbor. For MCP there are only two groups of neurons \(V^+\) and \(V^-\), so the competition is formatted as \(u_{ix}(k) - u_{yj}(k)\) or \(u_{ix}(k) - u_{yj}(k)\). \(\alpha(k)\) is regarded as a gradient damping coefficient to guarantee the network convergence finally, and \(\delta\) is a parameter controlling the speed of gradient descent.

In Eq. (5), the nonlinear function \(g(x)\) contains a useful superiority that should not change the fixed points of original updating rule \(u_{ix}(k) = -2 \sum_{j=0}^{n} t_{xy}v_{yj}(k)\) but the stability of the fixed points may be changed. These fixed points can not be modified due to the primary demand \(g(0) = 0\). In ICAM, when the states of neurons satisfy the condition \(u_{ix}(k-1) - u_{yj}(k-1) = 0\), generally competition intensity between \(u_{ix}(k)\) and \(u_{yj}(k)\) is great. At that time, the intrinsic competitive dynamics are large enough to drive those neurons to differentiate their values. The network retains its original dynamic characteristic, and needs no external dynamic irritation. Similarly, large values of \(u_{ix}(k-1) - u_{yj}(k-1)\) means the neurons have gotten distinct competition results. A neuron wins with clear advantage in the same group certainly. In this situation, the network also does not need external dynamic irritation because the competitive states have already reached saturation gradually.

Nevertheless, the ICAM function \(g(x)\) in the ICHN satisfies the requirement of the network specialty. And the ICAM increases the competitive intensity of ICHN among neurons of the same group. From a large threshold point of \(|x|\), as shown in Fig. 3, the absolute value of \(x\) is larger, \(g(x)\) is closer to zero. When the function \(g(x)\) is set with some specific parameter values of \(p_1\), \(p_2\) and \(|x| > x_c(x_c\) is the threshold value), the values of \(g(x)\) are almost equal to zero. Therefore, the nonlinear function \(g(x)\) of the ICAM does not modify the useful effecting dynamics of the original neural network in this situation. At the intermediate section of \(|x|\), \(g(x)\) obtains large values to motivate or inhibit the original network to produce complex dynamics. The complex dynamics lead to tremendous competitions. For instance, when \(u_{ix}(k-1) - u_{yj}(k-1)\) is larger than zero slightly, which means neuron \(u_{ix}\) is the winner at the previous stage competition. At the present stage, the latter term of Eq. (4) is expected to break the network stability of previous period by decreasing \(u_{ix}\) value and declining its competitive vantage. Contrarily, when \(u_{ix}(k-1) - u_{yj}(k-1) < 0\), similar situation occurs. The intermediate competitive stage reveals that the original competitive neural network cannot find final solution obviously because there are no existing preponderant vantage in neuron groups. So the introduction of term \(g(x)\) enhances the competitive dynamic of the ICHN, and helps the ICHN to jump out from
local minima region and search more and wider regions in the solution domains.

As shown in Fig. 3, the \( g(x) \) influencing effects can be adjusted by double parameters \( p_1 \) and \( p_2 \). Normally \( g(x) \) set with parameters \( p_1 = 5 \) and \( p_2 = 2 \) is adopted in the ICHN for it producing appropriate intermediate competition.

In the ICHN, the ICAM provides a nonlinear self-feedback correlative with the competitive inputting variants \( v_{x_i} \) and \( v_{x_j} \) but not \( v_{x_i} \) and \( v_{x_j} \) like other self-feedback networks normally. As the ICHN is a kind of discrete competitive neural network, the output variants \( v_{x_i} \) and \( v_{x_j} \) containing only two states 0 and 1 are not suitable to generate inhibitive competitive dynamics. Moreover, the ICAM can bring about suitable competitive dynamics according to the competitive intensity among neurons. It effectively overcomes the disadvantage that the competitive dynamics is too simple to make the original network trap into local minima easily. So the ICHN embedded with the ICAM is able to produce self-adaptive inhibitive competitive dynamics to escape from local minima and reach those optimal states corresponding to the minima of energy.

In the latter period, there are no self-connections in the network and no interconnections between neurons that belong to the group. Consequently, the constraint \( K_{x_i x_j} \) illustrated in the paper of Gloria et al. is equal to zero for every two neurons \( x_i \) and \( x_j \) in the same group \( x \), so the ICHN satisfies the convergent constraints and gradually reaches a saturated situation either on synchronized model or completed asynchronous model.

4. Simulations

In order to assess the performance of the proposed algorithm, simulations were implemented in C on a normal PC (Pentium4 2.40 GHz). In this section, we present the simulation results of the ICHN with different parameters setting on the instances of MCP benchmark problems and practical applications. We verified the effectiveness of the initial values setting strategy in ICHN by comparing two groups of running results. We also compare the results with other DCHOM algorithms with different modifications to demonstrate the effect of the inhibitive competitive activation mechanism.

The ICHN has an important weight parameter, which is parameter \( p \), to set the weights of vertex \( \#0 \). The parameter \( p \) controls the strength of the competitive energy among vertices groups of MCP. Appropriate parameter \( p \) values produce sufficient competitive dynamics to compel our algorithm to search global solution domains. To reveal the parameter influence on the aspect of solution quality, we simulated the ICHN with different parameter \( p \) values on two MCP benchmark problems hamming6-2 and Johnson8-4-4. The comparisons of solutions are shown in Table 1. In Table 1, the column ‘Average Solution’ means the average clique size gotten by the algorithm ICHN and DCHOM, and the column ‘Best Solution’ represents the clique size of the best solution among the simulations. From Table 1 we can see that the performances of both algorithms are greatly affected by the parameter \( p \). In the simulations, when \( p=1 \) is defined as in the original method [9,10], both algorithms easily produce unacceptable solutions for the insufficient competitive dynamics. However

### Table 1

<table>
<thead>
<tr>
<th>Weight parameter setting</th>
<th>DCHOM Average solution</th>
<th>DCHOM Best solution</th>
<th>ICHN Average solution</th>
<th>ICHN Best solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>hamming6-2</td>
<td>16.5</td>
<td>20</td>
<td>18.8</td>
<td>23</td>
</tr>
<tr>
<td>( p=1 )</td>
<td>16.5</td>
<td>20</td>
<td>18.8</td>
<td>23</td>
</tr>
<tr>
<td>( p=4 )</td>
<td>26.8</td>
<td>32</td>
<td>30.8</td>
<td>32</td>
</tr>
<tr>
<td>( p=8 )</td>
<td>30.6</td>
<td>32</td>
<td>31.4</td>
<td>32</td>
</tr>
<tr>
<td>Johnson8-4-4</td>
<td>9.4</td>
<td>12</td>
<td>11.9</td>
<td>14</td>
</tr>
<tr>
<td>( p=1 )</td>
<td>9.4</td>
<td>12</td>
<td>11.9</td>
<td>14</td>
</tr>
<tr>
<td>( p=2 )</td>
<td>10.8</td>
<td>14</td>
<td>12.6</td>
<td>14</td>
</tr>
<tr>
<td>( p=3 )</td>
<td>10.4</td>
<td>14</td>
<td>12.8</td>
<td>14</td>
</tr>
</tbody>
</table>

### Table 2

The domain about the general setting regularity of parameter \( p \).

<table>
<thead>
<tr>
<th>Density</th>
<th>Effective domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0.1</td>
<td>1-4</td>
</tr>
<tr>
<td>0.1-0.3</td>
<td>1-5</td>
</tr>
<tr>
<td>0.3-0.6</td>
<td>1-8</td>
</tr>
<tr>
<td>0.6-0.8</td>
<td>1-4</td>
</tr>
<tr>
<td>0.8-1.0</td>
<td>1-2</td>
</tr>
</tbody>
</table>

### Table 3

Simulation results gotten by ICHN with and without the initial values setting strategy.

<table>
<thead>
<tr>
<th>Problems</th>
<th>Random setting Average ratio</th>
<th>Random setting Best solution</th>
<th>Our strategy Average ratio</th>
<th>Our strategy Best solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>hamming6-2</td>
<td>29.4</td>
<td>32</td>
<td>31.4</td>
<td>32</td>
</tr>
<tr>
<td>hamming6-4</td>
<td>3.1</td>
<td>4</td>
<td>3.8</td>
<td>4</td>
</tr>
<tr>
<td>Johnson8-4-4</td>
<td>11.2</td>
<td>14</td>
<td>12.8</td>
<td>14</td>
</tr>
<tr>
<td>Johnson8-2-4</td>
<td>3.4</td>
<td>4</td>
<td>3.6</td>
<td>4</td>
</tr>
<tr>
<td>MANN_a9</td>
<td>14.4</td>
<td>16</td>
<td>14.8</td>
<td>16</td>
</tr>
<tr>
<td>e-fat200-1</td>
<td>9.4</td>
<td>11</td>
<td>10.8</td>
<td>12</td>
</tr>
</tbody>
</table>
in a special parameter domain, the larger the parameter value is, the better the near-optimal solution would be found. With the same parameter value, the ICHN could obtain superior solution quality to DCHOM in general. Unfortunately, the parameter value that is too large will cause the competitive algorithm to form unfeasible solutions whose saturated states cannot constitute a clique. So in practical applications parameter value should be limited in an effective domain. In the domain, the competitive algorithm not only could hold rich dynamics, but also could effectively get convergence to feasible saturated states. In order to verify the domain, we have done large numbers of experiments to summarize the general setting regularity. Generally, the lower limit of parameter value is 1, and the upper limit is relative with the problem density. Table 2 shows the general setting domain of parameter $p$.

Similarly, the initial values setting strategy was also validated by a large number of tests on different instances. The performances of ICHN with or without the strategy were compared and shown in Table 3. In the simulation, some instances of the MCP benchmark problems distributed with different densities are selected specially to validate our strategy effect. From Table 3, we can get an observation that the initial values setting strategy is effective to improve the solution quality. Embedded with the initial values setting strategy, the ICHN has more opportunity to position its initial values in the domains of the optimal solutions. The strategy greatly increases the successful probability of finding the optimal solutions.

Moreover, we tested the ICHN in some instances of the $p$-random graphs in MCP. The results presented in Table 4 compare the performances of conventional heuristics-SA, the original competitive network MNN [20], DCHOM [22], TCNN [7], and the proposed algorithm ICHN, in terms of MCP. For each graph, every algorithm ran 40 times with different initial states and random coefficients. In Table 4, “Best” represents the maximum clique found during these 40 runs, while “Avg” is the average clique size. From the simulations, we observe that the proposed algorithm could obtain the maximum clique in each case mostly. Though our algorithm sometimes used a longer computation time due to the additional exciting complex dynamics yielded by the ICAM, it had a high success rate, which meant our algorithm would have a high probability to find optimal solutions by using suitable parameters and relative strategies. Therefore, we can conclude that the ICHN has a superior ability for solving the MCP.

<table>
<thead>
<tr>
<th>Name</th>
<th>N</th>
<th>Density</th>
<th>C(G)</th>
<th>TCNN</th>
<th>DCHOM</th>
<th>ICHN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BS</td>
<td>AS</td>
<td>BS</td>
<td>AS</td>
<td>BS</td>
<td>AS</td>
</tr>
<tr>
<td>MBM8</td>
<td>45</td>
<td>0.927</td>
<td>15</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.905</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>0.349</td>
<td>4</td>
<td>3.8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>0.969</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.077</td>
<td>12</td>
<td>12</td>
<td>12.4</td>
<td>12.4</td>
</tr>
<tr>
<td>c-fat200-2</td>
<td>200</td>
<td>0.163</td>
<td>24</td>
<td>21</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>c-fat200-3</td>
<td>200</td>
<td>0.426</td>
<td>58</td>
<td>56</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>c-fat500-1</td>
<td>500</td>
<td>0.036</td>
<td>14</td>
<td>12</td>
<td>12.3</td>
<td>12.3</td>
</tr>
<tr>
<td>c-fat500-2</td>
<td>500</td>
<td>0.073</td>
<td>26</td>
<td>23</td>
<td>23.6</td>
<td>23.6</td>
</tr>
<tr>
<td>c-fat500-5</td>
<td>500</td>
<td>0.183</td>
<td>64</td>
<td>64</td>
<td>64.8</td>
<td>64.8</td>
</tr>
<tr>
<td>Johnsson3-4-4</td>
<td>28</td>
<td>0.556</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Johnsson8-4-4</td>
<td>70</td>
<td>0.768</td>
<td>12</td>
<td>11.6</td>
<td>11.6</td>
<td>11.6</td>
</tr>
<tr>
<td>Johnsson16-2-4</td>
<td>120</td>
<td>0.765</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Johnsson32-2-4</td>
<td>496</td>
<td>0.879</td>
<td>16</td>
<td>15</td>
<td>14.1</td>
<td>14.1</td>
</tr>
<tr>
<td>P-hat300-1</td>
<td>300</td>
<td>0.244</td>
<td>8</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
</tr>
<tr>
<td>P-hat300-2</td>
<td>300</td>
<td>0.489</td>
<td>25</td>
<td>23</td>
<td>20.3</td>
<td>20.3</td>
</tr>
<tr>
<td>P-hat300-3</td>
<td>300</td>
<td>0.744</td>
<td>36</td>
<td>32</td>
<td>30.5</td>
<td>30.5</td>
</tr>
<tr>
<td>P-hat500-1</td>
<td>500</td>
<td>0.253</td>
<td>9</td>
<td>7.1</td>
<td>7.1</td>
<td>7.1</td>
</tr>
<tr>
<td>P-hat500-2</td>
<td>500</td>
<td>0.505</td>
<td>36</td>
<td>33</td>
<td>30.8</td>
<td>30.8</td>
</tr>
<tr>
<td>P-hat500-3</td>
<td>500</td>
<td>0.752</td>
<td>49</td>
<td>44</td>
<td>42.3</td>
<td>42.3</td>
</tr>
<tr>
<td>P-hat700-1</td>
<td>700</td>
<td>0.249</td>
<td>12</td>
<td>7</td>
<td>5.8</td>
<td>5.8</td>
</tr>
<tr>
<td>P-hat700-2</td>
<td>700</td>
<td>0.498</td>
<td>43</td>
<td>43</td>
<td>40.2</td>
<td>40.2</td>
</tr>
<tr>
<td>P-hat700-3</td>
<td>700</td>
<td>0.748</td>
<td>62</td>
<td>57</td>
<td>54.6</td>
<td>54.6</td>
</tr>
</tbody>
</table>

Please cite this article as: G. Yang, et al., An improved competitive Hopfield network with inhibitive competitive activation mechanism for maximum clique problem, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2012.11.055
In order to evaluate not just the relative, but also the absolute performance quality, we tested and compared these algorithms on the second DIMACS benchmark graphs. From Table 5, we can see that it is efficient to get the optimum solutions by our ICHN. In all of the benchmark graphs, ICHN always find better or equal best solution (column ‘BS’) than the other algorithms (DCHOM and TCNN). Compared with DCHOM and TCNN, the average solutions (column ‘AS’) found by ICHN have high quality obviously in the test. The parameters are tuned to make ICHN search larger domains in order to find the optimal solution, and easily escape from local minima with the effect of inhibitive competitive activation mechanism. The results of numerical experiments suggest that the ICHN is superior to TCNN and DCHOM in light of the best solution and average solution on the benchmark problems.

As we know, analysis of networks and in particular discovering communities within networks has been a focus of recent work in several fields, with applications ranging from citation and friendship networks to food webs and gene regulatory networks. An important problem in community detection is finding the cliques in networks. The maximum clique in community networks can be considered as a core group effecting global influence and reflecting the general properties. Hence, the maximum clique problem in community detection is an important significant practical application. We selected four real world datasets that are popular in the field of community detection to verify the performance and significance of our algorithm. The datasets are shown as follows.

Karate Club Data Set: social network of friendships between 34 members of a karate club at a US university in the 1970s.

Dolphin Data Set: a undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [15].

Jazz Data Set: This dataset consists of the list of links of the network of Jazz musicians [18]. There are 198 nodes in this network.

Political Books Data Set: Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the “customers who bought this book also bought these other books” feature on Amazon. The number of nodes in this data set is 105. This data set is compiled by V. Krebs and is available on Krebs’ web site (http://www.orgnet.com/) [16].

In the experiments of practical applications, the results got by TCNN, DCHOM and ICHN are shown in Table 6. Table 6 lists the properties of four data sets and the best solution (Column ‘BS’) and the average solution (Column ‘AS’) found by the three algorithms. For all data sets, ICHN exhibits a significant average solution and best solution of maximum clique problem compared to TCNN and DCHOM. The superior ability of ICHN reveals very well on the Jazz data set for its complex relationship in reality. For Jazz data set, the average and best solutions of ICHN are larger than the ones for TCNN and DCHOM greatly. In order to show the ICHN ability clearly, we utilized a famous community detecting software, called ‘pajek’ [2], to draw the problem structure and the best solution gotten by ICHN. Fig. 4 shows the original relation in jazz musicians, and Fig. 5 shows the maximum clique ICHN found, which illustrates the core group in jazz musicians social networks. In Fig. 5, the vertices forming an arc in the right side are the maximum clique nodes. The maximum clique has 30 vertices including vertices #32, #33, #35, #40, #44, #58, #60, #62, #63,
In terms of the simulations on community detection
\#109, \#110, \#122, \#123, \#131, \#132, \#135, \#154, \#168 and \#179.
\#64, \#65, \#66, \#98, \#99, \#100, \#101, \#105, \#106, \#107, \#108,
overcome the shortcomings of the DCHOM, and have greatly
dynamics mechanism is embedded. The proposed algorithm has
setting strategy, we have proposed an effective improved compe-

5. Conclusions

In this paper we have shown that the DCHOM for MCP has
vital flaw in the aspect of weight definition method. By
weight the definition formula and introducing an initial value
strategy, we have proposed an effective improved competitive
Hopfield network (ICHN), in which an inhibitive competitive
dynamics mechanism is embedded. The proposed algorithm has
overcome the shortcomings of the DCHOM, and have greatly
enhanced its searching capability to find the optimal solutions
in MCP. Simulation results show that the ICHN has superior ability on
solving the MCP.

Acknowledgment

This research was partially supported by the Grants from
the Natural Science Foundation of China (No. 61003205); the
Qianjiang Talent Project of Zhejiang Province (No. 2011R10087);
the Fundamental Research Funds for the Central Universities and
the Research Funds of Renmin University of China.

References

[1] S. Bandyopadhyay, M. Bhattacharyya, A neuro-\textit{ga} approach for the maximum
fuzzy clique problem, in: Proceedings of the 15th International Conference on
Advances in Neuro-Information Processing - Volume Part I, Springer-Verlag,


maximum clique problem, IEEE Transactions on Evolutionary Computation


for the maximum clique problem, Computers & Operations Research


problems, Biological Cybernetics 52 (1985) 141–152, http://dx.doi.org/
10.1007/BF00339943.

for combinatorial optimization under a unifying framework, Neural Networks

bipartite subgraph problem, Transactions on Neural Networks 3 (1992)
139–145.

bottlenose dolphin community of doubtful sound features a large proportion

[16] R. Narayanan, Y. Narahari, A Shapley value-based approach to discover
influential nodes in social networks, IEEE Transactions on Automation Science

[17] N. Funahiki, S. Nishikawa, Comparisons of energy-descent optimization algo-
rithms for maximum clique problem, IEICE Transactions on Fundamentals

[18] P. Glesser, L. Donon, Community Structure in Jazz, Advanced complex systems

[19] K.A. Smith, Neural networks for combinatorial optimization: a review of more

take-all neuron model forcing the state of the system in a solution domain,

[21] J. Wang, Y. Cai, J. Yin, Multi-start stochastic competitive Hopfield neural
network for frequency assignment problem in satellite communications,
Gang Yang received the B.S. degree and M.S. degree from Shandong University, Shandong, China in 2002 and 2005 respectively. He received the Ph.D. degree from the University of Toyama, Japan, in 2009. He is currently an Assistant Professor in the Information School, the Renmin University of China, China. His main research interests are neural network, pattern recognition and optimizations.

Nan Yang received the B.Sc. degree and the M.Sc. degree in computer science from Xian Jiaotong University, China and the Ph.D. degree in computer science from Beijing Institute of Technology, in 1982, 1988 and 1999, respectively. He is currently an Associate Professor of computer science at the information school, Renmin University of China. His major research interests include data mining, web communities and social networks.

Zheng Tang received the B.S. degree from Zhejiang University, Zhejiang, China, in 1982 and an M.S. degree and a D.E. degree from Tsinghua University, Beijing, China, in 1984 and 1988, respectively. From 1988 to 1989, he was an Instructor in the Institute of Microelectronics at Tsinghua University. From 1990 to 1999, he was an Associate Professor in the Department of Electrical and Electronic Engineering, Miyazaki University, Miyazaki, Japan. In 2000, he joined Toyama University, Toyama, Japan, where he is currently a Professor in the Department of Intellectual Information Systems. His current research interests include intellectual information technology, neural networks, and optimizations.

Junyan Yi received the B.S. degree and M.S. degree from Shandong University, Shandong, China, in 2001 and 2005 respectively, and received the Ph.D. degree from University of Toyama, Toyama, Japan, in 2009, all in computer science. Between 2007 and 2009, she worked as a Research Assistant at University of Toyama, Japan. She joined the College of Computer Science & Technology, Zhejiang University of Technology, China in 2009 as an Associate Professor. Her main research interests include intellectual information technology, neural network, pattern recognition, image processing and optimization problems.