A columnar competitive model for solving multi-traveling salesman problem

Hong Qu a, Zhang Yi a,*, HuaJin Tang b

a School of Computational Intelligence Laboratory, Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, People’s Republic of China
b STMicroelectronics, SST Corporate R&D, Singapore

Accepted 19 October 2005

Communicated by Prof. Ji-Huan He

Abstract

This paper studies an optimization problem: multi-traveling salesman problem (MTSP), which is an extension of the well known TSP. A columnar competitive model (CCM) of neural networks incorporates with a winner-take-all learning rule is employed to solve the MTSP. Stability conditions of CCM for MTSP is exploited by mathematical analysis. Parameters settings of the network for guaranteeing the network converges to valid solutions are discussed in detail. Simulations are carried out to illustrate the performance of the columnar competitive model compare to the heuristic algorithms: Tabu Search.

© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

There has been an increasing interest in applying the Hopfield neural networks to combinatorial optimization problems [1–6], since the original work of Hopfield and Tank [1]. Several methods have been proposed to ensure the network converges to valid states. Aiyer et al. [5] have theoretically explained the dynamics of network for traveling salesman problems by analyzing the eigenvalues of the connection matrix. Abe [6] has shown the theoretical relationship between network parameters and solution quality based on the stability conditions of feasible solutions. Chaotic neural network provides another promising approach to solve those problems due to its global search ability and remarkable improvement with less local minima, see in [7–10]. Peng et al. [11] suggested the local minimum escape (LME) algorithm, which improves the local minimum of CHN by combining the network disturbing technique with the Hopfield network’s local minimum searching property. Otherwise, many papers have discussed efficient mapping approaches [12–17]. Talavan and Yanez [12] presented a procedure for parameters settings based on the stability conditions of the network. Cooper et al. [14] developed the higher-order neural networks (HONN) to solve TSP and study the stability conditions of valid

This work was supported by National Science Foundation of China under Grant 60471055.

* Corresponding author.

E-mail address: zhangyi@uestc.edu.cn (Z. Yi).

0960-0779/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.

solutions. Brandt et al. [15] presented a modified Lyapunov function for mapping the TSP problem. All of those works are noteworthy for the solving of TSP.

MTSP is another optimization problem related with TSP [18–21]. However, the former is more complex and interesting than the latter. This problem deals with some real world problems where there is a need to account for more then one salesman. Many interesting applications of MTSP have been found. For example, suppose there is a city that is suffering from floods, the government wants to send out some investigators starting from the government building to investigate the disaster areas (all villages and towns in this city) and return to the starting point. The problem is how to find out the nearly equal shortest tour for each investigator. Clearly, it can be mapped to a MTSP. In addition, MTSP can be generalized to a wide variety of routing and scheduling problems, such as the School Bus Problem [25,26] and the Pickup and Delivery Problem [27,28]. In [29], it has shown that MTSP was an appropriate model for the problem of bank messenger scheduling, where a crew of messengers pick up deposits at branch banks and returns them to the central office for processing. Some other problems such as: railway transportation, pipeline laying, routing choosing, computer network topology designing, postman sending mail, torch relay transfer, etc, all can be mapped to MTSP model.

This paper presents a method to solve the MTSP based on an existing maximum neuron model proposed by Takefuji et al. [22], which also be named as columnar competitive network recently [23,24]. Stability conditions of CCM for MTSP will be exploited by mathematical analysis. Parameters settings of the network to guarantee the CCM converges to valid solutions will be discussed.

The description of MTSP and its mathematic model are presented in Section 2. The MTSP mapping and the CCM model is described in Section 3. The dynamical stability convergence of CCM for MTSP and a criterion of parameters settings for CCM are studied in Section 4. The simulations are given in Section 5 to illustrate the theoretical finding. Finally, conclusions are drawn in Section 6.

2. The MTSP problem

The description of Multi-Travelling Salesman Problem involving with four problems are as follows:

Problem 1: Given a group of \( n \) cities and the distance between any two cities. Suppose there are \( m \) visitors starting from a city to visit the group of cities. Finding the nearly equal shortest tour for each visitor such that each city be visited only once and each visitor returns to the starting city at last.

Problem 2: Given a group of \( n \) cities and the distance between any two cities. Suppose there are \( m \) visitors starting from a city to visit the group of cities. Finding the nearly equal shortest tour for each visitor such that each city be visited only once and each visitor end to \( m \) different cities at last.

Problem 3: Given a group of \( n \) cities and the distance between any two cities. Suppose there are \( m \) visitors starting from \( m \) different cities to visit the group of cities. Finding the nearly equal shortest tour for each visitor such that each city be visited only once and each visitor end to the seem city at last.

Problem 4: Given a group of \( n \) cities and the distance between any two cities. Suppose there are \( m \) visitors starting from \( m \) cities to visit the group of cities. Finding the nearly equal shortest tour for each visitor such that each city be visited only once and each visitor end to \( m \) different cities at last.

Problem 1 is to find \( m \) closed tours which start from the same city, visit each city once, each tour contain the nearly equal number of city, and return to the starting city with a short total length. Problem 2 is to find \( m \) routes beginning from the same city and ending with \( m \) different cities. Problem 3 and Problem 4 is to find \( m \) routes with the \( m \) different starting cities. In this paper, all of our study are based on problem 1. The remains will be continue investigated later. The valid tour for those four problems are illuminated in Fig. 1.

We can describe problem 1 with a weighed, non-directed graph \( G(V,E) \). The city is represented by the vertex on the graph, denoted as \( v_c \). The road between each city is represented by the edge, denoted as \( e_{ij} \). \( w(e_{ij}) \) is the value weighed on edge \( e_{ij} \), which represents the real distance of the two cities. \( m \) loops should appear in graph \( G(V,E) \) to guarantee \( m \) closed tours: \( C_1, C_2, \ldots, C_m \), and \( v_0 \in C_i, i = 1, \ldots, m \). Then, MTSP can be described as a mathematic model as follows:

\[
\begin{align*}
\text{min} & \quad \sum_{j=1}^{m} \sum_{e_{ij} \in (C_j)} w(e_{ij}) + D \\
\text{subject to} & \quad v_0 \in C_1(V)(i = 1 \ldots m) \\
& \quad \bigcup_{i=1}^{m} C_i(V) = G(V)
\end{align*}
\]
where

\[ D = a \left\{ \sum_{j=1}^{m} n(C_j) - \bar{n} \right\}^2. \] (3)

In the objective function (1), \( \sum_{j=1}^{m} \sum_{e_i \in C_j} \omega(e_i) \) is the total length of \( m \) loops, \( D \) represent the degree of difference among respective loops. In Eq. (3), \( n(C_j) \) is the numbers of the vertex in loop \( C_j \), \( \bar{n} \) is the average vertex number of \( m \) loops, \( a \) is a scaling factor. It is clear that the difference between loops would be decreased when the value of \( D \) changes smaller.

3. MTSP mapping and CCM model

In this section, a neural network model is presented to solve MTSP. It is clear, they have \( m \) sub-tours in a valid solution of the \( n \)-city and \( m \)-loop MTSP problems, and each sub-tour starts with the starting city. To map this problem into the network, an appropriate representation scheme of neurons should be required. It is well known, a \( n \times n \) square array has been used in \( n \)-city TSP problem by Hopfield, but it is not enough for MTSP problem.

Here, we add \( m - 1 \) cites to the network, and we called those cites as the virtual cities. The virtual cities have the same connections and weights as the starting city. The location informations of a virtual city’s are specified by the output state of \( m - 1 \) neurons in the representation scheme. We define those neurons as the group neurons. Hence, we can distinguish \( m \) loops easily by those virtual cities using a \( (n + m - 1) \times (n + m - 1) \) square matrix. For example, a 5-city and 2-loop problem, an original feasible solution and the new one after adding the virtual cities as shown in Fig. 2. It needs \( (5 + 2 - 1) \times (5 + 2 - 1) = 36 \) neurons to represent the neurons’s output states.

![Fig. 2](image_url)
The above output states of neurons represent a multi-tour with 2 tours. \( A \) is the starting city to be visited, \( C \) is the second and \( D \) is the third of the first tour. \( F \) is the virtual city which represent city \( A \). The second tour starting from \( F(A) \), visited \( B, E \) in sequence. To sum up, the first tour is \( A - C - D - F(A) \) and the second is \( F(A) - E - D - A(F) \). \( E, F \) are the group neuron.

In the seminal work of Hopfield [1], it has been shown that an energy function of the form

\[
E = -\frac{1}{2}v^\top Wv - (\hat{\beta})^\top v
\]

(4)

can be minimized by the continuous-time neural network with the parameters: \( W = (W_{ij})_{neu} \) is the connection matrix, \( n \) is the number of neurons, \( v = (v_{x,i})_{neu} \) represents the output state of the neuron \( (x,i) \), and \( \hat{\beta} \) is the vector form of bias.

To solve MTSP efficiently, we employ a new neural network that has a similar structure as Hopfield network, but it obeys a different updating rule: the columnar competitive model (CCM), which incorporates winner-takes-all (WTA) in column-wise. The WTA mechanism can be described as [23]: Given a set of \( n \) neurons, the input to each neuron is calculated and the neuron with the maximum input value is declared the winner. The winner’s output is set to “1” while the remaining neurons will have their values set to “0”. The intrinsic competitive nature is in favor of the convergence of CCM to feasible solutions, since it can reduce the number of penalty terms compare to Hopfield’s formulation.

The energy function of CCM for MTSP can be written as

\[
E(v) = \frac{A}{2} \sum_{x} \sum_{i} \left( v_{x,i} \sum_{j \neq i} v_{x,j} \right) + \frac{B}{2} \sum_{i} (S_{i} - \bar{S})^2 + \frac{1}{2} \sum_{x} \sum_{y \neq x} \sum_{i} d_{xy} v_{x,i} (v_{y,i+1} + v_{y,i-1})
\]

(5)

where \( A > 0, B > 0 \) are scaling parameters, \( d_{xy} \) is the distance between city \( x \) and \( y \), \( \bar{S} = \frac{z}{2} \) is the average number of cities for \( m \) tours, \( S_i \) is the number of cities that have been visited by team \( i \). Let \( m_i (i = 1, \ldots, m-1) \) be the index of the virtual city in the whole city’s sequence which composed by \( m \) team’s tour concatenated end by end, and set \( m_0 = 0, m_m = n + m - 1 \), \( n \) is the total number of cites. For example in Fig. 1, \( m_0 = 0, m_1 = 4, m_2 = 6 \). Then

\[
\sum_{i=1}^{m} (S_i - \bar{S})^2 = \sum_{i=1}^{m} B \left( S_i^2 - 2\bar{S} S_i + \bar{S}^2 \right) = \sum_{i=1}^{m} \left\{ B \left( \sum_{x} \sum_{i=m_{i-1}}^{m_i-1} v_{x,i} \right) - \frac{Bn^2}{2m} \right\}
\]

(6)

Thus, comparing (5) with (4), the connection matrix and input basis are computed as

\[
\begin{cases}
W_{x,y} &= -\left\{ A\delta_{xy}(1 - \delta_{ij}) + d_{xy}(\delta_{i,j+1} + \delta_{i,j-1}) + B\theta_j \right\} \\
\hat{\beta} &= \frac{n}{m}
\end{cases}
\]

(7)

where

\[
\theta_j = \begin{cases} 1, & \text{if } m_{t-1} \leq j < m_t \\ 0, & \text{otherwise} \end{cases}
\]
then the input to neuron \((x,i)\) is calculated as

\[
\text{Net}_{x,i} = \sum_y \sum_j (W_{x,y} v_{y,j}) + i^b = -\sum_j d_{ij}(v_{y,j-1} + v_{y,j+1}) - A \sum_j v_{x,j} - B \sum_y \sum_{j=m_y-1}^{m_y-1} v_{y,j} + \frac{n}{m}
\]

\[
= -\sum_j d_{ij}(v_{y,j-1} + v_{y,j+1}) - A \sum_j v_{x,j} - B S_p + \frac{n}{m} \tag{8}
\]

The columnar competitive model based on winner-take-all (WTA) leaning rule, the neurons compete with others in each column, and the winner is with the largest input. The updating rule of outputs is given by

\[
v_{x,i}^{j+1} = \begin{cases} 
1, & \text{if } \text{Net}_{x,i} = \max\{\text{Net}_{x,i}^1, \text{Net}_{x,i}^2, \ldots, \text{Net}_{x,i}^{m_y-1}\} \\
0, & \text{otherwise}
\end{cases} \tag{9}
\]

For CCM, \(v_{x,i}\) is updated by the above WTA rule. The whole algorithm is summarized as follows:

1. Initialize the network, with each neuron having a small value \(v_{x,i}\).
2. Calculate \(m(l=0, \ldots, m)\).
3. Select a column (e.g., the first column), compute the input \(\text{Net}_{x,i}\) of each neuron in this column.
4. Apply WTA and update the outputs of the neurons in that column using Eq. (9).
5. Go to the next column, repeat step 2 until the last column in the network is done. This constitutes one epoch.
6. Go to step 2 until the network converges.

In the next section, by investigating the dynamical stability of the network, we present a theoretical results which give an analytic method to set the scaling parameters \(A\) and \(B\) optimally. As a consequence, the convergence to valid solutions can be assured.

4. Valid solutions and convergence analysis of CCM for MTSP

It is well known that the stability of original Hopfield network when it be applied to TSP is guaranteed by the Lyapunov energy function. However, the dynamics of CCM for MTSP is different from the original Hopfield network. In this section, we will investigate the stability of CCM.

The WTA updating rule ensure only one “1” per column, but it is not the case for the rows. The responsibility falls on scaling parameters \(A\) and \(B\) of the penalty-terms. The following analysis shows that the value of the \(A\) and \(B\) play a predominant role in ensuring the convergence of valid states.

In this section, our efforts are devoted to how to determine the critical value of \(A\) and \(B\) that can ensure the convergence of the valid solutions for the MSTP based on the stability analysis.

4.1. Parameters settings for the CCM when it be applied to MTSP

Consider the \(p\)-column of neuron outputs states matrix, suppose row \(b\) is an all-zero row and row \(a\) is not all-zero. According to Eq. (8), the input to neuron \((a,p)\) and \((b,p)\) is computed as

\[
\text{Net}_{a,p} = -A \sum_{j \neq a} v_{a,j} - \sum_y d_{ay}(v_{y,p-1} + v_{y,p+1}) - B S'_p + \frac{n}{m} \tag{10}
\]

\[
\text{Net}_{b,p} = -A \sum_{j \neq b} v_{b,j} - \sum_y d_{by}(v_{y,p-1} + v_{y,p+1}) - B S'_p + \frac{n}{m} \tag{11}
\]

where \(0 < S'_p < n\). Suppose row \(a\) contains \(l\) “1” \((1 \leq l \leq n + m - 1)\), then

\[
\text{Net}_{b,p} = -\sum_y d_{by}(v_{y,p-1} + v_{y,p+1}) - B S'_p + \frac{n}{m} > -\sum_y d_{by}(v_{y,p-1} + v_{y,p+1}) - B n^2 + \frac{n}{m} \tag{12}
\]

and

\[
\text{Net}_{a,p} = -A(l - 1) - \sum_y d_{ay}(v_{y,p-1} + v_{y,p+1}) - B S'_p + \frac{n}{m} < -A(l - 1) - \sum_y d_{ay}(v_{y,p-1} + v_{y,p+1}) + \frac{n}{m} \tag{13}
\]
**Theorem 1.** When $A - Bn^2 > 2d_{\text{max}} - d_{\text{min}}$, where $d_{\text{max}}$ and $d_{\text{min}}$ is the maximum and the minimum distance between any two cities, respectively, $n$ is the total number of cities, the neuron’s outputs under CCM always escape from the invalid states.

**Proof.** It is clear that only one neuron’s output in per column be set to ‘1’ under WTA undating rule. Assume the network reaches the following state after some updating:

\[
\begin{pmatrix}
0 & 0 & 0 & 0 \\
1 & v_{s,p} & 0 & \ldots & 0 \\
0 & v_{t,p} & 1 & \ldots & 0 \\
0 & v_{b,p} & 0 & \ldots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\end{pmatrix}
\]

According to Eqs. (12) and (13), the input to each neuron in the $p$th column is calculated as

\[
\begin{align*}
\text{Net}_{b,p} &> -(d_{bt} + d_{bh}) - Bn^2 + \frac{n}{m} \\
\text{Net}_{s,p} &< -(A + d_{at}) + \frac{n}{m} \\
\text{Net}_{t,p} &< -(A + d_{dt}) + \frac{n}{m} \\
\text{Net}_{v,p} &< -(A + d_{in} + d_{in}) + \frac{n}{m} \\
\end{align*}
\]

To ensure the neuron’s outputs reach the valid solution in the next updating, the neuron $v_{b,p}$ should be the winner, since all the other neurons in row $b$ is zero. In the other word, the input of the neuron $v_{b,p}$ should be the maximum one of all the inputs in $p$th column, $\text{Net}_{b,p} > \text{Net}_{a,p}, a \neq b$. Therefore it is ensured by

\[
\begin{align*}
&d_{bt} + d_{bh} + Bn^2 - \frac{n}{m} < A + d_{at} - \frac{n}{m} \\
&d_{bt} + d_{bh} + Bn^2 - \frac{n}{m} < A + d_{in} + d_{in} - \frac{n}{m} \\
\end{align*}
\]

It is well known that $d_{at} < d_{in} + d_{in}$ then it follows that

\[
A - Bn^2 > d_{bt} + d_{bh} - d_{at}
\]

which can be ensured by $A - Bn^2 > 2d_{\text{max}} - d_{\text{min}}$. The neuron outputs always escape from the invalid states. \(\square\)

### 4.2. Dynamical stability analysis

Suppose $E'$ and $E'^{+1}$ are two states before and after WTA update respectively. Consider $p$th column, let neuron $(a,p)$ be the only active neuron before updating, and neuron $(b,p)$ be the winning neuron, and $a$ row contains $l$ “1”, then

\[
\begin{align*}
\psi'_{s,p} &= \begin{cases} 
1, & x = a \\
0, & x \neq a
\end{cases} \quad \text{and} \quad \psi'^{+1}_{s,p} = \begin{cases} 
1, & x = b \\
0, & x \neq b
\end{cases}
\end{align*}
\]

The energy function (5) can be broken into three terms $E_p$, $E_q$ and $E_o$, that is, $E = E_p + E_q + E_o$. $E_p$ stands for the energy of the columns $p-1$, $p$ and $p+1$ of the rows $a$ and $b$. $E_q$ stands for the energy of the groups. $E_o$ stands for the energy of the rest columns and rows. Then $E_p$ is computed by

\[
E_p = A \left( \sum_i v_{a,i} \sum_j v_{b,j} \right) + \sum_x \sum_y d_{xy} \psi_{s,p}(v_{x,p+1} + v_{y,p+1})
\]
$E_o$ is computed as

$$E_o = \frac{A}{2} \sum_{i \neq j} \sum_{j} \left( v_{x,i} v_{y,j} \right) + \frac{1}{2} \sum_{i} \sum_{j \neq k} \sum_{k} d_{x,y} v_{x,i} (v_{y,j} + v_{y,k}) + \frac{1}{2} \sum_{i} \sum_{j \neq l} \sum_{l} d_{x,y} v_{x,i} v_{x,l}$$

$$+ \frac{1}{2} \sum_{i} \sum_{j} d_{x,y} v_{x,i} v_{x,j+2}$$

(17)

And $E_q$ is calculated as

$$E_q = \frac{B}{2} \sum_{r=1}^{m} S_r^2 - \frac{n(n + m - 1)}{m} + \frac{B n^2}{2m}$$

(18)

Now, we investigate how to change of $E$ under WTA learning rule of CCM.

**Theorem 2.** Let $A - B n^2 > 2d_{\max} - d_{\min}$, where $d_{\max}$ and $d_{\min}$ is the maximum and the minimum distance respectively, $n$ is the total city number, then CCM is always convergent to valid states.

**Proof.** Three cases are studied in the follow analysis.

*Case 1: (a, p) and (b, p) are both not group neuron*

In this case $0 < a, b < m$, it can be seen that only $E_p$ will be affected by the state of column $p$. According to Eq. (16), $E_p$ and $E_p^{+1}$ is computed by

$$E_p^+ = -\frac{A}{2}l(l-1) - \sum_{y} d_{ax}(v_{y,p-1} + v_{y,p+1})$$

(19)

$$E_p^{+1} = -\frac{A}{2}(l-1)(l-2) - \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1})$$

(20)

Then

$$E_p^{+1} - E_p^+ = -A(l-1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ax}(v_{y,p-1} + v_{y,p+1})$$

(21)

At the same time, the input to neuron $(a, p)$ and $(b, p)$ before updating are computed as follows:

$$\text{Net}^i_{a,p} = -A(l-1) - \sum_{y} d_{ax}(v_{y,p-1} + v_{y,p+1}) - BS_p' + \frac{n}{m}$$

(22)

$$\text{Net}^i_{b,p} = -\sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - BS_p' + \frac{n}{m}$$

(23)

According to the discussion of the above subsection, when

$$A - B n^2 > 2d_{\max} - d_{\min}$$

is guaranteed, then

$$\text{Net}_{b,p} > \text{Net}_{a,p}(a \neq b)$$

can be ensured. This implies that

$$-A(l-1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ax}(v_{y,p-1} + v_{y,p+1}) < 0$$

(24)

So we can clearly deduce that

$$E_p^{+1} - E_p^+ < 0$$

Thus, $E_p^{+1} - E_p^+ < 0$.

*Case 2: (a, p) is a group neuron while (b, p) is not*

In this case $n < a < n + m$ while $0 < b < n$, it can be concluded that not only $E_p$ but also $E_q$ would be changed before and after the WTA updating rule in $p$th column. $(a, p)$ is group neuron, it can be active before updating while nonactive after updating. This implies that two connected groups which be distinguished by neuron $(a, p)$ before updating merge into one group after updating. Suppose $s_{x}^{q}$ and $s_{x}^{b}$ represent the city’s number of those two connected groups before


Suppose \( a \) from \( A \). Applying (29) to (27), we get that
\[
E_q^{t+1} = \frac{B}{2} (s_q^{t+1})^2 + \frac{B}{2} \sum_{i=1}^{m} S_i^2 - \frac{n(n + m - 1)}{m} + \frac{Bn^2}{2m}
\]
(25)
\[
E_{q'}^{t+1} = \frac{B}{2} (s_{q'}^{t+1})^2 + \frac{B}{2} \sum_{i=1}^{m} S_i^2 - \frac{n(n + m - 1)}{m} + \frac{Bn^2}{2m}
\]
(26)

Now, we consider the change of \( E \):
\[
E_{p'}^{t+1} - E_{p}^{t+1} - E_{q'}^{t+1} = -A(l - 1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ay}(v_{y,p-1} + v_{y,p+1}) + B\delta_{q}^{t} s_{q}^{t}
\]
(27)

According to the discussion of above subsection (12) and (13), when \( A - Bn^2 > 2d_{\text{max}} - d_{\text{min}} \), \( \text{Net}_{b,p} > \text{Net}_{a,p} \) are ensured, it implies that
\[-A(l - 1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ay}(v_{y,p-1} + v_{y,p+1}) < -Bn^2
\]
(28)

that is
\[-A(l - 1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ay}(v_{y,p-1} + v_{y,p+1}) < -Bn^2
\]
(29)

Applying (29) to (27), we get that
\[E_{p'}^{t+1} + E_{q'}^{t+1} - E_{p}^{t+1} - E_{q}^{t+1} < -Bn^2 + B\delta_{q}^{t} s_{q}^{t} < -Bn\delta_{q}^{t} s_{q}^{t} + B\delta_{q}^{t} s_{q}^{t} < -B\delta_{q}^{t} (n - s_{q}^{t}) < 0
\]
(30)

It also implies that
\[E_{p'}^{t+1} - E_{p}^{t+1} < 0.
\]

\( \text{Case 3:} (b,p) \) is group neuron while \((a,p)\) is not

In this case \( n < b < n + m \) while \( 0 < a < n \). Like in case 2, both \( E_p \) and \( E_q \) are changed with the change of neuron’s state in column \( p \). \((b,p)\) is group neuron, it can be active after updating while nonactive before updating. This implies that one group which neuron \((a,p)\) be contained in before updating would be divided into two groups after updating. Suppose \( s_{q}^{t+1} \) and \( s_{q}^{t} \) represent the city number of those two connected groups after updating respectively, and \( s_{q}^{t} \) stand for the city number of the original group before updating. Then, \( s_{q}^{t+1} + s_{q}^{t} = s_{q}^{t+1} \).

As in case 2, \( E_q' \) and \( E_{q'}^{t+1} \) is computed as following, respectively.
\[
E_{q}^{t+1} = \frac{B}{2} (s_{q}^{t+1})^2 + \frac{B}{2} \sum_{i=1}^{m} S_i^2 - \frac{n(n + m - 1)}{m} + \frac{Bn^2}{2m}
\]
(31)
\[
E_{q'}^{t+1} = \frac{B}{2} (s_{q'}^{t+1})^2 + \frac{B}{2} \sum_{i=1}^{m} S_i^2 - \frac{n(n + m - 1)}{m} + \frac{Bn^2}{2m}
\]
(32)

Now we consider the change of \( E \):
\[E_{p}^{t+1} + E_{q'}^{t+1} - E_{p}^{t+1} - E_{q}^{t+1} = -A(l - 1) + \sum_{y} d_{by}(v_{y,p-1} + v_{y,p+1}) - \sum_{y} d_{ay}(v_{y,p-1} + v_{y,p+1}) - B\delta_{q}^{t} s_{q}^{t+1}
\]
(33)

from \( A - Bn^2 > 2d_{\text{max}} - d_{\text{min}} \), (24) and (33), it follows that
\[E_{p}^{t+1} + E_{q}^{t+1} - E_{p}^{t+1} - E_{q}^{t+1} < -B\delta_{q}^{t} s_{q}^{t+1} < 0
\]
(34)

since \( B > 0 \), \( s_{q}^{t+1} > 1 \), \( s_{q}^{t} > 1 \). This also implies that \( E_{p}^{t+1} - E_{p}^{t+1} < 0 \).

Observed from the discussion of above three cases, we can say that the energy decreases always during the process of WTA updating if \( A - Bn^2 > 2d_{\text{max}} - d_{\text{min}} \). In a other words, the CCM model always convergent under the WTA updating rule.

\[
\square
\]

5. Simulation results

An application of MTSP is studied in this section. Suppose there is a city that is suffering from floods. The government wants to send out three investigators starting from the government building to investigate the disaster areas (all
the town in this city) and return to the starting position. The city's road graph are given in Fig. 3. The question is to find the nearly equal shortest tour for each investigator.

There are 53 cities in this network, denoted as $A$–$R$ and 1–35, $O$ is the site of government. The arcs represent the road among cities. The length of each road are given by the number on the graph.

It is simple to say that the distance triangularity of the given network did not hold. In order to ensure the distance triangularity of network, the distances of shortest path between each two vertexes are used in our simulations. If $v_i$ and $v_j$ are not adjacent, then we can add a virtual arc between $v_i$ and $v_j$, denoted as $w_{ij}$, and associate the length of the shortest path from $v_i$ to $v_j$ with the value of $w_{ij}$. After this process, if the distance triangularity of network can hold, then this example can be solved by the method introduced in above sections. After getting the solutions, the original path should be resumed. The method used to resumed the result path is shown in the following example.

For example (as shown in Fig. 4), if $b$ and $c$ were not adjacent in the result path ($\cdots a b c \cdots$), and the shortest path between $b$ and $c$ is $b d c$, then the resumed path is ($\cdots a b d c \cdots$).

In this section we will validate the criteria of Theorems 1 and 2 by computer simulations. Here we adopted the algorithm introduced in Section 3. In this work, 500 simulations have been performed with different scale of parameters $A$ and $B$.

In this simulations, $\text{MAX} = 132.8$, $\text{MIN} = 1.8$ and $n = 53$. Three case are studied in the simulation based on the value of $A$ and $B$: (1) $A - Bn^2 < 2\text{MAX} - \text{MIN}$, set $A = 300$ AND $B = 0.05$; (2) $A - Bn^2 = 2\text{MAX} - \text{MIN}$, set $A = 400$ AND $B = 0.05$; (3) $A - Bn^2 > 2\text{MAX} - \text{MIN}$, chose $A = 400$ AND $B = 0.01$. The energy's changing of system is studied in detail. The energy decrease in all instance of simulation when $A - Bn^2 > 2\text{MAX} - \text{MIN}$ are guaranteed.
We show some good results of energy's changing as Fig. 5. At the same time, the simulation results of case (3) are given in Table 1.

Among the several significant developments of general local search methods [30] such as simulated annealing and tabu search, tabu search appears to be the most powerful methods [31,32]. To evaluate the efficacy of the model we presented, a heuristic algorithms: a tabu search algorithm with 4-exchange reference structure are considered to provide a compare with our work.

Table 2 shows the simulation results for CCM model and the 4-exchange Tabu Search algorithms. The simulation result shows that the CCM work more effective than Tabu search for solving MTSP in CPU's, while the results obtained by tabu search are better. It clearly shows that the balance's effect of optimal solution attained by CCM is greatly improved.

Table 1
The resulting paths

<table>
<thead>
<tr>
<th>Tour path</th>
<th>Length (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP 1</td>
<td>O → 2 → 5 → 6 → 7 → L → 19 → J → 11 → E → 9 → F → 10 → 8 → 4 → D → 3 → C → O</td>
</tr>
<tr>
<td>GROUP 2</td>
<td>O → 1 → A → B → 34 → 35 → 32 → 33 → 31 → 30 → R → 29 → Q → 28 → 27 → 24 → 26 → P → O</td>
</tr>
<tr>
<td>GROUP 3</td>
<td>O → M → 25 → 20 → 21 → K → 18 → I → 13 → G → 12 → H → 14 → 15 → 16 → 17 → 22 → 23 → N → O</td>
</tr>
</tbody>
</table>

Table 2
The computational result of CCM for MTSP with tabu search method

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>CCM</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>53</td>
<td>3</td>
<td>197.4</td>
<td>175.4</td>
</tr>
<tr>
<td>4</td>
<td>130.9</td>
<td>121.3</td>
<td>511.7</td>
</tr>
<tr>
<td>5</td>
<td>111.9</td>
<td>93.6</td>
<td>499.2</td>
</tr>
</tbody>
</table>

Max: the maximal cost of the m closed tour; Min: the least cost of the m closed tour; Dis: the total cost of the resumed solution; BL: the degree of balance; CPU: the CPU time.
6. Conclusions

In this paper, we presented a mathematic model to describe MTSP. Based on the description, a columnar competitive model was employed to solve MTSP. The stability condition of CCM for MTSP was exploited. According to the theoretical analysis, the critical values of the network parameters were found. The simulation result showed that WTA updating rule makes CCM an efficient and fast computational model for solving MTSP.

References