Available online at www.sciencedirect.com
SCIENCE DIRECT®

Neurocomputing 62 (2004) 267–292

NEUROCOMPUTING

www.elsevier.com/locate/neucom

An expanding self-organizing neural network for the traveling salesman problem

Kwong-Sak Leung[a], Hui-Dong Jin[b,*,1], Zong-Ben Xu[c]

[a]Department of Computer Science and Engineering, the Chinese University of Hong Kong, Shatin, N.T., Hong Kong

[b]Division of Mathematics and Information Sciences, CSIRO, Australia

[c]Faculty of Science, Institute for Information and System Sciences, Xi'an Jiaotong University, Xi'an, 710049, PR China

Received 11 February 2003; received in revised form 24 February 2004; accepted 26 February 2004

Abstract

The self-organizing map (SOM) has been successfully employed to handle the Euclidean traveling salesman problem (TSP). By incorporating its neighborhood preserving property and the convex-hull property of the TSP, we introduce a new SOM-like neural network, called the expanding SOM (ESOM). In each learning iteration, the ESOM draws the excited neurons close to the input city, and in the meantime pushes them towards the convex-hull of cities cooperatively. The ESOM may acquire the neighborhood preserving property and the convex-hull property of the TSP, and hence it can yield near-optimal solutions. Its feasibility is analyzed theoretically and empirically. A series of experiments are conducted on both synthetic and benchmark TSPs, whose sizes range from 50 to 2400 cities. Experimental results demonstrate the superiority of the ESOM over several typical SOMs such as the SOM developed by Budinich, the convex elastic net, and the KNIES algorithms. Though its solution accuracy is not yet comparable to some other sophisticated heuristics, the ESOM is one of the most accurate neural networks for the TSP in the literature.

Keywords: Self-organizing map; Traveling salesman problem; Neural networks; Elastic net; Convex-hull

[*] Corresponding author. CS&IT Building 108, North Road, ANU Campus, GPO Box 664, Canberra, ACT 2601, Australia. Tel.: +61-2-62167258; fax: +61-2-62167111.

E-mail addresses: ksleung@cse.cuhk.edu.hk (K.-S. Leung), hdjin@ieee.org (H.-D. Jin), zbxu@mail.xjtu.edu.cn (Z.-B. Xu).

[1] The work was submitted when he was with the Department of Computer Science and Engineering, the Chinese University of Hong Kong.

0925-2312/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.02.006

1. Introduction

As a classical combinatorial optimization problem, the traveling salesman problem (TSP) can be defined as follows: Given a graph $G=(V,A)$, where V is a set of n vertices and A is a set of arcs between vertices with each arc associated with a nonnegative distance, the TSP is to determine a minimum distance of closed circuits passing through each vertex once and only once [25]. In its simplest form, the Euclidean TSP (ETSP) is the TSP in which all vertices are on a two-dimensional plane with the distance between two vertices calculated according to the Euclidean metric. There are several real-life applications of the TSP, such as VLSI routing [11], hole punching [6,30], and wallpaper cutting [25]. The TSP is NP-hard, i.e., any algorithm for finding optimal tours must have a worst-case running time that grows faster than any polynomial if P $\neq$ NP [19,29]. Recent theoretical research shows a randomized algorithm can be designed to compute a $(1+1/c)$-approximation to the optimal tour in $O(n(\log n)^{O(c)})$ time for every fixed $c>1$ [5]. But its running time still depends exponentially on c. Because of these, researchers have been either attempting to develop optimization algorithms that work well on "real-world," rather than worst-case instances [6,15], or looking for heuristics that merely find *near*-optimal tours quickly. These heuristics include simulated annealing (SA) [20], genetic algorithms [19], tabu search [21], artificial ant systems [11], automata networks [35], and neural networks [4,10,17,22]. These diverse approaches have demonstrated various degrees of strength and success [19]. This paper will present an improved neural network that can generate near-optimal TSP solutions with quadratic computation complexity.

There are mainly two types of neural networks for the TSP: the Hopfield-type neural networks [17] and the Kohonen-type self-organizing map (SOM-like) neural networks [22]. The Hopfield-type neural networks get tours by searching for the equilibrium states of one dynamic system corresponding to the TSP under consideration. They have been successfully applied to handle small scale TSPs [1,10], but could not generate promising solutions for large scale TSPs [9,33,35]. In comparison, the SOM-like neural networks solve the TSP through unsupervised learning [22]. A SOM inspects the input cities for regularities and patterns, and then adjusts its neurons to fit the input cooperatively. The SOM finally brings about the localized response to the input, and thus reflects the neighborhood of the cities. This neighborhood preserving map then results in a tour. From each city, the resultant tour tries to visit its nearest city. The shortest subtours can intuitively lead to a near-optimal tour.

Due to their low computation complexity and promising performance, the SOM-like neural networks have attracted a large amount of research to explore and enhance the capability of handling the TSP [3,9,13,14]. Though the solution accuracy of the SOM-like neural networks are still not comparable to some state-of-the-art heuristics such as ant colony system [1], Lin–Kernighan [19,26], memetic algorithms [24], tabu search [19], and guided local search [34]. These heuristics usually have much higher computation complexity than a SOM [7,18,22]. Furthermore, improvements of neural networks for the TSP are being made [2,4,7,10,18]. To enhance a SOM, there are following three strategies.

Introducing the variable network structure: Instead of the static structure, the output neurons may be dynamically deleted/inserted. Typical examples are the SOM with a dynamic structure [3], the Kohonen network incorporating explicit statistics (KNIES) [4], and FLEXMAP with a growing structure [14].

Amending the competition criterion: Burke and Damany [9] have developed the guilty net by introducing a bias term into the competition criterion for inhibiting the too-often-winning neurons. In the work of Favata and Walker [13], the competition criterion is based on the inner product, which is slightly different from the Euclidean distance when all weights are normalized. In the SOMs developed by Budinich [7,8], the visiting order of a city is determined by not only its nearest neuron but also its neighbors.

Enhancing the learning rule: For example, the KNIES [4], besides drawing excited neurons closer to the input city as a classic SOM does, disperses other neurons so as to keep their mean unchanged. In the elastic net [12], an elastic force is embodied in its learning rule, which is often used to enhance SOMs [2].

Though the SOM-like neural networks may generate promising solutions by establishing a neighborhood preserving map, they have little considered some properties of optimal tours directly. In this paper we present a SOM-like neural network, called the expanding SOM (ESOM), for the TSP. The idea is to train the neural network tactfully so as to preserve the perfect topological relations, as well as to satisfy a necessary condition of any optimal tour—the convex-hull property of the TSP. Experimental results show that, with the similar computation complexity as a traditional SOM, the proposed ESOM can find very near-optimal tours for both synthetic and benchmark TSPs. Comparison results show that it outperforms such typical SOMs as Budinich's SOM [7], the convex elastic net (CEN) [2], and the KNIES [4]. We believe that our present results are among the most accurate ones in the neural networks literature.

After a brief review of the SOM implementation scheme in the next section, we present the details of our ESOM in Section 3. Its feasibility and complexity are analyzed in Section 4. In Section 5, we provide a comprehensive series of experimental results to demonstrate the effectiveness and efficiency of the ESOM in comparison with five algorithms. We conclude the paper in the last section.

2. The SOM-like neural networks for the TSP

A SOM-like neural network is an unsupervised competitive learning scheme. It simply inspects input data for regularities and patterns, and then organizes itself in such a way as to form a topology preserving ordered description. This description may lead to a solution of the problem under consideration. The SOM-like neural networks have successfully been applied to different applications, say, data visualization [32], clustering and data compression [23,4]. Through viewing a Hamiltonian circuit of the TSP as a particularly organized, topologically ordered path, SOMs can also be used to handle the TSP [2,7]. We give a brief description and outline several typical techniques of the SOMs for the TSP below.

When applied to the TSP, a SOM consists of a ring of output neurons and several input neurons. The output neurons, denoted by $1, 2, \ldots, M$, serve to characterize the feature map. The input neurons receive the coordinate values of an input city. At time t, their states are the vector $\vec{x}(t) = [x_1(t), x_2(t), \ldots, x_p(t)]^{\mathrm{T}} \in R^p$. Here p is the number of input neurons, and $p = 2$ for TSPs in a 2-D plane. The input neurons are fully connected to every output neuron. The synaptic weights between the jth output neuron and the input ones comprise a vector $\vec{w}_j(t) = [w_{j1}(t), \ldots, w_{jp}(t)]^{\mathrm{T}} \in R^p$ $(1 \leqslant j \leqslant M)$. Therefore, these output neurons are in two topological spaces. One lies on the ring of the output neurons to reflect a linear order of visiting the cities. The other one lies in the p-dimensional space where the coordinate of each output neuron is indicated by its synaptic weight vector. The basic idea of the SOM is to construct a topology preserving map from the high-dimensional synaptic weight space onto the one-dimensional ring space and then form a tour. The visiting order of each city is determined by the order of its associated output neuron on the ring.

The topology preserving map may preserve the neighborhood relationships between the output neurons. That is, the output neurons that are close on the ring space should be closely located on the synaptic weight space. This is accomplished by performing unsupervised learning on the output neurons from the data values of cities circularly. The coordinates of cities are fed into the input neurons iteratively in a random fashion. Then the output neurons compete with one another according to *a discriminant function*, say, the Euclidean metric. After that, *the winning neuron* $m(t)$, as well as its neighbors determined by a *neighborhood function* $h_{j,m(t)}$, are excited. All the excited neurons update their synaptic weights according to a learning rule. The learning process continues until all cities are fed into the SOM for certain times.

When the winner neuron and its neighbors updated simultaneously, this update aims at enhancing their responses to the subsequent application of any similar input vector $\vec{x}(t)$. In other words, this mechanism enables the excited neurons to decrease their individual values of the discriminant function in relation to the input vector $\vec{x}(t)$. This can be reached using the following rule [22]:

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \eta(t) h_{j,m(t)} (\vec{x}(t) - \vec{w}_j(t)), \tag{1}$$

where *the learning parameter* $\eta(t)$ takes values from $[0, 1]$ and decreases normally to zero as time t elapses [7,8]. Intuitively, the learning rule means that the weight vector $\vec{w}_j(t)$ moves towards the input $\vec{x}(t)$ with the movement quantity proportional to the distance between $\vec{w}_j(t)$ and $\vec{x}(t)$. One of its variants may be found in [13], where the weight vectors are normalized after learning in order to facilitate the algorithm. Figs. 1(a) and 1(b) show their conceptual views in a learning iteration, respectively. In the KNIES algorithms [4], the unexcited neurons are also uniformly moved away after each learning iteration so as to keep the mean of all the output neurons unchanged.

Another renowned learning rule, adopted originally by the elastic net [12], is to update the synaptic weights $\vec{w}_j(t)$ according to

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \eta(t)[\vec{x}(t) - \vec{w}_j(t)] + \eta_2(t)[\vec{w}_{j-1}(t) + \vec{w}_{j+1}(t) - 2\vec{w}_j(t)], \tag{2}$$

where $\eta_2(t)$ is another learning parameter. It is helpful to note that the last term in the above rule reflects the elastic force constraint that compels the length of the resulting

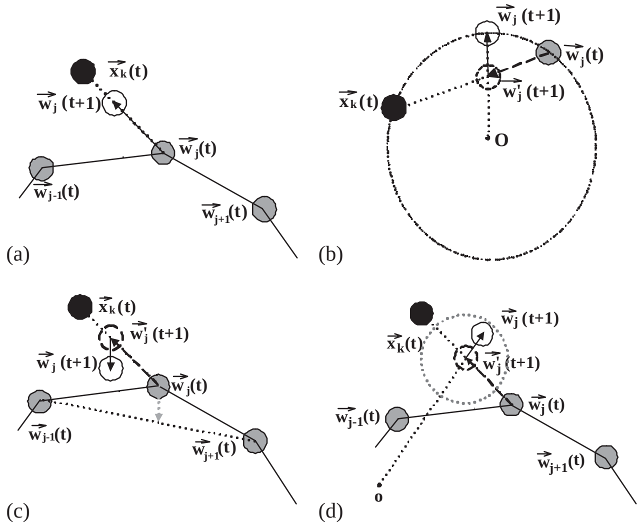


Fig. 1. Schematic view of four different learning rules. A black disc indicates an input city; a gray disc indicates a neuron; a solid line indicates the neuron ring; a circle indicates the new position of a neuron; a dashed circle indicates a neuron's interim position; a dashed arrow indicates a movement direction; a gray arrow indicates the elastic force which draws a neuron to the middle point of two neighboring neurons; and 'o' in (d) indicates the origin, i.e., the center of all the cities: (a) The learning rule in the conventional SOM; (b) the one for the SOM of Favata and Walker; (c) the one for the elastic net; (d) the learning rule for the ESOM.

ring to be as short as possible, as indicated in the CEN [2,28]. A schematic view for this learning mechanism is shown in Fig. 1(c).

In the next section, we will propose a new learning rule from the viewpoint of both acquiring the neighborhood preserving property and satisfying a necessary condition of any optimal tour. This rule is illustrated in Fig. 1(d). It will not only distinguish our proposed SOM from the existing SOM-like neural networks, but also lead to a substantial improvement on the performance.

3. The expanding SOM neural network

Upon repeated presentations of the input data, the synaptic weight vectors in a SOM tend to follow their distribution. The SOM forms a neighborhood preserving map of the input data in the sense that the output neurons adjacent on the ring will tend to have similar weight vectors [16]. Through this map, it can generate a good tour. This tour, from each city, tries to visit its nearest city. If all the subtours are locally shortest, the whole tour is definitely optimal. However, such a sufficient condition for optimal tours is not always achievable for any TSP. So there is no guarantee for the SOM to generate an optimal tour. Besides the neighborhood preserving property, we may as well consider some properties that any optimal tour must have. Through

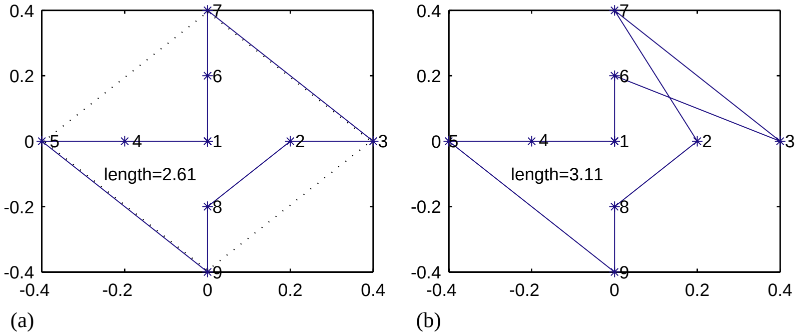


Fig. 2. The convex-hull property of a simple TSP: (a) The convex-hull and an optimal tour; (b) a tour without the convex-hull property.

learning, we may create a SOM to satisfy these necessary conditions too. Then, the new learning mechanism tries to satisfy the sufficient and the necessary conditions of optimal tours simultaneously, and finally the new SOM can substantially increase solution accuracy. This is the basic motivation of our research. In the following, we first show that *the convex-hull property* is one of such necessary conditions, and then give an improved SOM to satisfy the convex-hull and the neighborhood preserving properties simultaneously.

3.1. The convex-hull property

The convex-hull property says that, for any optimal tour of a 2-D TSP, the cities located on the convex-hull formed by the given cities must be visited in the same order as they appear on the convex-hull [2,25]. Let us take the 9-city TSP in Fig. 2(a) as an example, where the dotted lines constitute its convex-hull. The property says that, since cities 3, 9, 5 and 7 appear on the convex-hull, they should appear in clockwise or anti-clockwise order in any optimal tour, as in the one indicated by solid lines in Fig. 2(a). On the contrary, the tour in Fig. 2(b) conflicts with the property, so it is not an optimal one, as also indicated by its intersection. The property implies that, to get better tours, we should make better use of the cities on the convex-hull.

We give a rigorous proof of the convex-hull property. The basic idea is to deduce that a tour must intersect itself if it conflicts with the convex-hull property, as illustrated in Fig. 2(b).

Theorem 1. *The convex-hull property holds for any optimal tour.*

The proof of this theorem is presented in Appendix A. The convex-hull property has been used in [2,15]. In particular, Al-Mulhem and Al-Maghrabi incorporated this property into the elastic net and proposed their CEN, leading to a very encouraging application to the TSP [2]. The CEN needs the convex-hull for initialization, whose computation takes at least $O(n \ln n)$ additional time.

3.2. The expanding SOM: ESOM

The basic idea of our ESOM is to incorporate the convex-hull property implicitly in its learning rule with little additional computation. It acquires the convex-hull property gradually as the topological neighborhood among cities are being inspected and preserved. Therefore, our ESOM is trying to satisfy the sufficient and the necessary conditions of optimal tours. This is realized in the following way: In a single update iteration, besides drawing the excited neuron towards the input city, we expand it towards the convex-hull. The former adaptation, together with the cooperative adaptation of neighbor neurons, will gradually discover the topological neighborhood relationship of the input data, as widely believed in the neural networks community. In a similar way, the latter adaptation, also together with the cooperative adaptation, can approach the convex-hull property of the TSP. Accordingly, better solutions can be expected. Of course, in order to achieve this, the total adaptation still has to enhance the response of the excited neuron to the subsequent application of a similar input. On the other hand, to learn the convex-hull property appropriately, it is also necessary for the latter adaptation to reflect the convex-hull property in a single learning iteration. It is worthwhile noting that approaching the convex-hull may be approximated by pushing away from the center of the cities. So we can realize the second adaptation by expanding the excited neuron from the center, as illustrated in Fig. 1(d).

We propose the following specific algorithm to implement the above idea. We call it the expanding SOM (ESOM) because it yields the convex-hull property through expanding neurons outwards from the center of all the cities.

The ESOM

1. Map all the city coordinates $[x'_{k1}, x'_{k2}]^{\mathrm{T}}$ $(k=1,\ldots,n)$ into a circle C_R centered at the origin with radius $(R \leqslant 1)$. The new coordinates are denoted by $[x_{k1}, x_{k2}]^{\mathrm{T}}$ below. The center of the cities is mapped to the origin (the center of the circle).
2. Randomly set the weight vectors $\vec{w}_j(0)$ $(j=1,\ldots,M)$ within C_R, and set $t=0$.
3. Select a city at random, say $\vec{x}_k(t)=[x_{k1}(t), x_{k2}(t)]^{\mathrm{T}}$, and feed it to the input neurons.
4. Find the winning neuron, say $m(t)$, nearest to $\vec{x}_k(t)$ according to the Euclidean metric.
5. Train neuron $m(t)$ and its neighbors within *the effective width* $\sigma(t)$ by using the following formula:

$$\vec{w}_j(t+1) = c_j(t)\vec{w}'_j(t+1) \triangleq c_j(t)[\vec{w}_j(t) + \alpha_j(t)(\vec{x}_k(t) - \vec{w}_j(t))], \tag{3}$$

   where $1 \leqslant j \leqslant M$, *the learning rate* $\alpha_j(t)(\in[0,1])$ is specified by

$$\alpha_j(t) = \eta(t)h_{j,m(t)} \triangleq \eta(t)\max\left\{0, 1 - \frac{d_{j,m(t)}}{\sigma(t)+1}\right\} \tag{4}$$

   (*the learning parameter* $\eta(t)\in[0,1]$), and $c_j(t)$ $(\geqslant 1)$ is *the expanding coefficient* specified later.
6. Update the parameters $\sigma(t)$ and $\eta(t)$ according to a predefined decreasing scheme and, if the learning iterations are not terminated, go to Step 3 with $t := t+1$.

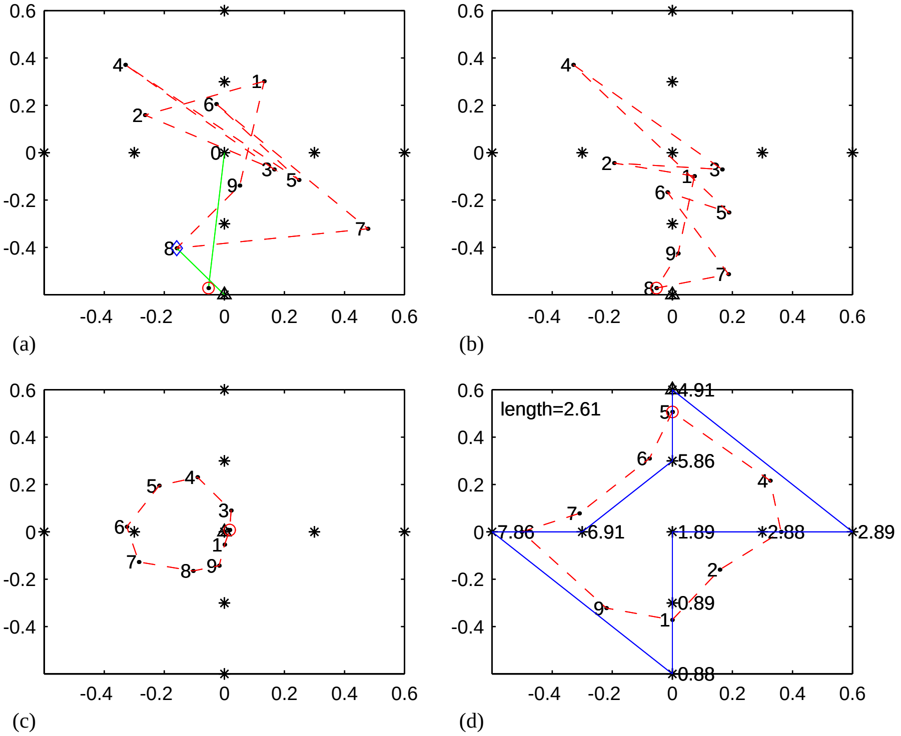


Fig. 3. A demonstration of the ESOM on a TSP: (a) the initialization output neuron ring and the winning neuron adaptation; (b) the output neuron ring after learning from the first input city; (c) the output neuron ring after learning from the first loop of inputs of 9 cities; (d) the final output neuron ring and the final tour.

7. Calculate the activity value of each city $\vec{x}_k$ according to the formula:

$$a(\vec{x}_k) = m_k - \tfrac{3}{26}\{\|\vec{x}_k - \vec{w}_{m_k}\| + \tfrac{2}{3}[\|\vec{x}_k - \vec{w}_{m_k+1}\| - \|\vec{x}_k - \vec{w}_{m_k-1}\|] + \tfrac{1}{2}[\|\vec{x}_k - \vec{w}_{m_k+2}\| - \|\vec{x}_k - \vec{w}_{m_k-2}\|]\}, \qquad (5)$$

where m_k is the winning neuron corresponding to $\vec{x}_k$.

8. Order the cities by their activity values, and then form a tour for the TSP.

Fig. 3 demonstrates an execution procedure of the ESOM on solving the TSP shown in Fig. 2. Here all cities have been subject to a linear transformation as in Step 1. One city overlaps with the origin ‘o’, i.e., the center of the 9 cities. In Fig. 3, the asterisks indicate the cities and the one marked by a triangle is the current input city. A dot represents an output neuron and the one circled is a winner. The integer beside each dot labels its index in the output neuron ring, shown by dashed lines. Fig. 3(a) shows the initialized output neuron ring. It also shows the winning neuron 8 learns from the input city and moves from the position indicated by the diamond to the one circled.

The movement is composed of two parts as directed by two gray lines in Fig. 3(a). The winning neuron 8 is drawn closer to the input city and then expanded towards the convex-hull. A clearer illustration may be found in Fig. 1(d). Figs. 3(b) and (c) show the output neuron rings after learning from, respectively, the first input city and the first input loop of the 9 cities. Fig. 3(d) gives the final output neuron ring and its generated tour. The number beside each asterisk is the activity value according to Eq. (5). On average, the ESOM may find the optimal tours three times within 10 runs.

Let us explain the ESOM in detail. Step 1 aims to preprocess the city data in such a way as to make all the cities lie inside the circle C_R. It should be noted that this linear transformation does not alter the characteristics of the tour space of the TSP. This preprocessing step is necessary for the following expanding procedure during learning.

The learning rule defined by Eq. (3) is the key point of the proposed ESOM. Differing from the previous learning rules, it has an additional multiplication factor—the expanding coefficient $c_j(t)$. If $c_j(t) \equiv 1$, we can derive a conventional SOM from the ESOM. The expanding coefficient is not less than 1.0 and its function is to push the excited neuron away from the origin. This expanding effect, together with the cooperative adaptation in the ESOM, is used to learn the convex-hull property of the TSP implicitly. There are many possible implementations of the expanding coefficient to reflect the convex-hull property. Motivated by Favata and Walker [13], whose SOM handles TSPs located on the unit sphere and the excited neurons have to be normalized after learning, we give an implementation as follows:

$$c_j(t) = [1 - 2\alpha_j(t)(1 - \alpha_j(t))\beta_j(t)]^{-1/2}, \tag{6}$$

where $\beta_j(t)$ is specified by

$$\begin{aligned}\beta_j(t) &= 1 - \vec{x}_k(t)^{\mathrm{T}}\vec{w}_j(t) - \sqrt{(1 - \|\vec{x}_k(t)\|^2)(1 - \|\vec{w}_j(t)\|^2)} \\ &= 1 - \left[x_{k1}(t), x_{k2}(t), \sqrt{1 - \|\vec{x}_k(t)\|^2}\right]\left[w_{j1}(t), w_{j2}(t), \sqrt{1 - \|\vec{w}_j(t)\|^2}\right]^{\mathrm{T}}. \end{aligned}\tag{7}$$

The convex-hull property is reflected in $\beta_j(t)$. The inner product between two unit vectors $[x_{k1}(t), x_{k2}(t), \sqrt{1 - \|\vec{x}_k(t)\|^2}]^{\mathrm{T}}$ and $[w_{j1}(t), w_{j2}(t), \sqrt{1 - \|\vec{w}_j(t)\|^2}]^{\mathrm{T}}$ contains information of the distance between $\vec{x}_k(t)$ and $\vec{w}_j(t)$, as well as their distances from the origin $\|\vec{x}_k(t)\|$ and $\|\vec{w}_j(t)\|$. Given $\vec{w}_j(t)$ and $\alpha_j(t)$, $\beta_j(t)$ roughly increases with $\|\vec{x}_k(t) - \vec{w}_j(t)$ or $\|\vec{x}_k(t)\|$ or $\|\vec{w}_j(t)\|$, and reaches its minimum 0 as $\vec{w}_j(t) = \vec{x}_k(t)$. Roughly speaking, the cities on the convex-hull are farther away from the center of the cities (i.e., the origin) than the others. The expanding coefficient $c_j(t)$ increases with $\beta_j(t)$, and then it is larger for the cities on the convex-hull than for the others. During learning, these cities have more influence on expanding, and then the ESOM more likely leads to tours with the convex-hull property. On the other hands, the expanding coefficient $c_j(t)$ is close to its minimum 1 when the weight vector $\vec{w}_j(t)$ is very near to the input city $\vec{x}_k(t)$, and it will not greatly affect the regularities obtained. We will discuss these further in Theorem 3 in Section 4. In Eq. (6), the term $2\alpha_j(t)(1 - \alpha_j(t))$ reflects the cooperation among adjacent neurons. The constant 1 restricts the value of

$c_j(t)$ to be around 1.0. The exponent $-\frac{1}{2}$ may ensure the excited neuron is not expanded out of the circle C_R. A bit larger value still works, but it facilitates our analysis. In addition, these terms together reach a good tradeoff between expanding the excited neuron towards the convex-hull and drawing it closer to the input city. The expanding effort does not influence the drawing effort too much in order to detect and preserve the neighborhood relationship. We will justify these statements in Theorem 2 in Section 4.

Steps 7 and 8 in the ESOM constitute a realization of the solution mapping procedure. It aims to yield a good tour from the ESOM after learning. In order to distinguish the local orders of the cities exciting the same neuron m_k, each city is mapped to, instead of the integer m_k, a real number according to Eq. (5). This mapping is designed not only for exploiting the information of the winning neuron but also for benefiting from that of its nearest neighbors. If there are more than one city exciting the same neuron m_k, then the city nearer to the preceding neurons on the ring and farther from the subsequent neurons should be visited earlier. This may form a shorter tour. In addition, these distances have different importance. The distance to the winning neuron, $\|\vec{x}_k - \vec{w}_{m_k}\|$, is most important, and its coefficient is set to 1 in Eq. (5). Similarly, $[\|\vec{x}_k - \vec{w}_{m_k+1}\| - \|\vec{x}_k - \vec{w}_{m_k-1}\|]$ is more important than $[\|\vec{x}_k - \vec{w}_{m_k+2}\| - \|\vec{x}_k - \vec{w}_{m_k-2}\|]$, and their coefficients, also based on some preliminary experiments, are set to $\frac{2}{3}$ and $\frac{1}{2}$ respectively. The coefficient $\frac{3}{26}$ in Eq. (5) is set to restrict the second term in the right side within $[-0.5, 0.5]$, so that this term only influences the local orders of the cities exciting the same winner m_k.

4. Feasibility and complexity analysis

In this section, we analyze the feasibility and the complexity of the proposed ESOM.

4.1. Feasibility

As explained before, the motivation of the developed ESOM is to approximate an optimal tour by trying to satisfy both a sufficient condition (i.e., the neighborhood preserving property) and a necessary condition (i.e., the convex-hull property). Thus, to show the feasibility of the ESOM rigorously, it should be verified that the ESOM does converge to a tour that possesses the neighborhood preserving property and the convex-hull property. But, unfortunately, we are frustrated in the present stage by our inability and very limited understanding on the behavior of the SOM itself. In fact, except the one-dimensional case, the convergence of the SOM-like neural networks in higher dimensional cases has been one of the long-standing open problems in the neural networks research [27,31]. Thus, instead of a rigorous convergence analysis, we conduct a one-step trend analysis to show the feasibility of the ESOM. Considering the linear transformation, we suppose that all cities are located within the circle C_R and their center overlaps with the origin in this section.

We first show how the ESOM can learn the neighborhood preserving property as the conventional SOM does.

Theorem 2. *Let C_R be the closed circle with radius R centered at the origin, $\vec{x}_k \in C_R$ $(k = 1, \ldots, n)$ be the input city and $\{\vec{w}_j(t)\}_{j=1}^{M}$ be the weight vectors of the ESOM at time t. Then, for any $t \geqslant 0$,*

(i) *for $j \in \{1, 2, \ldots, M\}$,*

$$\vec{w}_j(t) \in C_R; \tag{8}$$

(ii) *for any j,*

$$\|\vec{w}_j(t+1) - \vec{x}_k(t)\| \leqslant \rho_j(t)\|\vec{w}_j(t) - \vec{x}_k(t)\|, \tag{9}$$

where

$$\rho_j(t) = (1 - \alpha_j(t))(1 + \varsigma \cdot \alpha_j(t)), \tag{10}$$

$\alpha_j(t)$ is the learning rate as specified in Eq. (4), *and ς is the constant defined by*

$$\varsigma = \frac{1}{2\sqrt{1-R^2} - 1} - 1. \tag{11}$$

Furthermore, if $R < \frac{\sqrt{7}}{4}$, $\|\vec{w}_j(t) - \vec{x}_k(t)\| \neq 0$, and $\alpha_j(t) \neq 0$, there holds the following strict inequality for any j and k:

$$\|\vec{w}_j(t+1) - \vec{x}_k(t)\| < (1 - \alpha_j^2(t))\|\vec{w}_j(t) - \vec{x}_k(t)\|; \tag{12}$$

(iii) *As in Eq.* (3), *$\vec{w}_j'(t+1) = \vec{w}_j(t) + \alpha_j(t)(\vec{x}_k(t) - \vec{w}_j(t))$. We have*

$$1 - \varsigma\alpha_j(t) \leqslant \frac{\|\vec{w}_j(t+1) - \vec{x}_k(t)\|}{\|\vec{w}_j'(t+1) - \vec{x}_k(t)\|} \leqslant 1 + \varsigma\alpha_j(t). \tag{13}$$

The proof of this theorem can be found in Appendix B. Theorem 2(i) says that the weight vectors defined by the proposed ESOM are always bounded within the closed circle C_R. Theorem 2(ii) says that as long as the parameter R is appropriately chosen, say, $R < \frac{\sqrt{7}}{4}$, the expanding learning rule does enhance the responses of the excited neurons to the subsequent application of a similar input city. It is because that the updated weight vector $\vec{w}_j(t+1)$ is closer to the input city $\vec{x}_k(t)$ than the old one $\vec{w}_j(t)$. Furthermore, the formula $\vec{w}_j'(t+1) = \vec{w}_j(t) + \alpha_j(t)(\vec{x}_k(t) - \vec{w}_j(t))$ gives exactly the conventional SOM learning rule. Thus, Theorem 2(iii) shows that, as the learning rate $\alpha_j(t)$ tends to zero, the asymptotic behavior of the ESOM and the SOM will be equivalent. This implies that the ESOM will be convergent if the SOM does. It is important to note that the SOM has presented satisfactory convergence empirically. So, this theorem supports the feasibility of the ESOM.

We now turn to show how the ESOM is likely to capture the convex-hull property of the TSP. Due to the lack of thorough understanding on the asymptotic behavior of the SOM, we only present a one-step trend analysis.

Theorem 3. *Given $\alpha_j(t)$ and $\vec{w}_j(t)$, the expanding coefficient $c_j(t)$ specified by Eqs.* (6) *and* (7) *increases with $\|\vec{x}_k(t)\|$ (resp. $\|\vec{w}_j(t)\|$) if $\|\vec{x}_k(t)\| \geqslant \|\vec{w}_j(t)\|$ (resp. $\|\vec{w}_j(t)\| \geqslant$*

$\|\vec{x}_k(t)\|$) when $\vec{x}_k(t)$ and $\vec{w}_j(t)$ are within a one-fourth circle; it increases with $\|\vec{x}_k(t)\|$ when $\vec{x}_k(t)$ and $\vec{w}_j(t)$ are not within a one-fourth circle; and it attains the minimum 1 if $\vec{w}_j(t)=\vec{x}_k(t)$.

The proof is given in Appendix A. Note that the expanding coefficient $c_j(t)$ ($\geqslant 1$) characterizes the degree of the push of the weight vector $\vec{w}_j(t)$ outwards from the center of the cites or towards the convex-hull of the TSP, as shown in Fig. 1(d). Intuitively, a city $\vec{x}_k$ located on the convex-hull should be farther away from the origin than the others because the origin is also the center of the cities. Specifically, it is true when it is compared with the cities on the same radial from the origin. Thus, $\|\vec{x}_k\|$ is likely to be larger. In fact, for any point $\vec{x}$ within the convex-hull of the TSP with vertices $\{\vec{x}_c\}_{c\in C}$(C contains the cities on the convex-hull.), we have $\vec{x}=\sum_{c\in C}\{\lambda_c \times \vec{x}_c\}$ with $\sum_{c\in C}\lambda_c=1$ and $0\leqslant\lambda_c\leqslant 1$ ($\forall c\in C$). Then, $\|\vec{x}\|\leqslant\sum_{c\in C}\{\lambda_c\|\vec{x}_c\|\}\leqslant\left\{\sum_{c\in C}\lambda_c\right\}\max_{c\in C}\|\vec{x}_c\|=\max_{c\in C}\|\vec{x}_c\|$. That is, $\|\vec{x}_k\|$ reaches its maximum on some city located on the convex-hull. On the other hand, the neuron ring normally lies within the convex-hull as shown in Fig. 3, thus $\|\vec{w}_j(t)\|\leqslant\|\vec{x}_k(t)\|$ usually holds for the cities on the convex-hull. So, these cities normally push the excited neurons towards the convex-hull more strongly during learning, and this more likely results in tours with the convex-hull property. Furthermore, when the weight vector $\vec{w}_j(t)$ is very near to the input city $\vec{x}_k(t)$, the expanding coefficient is close to its minimum 1, and it cannot much affect the regularities obtained. This one-step trend analysis shows that the ESOM learning rule may extract and preserve the convex-hull property. It is also empirically confirmed in Section 5.

4.2. Complexity

The following theorem shows that the proposed ESOM is of the same computation complexity as the other SOM-like neural networks for the TSP [7,13].

Theorem 4. *The computation complexity of the ESOM is quadratic.*

Proof. Assume the total number of cities is n. According to the procedure of the ESOM, Step 1, 2, 7 or 8 inspects every neuron or city at most twice, so the computation time required for each step is $O(n)$. The number of iterations between Steps 3 and 6 is $O(n)$. (Normally, every city would be input to the SOM-like network with a fixed number of times, say, 100. This strategy is commonly used [7,13].) Within each iteration, Step 4 needs $O(n)$ computation to search the winner, and Step 5 involves at most all the n neurons for each input city. Thus, totally, the computation complexity of the ESOM is $O(n^2)$. □

5. Implementation and experimental results

To demonstrate the effectiveness and efficiency of the proposed ESOM, a series of experiments have been conducted on both synthetic and benchmark TSPs. Some experimental results are presented in comparison with five typical algorithms in this section.

5.1. Implementation

We have implemented the ESOM with C++ and executed it on a Sun Ultra 5/270 workstation. The source code and all the TSPs are accessible at www.cse.cuhk.edu.hk/∼hdjin/som. In order to make a fair comparison, we have set most parameters identical with Budinich's SOM in [7]. The parameters setting is as follows.

- *Radius R*: It is set to 0.6, which is less than $\frac{\sqrt{7}}{4}(0.66)$ as suggested by Theorem 2.
- *The number of output neurons M*: It is fixed and set equal to the number of cities n.
- *Training loops*: It is set to 100, i.e., there are $100 \times n$ iterations.
- *The learning parameter η*: It is initially set to 0.8 and is decreased linearly for each learning loop until it reaches zero at the last loop.
- *The width $\sigma(t)$*: It is set to $6.2 + 0.037n$ at the start and is decreased linearly to 1 in the first 65% of the iterations. For the subsequent 35% of the iterations, $\sigma(t)$ remains at 1.

To evaluate the performance of the ESOM, we have compared our algorithm with five typical algorithms. Besides Budinich's SOM [7], there are the enhanced CEN [2], the KNIES [4], as well as SA [20]. Like the implementation of SA in [7], we have taken the annealing factor 0.95, adopted the exchange method known as *2-Opt* inspired by Lin and Kernighan [26], and allowed total $20 \times n$ trials at each temperature level. It usually generates better tours than the heuristic *2-Opt* [7]. Our experimental results have been based on 10 independent runs for each TSP. For convenience, we have quoted the experimental results of the enhanced CEN [2] and the KNIES [4] from papers directly. It is partially due to the lack of the implementation details.

5.2. Experimental results

The first set of experiments we have performed is to compare the ESOM with Budinich's SOM and SA. The tested TSPs consist of 18 synthetic TSPs ranging from 50 to 2400 cities which have been generated randomly within a unit square.

Fig. 4(a) shows the experimental results of the ESOM, Budinich's SOM, and SA. The solution quality is in terms of the relative difference of the average tour length over a theoretical lower bound. The theoretical lower bound for a random TSP with n cities within the unit square is calculated by the Stein's formula [15], i.e., $0.765 \times \sqrt{n}$. It is shown in Fig. 4(a) that the performances of SA and Budinich's SOM are very similar, which coincides with [7]. For all the 18 TSPs, the tours obtained by the ESOM are much nearer to the theoretical bounds than those by Budinich's SOM and SA. Its solution qualities remain reasonably stable for different TSPs. For example, its tours for the TSP with 2400 cities are about 3.50% longer than the theoretical bound. Those of SA and Budinich's SOM are 5.22% and 5.99% respectively. More

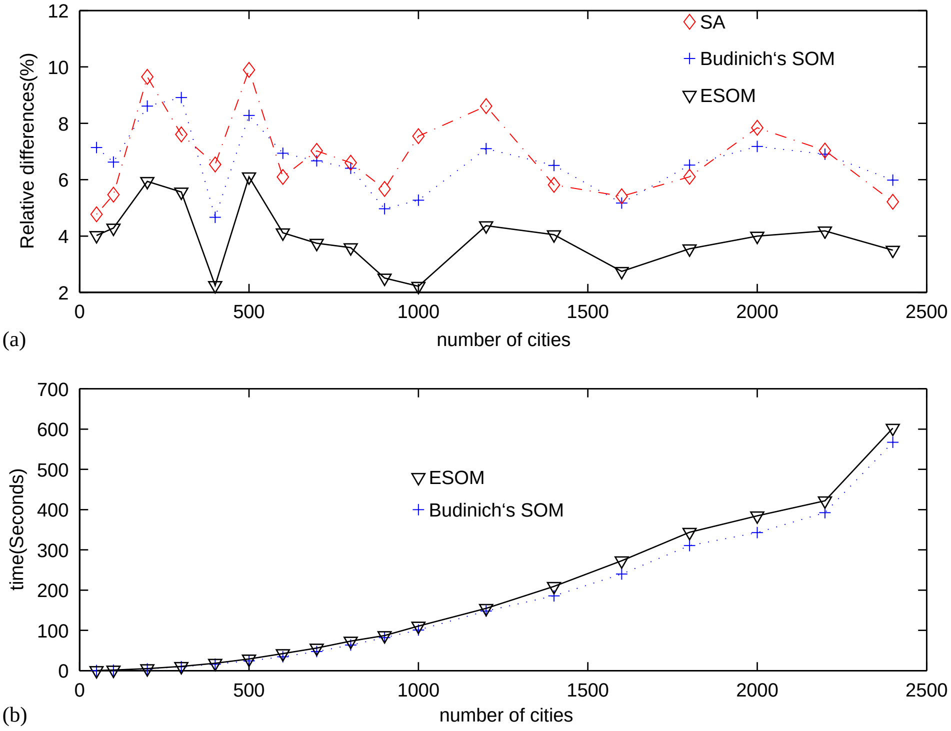


Fig. 4. Experimental results for 18 synthetic TSPs: (a) Comparison of the solution qualities yielded by the ESOM, Budinich's SOM and SA; (b) the average execution time comparison between Budinich's SOM and the ESOM.

specifically, we average all the results for the 18 TSPs, and find that the solution qualities are 3.93%, 6.65%, and 6.83%, respectively, for the ESOM, Budinich's SOM and SA. Consequently, the ESOM makes around 2.72 percentage improvement on Budinich's SOM and 2.90 on SA. Similar situations can also be observed from the other two sets of experiments, as listed in Tables 1 and 2, where the solution quality is the relative difference of the best generated tour length from the optimum. The ESOM makes at least 2.70 and 1.37 percentage improvement on its two counterparts for the two sets of benchmark TSPs, respectively. Based on these series of experiments, we conclude that the ESOM has made remarkable improvement on Budinich's SOM and SA.

Now we compare these algorithms in another measure. Fig. 4(b) shows the execution time of the ESOM and Budinich's SOM for the 18 TSPs. It is clear that their execution time is not significantly different, though the ESOM takes slightly longer than Budinich's SOM. It is noted, furthermore, that both the ESOM and Budinich's SOM run much faster than SA. For example, SA takes about 5400 s to generate a tour of the TSP with 2400 cities. However, only around 602 s are taken by the ESOM and Budinich's SOM.

Table 1
Experimental results of SA, Budinich's SOM, ESOM, and the enhanced CEN on 5 benchmark TSPs

TSP name	City no.	Optimum	Solution quality of 3 algorithms (%)			Solution quality of enhanced SOMs (%)		
			SA	Budinich	ESOM	CEN	Budinich	ESOM
GR96	96	51,231	4.12	2.09	**1.03**	4.39	**0.46**	**0.46**
GRID100	100	100	2.07	2.17	**0.83**	**0.80**	1.63	0.83
KROA100	100	21,282	5.94	3.68	**1.01**	1.60	0.93	**0.81**
GR137	137	69,853	8.45	8.61	**4.27**	3.29	4.51	**2.52**
LIN318	318	44,169	7.56	8.19	**4.11**	4.67	**2.81**	2.89
Average			5.63	4.95	**2.25**	2.95	2.07	**1.50**

Table 2
Experimental results of five algorithms on 20 benchmark TSPs

TSP name	City no.	Optimum	Solution quality of 5 algorithms (%)				
			KNIES-global	KNIES-local	SA	Budinich	ESOM
EIl51	51	426	2.86	2.86	2.33	3.10	**2.10**
ST70	70	675	2.33	**1.51**	2.14	1.70	2.09
EIL76	76	538	5.48	4.98	5.54	5.32	**3.89**
RD100	100	7910	2.62	2.09	3.26	3.16	**1.96**
EIL101	101	629	5.63	4.66	5.74	5.24	**3.43**
LIN105	105	14,379	1.29	1.98	1.87	1.71	**0.25**
PR107	107	44,303	**0.42**	0.73	1.54	1.32	1.48
PR124	124	59,030	0.49	**0.08**	1.26	1.62	0.67
BIER127	127	1,18,282	3.08	2.76	3.52	3.61	**1.70**
PR136	136	96,772	5.15	4.53	4.90	5.20	**4.31**
PR152	152	73,682	1.29	0.97	2.64	2.04	**0.89**
RAT195	195	2323	11.92	12.24	13.29	11.48	**7.13**
KROA200	200	28,568	6.57	5.72	5.61	6.13	**2.91**
PCB442	442	50,778	10.45	11.07	9.15	8.43	**7.43**
ATT532	532	27,686	6.80	6.74	5.38	5.67	**4.95**
Average on 15 TSPs			4.43	4.20	4.54	4.38	**3.01**
PR1002	1002	2,59,045			6.03	8.75	5.93
PCB1173	1173	56,892			11.14	11.38	9.87
FL1400	1400	20,127			4.74	5.57	4.05
D1655	1655	62,128			13.21	15.18	11.35
VM1748	1748	3,36,556			7.94	10.19	7.27
Average on 20 TSPs					5.56	5.84	4.18

It is interesting to point out that our implementation of SA is similar to the baseline implementation of SA in [19]. Its performance can be greatly enhanced with some sophisticated implementation techniques, such as the table lookup scheme of the exponential function, neighborhood pruning, intelligent temperature starts, and permutation-based neighbor generation. The enhanced SA (SA$_2^+$) may generate better near-optimal tours. For example, it generates tours only 1.90% longer than the optimum for PR1002 with 327 seconds [19]. Its solution quality is much higher than 5.93% and 8.75% of the ESOM and Budinich's SOM as listed in Table 2, but it needs twice longer time than the two SOMs. So, though the ESOM cannot generate tours as short as the enhanced SA, it takes less time.

Our second set of experiments has been primarily designed to compare the ESOM with the CEN [2], as well as Budinich's SOM and their enhanced versions. Here, an enhanced version of a SOM means that the algorithm is enhanced with the heuristics *NII*, which improves tours by using a rearrangement heuristics derived from *2-Opt* [2]. We have quoted the experimental results of the enhanced CEN[2] directly from [2], and we also present the results of the enhanced ESOM for a fair comparison. The benchmark TSPs are taken from the TSPLIB, collected by Reinelt [30].

Table 1 lists the experiment and comparison results of the plain and the enhanced SOM-like networks where the bold-faced text indicates the best one among 3 different algorithms for a TSP. The solution quality is in terms of the relative difference of the best generated tour length from the optimum. It may be observed from Table 1 that, for both the plain and the enhanced versions, the ESOM always yields much higher quality solutions than Budinich's SOM, SA, and the enhanced CEN do. More specifically, we can observe from Table 1 that the enhanced ESOM has improved the solution quality of the enhanced CEN from 2.95% to 1.50% on average. Even the plain ESOM can generate averagely shorter tours than the enhanced CEN. Note that the CEN takes at least $O(n \ln n)$ additional computation to construct the convex-hull explicitly, and hence the CEN would be slower than the ESOM. Therefore, we conclude that the ESOM beats the CEN in terms of both time and accuracy.

The third set of experiments has been conducted for the comparison between the ESOM and the KNIES, which had been claimed to be the most accurate neural network for the TSP currently reported in the literature [4]. Its basic idea is to, after the SOM learning, disperse output neurons to make their statistics equal to that of some cities. If all the cities participate, it leads to a global version, denoted by KNIES-global in Table 2. If only the represented cities are involved, it leads to a local version which is denoted by KNIES-local. Both of them perform quite well and their results for 15 TSPs in [4] are entirely listed in Table 2. Here, the solution quality data again are the relative differences of the best tour lengths from the optima, and the bold-faced text indicates the best one for a TSP. All the 20 TSPs can be obtained from the TSPLIB.

It may be observed from Table 2 that all the five algorithms can generate good tours for the first 15 TSPs. The solution qualities of the KNIES-global, the KNIES-local, SA, and Budinich's SOM are very close. On average, their solution qualities are 4.43%, 4.20%, 4.54%, and 4.38%, respectively. The ESOM may generate higher quality solutions than its four counterparts except ST70, PR107 and PR124. Its average solution quality is 3.01%, which makes at least 1.19 percentage improvement on its counterparts. The solution qualities of the ESOM range from 0.67% to 7.43%, and are reasonably stable. The solutions for ATT532 generated by SA, Budinich's SOM and

[2] The optimum of GR96 cited by Al-Mulhem and Al-Maghrali [2] is as long as 55,209.

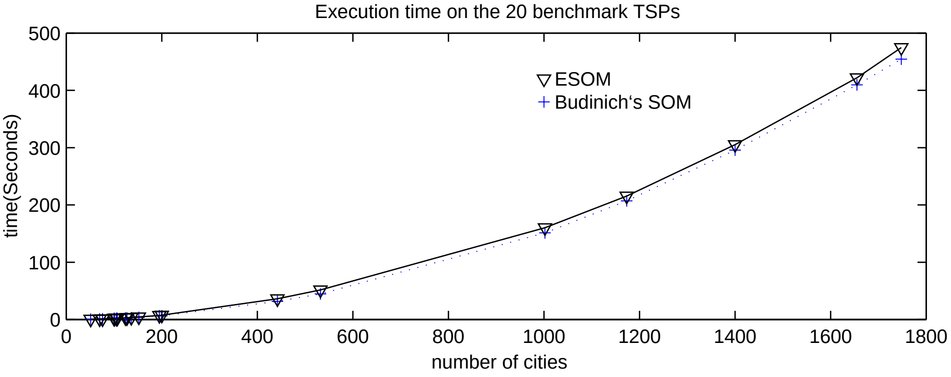


Fig. 5. The average execution time of Budinich's SOM and the ESOM for 20 benchmark TSPs.

the ESOM are shown in Fig. 6. It is worth pointing out that the ESOM can achieve the convex-hull property. On the other hand, both the KNIES-global and the KNIES-local need additional memory to indicate the status of each neurons. Furthermore, in order to maintain the overall statistics explicitly, they have to update each neuron in each learning iteration. So, the KNIES-global and the KNIES-local are at least more complicated than the ESOM. Consequently, the performance of the ESOM is better than the two KNIES algorithms.

Table 2 also lists the performance of SA, Budinich's SOM, and our ESOM for five large TSPs. For each TSP, the ESOM generates the most accurate tour among the three algorithms. It makes at least 1.51 percentage improvement on Budinich's SOM. The solution qualities of the ESOM ranges from 4.05% to 11.35%. Thus, its performance, though becomes a bit worse as the number of cities increases, is still stable. Fig. 5 illustrates the execution time of the ESOM and Budinich's SOM for the 20 TSPs. Similar to the performance for the 18 synthetic TSPs in Fig. 4(b), the ESOM runs as fast as Budinich's SOM. The execution time of both algorithm does not increase very quickly with the number of cities. For example, the ESOM takes about 475 s for VM1748, and Budinich's SOM takes 454 s. In comparison, SA spends 4777 seconds, which is about 10-fold longer. This performance comparison among the ESOM, Budinich's SOM, and SA on the benchmark TSPs is consistent with the one on the 18 synthetic TSPs.

6. Conclusion

We have developed an improved SOM-like neural network, the expanding self-organizing map (ESOM), for the traveling salesman problem (TSP). Through drawing the excited neurons closely towards the input city and pushing them outwards from the center of all the cities simultaneously in each learning iteration, the ESOM can satisfy the convex-hull property gradually as well as construct a good neighborhood preserving map. Our one-step trend analysis has shown its feasibility and its similar asymptotic

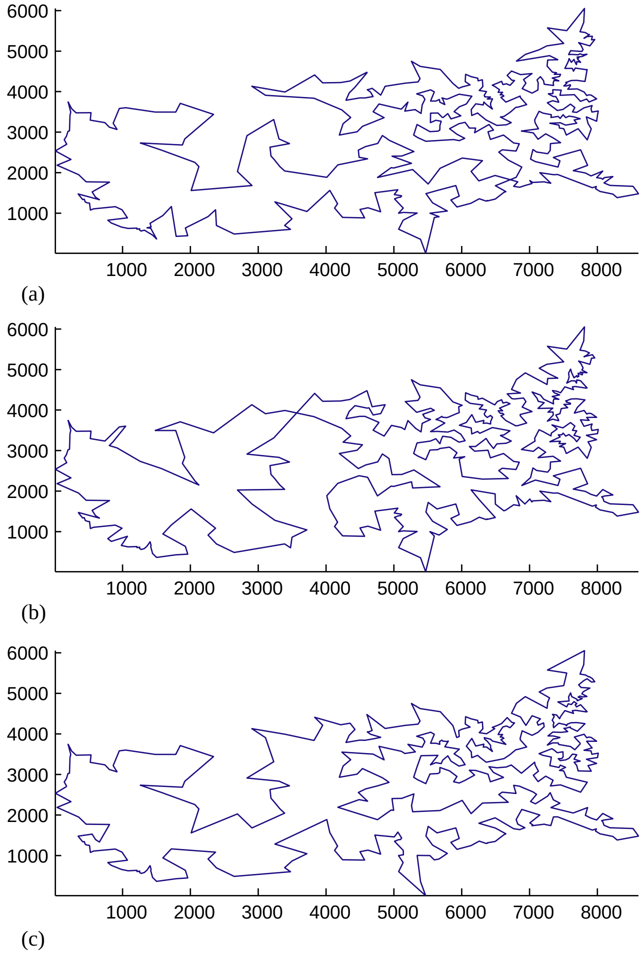


Fig. 6. Typical solutions obtained for ATT532 with 532 cities: (a) by SA; (b) by Budinich's SOM; (c) by the ESOM.

behavior with the conventional SOM. We have also shown that, like many SOMs for the TSP, the computation complexity of the ESOM is quadratic. A comprehensive series of experiments for both synthetic and benchmark TSPs have been conducted to illustrate the good performance the ESOM. They have particularly demonstrated that the proposed ESOM outperforms several typical SOM-like neural networks such as Budinich's SOM [7], the enhanced convex elastic net [2], and the KNIES algorithms [4]. Though its solution accuracy is not yet comparable to some sophisticated heuristics, we believe that the ESOM is one of the most accurate neural networks for the TSP in the literature.

There are several opportunities for further research in the line of the present work. The most challenging problems are the detailed convergence analysis of the ESOM as well as other SOM-like neural networks in a rigorous way. Another research direction of our interest is to design some new learning rules and tune the parameters involved for better performance. In addition, the idea of incorporating some necessary conditions into SOMs will be extended and applied to other applications, e.g., cluster analysis.

Acknowledgements

The authors appreciate the anonymous reviewers for their valuable comments to strengthen the paper. The authors also acknowledge the suggestions from Dr. Man-Leung Wong and Dr. Jiang-She Zhang. Special thanks go to Dr. Marco Budinich who kindly provided us several helpful references and implementation details. This research was partially supported by RGC Research Grant Direct allocation of the Chinese University of Hong Kong, the Hi-Tech R&D (863) Program (No. 2001AA113182) and the Nature Science Foundation Project (No. 10371097) of China.

Appendix A

Proof of Theorem 1. We prove the theorem by using the "reduction to absurdity" argument.

Let C_i $(i=1,2,\ldots,n)$ denote the cities of the TSP and C_{h_j} $(j=1,2,\ldots,m \geqslant 4)$ denote the cities located on the convex-hull. For simplicity, we assume that $C_{h_1}, C_{h_2}, \ldots, C_{h_m}$ are the order in which they appear clockwise on the convex-hull. Suppose that there exists an optimal tour T in which C_{h_j} $(j = 1,2,\ldots,m)$ do not appear in ascending or descending order with index j. Without loss of generality, we further assume that the cities C_{h_j} $(j = 1,2,\ldots, p < m-1)$ appear in the tour T in ascending order and the city C_{h_q} $(2 < p+1 < q \leqslant m)$ is the subsequent city in the tour T. Mark the path between C_{h_p} and C_{h_q} in the tour T as S_1 which only contains two cities located on the convex-hull. Then, we see that the path S_1 really separates the convex-hull into two parts, and furthermore, there exists a path $S_2 = C_{h_{p+1}} C_k \cdots C_{h_r}$ from one part to the other one such that both C_{h_p} and C_{h_q} are excluded. Indeed, consider the next city C_{h_i} in the tour T after C_{h_q}, if $i > q$, S_2 may be the path between $C_{h_{p+1}}$ and C_{h_i} in the tour; otherwise, S_2 is the path between $C_{h_{p+1}}$ and C_{h_1}. Then the paths S_1 and S_2 must intersect each other because the line segment $C_{h_p}C_{h_q}$ intersects with the line segment $C_{h_{p+1}}C_{h_r}$ inside of the convex-hull, and the paths S_1 and S_2 can be continuously transformed into the line segments $C_{h_p}C_{h_q}$ and $C_{h_{p+1}}C_{h_r}$, respectively. Therefore the tour T intersects itself. This contradicts the assumption that the tour T is optimal, since it is well known that any optimal tour of the TSP cannot intersect itself. This justifies Theorem 1. □

Proof of Theorem 2. (i) We first prove Eq. (8) by induction. This is trivially true for $t = 0$ according to Step 2 of the ESOM. If we assume that Eq. (8) holds for certain

t ($\geqslant 0$), then we find

$$1 - 2R^2 \leqslant -\tfrac{1}{2}(x_{k1}^2 + x_{k2}^2 + w_{j1}^2 + w_{j2}^2) + \sqrt{(1-R^2)^2} \tag{A.1}$$

$$\leqslant 1 - \beta_j = x_{k1}w_{j1} + x_{k2}w_{j2} + \sqrt{(1 - x_{k1}^2 - x_{k2}^2)(1 - w_{j1}^2 - w_{j2}^2)}$$

$$\leqslant \left[x_{k1}^2 + x_{k2}^2 + \sqrt{(1 - x_{k1}^2 - x_{k2}^2)^2}\right]\left[w_{j1}^2 + w_{j2}^2 + \sqrt{(1 - w_{j1}^2 - w_{j2}^2)^2}\right] = 1, \tag{A.2}$$

where the index t is omitted for simplicity. So, $0 \leqslant \beta_j(t) \leqslant 2R^2$. On the other hand, for any learning rate $\alpha_j(t) \in 0, 1$, it is seen that $0 \leqslant \alpha_j(t)(1 - \alpha_j(t)) \leqslant 0.25$. These show $1 \leqslant c_j(t) \leqslant 1/(\sqrt{1-R^2})$. Then,

$$1 - \|\vec{w}_j(t+1)\|^2 = \frac{(c_j(t))^{-2} - \|\vec{w}_j(t) + \alpha_j(t)(\vec{x}_k(t) - \vec{w}_j(t))\|^2}{(c_j(t))^{-2}}$$

$$= \frac{\left[(1 - \alpha_j(t))\sqrt{1 - ||\vec{w}_j(t)||^2} + \alpha_j(t)\sqrt{1 - ||\vec{x}_k(t)||^2}\right]^2}{(c_j(t))^{-2}}$$

$$\geqslant \left[(1 - \alpha_j(t))\sqrt{1 - R^2} + \alpha_j(t)\sqrt{1 - R^2}\right]^2 = 1 - R^2. \tag{A.3}$$

This implies, $\|\vec{w}_j(t+1)\| \leqslant R$ for any $j = 1, \ldots, M$. Thus, by induction, $\vec{w}_j(t) \in C_R$ for any j and t.

(ii) For simplicity, we also omit the index t below. Denote

$$\theta = \arcsin R,$$

$$\vec{u}_k = \left[x_{k1}, x_{k2}, \sqrt{1 - x_{k1}^2 - x_{k2}^2}\right]^{\mathrm{T}},$$

and

$$\vec{v}_j = \left[w_{j1}, w_{j2}, \sqrt{1 - w_{j1}^2 - w_{j2}^2}\right]^{\mathrm{T}}.$$

Clearly, $\|\vec{u}_k\| = \|\vec{v}_j\| = 1$. We proceed to show Eq. (9) with reference to Fig. 7. Let $\phi = \angle ADB$ and $\psi = \angle ADC$. First, we claim that $0 \leqslant \phi,\ \psi \leqslant \theta$ holds.

In fact, observing Fig. 7, we find $\phi = \angle ADB = \frac{1}{2}\angle AOD$ due to $|OA| = |OD| = 1$ and $DB \perp OA$. Since $\sin(\frac{1}{2}\angle AOD) = \frac{1}{2}|AD| = \frac{1}{2}\|\vec{u}_k - \vec{v}_j\|$ and

$$\|\vec{u}_k - \vec{v}_j\|^2 = 2\beta_j(t) \leqslant 4R^2,$$

we have $\sin\phi \leqslant R$. Note that $\theta, \phi \in [0, \pi/2)$ and $R = \sin\theta$. This implies $\phi \leqslant \theta$. To show $\psi \leqslant \theta$, we observe from Fig. 7 that

$$\tan\psi = \tan\angle ADC = \frac{|AC|}{|CD|} = \left|\frac{\sqrt{1 - w_{j1}^2 - w_{j2}^2} - \sqrt{1 - x_{k1}^2 - x_{k2}^2}}{\sqrt{(w_{j1} - x_{k1})^2 + (w_{j2} - x_{k2})^2}}\right|.$$

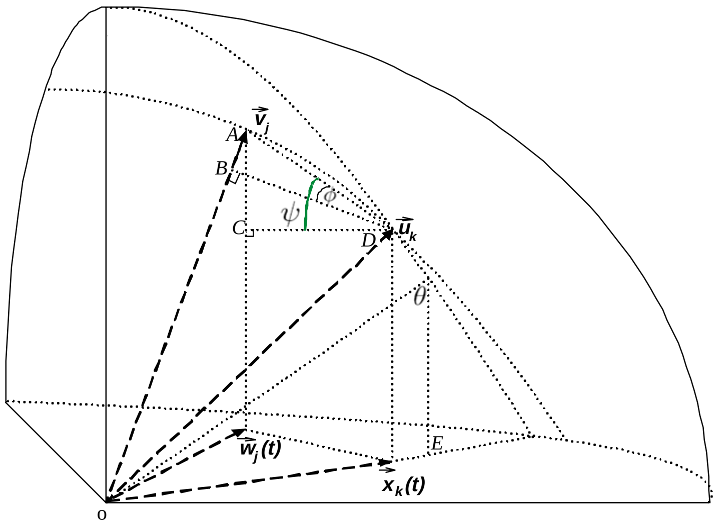


Fig. 7. A geometric illustration of some terms in the expanding coefficient.

Therefore $\tan\psi$ depends only on the relative position of $\vec{w}_j$ and $\vec{x}_k$. So, without loss of generality, we can assume $x_{k2} = 0$ and denote $(w_{j1}, w_{j2}) = (x_{k1} + l, \mu l)$ for the further verification. We now show that the function $\tan\psi$ decreases with l whenever $\|\vec{x}_k\| \leqslant R$ and $\|\vec{w}_j\| \leqslant R$. Due to the symmetry of $\vec{w}_j$ and $\vec{x}_k$ in $\tan\psi$, we need to achieve this only for the case when $x_{k1} \geqslant 0$ and $w_{j1}^2 + w_{j2}^2 \leqslant x_{k1}^2$ (i.e., $(x_{k1}+l)^2 + \mu^2 l^2 \leqslant x_{k1}^2$, and hence, $-2x_{k1}/(1+\mu^2) \leqslant l \leqslant 0$.). Furthermore, we can only consider the case that $\mu \geqslant 0$ because $\tan\psi$ is also symmetric with respect to μ. Thus we can calculate that

$$\begin{aligned}\frac{\partial \tan\psi}{\partial l} &= \frac{\partial\left(\frac{\sqrt{1-(x_{k1}+l)^2-(\mu l)^2}-\sqrt{1-x_{k1}^2}}{l\sqrt{1+\mu^2}}\right)}{\partial l}\\ &= \frac{\frac{-((x_{k1}+l)-\alpha^2 l)l}{\sqrt{1-(x_{k1}+l)^2-(\mu l)^2}-\sqrt{1-x_{k1}^2}} - (\sqrt{1-(x_{k1}+l)^2-(\mu l)^2}-\sqrt{1-x_{k1}^2})}{(l^2\sqrt{1+\mu^2})}\\ &\leqslant 0, \end{aligned} \tag{A.4}$$

where the last inequality is derived from the fact that

$$3(1-x_{k1}^2)^2 - l\cdot x_{k1}(1-x_{k1}^2) + l^2 + \mu l^2(1+x_{k1}^2) \geqslant 0 \tag{A.5}$$

and

$$\left[\sqrt{1-(x_{k1}+l)^2-(\mu l)^2}-\sqrt{1-x_{k1}^2}\right]^2 \geqslant ((x_{k1}+l)-\alpha^2 l)l. \tag{A.6}$$

Inequality (A.4) shows that ψ becomes larger as $\vec{w}_j$ moves to $\vec{x}_k$, and then this further shows that $\tan\psi$ will approach to the maximum when $\psi \to 0$. So we get

$$\tan 0 = \left.\frac{\frac{\partial\left(\sqrt{1-(x_1+l)^2-(\mu l)^2}-\sqrt{1-x_1^2}\right)}{\partial l}}{\frac{\partial\left(l\sqrt{1+\mu^2}\right)}{\partial l}}\right|_{l=0^-} = \frac{\frac{x_1}{\sqrt{1-x_1^2}}}{\sqrt{1+\mu^2}}$$

$$= \frac{x_1}{\sqrt{1+\mu^2}\sqrt{1-x_1^2}} \leqslant \frac{R}{\sqrt{1+\mu^2}\sqrt{1-R^2}}$$

$$\leqslant \frac{R}{\sqrt{1-R^2}} = \tan\theta. \tag{A.7}$$

This implies $\psi \leqslant \theta$ since $0 \leqslant \psi,\ \theta \leqslant \pi/2$, as claimed before.

From $0 \leqslant \phi,\ \psi \leqslant \theta$, we then have

$$\beta_j = 1 - \vec{u}_k^{\mathrm{T}}\vec{v}_j = |AB| = |AD|\sin\phi \leqslant |AD|\sin\theta$$

and

$$\|\vec{w}_j - \vec{x}_k\| = |CD| = |AD|\cos\psi \geqslant |AD|\cos\theta.$$

Since $0 \leqslant 2\alpha_j(1-\alpha_j)\beta_j \leqslant R^2$ and the function $F(y) = (1-\sqrt{1-y})/y$ monotonously increases for $y \in (0,1]$, one gets

$$\frac{1-\sqrt{1-R^2}}{R^2} \geqslant \frac{1-c_j^{-1}}{2\alpha_j(1-\alpha_j)\beta_j}.$$

Therefore,

$$c_j^{-1} \geqslant 1-(1-\sqrt{1-R^2})/R^2 \times 2\alpha_j(1-\alpha_j)\beta_j$$

$$= 1-2\alpha_j(1-\alpha_j)\beta_j(1-\sqrt{1-R^2})/R^2$$

$$\geqslant 1-\alpha_j(1-\alpha_j)|AD|\sin\theta(1-\sqrt{1-R^2})/R^2$$

$$\geqslant 1-2\alpha_j(1-\alpha_j)\tan\theta\|\vec{w}_j-\vec{x}_k\|(1-\sqrt{1-R^2})/R^2$$

$$= 1-2\alpha_j(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|(1-\sqrt{1-R^2})/\sqrt{1-R^2}/R. \tag{A.8}$$

Consequently, we obtain

$$\|\vec{w}_j(t+1)-\vec{w}'_j(t+1)\| = (c_j-1)\|\vec{w}'_j(t+1)\| \tag{A.9}$$

$$\leqslant \{[1-2\alpha_j(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|(1-\sqrt{1-R^2})$$

$$\times R/\sqrt{1-R^2}]^{-1}-1\}\|\vec{w}'_j(t+1)\|$$

$$
\begin{aligned}
&= \frac{2\alpha_j(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|}{1-2\alpha_j(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|(1-\sqrt{1-R^2})/R/\sqrt{1-R^2}}\\
&\quad \times\|\vec{w}_j+\alpha_j(\vec{x}_k-\vec{w}_j(t))\|(1-\sqrt{1-R^2})/R/\sqrt{1-R^2}\\
&\leqslant \frac{2R\times(1-\sqrt{1-R^2})(1-\alpha_j)\alpha_j\|\vec{w}_j-\vec{x}_k\|}{R\sqrt{1-R^2}-4\alpha_j(1-\alpha_j)R(1-\sqrt{1-R^2})}\\
&\leqslant \frac{2(1-\sqrt{1-R^2})}{\sqrt{1-R^2}-(1-\sqrt{1-R^2})}(1-\alpha_j)\alpha_j\|\vec{w}_j-\vec{x}_k\|\\
&= \varsigma(1-\alpha_j)\alpha_j\|\vec{w}_j-\vec{x}_k\|. \qquad \text{(A.10)}
\end{aligned}
$$

A schematic illustration of this inequality is shown in Fig. 1(d). Therefore,

$$
\begin{aligned}
\|\vec{w}_j(t+1)-\vec{x}_k\| &= \|[\vec{w}_j(t+1)-\vec{w}'_j(t+1)]+[\vec{w}'_j(t+1)-\vec{x}_k]\|\\
&\leqslant \|\vec{w}_j(t+1)-\vec{w}'_j(t+1)\|+\|\vec{w}'_j(t+1)-\vec{x}_k\|\\
&\leqslant \varsigma\alpha_j(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|+(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|\\
&= (1+\varsigma\alpha_j)(1-\alpha_j)\|\vec{w}_j-\vec{x}_k\|, \qquad \text{(A.11)}
\end{aligned}
$$

By the definition of ς, $\varsigma<1$ whenever $R<\frac{\sqrt{7}}{4}$. From Eq. (A.11), Eqs. (9) and (12) then follow. This justifies (ii).

(iii) Similar with the process leading to Eq. (A.11), we can easily obtain the following estimate:

$$
\begin{aligned}
\|\vec{w}_j(t+1)-\vec{x}_k\| &\geqslant \|\vec{w}'_j(t+1)-\vec{x}_k\|-\|\vec{w}_j(t+1)-\vec{w}'_j(t+1)\|\\
&\geqslant (1-\alpha_j)(1-\varsigma\alpha_j)\|\vec{w}_j(t)-\vec{x}_k\|. \qquad \text{(A.12)}
\end{aligned}
$$

Together with Eq. (A.11), this implies Eq. (13). With this, the proof of Theorem 2 is completed. □

Proof of Theorem 3. We only show the increasing property of the expanding coefficient $c_j(t)$ with $\|\vec{x}_k(t)\|$ when $\|\vec{x}_k(t)\|\geqslant\|\vec{w}_j(t)\|$ and $\vec{x}_k(t)$ and $\vec{w}_j(t)$ are within a one-fourth circle. The increasing property with $\|\vec{w}_j(t)\|$ can be deduced in a similar way because $c_j(t)$ is symmetric with respect to $\vec{x}_k(t)$ and $\vec{w}_j(t)$. For convenience, we express the vectors $\vec{x}_k(t)$ and $\vec{w}_j(t)$ in the polar coordinate form as follows:

$$
\begin{aligned}
\vec{w}_j(t) &= [w_{j1}(t),w_{j2}(t)]^{\mathrm{T}}=[r\sin\phi,r\cos\phi]^{\mathrm{T}},\\
\vec{x}_k(t) &= [x_{k1}(t),x_{k2}(t)]^{\mathrm{T}}=[\rho\sin\theta,\rho\cos\theta]^{\mathrm{T}}.
\end{aligned}
$$

Here $\vec{x}_k(t)$ and $\vec{w}_j(t)$ are within the one-fourth circle, thus $|\theta-\phi|\leqslant\pi/2$. Let

$$
\begin{aligned}
F(\rho,r) &= \vec{w}_j^{\mathrm{T}}(t)\vec{x}_k(t)+\sqrt{(1-||\vec{w}_j(t)||^2)(1-||\vec{x}_k(t)||^2)}\\
&= \rho r\cos(\theta-\phi)+\sqrt{(1-\rho^2)(1-r^2)}.
\end{aligned}
$$

From the definition of $c_j(t)$ in Eqs. (6) and (7), it is obvious that $F(\rho, r) = 1 - \beta_j = 1 - (1 - c_j^{-2}(t))/2\alpha_j(t)(1 - \alpha_j(t))$. $F(\rho, r)$ decreases with the expanding coefficient $c_j(t)$. So, to justify the increasing property of $c_j(t)$, it is sufficient to show that $F(\rho, r)$ decreases with ρ whenever $\rho \geqslant r$. A direct calculation shows

$$\frac{\partial F(\rho, r)}{\partial \rho} = r\cos(\theta - \phi) - \frac{\rho}{\sqrt{1-\rho^2}}\sqrt{1-r^2}. \tag{A.13}$$

This implies the decreasing property of $F(\rho, r)$ when $\rho \geqslant r$ because of the monotonic increasing property of $x/(\sqrt{1-x^2})$ and $0 \leqslant \cos(\theta - \phi) \leqslant 1$.

If $\vec{x}_k(t)$ and $\vec{w}_j(t)$ are not within a one-fourth circle, the two terms in the right-hand side of Eq. (A.13) are negative. So, the expanding coefficient $c_j(t)$ increases with $\|\vec{x}_k(t)\|$. From the proof of Theorem 2, we know $c_j(t) \geqslant 1.0$. We also know $c_j(t) = 1.0$ as $\vec{x}_k(t) = \vec{w}_j(t)$. That is, it reaches the minimum. This completes the proof of Theorem 3. □

References

[1] S. Abe, Convergence acceleration of the Hopfield neural network by optimization integration step sizes, IEEE Trans. Systems Man Cybernet.—B: Cybernet. 26 (1) (1996) 194–201.

[2] H. Al-Mulhem, T. Al-Maghrabi, Efficient convex-elastic net algorithm to solve the Euclidean traveling salesman problem, IEEE Trans. Systems Man Cybernet.—B: Cybernet. 28 (1998) 618–620.

[3] B. Angèniol, G.D.L.C. Vaubois, J.Y.L. Texier, Self-organizing feature maps and the travelling salesman problem, Neural Networks 1 (4) (1988) 289–293.

[4] N. Aras, B.J. Oommen, I.K. Altinel, Kohonen network incorporating explicit statistics and its application to the travelling salesman problem, Neural Networks 12 (9) (1999) 1273–1284.

[5] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, J. ACM 45 (5) (1998) 753–782.

[6] N. Ascheuer, M. Jünger, G. Reinelt, A branch and cut algorithm for the asymmetric traveling salesman problem with precedence constraints, Comput. Optim. Appl. 17 (1) (2000) 61–84.

[7] M. Budinich, A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing, Neural Comput. 8 (1996) 416–424.

[8] M. Budinich, B. Rosario, A neural network for the traveling salesman problem with a well behaved energy function, in: S.W. Ellacott, J.C. Mason, I.J. Anderson (Eds.), Mathematics of Neural Networks: Models, Algorithms and Applications, Kluwer Academic Publishers, Boston, MA, 1997, pp. 134–139.

[9] L.I. Burke, P. Damany, The guilty net for the traveling salesman problem, Comput. Oper. Res. 19 (1992) 255–265.

[10] C.Y. Dang, L. Xu, A globally convergent lagrange and barrier function iterative algorithm for the traveling salesman problem, Neural Networks 14 (2) (2001) 217–230.

[11] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1) (1997) 53–66.

[12] R. Durbin, D. Willshaw, An analogue approach to the traveling salesman problem, Nature 326 (1987) 689–691.

[13] F. Favata, R. Walker, A study of the application of Kohonen-type neural networks to the traveling salesman problem, Biol. Cybernet. 64 (1991) 463–468.

[14] B. Fritzke, P. Wilke, FLEXMAP—a neural network with linear time and space complexity for the travelling salesman problem, in: Proceeding of IJCNN-90 International Joint Conference on Neural Networks, 1991, pp. 929–934.

[15] B. Golden, W. Stewart, Empirical analysis of heuristics, in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinooy Kan, D.B. Shimoys (Eds.), The Travelling Salesman Problem, Wiley, New York, 1985.

[16] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[17] J.J. Hopfield, D.W. Tank, 'Neural' computation of decisions in optimization problems, Biol. Cybernet. 52 (1985) 141–152.

[18] H.-D. Jin, K.-S. Leung, M.-L. Wong, Z.-B. Xu, An efficient self-organizing map designed by genetic algorithms for the traveling salesman problem, IEEE Trans. System Man Cybernet.—B: Cybernet. 33 (6) (2003) 877–888.

[19] D.S. Johnson, L.A. McGeoch, The travelling salesman problem: a case study, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, New York, 1997, pp. 215–310.

[20] S.G. Kirkpatrick Jr., C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.

[21] J. Knox, Tabu search performance on the symmetric traveling salesman problem, Comput. Oper. Res. 21 (1994) 867–876.

[22] T. Kohonen, Self-Organizing Maps, Springer, New York, 1997.

[23] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, V. Paatero, A. Saarela, Organization of a massive document collection, IEEE Trans. Neural Networks 11 (3) (2000) 574–585.

[24] N. Krasnogor, J. Smith, A memetic algorithm with self-adaptive local search: TSP as a case study, in: D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, H.-G. Beyer (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), 2000, pp. 987–994.

[25] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms, Eur. J. Oper. Res. 59 (1992) 345–358.

[26] S. Lin, B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, Oper. Res. 21 (1973) 498–516.

[27] S. Lin, J. Si, Weight-value convergence of the SOM algorithm for discrete input, Neural Comput. 10 (4) (1998) 807–814.

[28] Y. Matsuyama, Harmonic competition: a self-organizing multiple criteria optimization, Neural Networks 7 (3) (1996) 652–668.

[29] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover Publications Inc.(Reprint), New York, July 1998.

[30] G. Reinelt, TSPLIB—A traveling salesman problem library, ORSA J. Comput. 3 (4) (1991) 376–384.

[31] A.A. Sadeghi, Self-organization property of Kohonen's map with general type of stimuli distribution, Neural Networks 11 (9) (1998) 1637–1643.

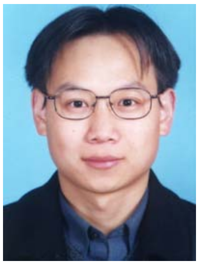
[32] W.H. Shum, H.-D. Jin, K.-S. Leung, M.-L. Wong, A self-organizing map with expanding force for data clustering and visualization, in: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, December 2002, pp. 434–441.

[33] K.A. Smith, An argument for abandoning the travelling salesman problem as a neural network benchmark, IEEE Trans. Neural Networks 7 (6) (1996) 1542–1544.

[34] C. Voudouris, E. Tsang, Guided local search and its application to the traveling salesman problem, Eur. J. Oper. Res. 113 (2) (1999) 469–499.

[35] Z.B. Xu, H.D. Jin, K.S. Leung, L. Leung, C.K. Wong, An automata network for performing combinatorial optimization, Neurocomputing 47 (2002) 59–83.

K.-S. Leung received his B.Sc. (Eng.) and Ph.D. degrees in 1977 and 1980, respectively, from the University of London, Queen Mary College. He worked as a senior engineer on contract R&D at ERA Technology and later joined the Central Electricity Generating Board to work on nuclear power station simulators in England. He joined the Computer Science and Engineering Department at the Chinese University of Hong Kong in 1985, where he is currently professor and chairman of the Department. Dr Leung's research interests are in soft computing including evolutionary computation, neural computation, probabilistic search, information fusion and data mining, fuzzy data and knowledge engineering. He has published over 180 papers and 2 books in fuzzy logic and evolutionary computation. He has been chair and member of many program and organizing committees of international conferences. He is in the Editorial Board of Fuzzy Sets and Systems and an associate editor of International Journal of Intelligent Automation and Soft Computing. He is a senior member of the IEEE, a chartered engineer, a member of IEE and ACM and a fellow of HKCS and HKIE.



H.-D. Jin received his B.Sc. and M.Sc. degrees in applied mathematics in 1995 and 1998, respectively, from Xi'an Jiaotong University, China. In 2002, he got his Ph.D. degree of Computer Science and Engineering from the Chinese University of Hong Kong, Hong Kong. He is currently with Division of Mathematical and Information Sciences, CSIRO, Australia. His research interests are neural networks, intelligent computation, and data mining. He is a member of ACM and IEEE.

Z.-B. Xu received the M.S. degree in mathematics in 1981 and the Ph.D. degree in applied mathematics in 1987 from Xi'an Jiaotong University, China. In 1988, he was a postdoctoral researcher in the Department of Mathematics, the University of Strathclyde, United Kingdom. He worked as a research fellow in the Information Engineering Department from February 1992 to March 1994, the Center for Environment Studies from April 1995 to August 1995, and the Mechanical Engineering and Automation Department from September 1996 to October 1996, at the Chinese University of Hong Kong. From January 1995 to April 1995, he was a research fellow in the Department of Computing at the Hong Kong Polytechnic University. He has been with the Faculty of Science and Research Center for Applied Mathematics at Xi'an Jiaotong University since 1982, where he was promoted to associate professor in 1987 and full professor in 1991, and now serves as an authorized Ph.D. supervisor in mathematics, dean of the Faculty of Science, and director of the Institute for Information and System Sciences. He has published two monographs and more than 80 academic papers on nonlinear functional analysis, numerical analysis, optimization techniques, neural networks, and genetic algorithms, most of which are in international journals. His current research interests include neural networks, evolutionary computation, and multiple objective decision making theory. Dr. Xu holds the title "Owner of Chinese Ph.D. Degree Having Outstanding Achievements" awarded by the Chinese State Education Commission and the Academic Degree Commission of the Chinese Council in 1991. He is a member of the New York Academy of Sciences and International Mathematicians Union (IMU).