An Integrated Self-Organizing Map for the Traveling Salesman Problem*

Hui-Dong Jin, Kwong-Sak Leung and Man-Leung Wong
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong, P.R. CHINA
{hdjin, ksleung, mlwong}@cse.cuhk.edu.hk, http://www.cse.cuhk.edu.hk/~hdjin

Abstract: - As a representative combinatorial optimization problem, the Traveling Salesman Problem (TSP) has attracted extensive research. In this paper, we develop a new Self-Organizing Map (SOM) network for the TSP and call it the Integrated SOM (ISOM) network. Its learning rule embodies the effective mechanisms of three typical learning rules. In its single learning activity, the excited neuron first is dragged close to the input city, and then is expanded towards the convex-hull of the TSP, and finally, it is drawn close to the middle point of its two neighbor neurons. The elaborate cooperation among these three learning mechanisms is evolved by a genetic algorithm. The simulation results show that the finally established ISOM can generate more promising solutions, with similar computation time, than other neural networks like the SOM network, the Expanded SOM, and the Convex Elastic Net.

Key-Words: - Self-organizing map, Traveling salesman problem, Genetic algorithms, Convex-hull

1 Introduction

The Traveling Salesman Problem (TSP) is one of the typical combinatorial optimization problems. It can be stated as a search for the shortest closed tour that visits each city once and only once. It has many real-life applications like VLSI routing, hole punching and wallpaper cutting problems [1,2]. On the other hand, it falls into a class of the NP-hard or NP-complete problems. Thus, during the past decades, the TSP has attracted extensive research and has repeatedly been used as the basis of comparison for different optimization algorithms, like Genetic Algorithms (GA) [3], tabu search [4], automata networks [5] and neural networks [6–8].

The Self-Organizing Map (SOM) networks, originally proposed by Kohonen, solve the TSP through unsupervised learning [6]. By simply inspecting the input city data for regularities and patterns, and then adjusting itself to fit the input data through cooperative adaptation of the synaptic weights, such a network creates the localized response to the input data, and thus reflects the topological ordering of the input cities. This neighborhood preserving map then results in an expected tour of the TSP under consideration. From each city, the resultant tour tries to visit its nearest city. The shortest subtour can intuitively lead to a good tour for the TSP. Such a property learned by the SOM is referred to as *the local optimality* hereafter.

*This research was partially supported by Hong Kong RGC CERG Grant CUHK 4161/97E.

Due to their acceptable computation complexity and promising performance, the SOM-like networks have attract a great amount of research to explore and enhance its capability on handling the TSP and generated some encouraging results [1,7,8]. There are three main streams to enhance the original SOM network as follows:

1. Using the variable structure network instead of the static structure [1].

2. Amending the competition criterion [9].

3. Enhancing the learning rule. The learning rule in the elastic net, proposed by Durbin and Willshaw [10], is often used to enhance the SOM-like networks [7]. Recently, an expanded learning rule is designed to learn both the local optimality and the global optimality, *the convex-hull property* [9] of the TSP.

These modified learning rules aim to improve the performance of SOM-like networks from different viewpoints and demonstrate more or less success in numerical experiments. In this paper, we integrate these learning rules together and develop a new SOM learning rule, termed as *the Integrated SOM* (ISOM) learning rule. In a single learning activity, the learning rule first follows the traditional learning rule in the SOM to drag the excited neuron close to the input city. Then the excited neuron is pushed towards the convex-hull of the TSP at hand. The pushing force is designed according to certain global features and

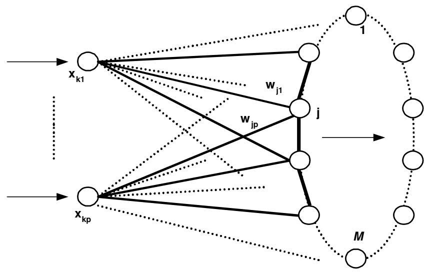

Fig. 1. A schematic SOM-like network for the TSP.

helps the SOM-like network to learn the global optimality. After that, according to the learning mechanism in the elastic net, the excited neuron is drawn close to the middle point of its two nearest neurons to keep the total length of the ring of the neurons as short as possible. We propose the ISOM network in next section. In section 3, the elaborate cooperation of these three learning mechanisms is evolved by GAs. The finally evolved ISOM then is tested and compared with serveral other SOM-like networks to demonstrate its favorable performance in section 4, followed by conclusion in section 5.

2 An Integrated SOM algorithm

We firstly give a brief description of SOM-like networks for the TSP and also outline several typical techniques involved. These pave the way for our Integrated SOM (ISOM) network.

Fig.1 shows a schematic view of a SOM-like network for the TSP. A ring of output neurons, denoted by $1, 2, \cdots, M$, is employed to characterize the feature map. The input neurons, receiving the data of the input city (say, coordinate values), are fully connected to every output neuron. If the state of input neurons at time t is the vector $\overrightarrow{x}_k(t) = [x_{k1}(t), x_{k2}(t), \cdots, x_{kp}(t)]^T \in R^p$, where p is the number of input neurons and usually equals 2, then the synaptic weights between the j-th output neuron and each of the input neurons form the vector $\overrightarrow{w}_j(t) = [w_{j1}(t), w_{j2}(t), \cdots, w_{jp}(t)]^T \in R^p$ $(1 \leq j \leq M)$. Therefore, these output neurons have two topological domains. 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 connection weight vector.

The underlying idea of the SOM-like networks is to construct a map from the high-dimensional connection weight space onto the one-dimensional ring space. In order to get a good tour, the map needs to preserve the neighborhood relations in the sense that the output neurons close to each other on the ring space should be located close to each other on the synaptic weight space. This is accomplished by executing the SOM unsupervised learning on the city data circularly. Normally, the city coordinate is fed to the input layer 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 excited neurons (the winning neuron, as well as its neighbors) update their synaptic weights according to a certain learning rule. The learning process continues until all cities are fed into the network with prespecified times.

Now we focus on the learning rules. The commonly used one is [6]:

$$\overrightarrow{w}_j(t+1) = \overrightarrow{w}_j(t) + \alpha_j(t)(\overrightarrow{x}_k(t) - \overrightarrow{w}_j(t)) \tag{1}$$

where $\alpha_j(t)$ is *a learning rate*, ranging between 0 and 1. Fig. 2(a) illustrates this update mechanism. Usually, the resultant solution tries to visit its nearest neighbor as far as possible. These shortest subtours hopefully leads to an optimal solution. However, such a local property does not always exist in the optimal tours of the TSP.

One way to improve its performance is to embody some global features of the TSP in its learning rule. Here, the global features refer to properties valid to all optimal solutions. The Expanded SOM (ESOM) [9] has taken into account of a global optimality — *the convex-hull property.* It says that, for any optimal tour of the TSP, the cities located on the convex-hull formed by given cities must be visited in the same order as that they appear on the convex-hull. Mathematically, *the expanded learning rule* in the ESOM has a form like:

$$\overrightarrow{w}_j(t+1) = c_j(t)\left[\overrightarrow{w}_j(t) + \alpha_j(t)\left(\overrightarrow{x}_k(t) - \overrightarrow{w}_j(t)\right)\right] \tag{2}$$

where the expanded coefficient $c_j(t)(\geq 1.0)$ is designed to reflect the convex-hull property of the TSP to some degrees. That is, the cities (and its corresponding neurons) nearer the convex-hull exercise more influence on the learning and attract the neuron more. As a result, the ESOM most likely achieves the convex-hull property of the TSP, as well as the local property, then improves its performance dramatically [9]. The schematic functionality of the expanded coefficient is illustrated in Fig. 2(b).

Another renowned learning rule, originally used by the elastic net [10], updates the connection weight vector according to

$$\begin{aligned}\overrightarrow{w}_j(t+1) = {} & \overrightarrow{w}_j(t) + \alpha_j(t)\left[\overrightarrow{x}_k(t) - \overrightarrow{w}_j(t)\right] \\ & + \frac{\beta_j(t)}{2}\left[\overrightarrow{w}_{j-1}(t) + \overrightarrow{w}_{j+1}(t) - 2\overrightarrow{w}_j(t)\right].\end{aligned} \tag{3}$$

where $\beta_j(t)$ is another learning rate parameter. It is worth noting that the last term in the right-hand side of Eq.(3) just reflects the elastic force constraint that forces the length of the resulting ring of neurons

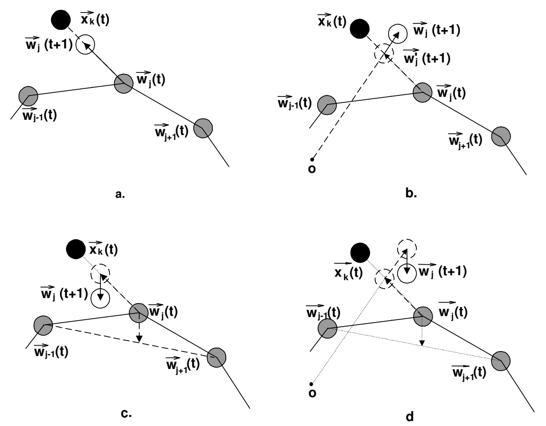

Fig. 2. A schematic view of four different learning rules (a). for the traditional SOM network; (b). for the ESOM algorithm; (c). for the elastic net; (d). for the proposed ISOM, respectively.

to be as short as possible [10]. In addition, this term may inhibit the intersection in the resultant tour. In fact, the last term and the first term intuitively attract the excited neuron to the middle point of its two neighbor neurons on the ring. A schematic view for the updated mechanism is shown in Fig.2(c).

These learning rules can lead to some good tours, which have been substantiated by numerous experiments. Furthermore, these underlying ideas emphasize different aspects. So we can integrate these ideas together and hold all their strengths, and then lead to a more effective SOM-like network. We give the ISOM network as follows.

Algorithm: the ISOM network

1. Map all the given city coordinates $(x'_{i1}, x'_{i2})^T (i = 1, \cdots, n)$ into a circle centered at the origin with radius $R\ (\leq 1)$, where n is the number of the cities. The center of the original cities is mapped onto the origin (the center of the circle). We still use $(x_{i1}, x_{i2})^T$ to denote the coordinate of $\overrightarrow{x}_i$ after transformation.

2. Set $t = 0$, $p = 2$, and the initial weight vectors $\overrightarrow{w}_j(t)$ ($j = 1, \cdots, n$, thus $M = n$) with random values within the circle.

3. Select a city at random, say $\overrightarrow{x}_k(t) = (x_{k1}(t), x_{k2}(t))^T$, and feed it to the input neurons.

4. Find the winning output neuron, say $m(t)$, nearest to $\overrightarrow{x}_k(t)$ according to the Euclidean metric. That is,
$$m(t) = \arg\min_j \|\overrightarrow{x}_k(t) - \overrightarrow{w}_j(t)\|. \qquad (4)$$

5. Train neuron $m(t)$ and its neighbors up to *the effective width* $\sigma(t)$ with the following formula:
$$\begin{aligned}\overrightarrow{w}_j(t+1) &= c_j(t)\left\{\overrightarrow{w}_j(t) + \alpha_j(t)\left[\overrightarrow{x}_k(t) - \overrightarrow{w}_j(t)\right]\right\} \\ &+ \frac{\beta_j(t)}{2}\left[\overrightarrow{w}_{j-1}(t) + \overrightarrow{w}_{j+1}(t) - 2\overrightarrow{w}_j(t)\right] \qquad (5)\end{aligned}$$
where $j = m(t), m(t)\pm 1, \cdots, m(t)\pm\sigma(t)$, and $c_j(t)$ is *the expanded coefficient*. Here, $\alpha_j(t)$ and $\beta_j(t)$ are *the learning rates*, respectively, specified by
$$\alpha_j(t) = \eta_1(t) \times h_{j,m(t)}, \qquad (6)$$
$$\beta_j(t) = \eta_2(t) \times h_{j,m(t)}, \text{ and} \qquad (7)$$
$$h_{j,m(t)} = \begin{cases} 1 - \frac{d_{j,m(t)}}{\sigma(t)+1} & d_{j,m(t)} \leq \sigma(t), \\ 0 & \text{otherwise,} \end{cases} \qquad (8)$$
where $\eta_1(t)$ and $\eta_2(t)$ are two *learning parameters* of the network, $h_{j,m(t)}$ a *neighborhood function* and $d_{j,m(t)}$ the distance between the neuron $m(t)$ and j on the ring.

6. Update the effective width $\sigma(t)$, $\eta_1(t)$ and $\eta_2(t)$ with presetted decreasing schemes, say, decreasing to 0 linearly. And, if the learning loop does not terminate, go to Step 3 with $t := t + 1$.

7. Calculate the activity value of each city $\overrightarrow{x}_k$ according to the formula:
$$\begin{aligned}a(\overrightarrow{x}_k) = m_k - \frac{3}{26}\Big\{&d(\overrightarrow{x}_k, \overrightarrow{w}_{m_k}) + \qquad (9)\\ &\sum_{i=1}^{2}\frac{2\left[d(\overrightarrow{x}_k, \overrightarrow{w}_{m_k+i}) - d(\overrightarrow{x}_k, \overrightarrow{w}_{m_k-i})\right]}{i+2}\Big\},\end{aligned}$$
where m_k is the winning neuron of $\overrightarrow{x}_k$.

8. Order the cities by their activity values, and then form a corresponding tour of the TSP on hand.

Steps 7 and 8 provide a mapping method which realizes how each city is mapped to a real number rather than an integer and then get a tour. This successfully avoids the confusion that several cities are mapped to the same neuron. Furthermore, the mapping method not only exploits the information of the winning neuron but also benefits from those of its nearest neurons.

The learning rule in Eq.(5) is the key point of the proposed ISOM network, which also makes it distinct from all previous SOM-like networks. The excited neuron first is dragged close to the input city according to the learning rule as specified by Eq.(1). Then it is pushed towards the convex-hull of the TSP under consideration and helps the ISOM to learn the convex-hull property, and finally, the neuron is drawn by the elastic force. The last operation helps to avoid intersection in the tour. Its schematic view is illustrated in Fig. 2(d). Apparently, it integrates three learning mechanisms.

Step 1 in the ISOM network mainly facilitates the implementation of the expanded coefficient $c_j(t)$ and makes it possible to reflect the convex-hull property based only on the input city and the excited neuron. For example, after this linear transformation, the distance between the city and the origin, namely the norm of the input vector, is proportional to the distance between the original city and the center of all the cities. Thus, the norm of the input vector can be used to reflect the location of the city. The larger the norm is, the more possibly the city is located on the convex-hull. Similarly, the inner product between the input city and the excited neuron can reflect the global information too. Thus, using the norm and the inner product, we can design some reasonable expanded coefficients to reflect the convex-hull property.

To simplify the design of the expanded learning mechanism, we divide the formula for the expanded coefficient into several terms with distinct functionalities:

$$c_j(t) = [1.0 + b_j(t) \times e_j(t)]^{a_4}\,. \tag{10}$$

Here, the constant 1.0 ensures that the expanded coefficient is close to 1.0 so as to make the learning rule incorporate well with the traditional one as in Eq.(1), and the constant a_4 helps to unify it with the learning rule in the ESOM [9]. The term $b_j(t)$, indicating the relative strength of the expanded force over the learning rate $\alpha_j(t)$, is formulated as:

$$b_j(t) = a_1 \times \alpha_j(t)^{a_2} \times (1 - \alpha_j(t))^{a_3} \tag{11}$$

where a_i $(i = 1, 2, 3)$ are positive real numbers. The term $e_j(t)$ in Eq.(10) aims to reflect the global optimality only in terms of the input city and the excited neuron. As mentioned above, the norm of a city or neuron in the ISOM embodies certain global information. So does their inner products. Based on them, we list some candidate formulae below:

- $e_j(t) = \sum_{i=1}^{2} [\alpha_j(t)x_{ti} + (1 - \alpha_j(t))w_{ji}(t)]^2 - |\sum_{i=1}^{2} x_{ti}w_{ji}(t)|$.
- $e_j(t) = \sum_{i=1}^{2} (x_{ti} - w_{ji}(t))^2 \times \sum_{i=1}^{2} x_{ti}^2$.
- $e_j(t) = \sum_{i=1}^{2} (w_{ji}(t) - x_{ti})\, w_{ji}(t)$.

3 Evolutionary design of ISOM

A well-defined Integrated SOM (ISOM) actually depends upon the elaborate cooperation of three learning mechanisms, choices of realization for the expanded coefficient, and the learning parameter setting as well. It seems impracticable to specify a perfect ISOM network by trial-and-error. However, Genetic Algorithms (GAs) provide us alternatives to design an effective ISOM version [3].

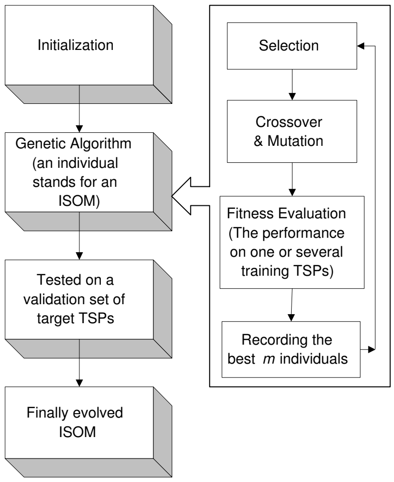

Fig. 3. Evolutionary design of the ISOM.

During the past two decades there has been growing interest in GAs that are based on Darwin's theory of evolution (survival of the fittest) [3]. GAs maintain a population of chromosomes, and manipulate them by several genetic operators. The most significant advantages lie in the gain of flexibility and adaptability to the task on hand, in combination with the robust performance and the global search characteristics. Thus GAs have been employed to handle these inherently hard or time-consuming problems.

Intuitively, we give a GA-based neural-evolutionary system for the proposed ISOM, as shown in Fig.3. After initialization, the GA is used to evolve a set of ISOM algorithms with best performance. The performance here relates the qualities of the generated solutions and their variance on some small-scale TSP problems. The top m different individuals are recorded as candidates. If the GA reaches the prespecified number of generations, the recorded individuals are verified on a validation set of the target TSPs. These problems can be somewhat large scale. At the end, the learning scheme with the best performance on these validation problems is selected as the finally evolved ISOM.

Our implementation of the GA is derived from the C++ library of Genetic Algorithms (GAlib) from http://lancet.mit.edu/ga/. In the implementation of the GA, an individual (or chromosome) represents a learning scheme. It encodes the type of formula to calculate the expanded coefficient, and its parameters a_i $(i = 1, \cdots, 4)$. It also includes other learning parameters in the ISOM network like the radius R, the total training loop L, the initial values and the de-

Alleles in chromosomes	Domains	eISOM
Formula for $e_j(t)$	{1,2,...,20}	1
Parameters for $e_j(t)$:		
a_1, a_2 and a_3	{0,0.25,...,5}	1,3,0.25
a_4	{0.2,0.4,...,5}	1.0
Radius R	(0.001,1.0)	0.61
Training loop L	{50,65,...,200}	160
Learning parameter $\eta_1(0)$	(0.001,1.0)	0.95
Learning parameter $\eta_2(0)$	(0.001,1.0)	0.12
$\eta_2(t)$ decreasing mode (be 0 after p_1 percent iterations)	p_1: (0,100)	48
The effective width	a {1,2, ...,14}	10
$\sigma(0) = a + b \cdot n$	b [0.001,0.6]	0.01
$\sigma(t)$ decreasing mode (be 1 after p_2 percent iterations)	p_2: (0,100)	62

Table 1: Alleles of a chromosome in the neural-evolutionary system, their domains and the parameter setting in the finally evolved ISOM (eISOM). The learning parameters $\eta_1(t)$, $\eta_2(t)$ and $\sigma(t)$ are decreased linearly for each learning iteration. $\eta_1(t)$ reaches zero at the last iteration.

creasing scheme of the effective learning width $\sigma(t)$, and the learning parameters $\eta_1(t)$ and $\eta_2(t)$. So, according to the proposed ISOM network given in Section 2, an individual actually determines a concrete ISOM network. In the chromosome, one argument is coded by one allele. In order to search for the parameter space efficiently, these alleles are limited to certain reasonable domains. The alleles and their domains are listed in Table 1 (Columns 1 and 2).

The fitness function is defined as 3 minus the mean of *the solution quality* of the solutions generated by an ISOM network and their average variance on target TSPs. The solution quality refers to the relative difference of the average tour length over the theoretic lower bound for the random TSPs, i.e., $0.765 \times \sqrt{n}$ [9].

The finally evolved ISOM[2] consists of the parameter setting or choices, as listed in the last column of Table 1. The expanded coefficient is specified by:

$$c_j(t) = 1 + \alpha_j(t)^3 \left(1 - \alpha_j(t)\right)^{0.25} \{\sum_{i=1}^{2} [\alpha_j(t) x_{ki}(t) + (1 - \alpha_j(t)) w_{ji}(t)]^2 - |\sum_{i=1}^{2} x_{ki}(t) w_{ji}(t)|\}.$$

4 Performance of the evolved ISOM

Using the parameter setting in the previous section, the computation complexity of the evolved ISOM is $O(n^2)$. It similar with ESOM [9] and the SOM developed by Budinich [8]. It is worth noting that almost all non-neural network methods, like [3,4], have higher order complexity in order to get solutions comparable with the SOM-like networks. The established ISOM is examined on two sets of TSPs. The simulation results are based on 10 independently runs for each TSP on a Sun Ultra 5/270 workstation.

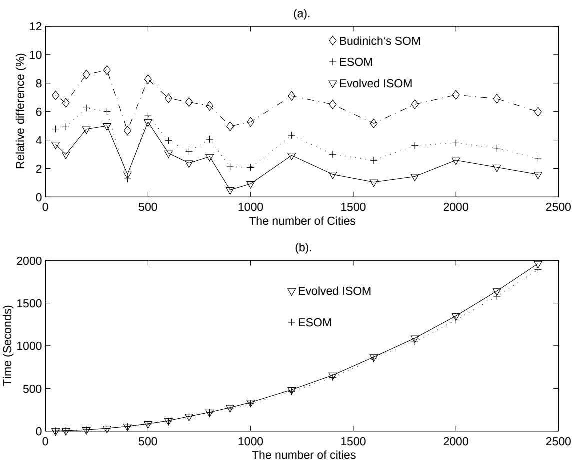

Fig. 4. (a) Comparison of the average solution quality yielded by the evolved ISOM, the ESOM and the Budinich's SOM network on 18 random TSPs; (b) The average execution time comparison between the ESOM and the evolved ISOM.

The first set of experiments conducted are based on a set of 18 TSPs ranging from 50 to 2400 cities. These TSPs are all generated randomly within the unit square. Fig.4(a) shows the comparison results in terms of the relative differences of the average tour lengths obtained by the networks from the theoretical lower bounds $0.765 \times \sqrt{n}$. It is seen from the figure that, for the 18 TSPs, the tours obtained by the evolved ISOM are shorter than the ones by the ESOM except for the TSP with 400 cities. We find that, averagely, the relative differences from the theoretical lower bounds are 6.67%, 3.76%, and 2.63% for the Budinich's SOM, the ESOM, and the evolved ISOM, respectively. That is, the evolved ISOM has made around 1.13% improvement over the ESOM, around 4.04% over the Budinich's SOM. From Fig.4(b), it is easily observed that the two SOM-like networks almost take the same amount of execution time for the 18 TSPs. Similar comparison conclusions can be drawn from the other set of experiments below. Thus we may say that the evolved ISOM generates better tours than the ESOM and the Budinich's SOM with similar execution time.

Our second set of experiments are designed to compare the evolved ISOM with the Convex Elastic Net (CEN) of Al-Malhem and Al-Maghrabi [7], the ESOM [9] and their enhanced versions. Here, an enhanced version of a SOM-like net-

[2] The finally evolved ISOM algorithm, as well as all random TSPs used below, can be downloaded from http://www.cse.cuhk.edu.hk/~hdjin/som/.

TSP name	Solution quality of plain SOMs (%)		Solution quality of enhanced SOMs (%)		
	ESOM	eISOM	CEN	ESOM	eISOM
GR96	1.0	0.8	4.4	0.5	0.5
KROA100	1.0	0.6	1.6	0.8	0.5
GR137	4.3	3.2	3.2	2.5	2.2
Lin318	4.1	2.1	4.7	2.9	2.0
Average	2.60	1.68	3.48	1.68	1.30

Table 2: Comparisons among the enhanced CEN, the ESOM and the evolved ISOM (eISOM) when applied to 4 TSP benchmarks.

work means that the network has incorporated with the heuristics NII trick presented by Al-Mulhem and Al-Maghrabi in [7]. Since the simulation results of the CEN reported in [7] have been enhanced by the local heuristics NII, the other SOM-like networks have also been enhanced to make the comparison fair. All the tested TSPs are taken from http://www.iwr.uni-heidelberg.de /iwr/comopt/software/TSPLIB95/tsp/. Table 2 lists the simulation and comparison results of the plain and enhanced SOM-like networks. The results are the relative differences of the best tour lengths obtained by the networks from the optimal tour length. The simulation results of the enhanced CEN are all quoted directly from [7]. Observing from Table 2, for both plain and enhanced versions, the evolved ISOM always yields higher quality solutions of TSPs than the Budinich's SOM, the ESOM, and the enhanced CEN do. Note that the CEN takes $O(n^2)$ computation time in its training process too. We therefore conclude that the evolved ISOM network outperforms the CEN.

Based on the complexity analysis and the comparison results, we can conclude that the evolved ISOM has better performance than ESOM, the Budinich's SOM, the CEN in terms of both accuracy and speed.

5 Conclusion

In this paper, we have developed the Integrated Self-Organizing Map (ISOM) network, a new Self-Organizing Map (SOM) network for the TSP. Its learning rule has integrated three effective mechanisms of several learning rules in the SOM-like network literature. That is, during the learning, it takes account of the local optimality of the traditional SOM network, the global optimality extracted by the Expanded SOM (ESOM) and the elastic force constraint in the elastic net simultaneously. We also have given several possible realizations for the expanded coefficient in its learning rule to reflect the global optimality of the TSP to some degrees. The elaborate synthesis of these three mechanisms, as well as the parameter setting involved are determined by a genetic algorithm automatically. The finally evolved ISOM algorithm has been examined on a wide spectrum of TSPs. Its performance is better, in terms of accuracy and speed, than other neural networks including the SOM developed by Budinich [8], the ESOM [9], and the Convex Elastic Net (CEN) [7]. The idea, combining the strengths of several methods on learning, could hopefully be used to handle other problems efficiently, too.

References:

[1] K. A. Smith, "Neural networks for combinatorial optimization: A review of more than a decade of research," *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 15–34, 1999.

[2] G. Reinelt, "Fast heuristic for large geometric traveling salesman problems," Tech. Rep. 185, Institut für Mathematick, Universotät Augsburg, 1989.

[3] David E. Goldberg, *Genetic algorithms in search optimization, and machine learning*, Addison-Wesley, 1989.

[4] J. Knox, "Tabu search performance on the symmetric traveling salesman problem," *Computers and Operations Research*, vol. 21, pp. 867–876, 1994.

[5] Z. B. Xu, H. D. Jin, K.S. Leung, Y. Lee, and C. K. Wong, "An automata network for performing combinatorial optimization," Accepted by Neurocomputing, Nov. 2000.

[6] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 2, pp. 59–69, 1982.

[7] H. Al-Mulhem and T. Al-Maghrabi, "Efficient convex-elastic net algorithm to solve the Euclidean traveling salesman problem," *IEEE Transactions on Systems, Man and Cybernetics — Part B: Cybernetics*, vol. 28, pp. 618–620, 1998.

[8] M. Budinich, "A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing," *Neural Computation*, vol. 8, pp. 416–424, 1996.

[9] K. S. Leung, H. D. Jin, and Z. B. Xu, "An improved self-organizing neural network for the traveling salesman problem," Resubmited to IEEE Transactions on Neural Networks, Aug. 2000.

[10] R. Durbin and D. Willshaw, "An analogue approach to the traveling salesman problem," *Nature(London)*, vol. 326, pp. 689–691, 1987.