11

Recurrent Neural Networks with the Soft 'Winner Takes All' Principle Applied to the Traveling Salesman Problem

Paulo Henrique Siqueira, Maria Teresinha Arns Steiner and Sérgio Scheer
Federal University of Paraná
Curitiba, Paraná,
Brazil

1. Introduction

This work shows the application of Wang's Recurrent Neural Network with the "Winner Takes All" (WTA) principle to solve the classic Operational Research problem called the Traveling Salesman Problem. The upgrade version proposed in this work for the 'Winner Takes All' principle is called soft, because the winning neuron is updated with only part of the activation values of the other competing neurons.

The problems in the TSPLIB (Traveling Salesman Problem Library - Reinelt, 1991) were used to compare the soft version with the 'Winner Takes All' hard version and they show improvement in the results using the 'Winner Takes All' soft version in most of the problems tested.

The implementation of the technique proposed in this paper uses the parameters of Wang's Neural Network for the Assignment problem (Wang, 1992; Hung & Wang, 2003) using the 'Winner Takes All' principle to form Hamiltonian circuits (Siqueira *et al*. 2007) and can be used for both symmetric and asymmetric Traveling Salesman Problems. The 2-opt technique is used to improve the routes found with the proposed Neural Network, thus becoming a technique that is competitive to other Neural Networks.

Other heuristic techniques have been developed recently to solve the Traveling Salesman Problem, and the work of Misevičius *et al*. (2005) shows the use of the ITS (Iterated Tabu Search) technique with a combination of intensification and diversification of solutions for the TSP. This technique is combined with the 5-opt and errors are almost zero in almost all of the TSPLIB problems tested.

The work of Wang *et al*. (2007) shows the use of Particle Swarm to solve the TSP with the use of the fraction (quantum) principle to better guide the search for solutions. The authors make comparisons with Hill Climbing, Simulated Annealing and Tabu Search, and show in a 14-cities case that the results are better than those of the other techniques.

In the area of Artificial Neural Networks, an interesting technique can be found in the work of Massutti & Castro (2009), who use a mechanism to stabilize winning neurons and the centroids of the groups of cities for the growing and pruning the network. The authors show modifications in the Rabnet's (real-valued antibody network) parameters for the Traveling Salesman Problem and comparisons made with TSPLIB problems solved by other techniques show that the Rabnet has better results.

Créput & Koukam (2007) show a hybrid technique with self-organizing maps (SOM) and evolutionary algorithms to solve the TSP, called Memetic Neural Network (MSOM). The results of this technique are compared with the CAN (Co-Adaptive Network) technique, developed by Cochrane & Beasley (2003), in which both have results that are regarded as satisfactory.

The Efficient and Integrated Self-Organizing Map (EISOM) was proposed by Jin *et al*. (2003), where a SOM network is used to generate a solution where the winning neuron is replaced by the position of the midpoint between the two closest neighboring neurons. The results presented by the authors show that the EISOM has better results than the Simulated Annealing and the ESOM network (Leung *et al*., 2004).

The Modified Growing Ring Self-Organizing Map (MGSOM) presented in the work of Bai *et al*. (2006) shows some changes in the adaptation, selection of the number of neurons, the network's initial weights and the winning neurons indexing functions, as well as the effects of these changes on the order of cities for the TSP. The MGSOM technique is easy to implement and has a mean error of 2.32% for the 12 instances of the TSPLIB.

The work of Yi *et al*. (2009) shows an elastic network with the introduction of temporal parameters, helping neurons in the motion towards the cities' positions. Comparisons with problems from the TSPLIB solved with the traditional elastic network show that it is an efficient technique to solve the TSP, with smaller error and less computational time than the other elastic networks.

In Li *et al*. (2009) a class of Lotka-Volterra neural networks is used to solve the Traveling Salesman Problem with the application of global inhibitions, analyzing the stability of the proposed neural network by means of equilibrium points. The results are analyzed and compared with several experiments where the equilibrium status of this network represents a feasible solution to the Traveling Salesman Problem.

The work of Hammer *et al*. (2009) shows a review of the most recent works in the area of Recurrent Neural Networks, including discussions about new paradigms, architectures and processing structures of these networks. The authors show the works of Recurrent Neural Networks applied in solving various Operational Research problems, such as the Traveling Salesman Problem, Quadratic Programming problems, training of Support Vector Machines and Winner Takes All.

This work is divided into four sections besides this introduction. In section 2 are shown Wang's Recurrent Neural Network and the soft 'Winner Takes All' technique applied to the Traveling Salesman Problem. Section 3 shows the comparative results and Section 4 the conclusions.

2. Wang's neural network with the soft winner takes all principle

The mathematical formulation of the Traveling Salesman Problem is the same of the Assignment problem (1) - (4), with the additional constraint (5) that ensures that the route starts and ends in the same city.

$$\text{Minimize } C = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1}$$

$$\text{Subject to } \sum_{i=1}^{n} x_{ij} = 1\,,\; j = 1,\ldots, n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ , } i = 1,..., n \tag{3}$$

$$x_{ij} \in \{0, 1\}, i, j = 1,\ldots, n \tag{4}$$

$$\tilde{x} \text{ forms a Hamiltonian circuit,} \tag{5}$$

where c_{ij} and x_{ij} are respectively the cost and decision variables associated with the assignment of vertex i to vertex j in the Hamiltonian circuit. The objective function (1) minimizes costs. The set of constraints (2) and (3) ensure that each city will be visited only once. Constraints (4) ensure the condition of integrality of the binary variables x_{ij}, and vector $\tilde{x}$ represents the sequence of a Traveling Salesman's route.

To obtain a first approximation for the TSP, Wang's Recurrent Neural Network is applied to the Assignment Problem, this is, the solution satisfies constraints (1) - (4), which can be written in matrix form (Hung & Wang, 2003):

$$\text{Minimize } C = c^{T}x \tag{6}$$

$$\text{Subject to } Ax = b \tag{7}$$

$$x_{ij} \in \{0, 1\}, i, j = 1,\ldots, n \tag{8}$$

where c^{T} is the vector with dimension n^2, which contains all of the cost matrix's lines in sequence, vector x contains the n^2 x_{ij} decision variables and vector b contains the number 1 in all of its positions. Matrix A has dimension $2n \times n^2$, with the following format:

$$A = \begin{bmatrix} I & I & \ldots & I \\ B_1 & B_2 & \ldots & B_n \end{bmatrix}$$

where I is the identity matrix with dimension n and each matrix B_i contains zeroes in all of its positions, with exception of the i-th line that contains "1" in all of its positions.

A Wang's Recurrent Neural Network is defined by the following differential equation (Wang, 1992; Hung & Wang, 2003):

$$\frac{du_{ij}(t)}{dt} = -\eta \sum_{k=1}^{n} x_{ik}(t) - \eta \sum_{l=1}^{n} x_{lj}(t) + \eta\theta_{ij} - \lambda c_{ij} e^{-\frac{t}{\tau}} \tag{9}$$

where x_{ij} = g (u_{ij} βt)), the equilibrium status of this network is a solution to the Assignment Problem (Wang, 1997) and function g is the sigmoid function with parameter β:

$$g(u) = \frac{1}{1 + e^{-\beta u}} \,. \tag{10}$$

The threshold is the vector of size n^2 $\theta = A^{T}b$, which has the value "2" in all of its positions. Parameters η, λ and τ are constant and empirically chosen (Hung & Wang, 2003). Parameter η penalizes violations of constraints (2) and (3). Parameters λ and τ control the minimization of the objective function (1). Considering $W = A^{T}A$, Wang's Neural Network's matrix form is the following:

$$\frac{du(t)}{dt} = -\eta(Wx(t) - \theta) - \lambda c e^{-\frac{t}{\tau}}, \tag{11}$$

The method proposed in this paper uses the "Winner Takes All" principle, which accelerates the convergence of Wang's Recurrent Neural Network, in addition to solve problems that appear in multiple solutions or very close solutions (Siqueira *et al.*, 2008).

The adjustment of parameter λ was done using the standard deviation between the coefficients of the rows in the problem's costs matrix and determining the vector

$$\bar{\lambda} = \eta\left(\frac{1}{\delta_1}, \frac{1}{\delta_2}, \ldots, \frac{1}{\delta_n}\right), \tag{12}$$

where δ_i is the standard deviation of row i of costs matrix c (Smith *et al.*, 2007).

The adjustment of parameter τ uses the third term of the definition of Wang's Neural Network (9), as follows: when $c_{ij} = c_{\max}$, term $-\lambda_i c_{ij} \exp(-t/\tau_i) = k_i$ must satisfy $g(k_i) \cong 0$, this is, x_{ij} will bear the minimum value (Siqueira *et al.*, 2007), considering $c_{ij} = c_{\max}$ and $\lambda_i = 1/\delta_i$, where $i = 1, \ldots, n$, τ is defined by:

$$\tau_i = \frac{-t}{\ln\left(\dfrac{-k_i}{\lambda_i c_{\max}}\right)}. \tag{13}$$

After a certain number of iterations, term $Wx(t) - \theta$ of equation (10) has no substantial alterations, thus ensuring that constraints (2) and (3) are almost satisfied and the 'Winner Takes All' method can be applied to establish a solution for the TSP.

The soft 'Winner Takes All' (SWTA) technique is described in the pseudo-code below, where the following situations occur with respect to parameter α:

- when $\alpha = 0$, the WTA update is nonexistent and Wang's Neural Network updates the solutions for the Assignment Problem with no interference;
- when $\alpha = 1$, the update is called hard WTA, because the winner gets all the activation of the other neurons and the losers become null. The solution is feasible for the TSP;
- in the other cases, the update is called soft WTA and the best results are found with $0.25 \leq \alpha \leq 0.9$.

Pseudo-code for the Soft 'Winner Takes All' (SWTA) technique

Choose the maximum number of routes r_{max}.

{

While $r < r_{max}$

{

While $Wx(t) - \theta > \phi$ *(where* $0 \leq \phi \leq 2$*):*

Find a solution x for the Assignment Problem using Wang's Neural Network.

}

Make $\bar{x} = x$ *and* $m = 1$;

Choose a line k from decision matrix $\bar{x}$;

Make $p = k$ *and* $\tilde{x}(m) = k$;

{

While $m < n$:

Find $\bar{x}_{kl}$ = argmax{ $\bar{x}_{ki}$, i = 1, …, n};

Do the following updates:

$$\bar{x}_{kl} = \bar{x}_{kl} + \frac{\alpha}{2}\left(\sum_{i=1}^{n} x_{il} + \sum_{j=1}^{n} x_{kj} \right) \quad (14)$$

$$\bar{x}_{kj} = (1-\alpha)\bar{x}_{kj}\ ,\ j = 1,\ldots,\ n,\ j \neq l,\ 0 \leq \alpha \leq 1 \quad (15)$$

$$\bar{x}_{il} = (1-\alpha)\bar{x}_{il}\ ,\ i = 1,\ldots,\ n,\ i \neq k,\ 0 \leq \alpha \leq 1 \quad (16)$$

Make $\tilde{x}$ $(m + 1) = l$ *and* $m = m + 1$;

to continue the route, make $k = l$.

}

Make $\bar{x}_{kp} = \bar{x}_{kp} + \frac{\alpha}{2}\left(\sum_{i=1}^{n} x_{ip} + \sum_{j=1}^{n} x_{kj} \right)$ *and* $\tilde{x}$ $(n + 1) = p$;

determine the cost of route C;

{

If $C < C_{min}$, *then*

Make $C_{min} = C$ *and* $x = \bar{x}$.

}

$r = r + 1$.

}

2.1 Example illustrating the SWTA technique applied to the TSP

Consider the 10-cities problem proposed in the work of Hopfield & Tank (1985). Considering α = 0.7 and the parameters defined by equations (12) and (13), after 32 iterations Wang's Neural Network shows the following solution for the Assignment Problem:

$$\bar{x} = \begin{pmatrix} 0 & 0.017 & \overline{\underline{0.914}} & 0.028 & 0.004 & 0.001 & 0.015 & 0.002 & 0.007 & 0.015 \\ 0.02 & 0 & 0.077 & 0.001 & 0 & 0 & 0 & 0 & 0.002 & 0.801 \\ 0.018 & 0.977 & 0 & 0.003 & 0 & 0 & 0 & 0 & 0.001 & 0.001 \\ 0.919 & 0.001 & 0.002 & 0 & 0.065 & 0.001 & 0.001 & 0 & 0 & 0 \\ 0.007 & 0 & 0 & 1.038 & 0 & 0.022 & 0.007 & 0.001 & 0 & 0 \\ 0.003 & 0 & 0 & 0.001 & 0.021 & 0 & 0.94 & 0.033 & 0.002 & 0.002 \\ 0.014 & 0 & 0 & 0.001 & 0.81 & 0.045 & 0 & 0.027 & 0.007 & 0.007 \\ 0.003 & 0 & 0 & 0 & 0.001 & 0.971 & 0.026 & 0 & 0.026 & 0.016 \\ 0.007 & 0.001 & 0.001 & 0 & 0 & 0.002 & 0.007 & 0.886 & 0 & 0.055 \\ 0.016 & 0.003 & 0.001 & 0 & 0 & 0.002 & 0.009 & 0.019 & 1.069 & 0 \end{pmatrix}$$

A city must be chosen so that the TSP's route can be formed, in this case city 1, this is, $p = k$ = 1. Element l = 3 satisfies $\bar{x}_{kl} = \bar{x}_{13}$ = argmax{ $\bar{x}_{1i}$, i = 1, …, n}, this is, the traveling salesman makes his route leaving city 1 towards city 3. Using equations (14)-(16), the elements in line 1 and column 3 are updated resulting in matrix $\bar{x}$:

$$\bar{x} = \begin{pmatrix} 0 & 0.005 & \overline{\underline{1.294}} & 0.008 & 0.001 & 0 & 0.005 & 0.001 & 0.002 & 0.004 \\ 0.02 & 0 & 0.023 & 0.001 & 0 & 0 & 0 & 0 & 0.002 & 0.801 \\ 0.018 & \overline{\underline{0.977}} & 0 & 0.003 & 0 & 0 & 0 & 0 & 0.001 & 0.001 \\ 0.919 & 0.001 & 0.001 & 0 & 0.065 & 0.001 & 0.001 & 0 & 0 & 0 \\ 0.007 & 0 & 0 & 1.038 & 0 & 0.022 & 0.007 & 0.001 & 0 & 0 \\ 0.003 & 0 & 0 & 0.001 & 0.021 & 0 & 0.94 & 0.033 & 0.002 & 0.002 \\ 0.014 & 0 & 0 & 0.001 & 0.81 & 0.045 & 0 & 0.027 & 0.007 & 0.007 \\ 0.003 & 0 & 0 & 0 & 0.001 & 0.971 & 0.026 & 0 & 0.026 & 0.016 \\ 0.007 & 0.001 & 0 & 0 & 0 & 0.002 & 0.007 & 0.886 & 0 & 0.055 \\ 0.016 & 0.003 & 0 & 0 & 0 & 0.002 & 0.009 & 0.019 & 1.069 & 0 \end{pmatrix}$$

In order to continue the route, update $k = l = 3$ is done and element $l = 2$ satisfies the condition in which $\bar{x}_{kl} = \bar{x}_{32} = \operatorname{argmax}\{\bar{x}_{3i}, i = 1, \ldots, n\}$. Proceeding this way, we obtain the matrix $\bar{x}$ in the form:

$$\bar{x} = \begin{pmatrix} 0 & 0.005 & \overline{\underline{1.294}} & 0.008 & 0.001 & 0 & 0.005 & 0.001 & 0.002 & 0.004 \\ 0.006 & 0 & 0.023 & 0 & 0 & 0 & 0 & 0 & 0.002 & \overline{\underline{0.87}} \\ 0.005 & \overline{\underline{0.993}} & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ \overline{\underline{1.296}} & 0 & 0.001 & 0 & 0.019 & 0 & 0 & 0 & 0 & 0 \\ 0.002 & 0 & 0 & \overline{\underline{1.064}} & 0 & 0.007 & 0.002 & 0 & 0 & 0 \\ 0.001 & 0 & 0 & 0 & 0.006 & 0 & \overline{\underline{0.984}} & 0.01 & 0.001 & 0.001 \\ 0.004 & 0 & 0 & 0 & \overline{\underline{0.877}} & 0.013 & 0 & 0.008 & 0.002 & 0.002 \\ 0.001 & 0 & 0 & 0 & 0 & \overline{\underline{1.022}} & 0.008 & 0 & 0.008 & 0.005 \\ 0.002 & 0 & 0 & 0 & 0 & 0.001 & 0.002 & \overline{\underline{0.941}} & 0 & 0.017 \\ 0.005 & 0.001 & 0 & 0 & 0 & 0.001 & 0.003 & 0.006 & \overline{\underline{1.476}} & 0 \end{pmatrix},$$

which is the solution in Fig. 1, at a cost of 2.7518 and a mean error of 2.27%.

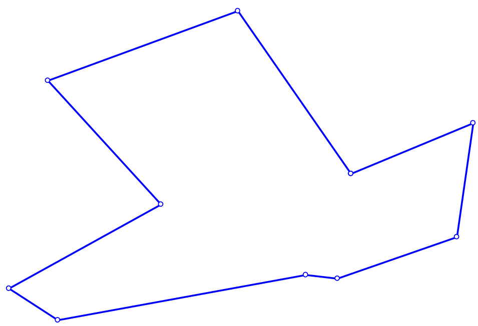


Fig. 1. Solution for the 10-cities problem of Hopfield & Tank, with cost 2.7518 and mean error 2.27%

The solution to the SWTA is once again applied to Wang's Neural Network and after other 13 iterations the following solution is found:

$$\overline{x} = \begin{pmatrix} 0 & 0.017 & 0.023 & 0.92 & 0.004 & 0.001 & 0.015 & 0.002 & 0.007 & 0.015 \\ 0.02 & 0 & 1.071 & 0.001 & 0 & 0 & 0 & 0 & 0.002 & 0.003 \\ 0.879 & 0.072 & 0 & 0.003 & 0 & 0 & 0 & 0 & 0.001 & 0.001 \\ 0.027 & 0.001 & 0.002 & 0 & 0.981 & 0.001 & 0.001 & 0 & 0 & 0 \\ 0.007 & 0 & 0 & 0.067 & 0 & 0.896 & 0.007 & 0.001 & 0 & 0 \\ 0.003 & 0 & 0 & 0.001 & 0.022 & 0 & 0.939 & 0.033 & 0.002 & 0.002 \\ 0.014 & 0 & 0 & 0.001 & 0.006 & 0.045 & 0 & 0.945 & 0.007 & 0.008 \\ 0.003 & 0 & 0 & 0 & 0.001 & 0.036 & 0.026 & 0 & 0.903 & 0.016 \\ 0.007 & 0.001 & 0.001 & 0 & 0 & 0.002 & 0.007 & 0.025 & 0 & 0.978 \\ 0.015 & 0.806 & 0.001 & 0 & 0 & 0 & 0.009 & 0.019 & 0.061 & 0 \end{pmatrix}$$

Using the SWTA technique an approximation to the optimal solution is found:

$$\overline{x} = \begin{pmatrix} 0 & 0.005 & 0.007 & \underline{1.298} & 0.001 & 0 & 0.005 & 0.001 & 0.002 & 0.005 \\ 0.006 & 0 & \underline{1.091} & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.001 \\ \underline{1.247} & 0.022 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.008 & 0 & 0.001 & 0 & \underline{1.004} & 0 & 0 & 0 & 0 & 0 \\ 0.002 & 0 & 0 & 0.02 & 0 & \underline{0.955} & 0.002 & 0 & 0 & 0 \\ 0.001 & 0 & 0 & 0 & 0.007 & 0 & \underline{0.983} & 0.011 & 0.001 & 0.001 \\ 0.004 & 0 & 0 & 0 & 0.002 & 0.014 & 0 & \underline{1.001} & 0.002 & 0.002 \\ 0.001 & 0 & 0 & 0 & 0 & 0.011 & 0.008 & 0 & \underline{0.96} & 0.005 \\ 0.002 & 0 & 0 & 0 & 0 & 0.001 & 0.002 & 0.008 & 0 & \underline{1.01} \\ 0.005 & \underline{1.158} & 0 & 0 & 0 & 0 & 0.001 & 0.006 & 0.018 & 0 \end{pmatrix},$$

which is shown in Fig. 2.

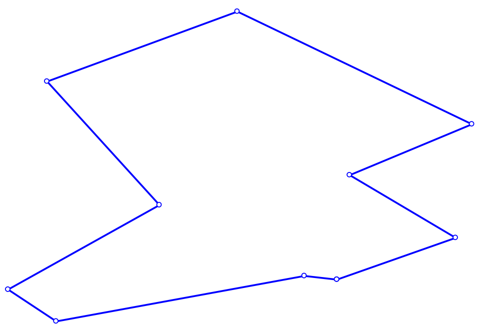
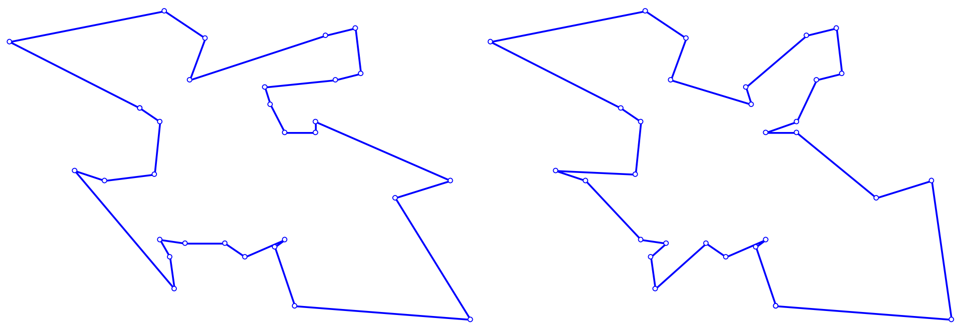


Fig. 2. Optimal solution for the 10-cities problem of Hopfield & Tank, cost 2.69

Applying the same SWTA technique to the 30-random-cities problem of Lin & Kernighan (1973), the solutions are shown below, where Fig. 3a shows the solution with cost 4.37 and 2.58% mean error, and Fig. 3b shows the optimal solution with cost 4.26



(a) (b)

Fig. 3. Solutions for the 10-random-cities problem of Hopfield & Tank: (a) cost 4.37 and 2.58% mean error; (b) optimal solution with cost 4.26

3. Results

The technique proposed in this paper, Soft Winner Takes All applied to Wang's Recurrent Neural Network, was used to solve 36 symmetric and 19 asymmetric problems from the TSPLIB (Traveling Salesman Problem Library) database. The results were compared with results from both the Hard Winner Takes All technique (Siqueira *et al.*, 2008) and from similar techniques published in the literature.

3.1 TSPLIB symmetric problems

Table 1 and Table 2 shows the comparison between the Soft and Hard WTA techniques, where the results of applying Wang's Neural Network with Soft WTA and the 2-opt (SWTA2) route improving technique in the final solutions have mean error ranging between 0 and 4.50%. The results without the application of the 2-opt (SWTA) vary between 0 and 14.39% and are better in almost all problems tested when compared to results obtained with the Hard WTA technique, and 95% confidence intervals (CI) for the mean were computed on the basis of standard deviations (SD) over 60 runs. Fig. 4 shows a comparison between the Soft and Hard WTA techniques applied to 36 problems from the TSPLIB, showing the best and worst results found with each technique.

Table 2 shows that from the 36 problems tested, only 6 have better results with the Hard WTA with 2-opt technique: att532, kroA200, u159, kroA150, pr124 and rd100. The best solutions with the Hard WTA technique without the 2-opt technique outweigh the results with the Soft WTA technique in only 3 problems: lin318, u159 and pr124.

Fig. 4 shows that from the 36 problems tested, 14 have the worst results with higher costs with the Soft WTA technique than with the Hard WTA: rd100, a280, ch130, bier127, kroA100, kroC100, kroE100, brazil58, pr107, eil51, gil262, lin318, fl417 and rat575.

In addition to comparing the technique proposed in this paper with the results of the TSPLIB with the Hard WTA technique, the results were compared with the following techniques:

- KNIESG method (Kohonen Network Incorporating Explicit Statistics Global) which uses statistical methods to determine the weights of neurons in a Self-Organizing Map, where all cities are used for the dispersion of neurons (Aras *et al.*, 1999);

TSP name	n	Optimal solution	Average error (%)							
			HWTA				SWTA			
			Best	Mean	SD	CI (95%)	Best	Mean	SD	CI (95%)
eil51	51	430	**1.16**	1.16	0.00	[1.16, 1.16]	0.47	1.59	0.78	[1.40, 1.79]
brazil58	58	16156	2.90	2.90	0.00	[2.90, 2.90]	**1.81**	2.65	1.18	[2.35, 2.94]
st70	70	678.6	2.71	3.55	0.64	[3.39, 3.71]	**1.68**	1.97	0.46	[1.86, 2.09]
eil76	76	545.4	1.03	2.00	0.76	[1.81, 2.20]	**0**	1.27	0.98	[1.02, 1.51]
kroA100	100	21282	3.68	4.22	0.58	[4.07, 4.36]	**3.05**	3.73	0.96	[3.49, 3.98]
kroB100	100	22141	8.27	8.54	0.25	[8.48, 8.60]	**4.73**	6.12	1.12	[5.83, 6.40]
kroC100	100	20749	5.20	5.20	0.01	[5.20, 5.21]	**3.35**	4.10	1.13	[3.81, 4.38]
kroD100	100	21294	8.57	8.85	0.47	[8.73, 8.97]	**4.64**	4.73	0.11	[4.71, 4.76]
kroE100	100	22068	6.18	6.37	0.25	[6.31, 6.44]	**4.07**	5.41	1.29	[5.09, 5.74]
rd100	100	7910	6.83	7.00	0.19	[6.95, 7.04]	**6.27**	7.02	1.28	[6.70, 7.34]
eil101	101	629	**3.02**	6.09	2.40	[5.49, 6.70]	**3.02**	5.86	1.32	[5.54, 6.20]
lin105	105	14383	4.33	5.41	0.83	[5.20, 5.62]	**3.70**	3.91	0.24	[3.85, 3.97]
pr107	107	44303	3.14	3.14	0.00	[3.14, 3.14]	**1.65**	2.89	0.77	[2.69, 3.08]
pr124	124	59030	**0.33**	1.44	1.22	[1.13, 1.75]	2.39	2.77	0.30	[2.69, 2.84]
bier127	127	118282	4.22	4.45	0.37	[4.35, 4.54]	**3.11**	5.14	1.61	[4.73, 5.55]
ch130	130	6110	5.06	5.97	0.83	[5.76, 6.18]	**4.52**	5.99	1.37	[5.64, 6.33]
pr136	136	96772	5.99	6.28	0.45	[6.16, 6.39]	**5.06**	5.66	0.55	[5.52, 5.80]
gr137	137	69853	9.09	9.14	0.13	[9.11, 9.18]	**6.65**	8.01	0.99	[7.76, 8.26]
kroA150	150	26524	8.85	9.53	0.98	[9.28, 9.78]	**8.50**	8.87	0.50	[8.74, 9.00]
kroB150	150	26130	7.33	8.43	0.92	[8.20, 8.67]	**6.80**	7.31	0.52	[7.17, 7.44]
pr152	152	73682	3.23	3.26	0.02	[3.25, 3.26]	**3.22**	3.23	0.00	[3.23, 3.23]
u159	159	42080	**6.33**	7.16	1.29	[6.84, 7.49]	6.40	7.57	1.07	[7.30, 7.84]
rat195	195	2323	5.55	6.63	1.37	[6.29, 6.98]	**5.42**	5.90	0.35	[5.81, 5.99]
d198	198	15780	10.43	10.75	0.31	[10.68, 10.83]	**6.86**	7.33	0.57	[7.18, 7.47]
kroA200	200	29368	8.95	10.57	1.42	[10.21, 10.93]	**8.03**	8.84	0.91	[8.61, 9.07]
tsp225	225	3859	7.64	8.40	0.92	[8.16, 8.63]	**5.73**	7.25	1.56	[6.86, 7.65]
gil262	262	2378	8.20	8.73	0.58	[8.58, 8.87]	**7.65**	8.33	0.86	[8.11, 8.55]
a280	280	2586	12.14	12.22	0.12	[12.19, 12.25]	**9.98**	12.01	1.96	[11.51, 12.50]
lin318	318	42029	**8.35**	8.50	0.16	[8.46, 8.54]	8.97	10.00	0.97	[9.75, 10.25]
fl417	417	11861	10.11	9.62	0.31	[9.54, 9.70]	**9.05**	10.02	1.10	[9.74, 10.30]
pr439	439	107217	**9.39**	10.95	1.17	[10.66, 11.25]	**9.39**	10.30	0.85	[10.09, 10.51]
pcb442	442	50783	9.16	10.50	2.01	[9.99, 11.01]	**8.76**	10.05	0.95	[9.81, 10.30]
att532	532	87550	14.58	14.83	0.34	[14.74, 14.91]	**9.10**	9.96	0.87	[9.74, 10.18]
rat575	575	6773	10.03	10.46	0.53	[10.33, 10.59]	**9.86**	10.73	0.59	[10.58, 10.88]
u724	724	41910	16.85	16.85	0.00	[16.85, 16.85]	**10.18**	10.56	0.44	[10.45, 10.67]
pr1002	1002	259045	15.66	15.91	0.30	[15.83, 15.99]	**14.39**	15.11	0.65	[14.95, 15.28]

Table 1. Comparisons between the results of 36 symmetric instances from the TSPLIB with the Hard WTA (HWTA) and Soft WTA (SWTA) techniques.

TSP name	*n*	Optimal solution	Average error (%)							
			HWTA2				SWTA2			
			best	Mean	SD	CI (95%)	Best	Mean	SD	CI (95%)
eil51	51	430	**0**	0.31	0.41	[0.21, 0.41]	**0**	0.09	0.21	[0.04, 0.15]
brazil58	58	16156	**0**	0.41	0.52	[0.28, 0.54]	**0**	0.30	0.51	[0.18, 0.43]
st70	70	678.6	**0**	0.14	0.28	[0.07, 0.21]	**0**	0.14	0.26	[0.08, 0.21]
eil76	76	545.4	**0**	0.12	0.28	[0.05, 0.19]	**0**	0.37	0.65	[0.21, 0.54]
kroA100	100	7910	0.84	2.12	1.18	[1.82, 2.42]	**0**	0.47	0.81	[0.26, 0.67]
kroB100	100	21282	0.71	1.47	0.72	[1.28, 1.65]	**0.47**	1.76	0.97	[1.52, 2.01]
kroC100	100	22141	**0**	0.29	0.40	[0.19, 0.39]	**0**	0.73	0.88	[0.51, 0.95]
kroD100	100	20749	0.73	1.23	0.58	[1.08, 1.38]	**0.59**	0.89	0.41	[0.78, 0.99]
kroE100	100	21294	0.84	1.65	0.83	[1.44, 1.86]	**0.32**	1.74	2.25	[1.77, 2.31]
rd100	100	22068	**0.08**	0.49	0.50	[0.36, 0.61]	0.49	1.48	0.90	[1.25, 1.71]
eil101	101	629	0.48	1.63	1.35	[1.29, 1.97]	**0.16**	0.95	1.20	[0.65, 1.26]
lin105	105	14383	0.20	0.73	0.77	[0.53, 0.93]	**0**	1.46	1.33	[1.13, 1.80]
pr107	107	44303	**0**	0.53	0.93	[0.29, 0.76]	**0**	0.11	0.15	[0.07, 0.14]
pr124	124	59030	**0**	0.45	0.82	[0.24, 0.66]	0.09	0.98	1.08	[0.71, 1.25]
bier127	127	118282	0.37	1.08	0.65	[0.92, 1.24]	**0.25**	1.55	1.03	[1.29, 1.81]
ch130	130	6110	1.39	1.85	0.72	[1.67, 2.03]	**0.80**	2.14	1.30	[1.81, 2.47]
pr136	136	96772	1.21	1.25	0.05	[1.24, 1.26]	**0.58**	1.18	0.48	[1.06, 1.30]
gr137	137	69853	2.07	2.96	1.83	[2.50, 3.42]	**0.21**	1.38	1.75	[0.94, 1.82]
kroA150	150	26524	**1.17**	2.35	1.21	[2.04, 2.65]	1.39	2.77	1.30	[2.44, 3.10]
kroB150	150	26130	2.16	3.48	1.22	[3.18, 3.79]	**1.48**	3.77	2.27	[3.20, 4.35]
pr152	152	73682	**0**	0.00	0.00	[0.00, 0.00]	**0**	0.51	1.01	[0.25, 0.76]
u159	159	42080	**0**	0.51	0.79	[0.31, 0.71]	0.79	1.71	0.86	[1.49, 1.93]
rat195	195	2323	3.32	3.95	1.11	[3.67, 4.24]	**2.71**	3.24	0.65	[3.08, 3.41]
d198	198	15780	1.22	1.95	1.25	[1.64, 2.27]	**0.73**	0.81	0.11	[0.78, 0.83]
kroA200	200	29368	**0.62**	6.02	5.87	[4.54, 7.51]	0.75	1.37	0.65	[1.20, 1.53]
tsp225	225	3859	2.54	3.10	0.66	[2.94, 3.27]	**1.06**	1.75	0.96	[1.51, 1.99]
gil262	262	2378	2.90	3.57	0.96	[3.32, 3.81]	**1.89**	3.02	1.31	[2.69, 3.35]
a280	318	42029	4.02	4.07	0.10	[4.04, 4.09]	**2.01**	2.87	1.02	[2.61, 3.13]
lin318	280	2586	1.90	2.38	0.84	[2.16, 2.59]	**1.89**	3.25	1.35	[2.91, 3.59]
fl417	417	11861	1.58	1.96	0.46	[1.85, 2.08]	**1.43**	1.61	0.32	[1.53, 1.69]
pr439	439	107217	2.39	3.26	0.84	[3.05, 3.48]	**1.99**	3.23	1.43	[2.86, 3.59]
pcb442	442	50783	2.87	3.18	0.52	[3.05, 3.32]	**2.79**	3.63	0.97	[3.39, 3.88]
att532	532	87550	**1.28**	1.91	1.22	[1.60, 2.22]	1.48	2.12	0.92	[1.89, 2.35]
rat575	575	6773	4.98	5.89	0.96	[5.65, 6.14]	**4.50**	5.33	0.78	[5.13, 5.53]
u724	724	41910	6.28	6.53	0.35	[6.44, 6.62]	**4.06**	4.47	0.51	[4.34, 4.60]
pr1002	1002	259045	4.68	5.58	0.72	[5.40, 5.76]	**4.39**	5.24	1.20	[4.94, 5.55]

Table 2. Comparisons between the results of 36 symmetric instances from the TSPLIB with the Hard WTA with 2-opt (HWTA2) and Soft WTA with 2-opt (SWTA2) techniques.

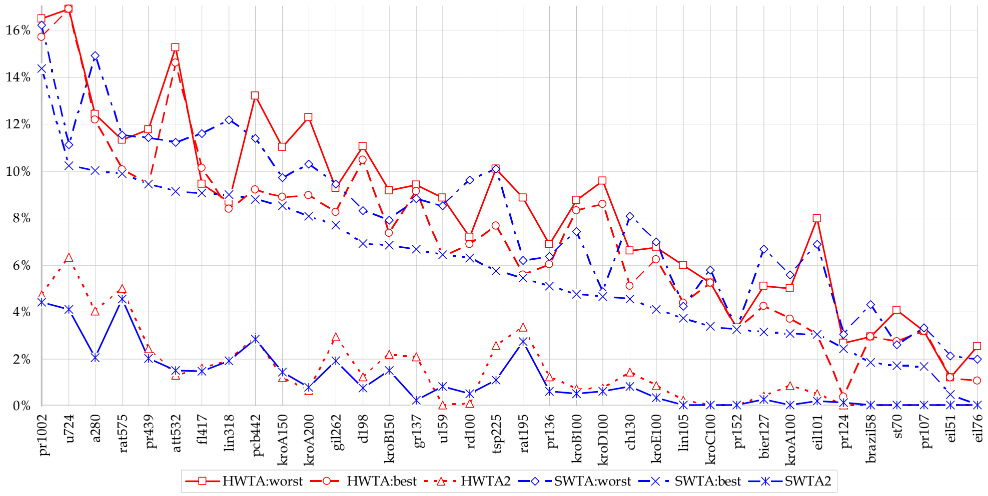


Fig. 4. Comparison between the results of the Hard WTA (HWTA) and Soft WTA (SWTA) techniques for 36 symmetrical problems from the TSPLIB

- the KNIESL technique, which consists of a local version of KNIESG, where only some cities are used for the neurons dispersion phase;
- in the Efficient and Integrated Self-Organizing Method (EISOM), a SOM network is used to generate a solution in which the winning neuron is substituted by the position of the mid-point between the two closest neighboring neurons (Jin *et al.*, 2003);
- the Co-Adaptive Network (CAN), which uses the idea of cooperation among neighboring neurons and uses a number of neurons that is higher than the number of cities in the problem (Cochrane & Beasley, 2003);
- the Real-Valued Antibody Network (RABNET), which uses a mechanism to stabilize the winning neurons and the centroids of the groups of cities for growth and pruning of the network (Massutti & Castro, 2009);
- the Modified Growing Ring Self-Organizing Network (MGSOM) incorporates other initialization methods for the weights in the network, with other adaptation parameters proposed for the SOM network and other indexing forms for the order of cities (Bai *et al.*, 2006);
- the MSOM, which consists in a hybrid technique with Self-Organizing Maps (SOM) and evolutionary algorithms to solve the TSP, called Memetic Neural Network (Créput & Kouka, 2007); and
- the technique of building a decision tree with minimum amplitude to choose the candidate cities for path exchange with the Lin-Kernighan of 2 up to 5-opt techniques (Kelsgaun, 2000).

The comparisons are shown in Table 3, where 16 of the 24 problems tested have better results with the technique proposed using the 2-opt route improving technique.

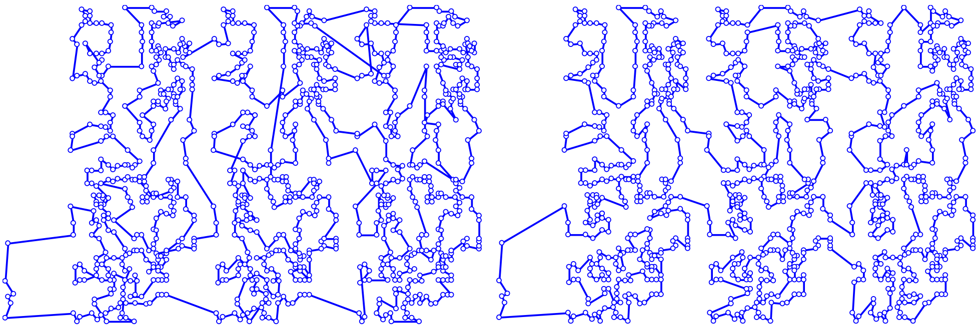
The order of computational complexity of the proposed technique is $O(n^2 + n)$ (Wang, 1997), considered competitive when compared with the complexity of Self-Organizing Maps, which have an $O(n^2)$ complexity (Leung *et al.*, 2004).

TSP name	Average error (%)									
	5OPT	KNIESG	KNIESL	EISOM	MGSOM	RABNET	CAN	MSOM	SWTA	SWTA2
eil51	0.85	2.86	2.86	2.56	1.40	0.56	0.94	1.64	0.47	**0**
st70	0.61	2.33	1.51	-	1.18	-	0.89	0.59	1.68	**0**
eil76	0.2	5.48	4.98	-	3.38	**0**	2.04	1.86	**0**	**0**
rd100	**0.16**	2.62	2.09	-	1.17	0.91	1.19	0.43	6.27	0.49
kroA100	1.65	-	-	-	-	0.24	0.57	0.18	3.05	**0**
kroB100	1.42	-	-	-	-	0.91	1.53	0.62	4.73	**0.47**
kroC100	1.35	-	-	-	-	0.80	0.80	0.30	3.35	**0**
kroD100	0.72	-	-	-	-	**0.38**	0.80	0.61	4.64	0.59
eil101	0.27	5.63	4.66	3.59	-	1.43	1.11	2.07	3.02	**0.16**
lin105	0.06	1.29	1.98	-	0.03	**0**	**0**	**0**	3.70	**0**
pr107	10.78	0.42	0.73	-	0.17	-	0.18	0.14	1.65	**0**
pr124	1.67	0.49	0.08	-	-	-	2.36	**0**	2.39	0.09
bier127	0.73	3.08	2.76	-	1.09	0.58	0.69	1.25	3.11	**0.25**
ch130	0.58	5.63	4.66	-	-	**0.57**	1.13	0.80	4.52	0.80
pr136	0.96	5.15	4.53	-	2.15	-	3.93	0.73	5.06	**0.58**
kroA150	0.88	-	-	1.83	-	**0.58**	1.55	1.75	8.50	1.40
pr152	2.1	1.29	0.97	-	0.74	-	0.74	1.07	3.23	**0**
rat195	**1.35**	11.92	12.24	-	5.98	-	4.69	4.69	5.42	2.71
kroA200	1.07	6.57	5.72	1.64	1.97	0.79	0.92	**0.70**	8.03	0.75
lin318	**0.35**	-	-	2.05	-	1.92	2.65	3.48	8.97	1.89
pcb442	**0.62**	10.45	11.07	6.11	8.58	-	5.88	3.57	8.76	2.79
att532	**0.99**	6.80	6.74	3.35	-	-	4.24	3.29	9.10	1.48
rat575	**0.74**	-	-	2.18	-	4.05	4.89	4.31	9.86	4.50
pr1002	**0.9**	-	-	4.82	-	-	4.18	4.75	14.39	4.39
mean	4.50	1.29	4.22	3.29	2.32	1.00	2.16	1.78	5.13	1.04

Table 3. Comparisons between the results of 24 symmetric problems from the TSPLIB with the techniques: Soft WTA (SWTA), Soft WTA with 2-opt (SWTA2), 5-OPT (decision tree with minimum amplitude of 2 up to 5-opt), KNIESG (Kohonen Network Incorporating Explicit Global Statistics), KNIESL (Kohonen Network Incorporating Explicit Statistics Local), EISOM (Efficient and Integrated SOM), MGSOM (Modified Growing Ring SOM), RABNET (Real-valued Antibody Network), CAN (Co-Adaptive Network) and MSOM (Memetic SOM)

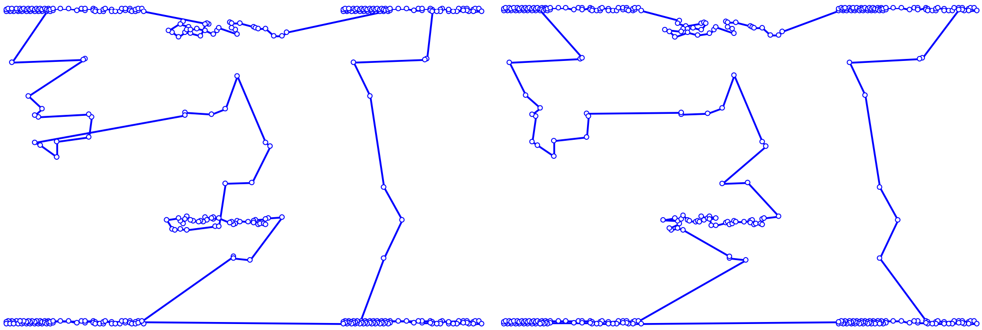
Fig. 5a shows the best result found with the Soft WTA technique for the 1002-cities problem, by Padberg and Rinaldi, and Figure 5b shows the best result found with the same technique with the 2-opt route improvement. In Fig. 6 are the best results for drilling problem fl417 by Reinelt. In Fig. 7 are shown the best results for the of 439-cities problem by Padberg and Rinaldi.

Fig. 8 and 9 show the comparison between the best results for Wang's Recurrent Neural Network with the Hard and Soft WTA techniques for drilling problems d198 (Reinelt) and pcb442 (Groetschel, Juenger and Reinelt), respectively. Fig. 10 show the best result found with the Soft WTA technique for 399-cities problem of Parana, Brazil, where the optimal solution to this problem is 7,086,615.



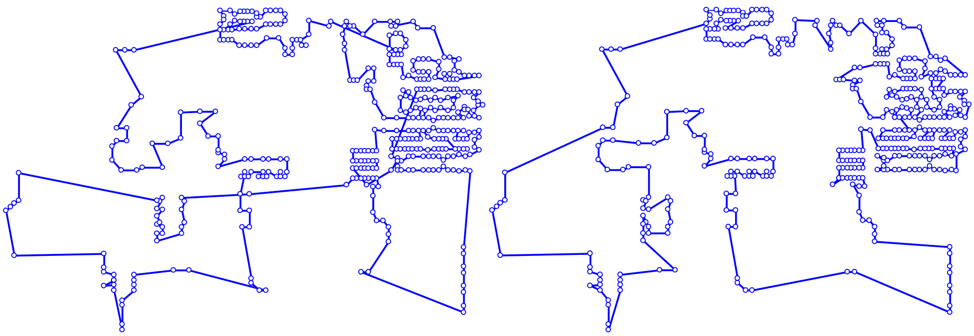
(a) (b)

Fig. 5. Example of the pr1002 problem with the application of Wang's Neural Network: (a) with Soft WTA and average error of 14.39%, (b) with Soft WTA and 2-opt improvement with average error of 4.39%



(a) (b)

Fig. 6. Example of the fl417 problem with the application of Wang's Neural Network: (a) with Soft WTA and average error of 9.05%, (b) with Soft WTA and 2-opt improvement with average error of 1.43%



(a) (b)

Fig. 7. Example of the pr439 problem with the application of Wang's Neural Network: (a) with Soft WTA and average error of 9.39%, (b) with Soft WTA and 2-opt improvement with average error of 1.99%

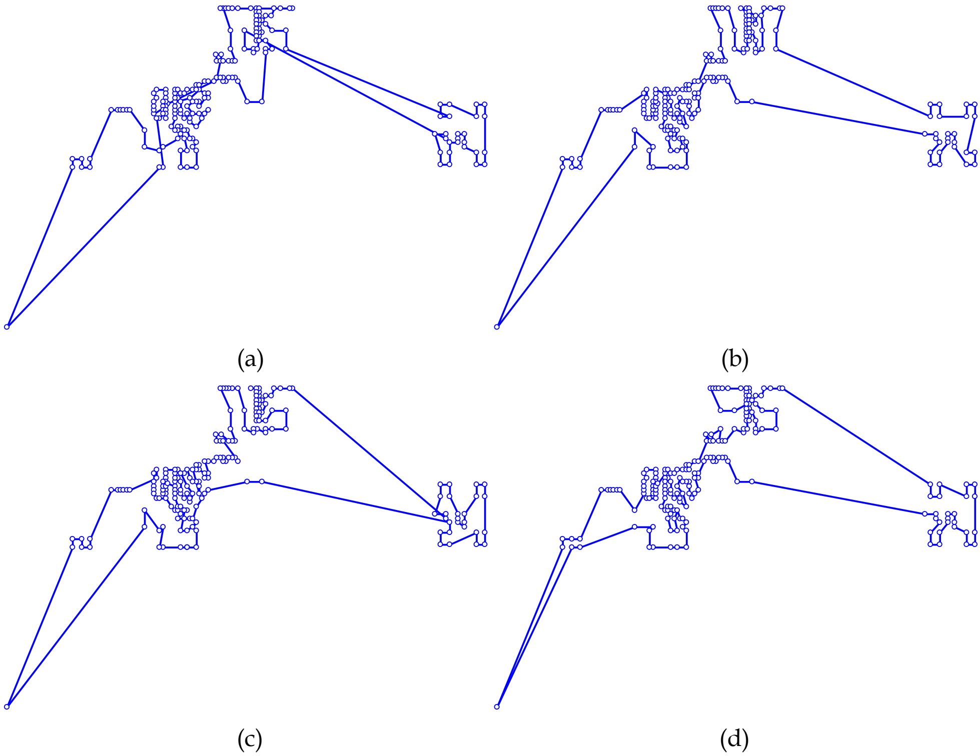


Fig. 8. Example of the d198 problem with the application of Wang's Neural Network: (a) with Hard WTA and average error of 10.43%, (b) with Hard WTA and 2-opt improvement with average error of 1.22% (c) with Soft WTA and average error of 6.86%, (d) with Soft WTA and 2-opt improvement with average error of 0.73%

3.2 Asymmetric problems from the TSPLIB

Table 4 and Table 5 shows the comparison between the Hard and Soft WTA techniques applied to asymmetric problems from the TSPLIB and demonstrates that the Soft WTA technique exceeds or equals the Hard WTA technical in all problems using 2-opt technique. The average error of the Soft WTA technique with 2-opt (SWTA2) varies between 0 and 10.56%, and with the Hard WTA technique with 2-opt (HWTA2) this error varies between 0 and 16.14%. The 95% confidence intervals for the mean were computed on the basis of standard deviations (sd) over 60 runs.

Fig. 11 shows a comparison between the Soft and Hard WTA techniques, showing the best and worst results from each asymmetric problem from the TSPLIB.

The techniques compared with the TSP asymmetric problems are described in the work of Glover *et al*. (2001):

- the Karp-Steele Path method (KSP) and the General Karp-Steele (GKS) method start with a cycle by removing paths and placing new ones to find a Hamiltonian cycle. The difference between these methods is that the GKS uses all vertices in the cycle to change paths.

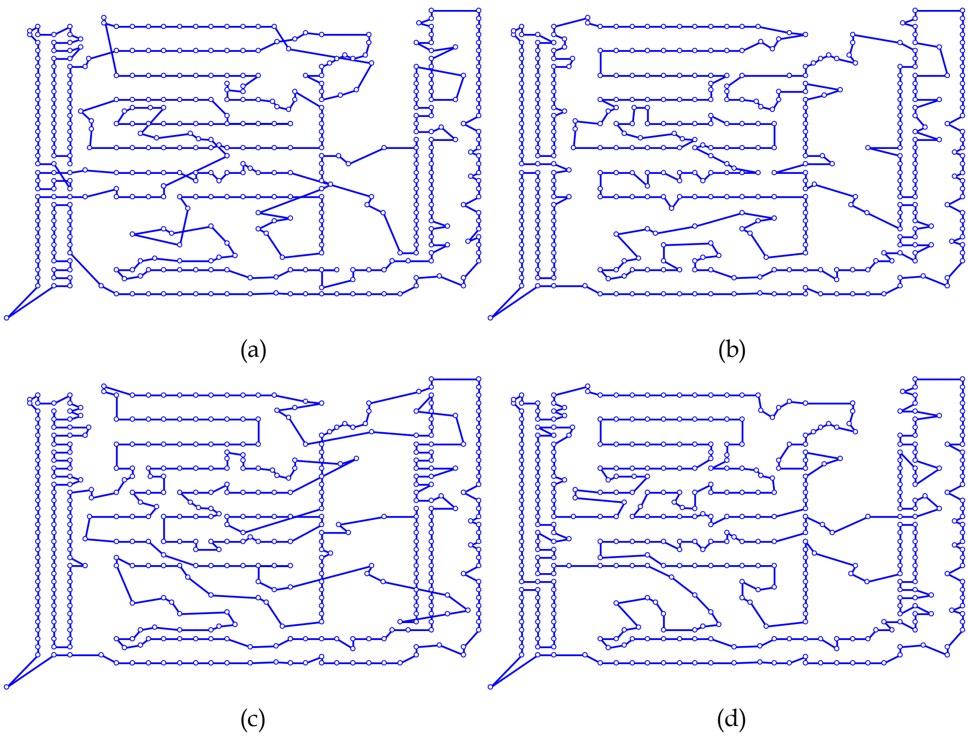


Fig. 9. Example of the pcb442 problem with the application of Wang's Neural Network: (a) with Hard WTA and average error of 9.16%, (b) with Hard WTA and 2-opt improvement with average error of 2.87%, (c) with Soft WTA and average error of 8.76%, (d) with Soft WTA and 2-opt improvement with average error of 2.79%

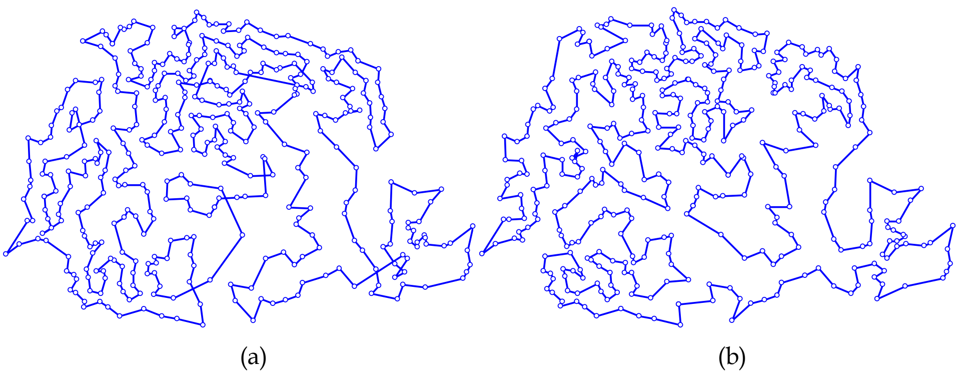


Fig. 10. Example of 399-cities problem of Paraná, Brazil (a) with Soft WTA and average error of 10.14%, and (b) with Soft WTA and 2-opt improvement with average error of 4.43%

TSP name	n	Optimal solution	Average error (%)							
			HWTA				SWTA			
			Best	Mean	SD	CI (95%)	Best	Mean	SD	CI (95%)
br17	17	39	**0**	0.73	1.25	[0.42, 1.05]	**0**	0.00	0.00	[0.00, 0.00]
ftv33	33	1286	**0**	5.26	0.77	[5.06, 5.45]	**0**	3.63	1.64	[3.22, 4.05]
ftv35	35	1473	3.12	4.84	1.33	[4.51, 5.18]	**0.61**	4.14	3.18	[3.34, 4.95]
ftv38	38	1530	3.73	3.77	0.03	[3.76, 3.78]	**2.94**	5.85	2.13	[5.32, 6.39]
pr43	43	5620	0.29	0.41	0.07	[0.39, 0.43]	**0.20**	0.28	0.06	[0.26, 0.29]
ftv44	44	1613	2.60	3.44	0.92	[3.21, 3.67]	**2.23**	5.29	2.64	[4.62, 5.95]
ftv47	47	1776	**3.83**	6.64	2.18	[6.09, 7.20]	5.29	7.32	2.86	[6.60, 8.04]
ry48p	48	14422	5.59	5.99	0.44	[5.88, 6.10]	**2.85**	4.16	0.91	[3.93, 4.40]
ft53	53	6905	**2.65**	3.04	0.30	[2.96, 3.11]	3.72	5.09	1.53	[4.70, 5.47]
ftv55	55	1608	11.19	8.00	3.23	[7.18, 8.81]	**2.11**	5.02	2.91	[4.29, 5.76]
ftv64	64	1839	2.50	2.50	0.00	[2.50, 2.50]	**1.41**	2.00	0.55	[1.86, 2.14]
ft70	70	38673	1.74	2.79	1.01	[2.53, 3.04]	**1.70**	1.94	0.27	[1.87, 2.01]
ftv70	70	1950	8.77	7.61	1.96	[7.11, 8.10}	**4.10**	8.01	3.16	[7.21, 8.81]
kro124p	124	36230	7.66	9.24	1.44	[8.88, 9.61]	**7.27**	8.25	1.00	[8.00, 8.50]
ftv170	170	2755	12.16	13.72	1.24	[13.41, 14.03]	**10.56**	12.63	2.34	[12.04, 13.22]
rbg323	323	1326	16.14	16.24	0.16	[16.20, 16.28]	**3.02**	3.19	0.27	[3.12, 3.26]
rbg358	358	1163	12.73	17.52	4.54	[16.37, 18.67]	**5.76**	7.57	2.26	[6.99, 8.14]
rbg403	403	2465	4.71	4.71	0.00	[4.71, 4.71]	**3.53**	3.93	0.66	[3.76, 4.10]
rbg443	443	2720	8.05	8.05	0.00	[8.05, 8.05]	**2.98**	3.33	0.55	[3.19, 3.47]

Table 4. Comparisons between the results of the 20 asymmetric problems from the TSPLIB for the techniques Hard WTA (HWTA) and Soft WTA (SWTA).

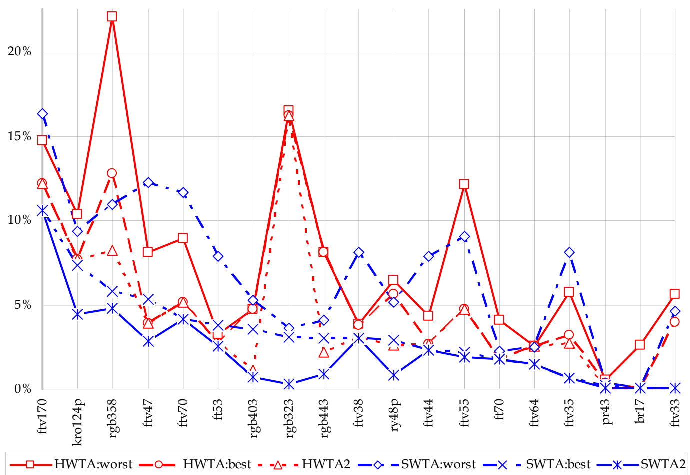


Fig. 11. Comparison between the results of the Hard WTA (HWTA) and Soft WTA (SWTA) techniques for the 19 asymmetric problems from the TSPLIB

TSP name	*n*	Optimal solution	Average error (%)							
			HWTA2				SWTA2			
			Best	**Mean**	**SD**	**CI (95%)**	**Best**	**Mean**	**SD**	**CI (95%)**
br17	17	39	**0**	0.00	0.00	[0.00, 0.00]	**0**	0.00	0.00	[0.00, 0.00]
ftv33	33	1286	**0**	2.68	2.13	[2.14, 3.22]	**0**	1.64	2.08	[1.12, 2.17]
ftv35	35	1473	3.12	3.22	1.28	[2.90, 3.54]	**0.61**	1.85	2.48	[1.22, 2.48]
ftv38	38	1530	3.01	3.71	0.37	[3.62, 3.81]	**2.94**	4.86	2.66	[4.19, 5.54]
pr43	43	5620	0.05	0.09	0.05	[0.07, 0.10]	**0**	0.16	0.17	[0.12, 0.21]
ftv44	44	1613	2.60	3.21	1.36	[2.87, 3.56]	**2.23**	4.23	1.89	[3.75, 4.71]
ftv47	47	1776	3.83	4.35	1.04	[4.09, 4.61]	**2.82**	5.38	2.43	[4.76, 5.99]
ry48p	48	14422	1.24	2.82	0.57	[2.67, 2.96]	**0.76**	1.65	1.31	[1.32, 1.98]
ft53	53	6905	2.65	3.21	0.58	[3.06, 3.36]	**2.49**	2.79	0.46	[2.67, 2.90]
ftv55	55	1608	6.03	5.97	1.65	[5.55, 6.39]	**1.87**	2.31	0.93	[2.08, 2.55]
ftv64	64	1839	2.50	3.50	1.73	[3.06, 3.94]	**1.41**	1.73	0.34	[1.65, 1.82]
ft70	70	38673	1.74	1.74	0.00	[1.74, 1.74]	**1.70**	2.19	0.34	[2.05, 2.33]
ftv70	70	1950	8.56	7.54	2.70	[6.85, 8.22]	**4.10**	7.32	4.98	[6.06, 8.58]
kro124p	124	36230	7.66	8.19	1.18	[7.89, 8.48]	**4.36**	4.94	1.29	[4.61, 5.26]
ftv170	170	2755	12.16	14.03	2.71	[13.34, 14.71]	**10.56**	11.23	1.32	[10.89, 11.56]
rbg323	323	1326	16.14	16.34	0.16	[16.30, 16.38]	**0.23**	1.71	1.29	[1.38, 2.03]
rbg358	358	1163	8.17	8.91	1.36	[8.57, 9.25]	**4.73**	6.29	1.55	[5.90, 6.69]
rbg403	403	2465	4.71	1.54	0.41	[1.44, 1.65]	**0.65**	0.91	0.33	[0.83, 1.00]
rbg443	443	2720	2.17	3.93	3.95	[2.94, 4.93]	**0.85**	0.91	0.06	[0.90, 0.93]

Table 5. Comparisons between the results of the 19 asymmetric problems from the TSPLIB for the techniques Hard WTA with 2-opt (HWTA2) and Soft WTA with 2-opt (SWTA2).

- the Path Recursive Contraction method (PRC) forms an initial cycle, removing sub-cycles to find a Hamiltonian cycle
- the Contraction of Paths (COP) heuristic is a combination of the GKS and PRC techniques;
- the Random Insertion (RI) heuristic starts with two vertices, inserting a vertex not yet chosen, creating a cycle. This procedure is repeated to create a route that contains all vertices;
- the Greedy heuristic (GR) choose the smallest path in the graph, contracts this path to create a new graph, maintaining this procedure up to the last path, forming a route.

Table 6 shows that the technique proposed in this paper has equal or better results than the techniques mentioned in 11 of the 19 tested asymmetric problems from the TSPLIB: br17, ftv33, ftv35, pr43, ftv44, ry48p, ft53, ftv55, ftv64, ft70 and kro124p.

Considering the techniques without the 2-opt improvement, Fig. 11 shows that the best solutions for the Hard WTA technique are better than the Soft WTA in only 3 problems: ftv47, ft70 and ft53. A great improvement can also be seen with the Soft WTA technique in the quality of the solutions for problems rgb443, rgb323 and rgb358.

The worst solutions for problems ftv35, ftv38, ftv44, ftv47, ft53, ftv70, ftv170 and rgb403 with the Soft WTA technique have higher costs than those found with the Hard WTA.

TSP name	Average error (%)							
	GR	RI	KSP	GKS	PRC	COP	SWTA	SWTA2
br17	102.56	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ftv33	31.34	11.82	13.14	8.09	21.62	9.49	**0**	**0**
ftv35	24.37	9.37	1.56	1.09	21.18	1.56	**0.61**	**0.61**
ftv38	14.84	10.20	1.50	**1.05**	25.69	3.59	2.94	2.94
pr43	3.59	0.30	0.11	0.32	0.66	0.68	0.20	**0**
ftv44	18.78	14.07	7.69	5.33	22.26	10.66	**2.23**	**2.23**
ftv47	11.88	12.16	3.04	**1.69**	28.72	8.73	5.29	2.82
ry48p	32.55	11.66	7.23	4.52	29.50	7.97	2.85	**0.76**
ft53	80.84	24.82	12.99	12.31	18.64	15.68	3.72	**2.49**
ftv55	25.93	15.30	3.05	3.05	33.27	4.79	2.11	**1.87**
ftv64	25.77	18.49	3.81	2.61	29.09	1.96	**1.41**	**1.41**
ft70	14.84	9.32	**1.88**	2.84	5.89	1.90	4.10	4.10
ftv70	31.85	16.15	3.33	2.87	22.77	1.85	**1.70**	**1.70**
kro124p	21.01	12.17	16.95	8.69	23.06	8.79	7.27	**4.36**
ftv170	32.05	28.97	2.40	**1.38**	25.66	3.59	10.56	10.56
rbg323	8.52	29.34	**0**	**0**	0.53	**0**	3.02	0.23
rbg358	7.74	42.48	**0**	**0**	2.32	0.26	5.76	4.73
rbg403	0.85	9.17	**0**	**0**	0.69	0.20	3.53	0.65
rbg443	0.92	10.48	**0**	**0**	**0**	**0**	2.98	0.85
mean	25.80	15.07	4.14	2.94	16.39	4.30	3.17	2.22

Table 6. Comparisons between the results of the 19 asymmetric problems from the TSPLIB with the techniques Soft WTA (SWTA), Soft WTA with 2-opt (SWTA2) Random Insertion (RI), Karp-Steele Path (KSP), General Karp-Steele path (GKS), Path Recursive Contraction (PRC), Contraction or Path (COP) and Greedy heuristic (GR).

7. Conclusions

This paper presents a modification in the application of the 'Winner Takes All' technique in Wang's Recurrent Neural Network to solve the Traveling Salesman Problem. This technique is called Soft 'Winner Takes All', because the winning neuron receives only part of the activation of the other competing neurons.

The results were compared with the Hard 'Winner Takes All' variation, Self-Organizing Maps and the Path insertion and removal heuristics, showing improvement in most of the problems tested from the TSPLIB. The average errors for symmetric problems were between 0 and 4.50%, and for the asymmetric ones, between 0 and 10.56%.

The proposed technique was implemented with the 2-opt route improvement and the results shown in this study were compared both with and without the 2-opt technique.

8. References

Aras, N.; Oommen, B.J. & Altinel, I.K. (1999). The Kohonen network incorporating explicit statistics and its application to the traveling salesman problem. *Neural Networks*, Vol. 12, No. 9, November 1999, pp. 1273-1284, ISSN 0893-6080

Bai, Y.; Zhang, W. & Jin, Z. (2006). An new self-organizing maps strategy for solving the traveling salesman problem, *Chaos, Solitons and Fractals*, Vol. 28, No. 4, May 2006, pp. 1082–1089, ISSN 0960-0779

Cochrane, E.M. & Beasley, J.E. (2003). The Co-Adaptive Neural Network Approach to the Euclidean Travelling Salesman Problem. *Neural Networks*, Vol. 16, No. 10, December 2003, pp. 1499-1525, ISSN 0893-6080

Créput, J.C. & Koukam, A. (2009). A memetic neural network for the Euclidean travelling salesman problem. *Neurocomputing*, Vol. 72, No. 4-6, January 2009, pp. 1250-1264, ISSN 0925-2312

Glover, F.; Gutin, G.; Yeo, A. & Zverovich, A. (2001). Construction heuristics for the asymmetric TSP. *European Journal of Operational Research*, Vol. 129, No. 3, March 2001, pp. 555-568, ISSN 0377-2217

Hammer, B.; Schrauwen, B. & Steil, J.J. (2009), Recent advances in efficient learning of recurrent networks, *Proceedings of ESANN'2009 - European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*, pp. 213-226, ISBN 2-930307-09-9, Bruges (Belgium), April 2009, D-side public., Bruges

Hopfield, J.J.. & Tank, D.W. (1985). "Neural" computation of decisions in optimization problems, *Biological cybernetics*, Vol. 52, No. 3, July 1985, pp. 141-152, ISSN 0340-1200

Hung, D.L. & Wang, J. (2003). Digital Hardware realization of a Recurrent Neural Network for solving the Assignment Problem. *Neurocomputing*, Vol. 51, Abril 2003, pp. 447-461, ISSN 0925-2312

Jin, H.D.; Leung, K.S.; Wong, M.L. & Xu, Z.B. (2003). An Efficient Self-Organizing Map Designed by Genetic Algorithms for the Traveling Salesman Problem. *IEEE Transactions On Systems, Man, And Cybernetics - Part B: Cybernetics*, Vol. 33, No. 6, December 2003, pp. 877-887, ISSN 1083-4419

Kelsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research,* Vol. 126, No. 1, October 2000, pp. 106-130, ISSN 0377-2217

Leung, K.S.; Jin, H.D. & Xu, Z.B. (2004). An expanding self-organizing neural network for the traveling salesman problem. *Neurocomputing*, Vol. 62, December 2004, pp. 267-292, ISSN 0925-2312

Li, M.; Yi, Z. & Zhu, M. (2009). Solving TSP by using Lotka-Volterra neural networks, *Neurocomputing*, Vol. 72, No. 16-18, October 2009, pp. 3873-3880, ISSN 0925-2312

Lin, S. & Kernighan, B.W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research,* Vol. 21, No. 2, March-April 1973, pp. 498-516, ISSN 0030-364X

Masutti, T.A.S. & Castro, L.N. (2009). A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Information Sciences*, Vol. 179, No. 10, April 2009, pp. 1454-1468, ISSN 0020-0255

Misevičius, A.; Smolinskas, J. & Tomkevičius A. (2005). Iterated Tabu Search for the Traveling Salesman Problem: new results. *Information Technology And Control*, Vol. 34, No. 4, 2005, pp. 327-337, ISSN 1392-124X

Reinelt, G. (1991). TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, Vol. 3, No. 4, 1991, pp. 376-384, ISSN 0899-1499

Siqueira, P.H.; Steiner, M.T.A. & Scheer, S. (2007). A new approach to solve the Traveling Salesman Problem. *Neurocomputing*, Vol. 70, No. 4-6, January 2007, pp. 1013-1021, ISSN 0925-2312

Siqueira, P.H.; Scheer, S. & Steiner, M.T.A. (2008). A Recurrent Neural Network to Traveling Salesman Problem. In: *Travelling Salesman Problem*, Greco, F. (Ed.), pp. 135-156, In-the, ISBN 978-953-7619-10-7, Croatia

Wang, J. (1992). Analog Neural Network for Solving the Assignment Problem. *Electronic Letters*, Vol. 28, No. 11, May 1992, pp. 1047-1050, ISSN 0013-5194

Wang, J. (1997). Primal and Dual Assignment Networks. *IEEE Transactions on Neural Networks*, Vol. 8, No. 3, May 1997, pp. 784-790, ISSN 1045-9227

Wang, Y.; Feng, X.Y.; Huang, Y.X.; Pu, D.B.; Liang, C.Y. & Zhou, W.G. (2007). A novel quantum swarm evolutionary algorithm and its applications, *Neurocomputing*, Vol. 70, No. 4-6, January 2007, pp. 633-640, ISSN 0925-2312

Yi, J.; Yang, G.; Zhang, Z. & Tang, Z. (2009). An improved elastic net method for travelling salesman problem, *Neurocomputing*, Vol. 72, No. 4-6, January 2009, pp. 1329-1335, ISSN 0925-2312