A new heuristic procedure to solve the Traveling Salesman problem

Paulo Henrique Siqueira[1], Sérgio Scheer[2] and Maria Teresinha Arns Steiner[3]

[1]Department of Technical Drawing, [2]Department of Civil Construction, [3]Department of Mathematics, Federal University of Paraná, PO BOX 19081, 81531-990, Curitiba, Brazil

E-mail: paulohs@ufpr.br, scheer@ufpr.br, tere@ufpr.br

Abstract – This paper presents a heuristic technique that uses the Wang Recurrent Neural Network with the "Winner Takes All" principle to solve the Traveling Salesman Problem. When the Wang Neural Network presents solutions for the Assignment Problem with all constraints satisfied, the "Winner Takes All" principle is applied to the values in the Neural Network's decision variables, with the additional constraint that the new solution must form a feasible route for the Traveling Salesman Problem. The results from this new technique are compared to other heuristics, with data from the TSPLIB (Traveling Salesman Problem Library). The 2-opt local search technique is applied to the final solutions of the proposed technique and shows a considerable improvement of the results.

Keywords – Recurrent Neural Network, Assignment problem, Traveling Salesman Problem

I. INTRODUCTION

The Traveling Salesman Problem (TSP) is a classical problem of combinatorial optimization in the area of Operations Research. The purpose is to find a minimum total cost Hamiltonian cycle [2]. There are several practical uses for this problem, such as Vehicle Routing [12] and Drilling Problems [15].

This problem has been extensively used as a basis for comparison in order to improve several optimization techniques, such as Genetic Algorithms [1], Simulated Annealing [6], Tabu Search [14], Local Search [5], Ant Colony [7] and Neural Networks [13], the latter used in this work.

The main types of Neural Network used to solve the TSP are: Hopfield's Recurrent Networks [21] and Kohonen's Self-Organizing Maps [13]. In a Hopfield Network, the main idea is to automatically find a solution for the TSP by means of a equilibrium state of the equation system defined for the TSP. By using Kohonen's Maps for the TSP, the final route is determined through the cities corresponding to those neurons that have weights that are closest to the pair of coordinates ascribed to each city in the problem.

The Wang Recurrent Neural Network (WRNN) with the "Winner Takes All" (WTA) principle [17] can be applied to solve the TSP, solving this problem as if it were an Assignment Problem (AP) by means of the WRNN, and, furthermore, using the WTA principle on the solutions found with the WRNN, with the constraint that the solutions found must form a feasible route for the TSP. The parameters used for the WRNN are those that show the best solutions for the AP.

The solutions found with the heuristic technique proposed in this work (WRNN+WTA) are compared with the solutions from the Self-Organizing Maps (SOM) and the Simulated Annealing (SA) for the symmetrical TSP, as well as with other heuristics for the asymmetrical TSP. The 2-opt Local Search technique [5] is used to improve the solutions found with the technique proposed in this work. The data used for the comparisons are from the TSPLIB database [16].

This paper is divided into 6 sections, including this introduction. In Section 2, the TSP is defined. In Section 3, the WRNN and the WTA principle are presented. In Section 4, the use of the technique proposed in this work is demonstrated through the TSPLIB example of 42-city instance by Dantzig. In Section 5, the results of the proposed technique are presented and comparisons between this technique and other heuristics in the literature are made. In Section 6, conclusions are presented.

II. FORMULATION OF THE PROBLEM

The TSP can be mathematically formulated as follows [2, 4]:

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

Subject to $\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, ..., n$ (2)

$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, ..., n$ (3)

$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, ..., n$ (4)

$\tilde{x}$ forms a Hamiltonian cycle (5)

where c_{ij} and x_{ij} are, respectively, the costs and the decision variables associated to the assignment of element i to position j. When $x_{ij} = 1$, the element i is assigned to the position j and this means that the route has a stretch that is the sequence from city i to city j. Vector $\tilde{x}$ has the whole sequence of the route that was found, i.e., the solution for the TSP.

The AP has the same formulation, with exception of (5). The objective function (1) represents the total cost to be minimized. The constraints sets (2) and (3) ensure that each city i will be assigned to exactly one city j. Set (4) represents the integrality constraints of zero-one variables x_{ij}, and can be replaced by constraints of the following type [2, 4]:

$$x_{ij} \geq 0, \; i, j = 1, 2, ..., n. \tag{6}$$

Constraint (5) assures that in the final route each city will be visited once and that no sub-routes will be formed. The matrix form of the problem described in (1)-(5) is the following [10]:

Minimize $C = c^{\mathrm{T}}x$ (7)

Subject to $Ax = b$ (8)

$x_{ij} \geq 0, \quad i, j = 1, 2, ..., n$

$\tilde{x}$ forms a Hamiltonian cycle

where vectors c^{T} and x contain all the rows from the cost matrix and from the matrix with decision elements x_{ij}, this is, $c^{\mathrm{T}} = (c_{11}, c_{12}, ..., c_{1n}, c_{21}, c_{22}, ..., c_{2n}, ..., c_{n1}, c_{n2}, ..., c_{nn})$ and $x = (x_{11}, x_{12}, ..., x_{1n}, x_{21}, x_{22}, ..., x_{2n}, ..., x_{n1}, x_{n2}, ..., x_{nn})$. Vector b has the number 1 in all of its positions and matrix A has the following form:

$$A = \begin{bmatrix} I & I & ... & I \\ B_1 & B_2 & ... & B_n \end{bmatrix} \in \Re^{2n \times n^2}$$

where I is the identity matrix, of order $n \times n$, and each matrix B_i, for $i = 1, 2, ..., n$, has zeros, with the exception of the i^{th} row, which has the number 1 in all of its positions.

III. WRNN AND WTA TO SOLVE THE TSP

The Recurrent Neural Network proposed by Wang, published in [10], [19] and [20], is characterized by the following differential equation:

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

where $x_{ij} = g(u_{ij}(t))$ and this Neural Network's equilibrium state is a solution for the AP. Function g is the sigmoid function, with a β parameter, i.e.,

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

The boundary vector is defined as $\theta = A^{\mathrm{T}}b = (2, 2, ..., 2) \in \Re^{n^2 \times 1}$. Parameters η, λ and τ are constant and empirically chosen [10], thus affecting the Neural Network's convergence. Parameter η penalizes violations of the constraint set of the problem defined by (1)-(4). Parameters λ and τ control the AP's objective function minimization (1). The matrix form for this Neural Network can be written as follows:

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

where $x = g(u(t))$ and $W = A^{T}A$.

The heuristic technique proposed in this work uses the WTA principle, accelerating the WRNN's convergence and correcting eventual problems that may appear due to multiple optimal solutions or optimal solutions that are very close to each other [17].

The parameters chosen for the WRNN are those that determine the best results for the AP [17]. Parameter η is considered equal to 1 in all cases tested within this work. Parameter λ is taken as a vector defined by:

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

where δ_i, for $i = 1, 2, \ldots, n$, represents the standard deviation for each row in the cost matrix c. Each element of vector $\bar{\lambda}$ is used to update the corresponding row in decision matrix x.

The best choice for parameter τ uses the WRNN's definition's fourth term (9). When $c_{ij} = c_{\max}$, the term $\lambda c \exp(-t/\tau) = \alpha$ must be such that $g(\alpha) = \phi \cong 0$.

$$\alpha = \frac{-\ln\left(\frac{1}{\phi} - 1\right)}{\beta}, \tag{13}$$

Using the value of $\alpha = \lambda c \exp(-t/\tau)$ and expression (13), the value of parameter τ is obtained.

$$\tau_i = \frac{-t}{\ln\left(\frac{-\alpha}{\lambda c_{\max}}\right)}. \tag{14}$$

In equation (11), $Wx(t) - \theta$ measures the AP's constraints violation. After a certain number of iterations this term suffers no substantial changes of its value. This evidences the fact that the problem's constraints are almost satisfied. At this moment, the WTA principle can be applied.

When all the elements of x satisfy the condition $Wx(t) - \theta \leq \phi$, where $\phi \in [0, 2]$, the proposed technique can be used and its algorithm is presented below:

Step 1: Determine a maximum number of routes $r_{\max}$. Find an AP's x solution using the WRNN. If $Wx(t) - \theta \leq \phi$, then go to Step 2. Otherwise, find another solution x.

Step 2: Given the decision matrix, consider matrix $\bar{x}$, where $\bar{x} = x$, $m = 1$ and go to Step 3.

Step 3: Choose a row k in decision matrix $\bar{x}$. Do $p = k$, $\tilde{x}$ (m) = k and go to Step 4.

Step 4: Find the biggest element of row k, $\bar{x}_{kl}$. This element's value is given by the half of the sum of all elements of row k and of column l of matrix x, this is,

$$\bar{x}_{kl} = \frac{1}{2}\left(\sum_{i=1}^{n} x_{il} + \sum_{j=1}^{n} x_{kj}\right). \tag{15}$$

The other elements of row k and column l become null. So that sub-routes are not formed, the other elements of column k must also be null. Do $\tilde{x}(m + 1) = l$; to continue the TSP's route, make $k = l$ and go to Step 5.

Step 5: If $m < n$, then make $m = m + 1$ and go to Step 4. Otherwise, do

$$\bar{x}_{kp} = \frac{1}{2}\left(\sum_{i=1}^{n} x_{ip} + \sum_{j=1}^{n} x_{kj}\right), \tag{16}$$

$\tilde{x}\ (n + 1) = p$, determine the route's cost, C, and go to Step 6.

Step 6: If $C < C_{min}$, then do $C_{min} = C$ and $x = \bar{x}$. Make $r = r + 1$. If $r < r_{max}$, then run the WRNN again and go to Step 2, otherwise Stop.

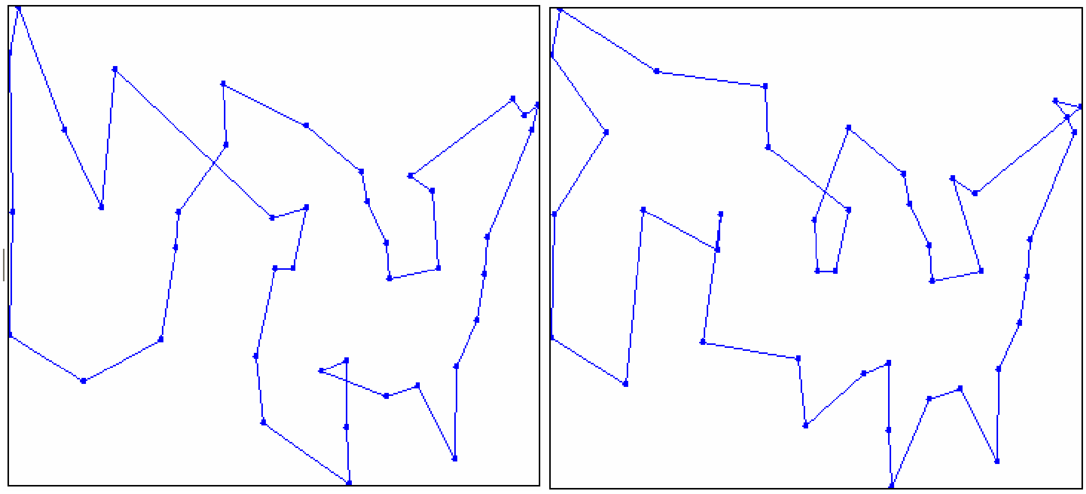
The technique proposed in this work can be applied to symmetrical or asymmetrical TSPs. The following Section shows an example of application of this heuristic to symmetrical TSP dantzig42 [16].

IV. PROPOSED HEURISTIC TECHNIQUE APPLIED TO A TSPLIB PROBLEM

Consider the symmetrical TSP with 42-city instance by Dantzig [16], as shown in Fig. 1 and 2.

This problem contains coordinates of cities in The United States, and after 25 epochs the condition $Wx(t) - \theta \leq \phi$ is satisfied with $\phi = 0.01$ and the WRNN presents the first solution $\tilde{x}_1$ for the TSP, as shown in Fig. 1 (a).

The solution $\tilde{x}_1$ is presented to WRNN, and after 20 iterations an improved solution is reached, with the average error decreasing from 6.29% to 5.58% as shown in Fig. 1 (a) and (b).

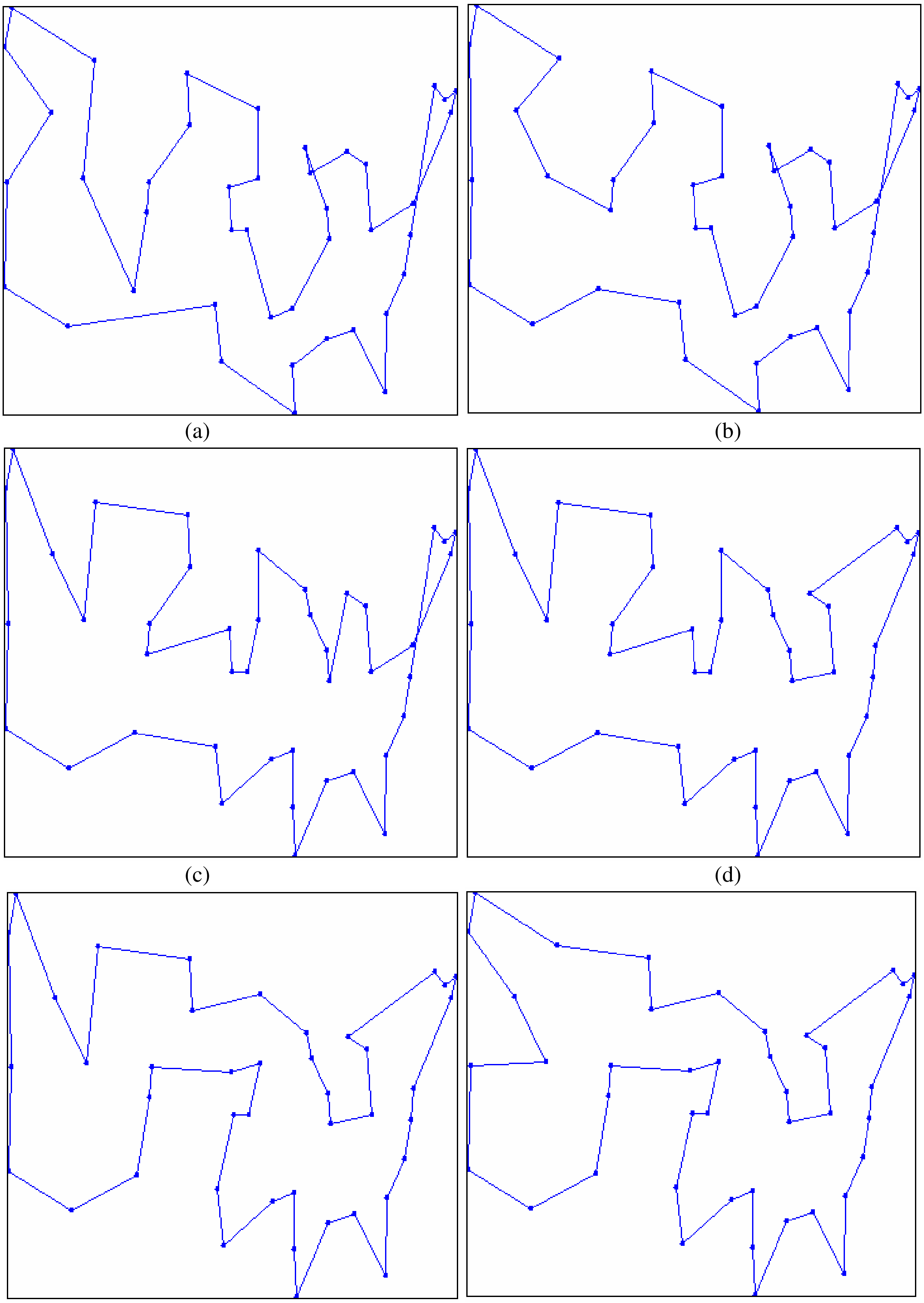


(a) (b)

Fig. 1 – (a) First feasible tour found through the proposed heuristic, with an average error of 6.29%.
(b) Tour with 5.58% of average error.

An improvement to heuristic WRNN is the application of local search 2-opt heuristic on Step 5 of the algorithm shown on previous section. This application is made after the expression (16), to the WRNN solution, just as an improvement. The results of WRNN with 2-opt on problem dantzig42 is shown in Fig. 2, where after 76 epochs a optimal solution is found.

Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 5, No. 1
pp. 1-9, 2007 ©Sociedade Brasileira de Redes Neurais



(a) (b) (c) (d) (e) (f)

Fig. 2 – (a) First feasible solution found, in 26 epochs and average error of 6.44%. (b) 34 epochs and error 4.01%. (c) 35 epochs and error 3.29%. (d) 37 epochs and error 2.15%. (e) 53 epochs and error 0.29%. (f) 76 epochs and optimal solution found.

The next Section shows the results of applying this technique to some of the TSPLIB's problems for symmetrical and asymmetrical TSPs.

Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 5, No. 1
pp. 1-9, 2007 ©Sociedade Brasileira de Redes Neurais

V. RESULTS OF APPLYING THE PROPOSED HEURISTIC TO SOME OF THE TSPLIB'S PROBLEMS

The results found with the technique proposed in this work for the TSPLIB's TSP's symmetrical cases are compared with SOM and SA results, and the asymmetrical cases are compared to removal and insertion arc heuristic.

For symmetrical TSPs, the following methods were used to compare with the technique presented in this work: the method that involves statistical methods between a SOM's neurons' weights [3] and has a global version (KniesG: Kohonen Network Incorporating Explicit Statistics Global), where all cities are used in the neuron dispersion process, and a local version (KniesL), where only some represented cities are used in the neuron dispersion step; the SA technique [6], using the 2-opt improvement technique; Budinich's SOM, which consists of a traditional SOM applied to the TSP, presented in [6]; the expanded SOM (ESOM) [13], which, in each iteration, places the neurons close to their corresponding input data (cities) and, at the same time, places them at the convex contour determined by the cities; the efficient and integrated SOM (eISOM) [11], where the ESOM procedures are used and the winning neuron is placed at the mean point among its closest neighboring neurons; the efficient SOM technique (SETSP) [18], which defines the updating forms for parameters that use the TSP's number of cities; and Kohonen's cooperative adaptive network (CAN) [8] uses the idea of cooperation between the neurons' close neighbors and uses a number of neurons that is larger than the number of cities in the problem.

The computational complexity of the proposed heuristic is $O(n^2 + n)$ [20], considered competitive when compared to the complexity of mentioned SOM neural network, which have complexity $O(n^2)$ [13]. The CAN technique has a computational complexity of $O(n^2\log(n))$ [8], while the SA technique has a complexity of $O(n^4\log(n))$ [14].

In Table 1 are shown the average errors of the techniques mentioned above. The "pure" technique proposed in this work, the proposed technique with the 2-opt improvement algorithm, as well as the best (max) and worst (min) results of each problem considered are also shown.

Table 1 – Results of the experiments for the symmetrical TSP, with techniques presented on TSPLIB: KniesG, KniesL, SA, Budinich's SOM, ESOM, EISOM, SETSP, CAN and a technique presented on this paper: WRNN with WTA. The solutions presented in bold characters show the best results for each problem, disregarding the results with the 2-opt technique. (NC = not compared)

TSP's name	number of cities	optimal solution	average error (%)										
			for 8 algorithms presented on TSPLIB								WRNN with WTA		
			KniesG	KniesL	SA	Budinich	ESom	EiSom	Setsp	CAN	Max	Min	2-opt
eil51	51	430	2.86	2.86	2.33	3.10	2.10	2.56	2.22	**0.94**	1.16	1.16	0
st70	70	678.6	2.33	1.51	2.14	1.70	2.09	NC	1.60	**1.33**	4.04	2.71	0
eil76	76	545.4	5.48	4.98	5.54	5.32	3.89	NC	4.23	2.04	2.49	**1.03**	0
gr96	96	514	NC	NC	4.12	2.09	**1.03**	NC	NC	NC	6.61	4.28	0
rd100	100	7,910	2.62	2.09	3.26	3.16	1.96	NC	2.60	**1.23**	7.17	6.83	0.08
eil101	101	629	5.63	4.66	5.74	5.24	3.43	3.59	NC	**1.11**	7.95	3.02	0.48
lin105	105	14,383	1.29	1.98	1.87	1.71	0.25	NC	1.30	**0**	5.94	4.33	0.20
pr107	107	44,303	0.42	0.73	1.54	1.32	1.48	NC	0.41	**0.17**	3.14	3.14	0
pr124	124	59,030	0.49	**0.08**	1.26	1.62	0.67	NC	NC	2.36	2.63	0.33	0
bier127	127	118,282	3.08	2.76	3.52	3.61	1.70	NC	1.85	**0.69**	5.08	4.22	0.37
pr136	136	96,772	5.15	4.53	4.90	5.20	4.31	NC	4.40	**3.94**	6.86	5.99	1.21
pr152	152	73,682	1.29	0.97	2.64	2.04	0.89	NC	1.17	**0.74**	3.27	3.23	0
rat195	195	2,323	11.92	12.24	13.29	11.48	7.13	NC	11.19	**5.27**	8.82	5.55	3.31
kroa200	200	29,368	6.57	5.72	5.61	6.13	2.91	1.64	3.12	**0.92**	12.25	8.95	0.62
lin318	318	42,029	NC	NC	7.56	8.19	4.11	**2.05**	NC	2.65	8.65	8.35	1.90
pcb442	442	50,784	10.45	11.07	9.15	8.43	7.43	6.11	10.16	**5.89**	13.18	9.16	2.87
att532	532	27,686	6.8	6.74	5.38	5.67	4.95	3.35	NC	**3.32**	15.43	14.58	1.28

The results of the heuristic proposed in this paper, together with the 2-opt improvement, presented an average error range from 0 to 3.31%, as shown in the 2-opt column of Table 1. The methods that use improvement techniques to their solutions are SA, CAN and WRNN with WTA. The techniques proposed in this paper, with 2-opt, present better results that SA and CAN methods in almost every problem, with the only exception being the *lin105* problem. Without the improvement 2-opt, the results of problems *eil76, eil51, eil101* and *rat195* are better than the results of the other neural networks that do not use improvement techniques.

For the asymmetrical TSP, the techniques used to compare with the technique proposed in this work were [9]: the Karp-Steele path methods (KSP) and general Karp-Steele (GKS), which begin with one cycle and by removing arcs and placing new arcs, transforming the initial cycle into a Hamiltonian one. The difference between these two techniques is that the GKS uses all of the cycle's vertices for the changes in the cycle's arcs. The following techniques were also used: the path recursive contraction (PRC) that consists in forming an initial cycle and transforming it into a Hamiltonian cycle by removing arcs from every sub-cycle; the contraction or path heuristic (COP), which is a combination of the GKS and RPC techniques; the "greedy" heuristic (GR) that chooses the smallest arc in the graph, contracts this arc creating a new graph, and keeps this procedure up to the last arc, thus creating a route; and the random insertion heuristic (RI) that initially chooses 2 vertices, inserts one vertex that had not been chosen, thus creating a cycle, and repeats this procedure until it creates a route including all vertices.

Table 2 shows the average errors of the techniques described, as well as those of the "pure" technique presented in this work and of the proposed technique with the 2-opt technique.

The results of the "pure" technique proposed in this work are better or equivalent to those of the other heuristics mentioned above, for problems br17, ftv33, ftv44, ft53, ft70 and kro124p, as shown in Table 2. By using the 2-opt technique on the proposed technique, the best results were found for problems br17, ftv33, pr43, ry48p, ftv44, ft53, ft70 and kro124p, with average errors ranging from 0 to 16.14%.

Table 2 – Results of the experiments for the asymmetrical TSP with techniques presented on TSPLIB: GR, RI, KSP, GKS, RPC, COP and a technique presented on this paper: WRNN with WTA. The solutions presented in bold characters show the best results for each problem, disregarding the results with the 2-opt technique.

TSP's name	number of cities	optimal solution	average error (%)								
			for 5 algorithms						WRNN with WTA		
			GR	RI	KSP	GKS	PRC	COP	max	Min	2-opt
Br17	17	39	102.56	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0
Ftv33	33	1,286	31.34	11.82	13.14	8.09	21.62	9.49	7.00	**0**	0
Ftv35	35	1,473	24.37	9.37	1.56	**1.09**	21.18	1.56	5.70	3.12	3.12
Ftv38	38	1,530	14.84	10.20	1.50	**1.05**	25.69	3.59	3.79	3.73	3.01
Pr43	43	5,620	3.59	0.30	**0.11**	0.32	0.66	0.68	0.46	0.29	0.05
Ftv44	44	1,613	18.78	14.07	7.69	5.33	22.26	10.66	2.60	**2.60**	2.60
Ftv47	47	1,776	11.88	12.16	3.04	**1.69**	28.72	8.73	8.05	3.83	3.83
ry48p	48	14,422	32.55	11.66	7.23	**4.52**	29.50	7.97	6.39	5.59	1.24
Ft53	53	6,905	80.84	24.82	12.99	12.31	18.64	15.68	3.23	**2.65**	2.65
Ftv55	55	1,608	25.93	15.30	**3.05**	**3.05**	33.27	4.79	12.19	11.19	6.03
Ftv64	64	1,839	25.77	18.49	3.81	2.61	29.09	**1.96**	2.50	2.50	2.50
Ft70	70	38,673	14.84	9.32	1.88	2.84	5.89	1.90	2.43	**1.74**	1.74
Ftv70	70	1,950	31.85	16.15	3.33	2.87	22.77	**1.85**	8.87	8.77	8.56
kro124p	100	36,230	21.01	12.17	16.95	8.69	23.06	8.79	10.52	**7.66**	7.66
ftv170	170	2,755	32.05	28.97	2.40	**1.38**	25.66	3.59	14.66	12.16	12.16
rbg323	323	1,326	8.52	29.34	**0**	**0**	0.53	**0**	16.44	16.14	16.14
rbg358	358	1,163	7.74	42.48	**0**	**0**	2.32	0.26	22.01	12.73	8.17
rbg403	403	2,465	0.85	9.17	**0**	**0**	0.69	0.20	4.71	4.71	4.71
rbg443	443	2,720	0.92	10.48	**0**	**0**	**0**	**0**	8.05	8.05	2.17

VI. CONCLUSIONS

This work presented the WRNN with the WTA principle to solve the TSP. By means of the WRNN, a solution for the AP is found and the WTA principle is applied to this solution, transforming it into a feasible route for the TSP. These technique's solutions were considerably improved when the 2-opt technique was applied on the solutions presented by the technique proposed in this work. The data used for testing were obtained at the TSPLIB and the comparisons that were made with other heuristics show that the technique proposed in this work achieves better results in several of the problems tested, with average errors below 16.14%. A great advantage of implementing the technique presented in this work is the possibility of using the same technique to solve both symmetrical and asymmetrical TSPs as well.

REFERENCES

[1] M. Affenzeller, S. Wanger, A Self-Adaptive Model for Selective Pressure Handling within the Theory of Genetic Algorithms, EUROCAST 2003, Las Palmas de Gran Canaria, Spain, 2003, Lecture Notes in Computer Science 2809 (1) (2003) 384-393.

[2] R.K. Ahuja, T.L. Mangnanti, J.B. Orlin, Network Flows, Prentice Hall, New Jersey, 1993.

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

[4] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, Linear Programming and Network Flows, Wiley, New York, 1990.

[5] L. Bianchi, J. Knowles, J. Bowler, Local search for the probabilistic traveling salesman problem: Correction to the 2-p-opt and 1-shift algorithms, European Journal of Operational Research 162 (1) (2005) 206-219.

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

[7] S.C. Chu, J.F. Roddick, J.S. Pan, Ant colony system with communication strategies, Information Sciences 167 (1-4) (2004) 63-76.

[8] E.M. Cochrane, J.E. Beasley, The Co-Adaptive Neural Network Approach to the Euclidean Travelling Salesman Problem. Neural Networks 16 (10) (2003) 1499-1525.

[9] F. Glover, G. Gutin, A. Yeo, A. Zverovich, Construction heuristics for the asymmetric TSP, European Journal of Operational Research 129 (3) (2001) 555-568.

[10] D.L. Hung, J. Wang, Digital Hardware realization of a Recurrent Neural Network for solving the Assignment Problem, Neurocomputing 51 (2003) 447-461.

[11] 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 Transactions On Systems, Man, And Cybernetics - Part B: Cybernetics 33 (6) (2003) 877-887.

[12] G. Laporte, The vehicle routing problem: An overview of exact and approximate algorithms, European Journal of Operational Research 59 (2) (1992) 345-358.

[13] K.S. Leung, H.D. Jin, Z.B. Xu, An expanding self-organizing neural network for the traveling salesman problem, Neurocomputing 62 (2004) 267-292.

[14] G. Liu, Y. He, Y. Fang, Y. Oiu, A novel adaptive search strategy of intensification and diversification in tabu search, Proceedings of Neural Networks and Signal Processing, Nanjing, China, 2003.

[15] G.C. Onwubolu, M. Clerc, Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization, International Journal of Production Research 42 (3) (2004) 473-491.

[16] G. Reinelt, TSPLIB – A traveling salesman problem library, ORSA Journal on Computing 3 (4) (1991) 376-384.

[17] P.H. Siqueira, S. Scheer, M.T.A. Steiner, Application of the "Winner Takes All" Principle in Wang's Recurrent Neural Network for the Assignment Problem, Advances in Neural Networks - ISNN 2005, Chongqing, China, 2005, Lecture Notes in Computer Science 3496 (1) (2005) 731-738.

[18] F.C. Vieira, A.D. Doria Neto, J.A. Costa, An Efficient Approach to the Travelling Salesman Problem Using Self-Organizing Maps, International Journal Of Neural Systems 13 (2) (2003) 59-66.

[19] J. Wang, Analog Neural Network for Solving the Assignment Problem, Electronic Letters 28 (11) (1992) 1047-1050.

[20] J. Wang, Primal and Dual Assignment Networks, IEEE Transactions on Neural Networks 8 (3) (1997) 784-790.

[21] R.L. Wang, Z. Tang, Q.P. Cao, A learning method in Hopfield neural network for combinatorial optimization problem, Neurocomputing 48 (4) (2002) 1021-1024.