RESEARCH ARTICLE OPEN ACCESS

A hybrid Recurrent Neural Network with Winner Takes All and Ant Colony Optimization principles for the Traveling Salesman Problem

Paulo Henrique Siqueira
Department of Graphic Expression, Federal University of Paraná, Curitiba, Brazil.

ABSTRACT
This paper presents the application of 'Winner Takes All' (WTA) and Ant Colony Optimization (ACO) principles in Wang's Recurrent Neural Network to solve the Traveling Salesman Problem. Each competing neuron is updates with a part of the value of decision variable using WTA principle. The choice of each city of the route, represented by the Neural Network neurons, is made using the ACO transition rule. The pheromones used in this new technique are the values of the decision variables of the model. The traditional WTA method uses the best value of decision matrix for each iteration. The results are compared with traditional WTA principle and others heuristics with instances of the TSPLIB (Traveling Salesman Problem Library) and 3 instances of spatial missions of debris removal. The presented results show that this new hybrid method assure equal or better results in most of the problems tested compared with traditional WTA.
Keywords - Ant Colony Optimization, Recurrent Neural Network, Traveling Salesman Problem, Winner Takes All principle

Date of Submission: 04-11-2019 Date Of Acceptance: 25-11-2019

I. INTRODUCTION

One technique that uses Wang's Recurrent Neural Networks with the "Winner Takes All" (WTA) principle [1] is presented to solve the classical problem of combinatorial optimization of Traveling Salesman problem (TSP). With proper choices for the parameters of the Wang's Recurrent Neural Network, this technique reveals to be efficient solving TSP instances in real time.

The search for the winning neuron of WTA can be done using the maximum value of decision matrix, in the traditional form, or using another heuristic. In this paper, a hybrid method is proposed using the Ant Colony Optimization (ACO) transition rule [2] to define TSP's routes. Some problems of the TSPLIB [3] were used to compare the new proposed method with others versions of WTA and they show improvement in the results with the hybrid technique ACO and WTA in a Recurrent Neural Network.

The implementation of the proposed technique in this paper uses the parameters of Wang's Neural Network for the Assignment problem [4], [5] with the WTA principle to form Hamiltonian circuits [6] and can be used both in symmetrical and asymmetrical TSP problems.

The present paper is divided into 5 sections, including this introduction. Section 2 presents a brief literature review on recent techniques used to solve symmetric and asymmetric TSPs. In section 3 are shown Wang's Recurrent Neural Network with WTA and ACO principles to solve TSP. Section 4 shows the comparative results applied to symmetric and asymmetric instances of TSP, and in Section 5 the conclusions are presented.

II. LITERATURE REVIEW

The TSP is one of the most challenging NP-hard problems in Operational Research area. Many current papers show applications of TSP for transportation and delivery problems, and recently to drones and space mission routing.

Debris in orbit poses a growing threat to space missions. In the near future, special missions for debris removal should be constant. The decision of the order of cleaning of each debris is a case of TSP. The application of 3 algorithms to solve real instances of spatial debris removal is shown in [7], considering the static and perturbed cases of TSP, simulating the debris movements.

An adapted ACO to solve a dynamic TSP is presented in [8], and applied on surveillance of deers by drones. Adaptive Large Neighborhood Search (ALNS) is used by authors and compared with ACO-based immigrant schemes existents, with efficient numerical results.

The paper [9] shows a hybrid heuristic model called Frequency Graph (FG) with ACO heuristic. Frequencies on the vertices are compared with weighted graphs for TSP instances. The results shown by the authors prove that proposed heuristic of frequency graphs is better than weighted graphs.

An artificial immune (IA) algorithm with estimation of distribution algorithm (EDA) to symmetric cases of TSP is presented in [10]. Population-based incremental learning and univariate marginal distribution algorithm are used to representation of TSP. Authors used local search operator repair infeasible solutions find by proposed technique. Results of benchmarks show that the IA with EDA are better or competitive than conventional IA and other hybrid algorithms.

A Variable Neighborhood Search (VNS) with stochastic approach to solve TSP is show in [11]. The authors compare the proposed technique with VNS-1 and VNS-2 algorithms reported in literature. The proposed technique has been best performance than the conventional algorithms, and has also been tested on 60 symmetric benchmarks of TSPLIB.

A Hybrid Immune Algorithm (HIA) to solve TSP is presented in [12]. The initial routes are found by a greedy algorithm, and utilize delete-cross operator (DO) and dynamic mutation operator (DMO) to improve routes. Experimental results of TSPLIB benchmark show that HIA is able to yield better solutions that of other techniques.

The paper [13] show a hybrid technique with a local search based on 2-opt and Simulated Annealing (SA). The authors compare SA technique with and without the proposed local search in benchmarks of TSPLIB, and show improvement average errors about 27% with the hybrid technique presented.

The discrete version of BAT algorithm to solve symmetric and asymmetric in-stances of TSP is presented in [14]. The proposed technique is based on echolocation characteristics of microbats. The authors compare BAT with other methods of TSPLIB instances, and results show improvements significantly in most of cases.

The paper [15] shown preprocessing methods used are Cross-Entropy (CE) and Particle Swarm Optimization (PSO) with Tabu Search to solve moderate sized asymmetric TSP. Methods are compared by authors with traditional TS algorithm, with better and faster results.

Efficient methods of combining preprocessing methods and Tabu Search (TS) for solving asymmetric TSP instances are presented in [16]. The authors use TS with two algorithms: randomized greedy contract (RGC) and Genetic Algorithm (GA). The proposed techniques reach optimal solutions in most of tested cases of TSPLIB benchmarks.

The paper [17] show comparison of 3 operators to Genetic Algorithm to solve asymmetric benchmarks of TSPLIB: partially mapped crossover (PMX), order crossover (OX) and new proposed cycle crossover (CX2). The PMX operator has better solutions compared with OX and CX2 in all tested instances by authors. An improved operator has been presented in [18] called by ICX, with better results than CX2 in most of tested benchmarks.

The adapting of Harmony Search algorithm (HS) to solve asymmetric TSP instances is presented in [19]. The authors found effectiveness of 59% compared with previous works, and routes imperfectness are eliminate with this new proposed technique.

III. WANG'S NEURAL NETWORK WITH WTA AND ACO PRINCIPLES

The mathematical formulation of TSP can be described as a complete graph G = (V, E) where V = {1,…,n} is a vertex set, and E = {(i, j), i,j ∈ V} is an edge set. The set V represent the city vertices and each edge (i, j) has the associated cost c_{ij} from city i to city j.

This formulation is the same as the problem of Assignment, with the additional constraint (5) that ensures that the route starts and ends in the same city.

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

$$\text{Subject to: } \sum_{i=1}^{n} x_{ij} = 1,\ j = 1,\dots,n \qquad (2)$$

$$\sum_{j=1}^{n} x_{ij} = 1,\ i = 1,\dots,n \qquad (3)$$

$$x_{ij} \in \{0,1\},\ i, j = 1,\dots,n \qquad (4)$$

$$\tilde{x} \text{ forms a Hamiltonian circuit} \qquad (5)$$

The objective function (1) minimizes costs. The set of constraints (2) and (3) ensures that each city will be visited only once. Constraints (4) guarantee the condition of integrality of the x_{ij} binary variables. Vector $\tilde{x}$ represents the sequence of the TSP's route.

To obtain a first approximation for the TSP, Wang's Recurrent Neural Network is applied to the problem of Assignment, this is, the solution satisfies constraints (1)-(4), which can be written in matrix form [5]:

$$\text{Minimize: } C = c^T x \qquad (6)$$

$$\text{Subject to: } Ax = b \qquad (7)$$

$$x_{ij} \in \{0,1\},\ i, j = 1,\dots,n \qquad (8)$$

where c is the vector with dimension n^2 that contains all rows of the cost matrix c in sequence, vector x contains the n^2 decision variables x_{ij} and vector b contains the number 1 in all positions. The matrix A has dimension $2n \times n^2$ and has the following format:

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

where I is the identity matrix of order n and each matrix B_i has zeroes in all of its positions with the exception of the i^{th} line, which has the number 1 in all of its positions. Wang's Recurrent Neural Network is defined by the following differential equation [4], [5]:

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

where $x_{ij} = g(u_{ij}(t))$, the equilibrium state of this network is a solution for the problem of Assignment [20] and g is the sigmoidal function with parameter ε:

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

The threshold is the vector $\theta = A^T b$ of order n^2, which has the number 2 in all of its positions. Parameters λ, η and τ are constant and chosen empirically [5], where η penalizes the violations to constraints (2) and (3) and parameters λ and τ control the minimization of the objective function (1). Considering $W = A^T A$, the matrix form of Wang's Neural Network is the following:

$$\frac{du(t)}{dt} = -\eta\left(Wx(t) - \theta\right) - \lambda c e^{-\frac{t}{\tau}} \qquad (9)$$

The method proposed in this paper uses the 'Winner Takes All' principle, which accelerates the convergence of Wang's Recurrent Neural Network and solves problems that appear in multiple solutions or very close solutions [1].

The adjustment of parameter λ was made using the standard deviation of the problem's costs matrix's rows coefficients, determining the vector:

$$\bar{\lambda} = \eta\begin{pmatrix} \frac{1}{\delta_1} & \frac{1}{\delta_2} & \cdots & \frac{1}{\delta_n} \end{pmatrix},$$

where δ_i is the standard deviation of row i of matrix c [6].

The adjustment of parameter τ uses the third term of Wang's Neural Network definition (9), as follows: when $c_{ij} = c_{max}$, the term $\lambda c_{ij} e^{-t/\tau} = v_i$, must satisfy $g(v_i) \cong 0$, this is, x_{ij} will have minimal value [6]; considering $c_{ij} = c_{max}$ and $\lambda_i = 1/\delta_i$, where i = 1,…, n, τ is defined by:

$$\tau_i = \frac{-t}{\ln\left(\frac{-v_i}{\lambda_i c_{\max}}\right)}.$$

After a certain number of iterations, the term $Wx(t) - \theta$ of equation (9) has no further substantial alterations, thus assuring that constraints (2) and (3) are almost satisfied and the WTA method can be applied to determine a solution for the TSP.

In this step of the method, the search for the winning neuron can be done in the traditional way, using the maximum value of each line of decision matrix x(t) [1]. The purpose of this paper is to modify this search, using a principle based on Ant Colony Optimization algorithm.

The matrix x(t) is considered as the pheromone matrix of the ACO method, whose respective cost values c are used for the transition rule calculations to construction of a TSP route.

Consider the construction of route r in iteration t by randomly choosing a city (neuron) k. The probability of choosing a city l for continuation of route r is given by the equation:

$$p_{kl} = \begin{cases} \dfrac{\bar{x}_{kl}^{\alpha}\left(c_{\max} - c_{kl}\right)^{\beta}}{\sum_{h \in V_r} \bar{x}_{kh}^{\alpha}\left(c_{\max} - c_{hl}\right)^{\beta}} & \text{if } l \in V_r \\ 0 & \text{otherwise} \end{cases},$$

where the parameters α and β control the greatest emphasis on the decision matrix x (pheromones) or the cost matrix c, respectively, and V_r represents the set of vertices (neurons) not yet visited on route r. The ACO transition rule [2] can analyze all possibilities of vertices choices to TSP with the V_r set analysis, improving the vertices search for the method Winner Takes All.

Fig. 1 shows the analysis of the choice of arc from k to l, taking into account the decision x_{kh} between k and h, and the cost $c_{max} - c_{hl}$ between h and l, where h is one of the unvisited vertices on present route. By analyzing all unvisited vertices with transition rule, the choice for the WTA method becomes more effective.

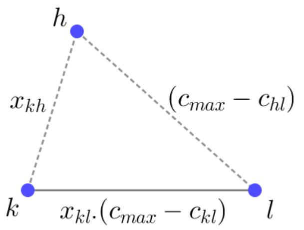

Fig. 1. Neighborhood analysis of ACO method to decide the arc choice between cities k and l.

In WTA algorithm the following situations occur: when φ = 0 updating of the WTA is nonexistent and Wang's Neural Network updates the solutions for the Assignment problem without interference; and when φ = 1 the update is called hard WTA, because the winner gets all the activation of the other neurons, the losers become null and the solution found is feasible for the TSP. In other cases, the update is called soft WTA [1] and the best results are found empirically with 0.25 ≤ φ ≤ 0.9. In this paper, experiments for each problem were made 20 times with each of the following values for the parameter φ: 0.25, 0.5, 0.7 and 0.9.

An improvement of the technique applied to results of SWTA is the application of 2-opt after determining routes for WTA. In pseudo-code this improvement is made before determining the cost of route made by WTA.

The pseudo-code of proposed technique WTA with ACO and 2-opt is described below:

Let r_{max} maximum number of routes and r = 1.
{While $r < r_{max}$
{While $Wx(t) - \theta > \phi$, $0 \le \phi \le 2$:
Find a solution x for the problem of Assignment using Wang's Neural Network.
}
Let $\bar{x} = x$ and m = 1;
Choose a row k in matrix x;
Do q = k and $\bar{x}(m) = k$;
{While m < n:
Let V_r is the set of cities not visited in route r;
To each $l \in V_r$ calcule:

$$p_{kl} = \begin{cases} \dfrac{\bar{x}_{kl}^{\alpha}\left(c_{\max} - c_{kl}\right)^{\beta}}{\sum\limits_{h \in V_r} \bar{x}_{kh}^{\alpha}\left(c_{\max} - c_{hl}\right)^{\beta}} & \text{if } l \in V_r \\ 0 & \text{otherwise} \end{cases};$$

Find $\bar{p}_{kl} = \text{argmax}\{\, p_{kl}, l \in V_r\}$;
Do the following updates:

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

$\bar{x}_{kj} = (1-\varphi)\bar{x}_{kj}$, j = 1, .., n, $j \neq l$, $0 \le \varphi \le 1$;

$\bar{x}_{il} = (1-\varphi)\bar{x}_{il}$, i = 1, .., n, $i \neq k$, $0 \le \varphi \le 1$;

Let $\tilde{x}(m+1) = l$ and m = m+1;
To continue the route, do k = l.
}

Do $$\bar{x}_{kq} = \bar{x}_{kq} + \frac{\varphi}{2}\left(\sum_{i=1}^{n} x_{iq} + \sum_{i=1}^{n} x_{kj}\right);$$

Let $\tilde{x}(n+1) = q$;
Improve the solution $\bar{x}$ using 2-opt:
$\bar{\bar{x}} = 2opt(\bar{x})$.
Determine the cost of route C;
{If $C < C_{min}$, then
$C_{min} = C$ and $x = \bar{x}$;
}
r = r + 1.
}

The computational complexity of the proposed technique is O(n^2 + n) [20], considered competitive when compared to other heuristic methods.

IV. RESULTS

The results of the technique proposed in this paper to solve the symmetric TSP were compared with the results obtained using 6 different methods cited on Section 2 for TSPLIB problems: FG: Frequency Graph with ACO [9]; IA: Immune Algorithm with estimation of distribution algorithm [10]; VNS: Variable Neighborhood Search with stochastic approach [11]; HIA: Hybrid Immune Algorithm [12]; BAT algorithm [14]; and SA: Simulated Annealing with 2-opt [13].

These comparisons are shown in Table 1, including TSP name and number of cities (n) to each tested instance. According to the results presented, 12 of the 17 problems tested showed better or equal results using the technique proposed in this paper. Average errors of proposed technique vary between 0 and 1.77%. Fig. 2 show the best results found with the Soft WTA and ACO principles for the kroA200 problem of the TSPLIB.

The results of the technique proposed in this paper to solve the asymmetric TSP were compared with the results obtained using 6 different methods cited on Section 2 for TSPLIB problems: BAT algorithm [14]; HS: Harmonic Search [19]; CX2: Genetic Algorithm with Cycle Crossover [17]; CE: Tabu Search with Cross-Entropy [15], ICX: Genetic Algorithm with Improve Cycle Crossover [18] and RGC: Tabu Search with Randomized Greedy Contract [16].

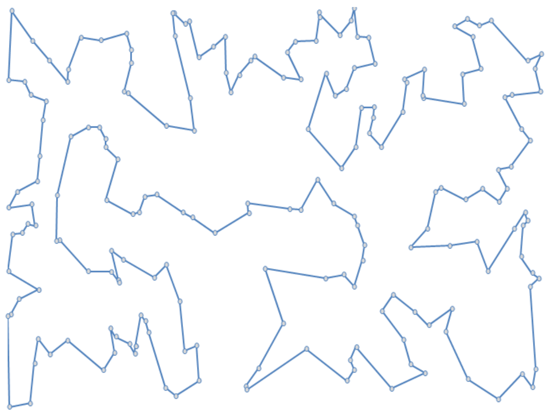

Fig. 2. Example of the kroA200 problem with application of Wang's Neural Network with soft WTA and ACO principles: cost of 29542 and average error of 0.59%.

Table 2 shows that the technique proposed in this paper have equal or better results than the techniques cited in Section 1 in 7 of 18 tested asymmetric problems of TSPLIB. Average errors of proposed technique vary between 0 and 5.37%.

Table 3 shows the comparison between technique proposed with ACO principles, Soft WTA [21] and Hard WTA [6] techniques. The parameter φ represent the best value for each tested instance of TSPLIB, and Hard WTA use φ = 0. Results of applying Wang's Neural Network with Soft WTA and ACO principles improving technique have average error ranging between 0 and 2.73%.

The results are better in almost all problems tested when compared to the results obtained with the Soft and Hard WTA technique. The average of all errors decreased from 1.91 to 0.83%.

Table 1. Comparisons between the results for symmetrical instances of TSPLIB using techniques: Soft WTA and ACO principles (WTA), FG, IA, VNS, HIA, BAT and SA.

TSP name	n	Average error (%)						
		FG	IA	VNS	HIA	BAT	SA	WTA
bier127	127	-	0.04	0.58	-	-	-	0.03
kroA200	200	2.84	0.67	3.17	2.51	-	-	0.59
pr107	107	0.89	0	0	-	0	0	0
pr124	124	1.35	0.1	0	-	0	0	0
pr136	136	-	0.2	1.24	-	0.8	0.37	0.85
pr144	144	1.93	-	0	0.75	0	0	0.47
u159	159	0.34	-	0.84	-	-	0	0
lin318	318	-	1.55	4.51	6.95	-	1.2	1.77
lin105	105	0.06	0	0	-	-	0	0
rd100	100	0.46	0	0	-	-	-	0
eil101	101	2.54	0	2.11	-	0.79	-	0
eil51	51	0	0	0.69	-	0	-	0
st70	70	0.44	0	0.31	-	0	-	0
pr76	76	0.5	-	0	-	-	-	0
rat195	195	2.37	0.77	5.47	3.44	-	0.6	0.77
pcb442	442	-	2.71	0.01	8.54	-	1.86	1.71
eil76	76	1.49	0	1.37	-	0.19	-	0
average		**1.17**	**0.43**	**1.19**	**4.44**	**0.22**	**0.45**	**0.36**

Table 2. Comparisons between the results for asymmetric instances of TSPLIB using techniques: Soft WTA with ACO principles (WTA), BAT, HS, CX2, CE, ICX and RGC.

TSP name	n	Average error (%)						
		BAT	HS	CX2	CE	ICX	RGC	WTA
ftv33	34	0	2.86	40.82	-	39.27	-	0
ftv35	36	0	1.22	-	-	-	-	0
ftv38	39	0	1.51	42.98	-	34.6	-	0
ftv44	45	0	1.67	-	-	-	-	0.68
ftv47	48	1.13	1.83	-	-	-	-	1.35
ftv55	56	0	1.34	-	-	-	-	1.55
ftv64	65	2.18	2.47	-	-	-	-	1.20
ftv70	71	8.26	4.94	-	-	-	-	2.46
ry48p	48	0	0.63	-	-	-	-	2.72
ft53	53	1.39	6.09	59.12	-	64.66	-	0
kro124p	100	3.61	7.54	155.18	1.24	-	-	3.04
ft70	70	3.18	3.45	-	-	-	-	1.19
ftv170	171	-	20.65	110.8	2.8	-	1.52	5.37
pr43	43	0	0.02	-	-	-	-	0.05
rbg323	323	21.52	19.94	216.93	2.2	126.34	0.1	0.6
rbg358	358	-	28.32	343.68	2.78	-	0	2.84
rbg403	403	-	10	152.2	1.74	-	0	0.04
rbg443	443	-	6.74	149.87	1.79	65.26	0.04	0.07
average		**1.17**	**2.95**	**6.73**	**141.29**	**2.09**	**66.02**	**0.33**

Figures 3 and 4 shows the best results found with the Soft WTA and ACO principles for the pr1002 and pr439 problems of the TSPLIB, respectively.

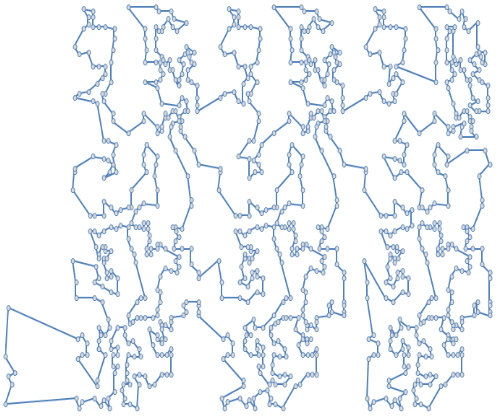

Fig. 3. Example of the pr1002 problem with application of Wang's Neural Network with Soft WTA and ACO principles: cost of 266118 and average error of 2.73%.

Table 4 compares the proposed technique with Hard and Soft WTA techniques applied to asymmetric problems of TSPLIB. Results demonstrate that the new pro-posed method with ACO principle exceeds or equals the Soft and Hard WTA techniques in most problems. The average of all errors decreased from 4.54 to 1.34%.

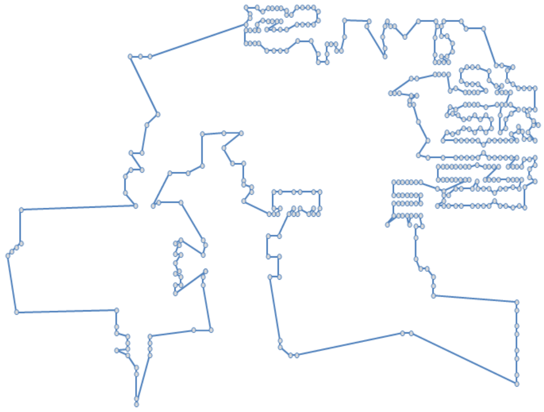

Fig. 4. Example of the pr439 problem with application of Wang's Neural Network with soft WTA and ACO principles: cost of 108149 and average error of 0.87%.

Table 5 compares results of Soft WTA with ACO and techniques proposed in paper [7] to solve instances of application of static TSP to removal orbital debris. The data considered by authors are from the colision occurred in 2009 of Iridium-33 and Cosmos-2251 satellites and the event caused by a Chinese anti-satellite missile test in 2007 in Fengyun-1C.

Table 3. Comparisons between the results for symmetric instances in the TSPLIB of the techniques Hard WTA (HWTA), Soft WTA (SWTA) and Soft WTA with ACO principle (WTA-ACO).

TSP name	n	Optimal solution	φ	Average error (%)		
				HWTA	SWTA	SWTA-ACO
eil51	51	430	0.7	0	0	0
rd100	100	7910	0.25	0.08	0.49	0.01
eil101	101	629	0.9	0.48	0.16	0
lin105	105	14383	0.9	0.2	0	0
pr107	107	44303	0.7	0	0	0
bier127	127	118282	0.7	0.37	0.25	0.03
pr124	124	59030	0.7	0	0.09	0
ch130	130	6110	0.25	1.39	0.8	0.54
pr136	136	96772	0.25	1.21	0.58	0.85
gr137	137	69853	0.7	2.07	0.21	0
rat195	195	2323	0.5	3.32	2.71	0.9
d198	198	15780	0.7	1.22	0.73	0.67
kroA200	200	29368	0.5	0.62	0.75	0.59
gil262	262	2378	0.25	2.9	1.89	1.81
a280	280	2586	0.7	4.02	2.01	0
fl417	417	11861	0.25	1.58	1.43	1.16
pcb442	442	50783	0.5	2.87	2.79	1.71
pr439	439	107217	0.7	2.39	1.99	0.87
att532	532	87550	0.25	1.28	1.48	0.75
rat575	575	6773	0.25	4.98	4.5	2.14
u724	724	41910	0.5	6.28	4.06	2.58
pr1002	1002	259045	0.7	4.68	4.39	2.73
average				**1.91**	**1.44**	**0.83**

Table 4. Comparisons between the results for asymmetric instances in the TSPLIB of the techniques Hard WTA (HWTA), Soft WTA (SWTA) and Soft WTA with ACO principle (WTA-ACO).

TSP name	n	Optimal solution	φ	Average error (%)		
				HWTA	SWTA	SWTA-ACO
br17	17	39	0.7	0	0	0
ftv33	34	1286	0.7	0	0	0
ftv35	36	1473	0.5	3.12	0.61	0
ftv38	39	1530	0.9	3.01	2.94	2.35
pr43	43	5620	0.7	0.05	0	0.05
ftv44	45	1613	0.25	2.6	2.23	0.68
ftv47	48	1776	0.9	3.83	2.82	1.35
ry48p	48	14422	0.5	1.24	0.76	2.72
ft53	53	6905	0.5	2.65	2.49	0
ftv55	56	1608	0.7	6.03	1.87	1.55
ftv64	65	1839	0.9	2.5	1.41	1.2
ft70	70	38673	0.7	1.74	4.1	2.46
ftv70	71	1950	0.5	8.56	1.7	1.19
kro124p	100	36230	0.7	7.66	4.36	3.04
ftv170	171	2755	0.25	12.16	10.56	5.37
rbg323	323	1326	0.7	16.14	0.23	0.6
rbg358	358	1163	0.7	8.17	4.73	2.84
rbg403	403	2465	0.9	4.71	0.65	0.04
rbg443	443	2720	0.9	2.17	0.85	0.07
average				**4.54**	**2.23**	**1.34**

Table 5. Comparisons between the results for symmetric instances to active debris removal problem with Soft WTA and ACO principles (WTA-ACO), IOT, RAAN and NNS.

TSP name	n	IOT	RAAN	NNS	WTA-ACO
Iridium-33	232	47.169	47.424	53.46	42.83
Cosmos-2251	533	63.19	65.63	89.21	63.22
Fengyun	1,550	155.29	186.9	199.39	171.5

The authors used 3 techniques to solve static case of TSP in debris removal problem [7]: inver-over operator (IOT), an evolutionary technique that provides a powerful genetic operator; the right ascension of the ascending node walk (RAAN); and the nearest neighbour search (NNS).

V. CONCLUSIONS

This paper presents an application of the 'Winner Takes All' technique in Wang's Recurrent Neural Network to solve the Traveling Salesman Problem, with some modifications to choose the winner neuron in each iteration. This technique use Soft 'Winner Takes All', with transition rule of Ant Colony Optimization.

The results were compared with the traditional 'Winner Takes All' versions, Ant Colony Optimization, Immune Algorithm, Variable Neighborhood Search, Bat Algo-rithm, Simulated Annealing, Harmonic Search, Genetic Algorithm and Tabu Search. The most of the tested symmetric and asymmetric problems from the TSPLIB show improvement in results of proposed new hybrid technique compared with these heuristics.

Comparing the new method with traditional WTA methods applied in symmetric problems, the average of its errors decreases from 1.91 to 0.83%. In asymmetric problems, the average error reduced from 4.54 to 1.34%. The results shown in this paper prove that the new hybrid WTA with ACO technique can improve both sym-metric and asymmetric TSP built routes.

REFERENCES

[1]. P. H. Siqueira, M. T. A. Steiner, and S. Scheer, Recurrent Neural Network with Soft 'Winner Takes All' principle for the TSP. Proceedings of ICNC 2010, 2nd International Conference on Neural Computation, Valencia, 2010, 265-270.

[2]. M. Dorigo, and L. M. Gambardella, Ant colonies for the travelling salesman problem. Biosystems, 43, 1997, 73-81.

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

[4]. J. Wang, Analog Neural Network for Solving the Assignment Problem. Electronic Letters, 28, 1992, 1047-1050.
[5]. D. L. Hung, and J. Wang, Digital Hardware realization of a Recurrent Neural Network for solving the Assignment Problem. Neurocomputing, 51, 2003, 447-461.
[6]. P. H. Siqueira, M. T. A. Steiner, and S. Scheer, A new approach to solve the Traveling Salesman Problem. Neurocomputing, 70, 2007, 1013-1021.
[7]. D. Izzo, I. Getzner, D. Hennes, and L. M. Simões, Evolving Solutions to TSP Variants for Active Space Debris Removal. Proceedings of GECCO'15, 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, 2015, 1207-1214.
[8]. S. Chowdhury, M. Marufuzzaman, H. Tunc, L. Bian, and W. Bullington, A modified Ant Colony Optimization algorithm to solve a dynamic traveling salesman problem: A case study with drones for wildlife surveillance. Journal of Computational Design and Engineering, 6, 2019, 368-386.
[9]. Y. Wang, and Y. Wu, Frequency Graphs for Travelling Salesman Problem Based on Ant Colony Optimization. International Journal of Computational Intelligence and Applications, 18, 2019, 1-16.
[10]. Z. Xu, Y. Wang, S. Li, Y. Liu, Y. Todo, and S. Gao, Immune algorithm combined with estimation of distribution for traveling salesman problem. IEEJ Transactions on Electrical and Electronic Engineering, 11, 2016, 142-154.
[11]. S. Hore, A. Chatterjee, and A. Dewanji, Improving variable neighborhood search to solve the traveling salesman problem, Applied Soft Computing, 68, 2018, 83-91.
[12]. G. Pan, K. Li, A. Ouyang, and K. Li, Hybrid immune algorithm based on greedy algorithm and delete-cross operator for solving TSP. Soft Computing, 20, 2016, 555-566.
[13]. K. S. Ghandeshtani, M.B. Taghadosi, S. M. H. Seyedkashi, and K. Shojaii, Dynamic Local Search Algorithm for Solving Traveling Salesman Problem. Proceedings of ADVCOMP 2010, Fourth International Conference on Advanced Engineering Computing and Applications in Sciences. Florence, 2010, 53-58.
[14]. E. Osaba, X. S. Yang, F. Diaz, P. Lopez-Garcia, and R. Carballedo, An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Engineering Applications of Artificial Intelligence, 48, 2016, 59-71.
[15]. S. Basu, and M. Sharma, Preprocessing Schemes for Tabu search on Asymmetric Traveling Salesman Problem. Working Paper Series, 735, 20136, 1-25.
[16]. S. Basu, M. Sharma, and P. S. Ghosh, Efficient preprocessing methods for tabu search: an application on asymmetric travelling salesman problem. INFOR: Information Systems and Operational Research, 55, 2017, 134-158.
[17]. A. Hussain, Y. S. Muhammad, M. N. Sajid, I. Hussain, A. M. Shoukry, and S. Gani, Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Computational intelligence and neuroscience, 2017, 2017, 1-7.
[18]. A. Hussain, Y. S. Muhammad, and M. N. Sajid, An Improved Genetic Algorithm Crossover Operator for Traveling Salesman Problem. Turkish Journal of Mathematics and Computer Science, 9, 2018, 1-13.
[19]. U. Boryczka, and K. Szwarc, The Harmony Search algorithm with additional improvement of harmony memory for Asymmetric Traveling Salesman Problem. Expert Systems with Applications, 122, 2019, 43-53.
[20]. J. Wang, Primal and Dual Assignment Networks. IEEE Transactions on Neural Networks, 8, 1997, 784-790.
[21]. P. H. Siqueira, S. Scheer, and M. T. A. Steiner, Recurrent Neural Networks with the Soft 'Winner Takes All' Principle Applied to the Traveling Salesman Problem. In Davendra, D., Ed., Traveling Salesman Problem, Theory and Applications, In-teh: Croatia, 2010, 177-196.

Paulo Henrique Siqueira "A hybrid Recurrent Neural Network with Winner Takes All and Ant Colony Optimization principles for the Traveling Salesman Problem" International Journal of Engineering Research and Applications (IJERA), vol. 9, no. 11, 2019, pp 85-92