Journal of Maritime Research, Vol. I. No. 3, pp. 51-66, 2004

ISSN: 1697-4840

MODELING SHIP'S ROUTE BY THE ADAPTATION OF HOPFIELD -TANK TSP NEURAL ALGORITHM

Sanja Bauk y Nataša Kova

ABSTRACT

This paper considers determining the optimal linear ship's route by the application of the Hopfield-Tank neural network algorithm adaptation for solving well known traveling salesman problem (TSP). In the paper proposed mathematical approach to the Hopfield-Tank TSP neural algorithm realization is primarily based upon faster generating zero-one matrices of suboptimal solutions.

1. INTRODUCTION

It is known that the route planing is the beginning of all operations in marine shipping, particularly linear one. Since route of linear ship provides a cycle voyage, it is naturally compared with well-defined general traveling salesman problem (TSP) [1]. According to this problem, navigator has to complete a round trip of a set of ports, visiting each one only once in such a way as to minimize total sailing distance. This kind of problem is computationally very difficult and it is shown that the time to find a solution grows exponentially with number of visiting ports. The solution of the problem is in the *nutshell* and that is where the application of the artificial intelligence becomes interesting. Besides simulated annealing and genetic algorithms, Hopfield-Tank recurrent neural network application to the TSP is one of the most interesting methodologies. This solution may not be the best and the fastest obtained one, but undoubtedly gains insight to the marrow of the problem.

The remaining part of the paper is organized in the following manner:

(1) the second part is addressed to some basic remarks to the TSP formulation and chronology of its most important solutions;
(2) the third part describes TSP model adaptation to Hopfield recurrent neural network architecture;
(3) the fourth part considers the shortest sailing path between two distant ports on the Earth surface;
(4) the fifth part contains the original mathematical approach to the Hopfiel-Tank TSP neural algorithm implementation and the numerical results for TSP in the case of relatively large number of ports being arbitrary chosen, and finally
(5) the last one contains some conclusion remarks and further investigation directions.

Faculty of Maritime Studies, University of Montenegro
e-mails: bsanja@cg.yu, dnzkovac@cg.yu

2. TSP OVERVIEW

The origins of the traveling salesman problem (TSP) are obscure. The mathematical problems related to it were treated firstly by the Irish mathematician William Rowan Hamilton and by British mathematician Thomas Penyngton Kirkman (1800's). The description of this early work of Hamilton and Kirkman can be found in [2]. The mathematician and economist Karl Menger publicized it among his colleagues in Vienna, in 1930's [3]. Later, the chronology of the most significant TSP solutions has been going as follows: Dantzing, Fulkerson and Johnson (1954) solved it for 49 nodes; Held and Karp (1971) for 64 nodes; Camerini, Fratta and Maffioli (1975) for 100 nodes; Grotschel (1977) for 120 nodes; Crowder and Padberg (1980) for 318 nodes; Padberg and Rinaldi (1987) for 532 nodes; Grotschel and Holland (1987) for 666 nodes; Padberg and Rinaldi during the same year for 2 392 nodes; Applegate, Bixby, Chvatal and Cook (1994) for 7 397 nodes, and four years later, for even 13 509 nodes, etc. In the recent years Applegate, Bixby, Chvatal and Cook (2001) have solved TSP for impressive number of 15 112 nodes. But nobody was able to come up with an algorithm for solving the traveling salesman problem (TSP) that does not show an exponential growth of run time with a growing number of nodes. There is a strong belief that there is no algorithm that will not show this behavior, but no one was able to prove it completely. But one was able to prove that the TSP is a kind of prototypical problem for a big class of nondeterministic polynomial (NP) time hardness problem and a lot of artificial intelligent methods have been developed in aim to solve it exactly or approximately. Hopfield (1986) has explored an innovative method to solve it by the electronic circuit that produces approximate solutions quite effectively. Later, Hopfield and Tank (1987) have improved this neural network based method for TSP implementation and its more efficiently solving.

Our intention here is the Hopfield-Tank TSP neural algorithm adaptation to the liner ship's routing problem, in a mathematical sense. But firstly, some remarks to the functional equivalence between Hopfield-Tank network energy function and TSP model are given. Then some elements of the shortest path estimation between a pair of ports on the Earth surface are examined, by the roles of sphere trigonometry. Finally, some remarks to the software realization of the Hopfield-Tank TSP neural algorithm are discussed. The results being obtained through the adequate example(s) are also presented, numerically and graphically.

3. FUNCTIONAL EQUIVALENCE BETWEEN HOPFIELD-TANK NEURAL NETWORK AND TSP MODEL

Neural networks are very important in many scientific disciplines in solving previously *unsolvable* problems, like in a way the TSP of larger dimensions is. Among many neural network schemes that have been proposed and investigated, the Hopfield-type neural network remains an important one due its applicability in solving associative memory, pattern recognition, and optimization problems, with ease of VLSI implementation [4]. The first neural algorithm for combinatorial optimization was the simulated annealing method.

In this algorithm one of the neurons is selected randomly and the state of the selected neuron is updated to reduce the energy of the network. Therefore the state transitions of the system must be operated serially though the network itself has parallel architecture. Hopfield and Tank used analog neurons and continuous dynamics for energy reduction. In this manner becomes possible to operate calculation in parallel. Although to simulate continuous dynamics with digital computers, many iterations are required before reaching low or minimum energy state [5].

Namely, the Hopfield-Tank neural network optimization algorithm is based on the fact that weights of the network are to be made so that the optimal solution is located in the lower energy area of the state space. In other words, the neural network optimization problem is based upon minimization of energy function given by (1):

$$E(x) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N}\theta_i x_i \tag{1}$$

where x_i - state of the i-th neuron; θ_i - threshold of the i-th neuron and w_{ij} - weight of the connection from the j-th neuron to i-th neuron ($w_{ij} = w_{ji}$). A candidate of the solution is represented by one of the state vectors $x = (x_1, x_2, \ldots x_N)$. The energy function (1) consists of two parts, cost (2) and penalty (3):

$$E_c(x) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}^{(c)}x_i x_j + \sum_{i=1}^{N}\theta_i^{(c)}x_i \tag{2}$$

$$E_p(x) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}^{(p)}x_i x_j + \sum_{i=1}^{N}\theta_i^{(p)}x_i \tag{3}$$

Both of them, in addition, are to be reduced as much as possible, according to the aim of $E(x)$ minimization. The second one is to be reduced to zero in the optimal solution. TSP could be formulated in a following way: the shortest round tour, which covers all P nodes (here ports) salesman (here navigator) is to visit, has to be found. The problem of P nodes may be coded into P -by- P network. Each row of the network corresponds to a node and the ordinal position of the node in the tour is given by the node at the place outputting a high value (i.e. one), while rest are all at very low values (i.e. zero). The nodes of the network are unipolar sigmoidal activation units where ai-th unit has output $x_{ai}=1$, if, and only if, node a is visited i-th in the tour and output $x_{ai}=0$, if a is not visited i-th in the tour [6.7]. The distance between nodes a and b is denoted as d_{ab} and the energy function, that is its cost part, takes the form (4):

$$E_c(x) = \frac{D}{2}\sum_{a}\sum_{b\neq a}\sum_{i} d_{ab}x_{ai}\left(x_{b,i+1} + x_{b,i-1}\right) \tag{4}$$

where indexes are cyclic, that is $P+1=1$, $1-1=P$, and D is positive constant. Constrains that must be satisfied are: only one node can be visited at the same time; each node is

visited only once and every node must be visited in the tour. These constrains could be formulated in the following way (5):

$$E_p(x)=\frac{A}{2}\sum_a\sum_i\sum_{j\neq i}x_{ai}x_{aj}+\frac{B}{2}\sum_i\sum_a\sum_{b\neq a}x_{ai}x_{bi}+\frac{C}{2}\left(\sum_a\sum_i x_{ai}-P\right)^2 \quad \textbf{(5)}$$

where A, B and C are positive coefficients. From the energy function weight matrix can be obtained (6):

$$w_{aibj}=-A\delta_{ab}(1-\delta ij)-B\delta_{ij}(1-\delta_{ab})-C-Dd_{ab}(\delta_{j,i+1}+\delta_{j,i-1}) \quad (6)$$

where $\delta i_1 i_2=1$, if $i_1=i_2$ and $\delta i_1 i_2=0$, if $i_1\neq i_2$. In accordance with original Hopfield and Tank adaptation [8] coefficients A, B, C and D take values 500, 500, 200 and 500, respectively. The neural network with the above weights has a tendency to give solutions that do not cover all P nodes (here ports). Kakeya [5] suggested certain improvements by introducing new parameters r and α into the original Hopfield and Tank recurrent neural network optimization model. Penalty term of energy function $(C/2)(\Sigma_a \Sigma_i x_{ai} - P)^2$ was replaced by $(C/2)(\Sigma_a \Sigma_i x_{ai} - \alpha P)^2$ where $\alpha>1$, in order to increase firing rate of the neuron. While the weight vector takes new form $w^{(c)}=D(r-d_{ab})(\delta_{j,i+1}+\delta_{j,i-1})$, where $r > 0$. By allowing the network to run under its dynamics, the energy minimum is to be reached. This energy minimum corresponds to the solution of the problem. What is meant here under solution is to be qualified. Hopfield-Tank network is not guaranty to produce the shortest round tour, but only those that are close to the shortest one. The TSP requires large number of iterations because number of possible tours is equal to $(P-1)!$ when the problem is asymmetrical and $(P-1)!/2$ when it is symmetrical, that is when $d_{ab}=d_{ba}$. Besides subcycles are to be dismissed because the continual round tour has to be found as the optimal one. Thus, the computational complexity of the problem, particularly when the number of visiting ports is large, does not vanish so simply [8,9]. In aim to find the shortest linear ship's route, in the next sections we shall firstly give some remarks to the manner(s) of estimating the shortest path between each pair in the given set of ports on the Earth sphere surface. Then, we shall realize the software adaptation of the Hopfield-Tank TSP neural algorithm to the problem of the optimal linear ship's route modeling.

4. THE SHORTEST PATH ON THE EARTH BETWEEN EACH PAIR IN THE GIVEN SET OF PORTS

The navigation between two distant points (here ports) on the Earth is possibly done in three ways: orthodrome navigation, loxodrome navigation and combined navigation. The shortest way from departure to arrival position is a shorter section of the great circle arc, that is orthodrome. Such navigation is difficult to achieve since the orthodrome intersects the meridians at different angles, so it would require constant, precisely defined change of course. In the case when course is constant, that is easily achievable in practice, navigator follows a curve asymptotically approaching nearer pole. Such a curve on the surface of the Earth is called a loxodrome. In the case of combined navigation, a standard orthodrome is

replaced by two orthodrome tangents to the boundary parallel and a loxodrome between them. The dangerous areas that could be reached by following strictly the orthodrome navigation are avoided by applying the technique of combined navigation [10].

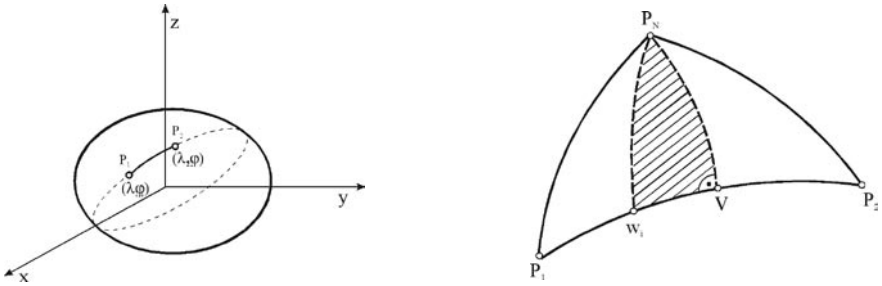


Figure 1. Orthodrome and waypoint sphere triangle

4.1. The orthodrome approximation by the loxodrome

The orthodrome is the shortest path between two distant points on the surface of the Earth (figure 1.a). Therefore, the aim of orthodrome navigation is the shortest path and the least traveling time, resulting invariably in cost effectiveness. Since strict orthodrome navigation is difficult to achieve in practice, it is divided into a finite number of waypoints between which loxodrome navigation is applied. Thus certain number of shorter loxodromes approximates the orthodrome, where the common positions are determined waypoints. For the purpose of waypoints coordinates calculation the right angle sphere triangle whose vertices are: the pole closer to the orthodrome vertex, the orthodrome vertex and the waypoint, is taken into consideration (figure 1.b). The labels of the sphere triangle in figure are: P_N - North Pole; P_1 -departure port; P_2- port of arrival; w_i -i-th waypoint ($i=\overline{1,N}$, where N is number of waypoints) and V - the orthodrome vertex. The waypoints (w_i) may be selected symmetrically to the orthodrome vertex, or by orthodrome division into finite number of equal sections. The orthodrome division is arbitrary and may be 1,2,3,... degrees, depending on the orthodrome length. We suggest the second approach, since it is much more appropriate to computation. Upon the determination of waypoints number through adequate simulation process [10,11], waypoints geographical coordinates, the appropriate loxodrome courses and distances can be calculated by application of the adequate sphere trigonometry rules. The difference (d) between the sum of the loxodrome distances and orthodrome one, expressed in nautical miles, is to be reduced as much as possible (7):

$$min\, d = \sum_{j=1}^{n} d_{lox}(j) - d_{ort} \tag{7}$$

where d_{lox} is loxodrome distance for $j=\overline{1,N}$, ($n=N+1,N$ is number of waypoints) and d_{ort} is orthodrome distance between endpoints. Through the process of waypoints number optimization this could be achieved [10]. In case of orthodrome intersection with Equator or Greenwich (figure 2), first of all points of intersection are to be determined, then on each orthodrome segment, former meant procedure of waypoints number optimization is to be done.

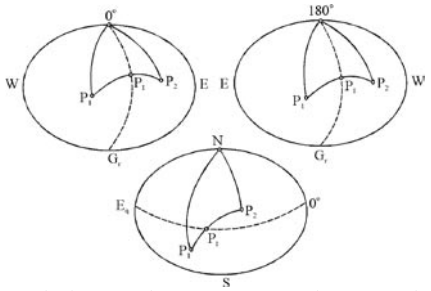


Figure 2. The orthodrome and its intersection with Greenwich and Equator

In the paper has been assumed that the distances between each pair of ports are orthodrome, that is absolutely the shortest ones, but in the practical navigation, they should be replaced with the optimal number of the loxodrome, with the appropriate loxodrome courses. The deviations are also to be involved in aim to avoid obstacles and to enable the eventually predicted route tracking.

Now, let us introduce some mathematical adaptations of the Hopfield-Tank TSP neural algorithm applied to the process of linear ship's route modeling. It must be pointed out that through the detailed survey of ship routing and optimization problems given in [12] by Christiansen, Fagerholt and Ronen (2003), becomes obvious that nobody has treated linear ship's TSP routing problem and its proper modifications by means of neural networks. Namely, this problem has been mostly treated as integer that is binary programming one. Although this approach based upon neural networks is in a way more sophisticates, since it reduces number of boundaries and enables easier subcycles dismissing.

5. THE MATHEMATICAL APPROACH TO THE HOPFIELD-TANK TSP NEURAL ALGORITHM

The traveling salesman problem (TSP) is a classic example of a nondeterministic polynomial *(NP) complete* problem. A problem is assigned to the NP class if it is verifiable in polynomial time by a nondeterministic *turing machine*. While, a nondeterministic *turing machine* is a "parallel" *turing machine* which can take many computational paths simultaneously, with the restriction that the parallel *turing machines* cannot communicate. A problem is said to be NP hard if an algorithm for solving it can be translated into one for solving any other NP problem. It is much easier to show that a problem is NP than to show that it is NP hard. A problem which is both NP and NP hard is called a NP complete problem. Essentially, the only way known to solve NP complete problem, is to compute the costs (here distances) of all tours [13].

There are fiew assumptions in this paper:

1. P represents the number of nodes, and mark the nodes with numbers 1,2,3,... P;
2. We assume that the distances between P nodes are specified in a matrix of distances that specifies the non-negative orthodrome distances between any pair of ports and this matrix is symmetrical: for each pair $(i,j) \in PxP$ distance between node i and node

j is same as distance between node *j* and node *i*. All entries in the matrix have some distance and the distance on the main diagonal are set to zero.

3. Each port must be visited exactly once and no port can be skipped.

The tour is represented as 0 - 1 matrix with *P* rows and *P* columns. If number 1 is in matrix on positin *(i,j)*, that means that the *i*-th node is on the *j*-th position in the tour. There is a variable named *minControl* representing a distance that is assossiated to the tour *1,2,3K,P* which is propagated as the optimal one, in the first run. This is a first tour that is examinated with the algorithm and it is represented with matrix with ones on the main diagonal. After that, algorithm find a next tour, compute the distance, and compare it with a *minControl* that is find so far. If a new generated tour has a distance *d,* and *d* is a smaller distance then *minControl*, we will introduce *d* as a best tour and refresh *minControl* with this computed minimum distance. This proccess continues for all possible tours. If there are *P* ports, then there are *P!* possible tours, so the number of tours to be checked grows very large and very quickly. Exploring all tours is called a "*brute force"* approach or exhaustive search. This algorithm generate a 0 - 1 matrix so that every row and every column has exactly one 1, and every generated matrix is treated as a subomptimal solution. This aproach is implemented within the next pseudo-code:

```
const
  MaxPortNumber = 9;
  D = 500;
  C = 200;
Type
  PortMatrix = array [1..MaxPortNumber, 1..MaxPortNumber] of integer;
  PortMatrixReal = array [1..MaxPortNumber, 1..MaxPortNumber] of real;
  order = array [1..MaxPortNumber] of integer;
var
  matrix : PortMatrix;
  distance : PortMatrixReal;
  tour : order;
  k, m, P: integer;
  s, min, Ec, w: real;
```

{mark that in the algorithm s represents a distance of courent tour, previously in the work labeld as d}

```
procedure Visit (var matrix : PortMatrix; row, col : integer);
var i, j : integer;
      help: order; {generated tour derived from 0-1 matrix}
begin
  if done one solution then
    begin
      for i := 1 to P do
```

```
            for j := 1 to P do
                if matrix[i,j]=1 then
                    help[i] := j;
            s := 0; {compute the curent cost}
            for i := 1 to P-1 do
                s := s + distance[help[i],help[i+1]];
            s := s + distance[help[1],help[n]];
            if s < minControl then
              begin
                min := s;
                best := help;
              end;
        end
    else
repeat
        clear all colums after used column
        if Put (row, col) then
          begin
    matrix [row, col] := 1;
    Put (1, next column in matrix);
          end;
        go to next row in matrix
until last row;
end;
```

Put(i,j) is a function implemented to check if 1 can be in the 0 - 1 matrix on the *(i,j)* position. *Help* array is formed only for better understanding of the algorithm, since the total distance of tour can be computed from the *matrix*. In the main program we must specify *P* and input the distances into distance matrix. After that program will call the esential procedure *Visit* wich will find the best tour. The exponential search space can be seen in the figures 3, 4 and 5, for ten, twelve and fourteen nodes with linear and semilogarithm coordinate axis, respectively.

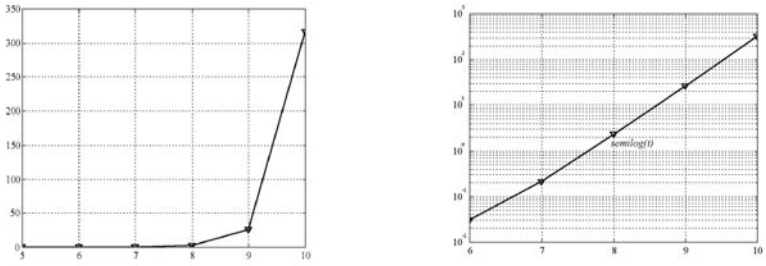


Figure 3. The exponential search space in the case of ten nodes

Lets remark that the 0 - 1 matrix represents a permutation of ports. It is not important which port will be the first one in the optimal tour, since we can rearrange the optimal tour to start from the desired one. If *P=5* and the optimal tour is e.g. 1 5 3 2 4, but if it is necessary to start from node 3, the rearranged tour is 3 2 4 1 5. Here, we will use only those tours that start from node 1. The goal is to find the sequence of ports that starts and ends with city 1 such that the overall passed distance of the tour is minimized. In this case, for *P* ports, it is enough to examine *(P-1)!* suboptimal 0-1 matrices. Pseudo-cod for this improvement is same as the previous one because this is the case where the program generates all permutations for ports *2,3,4,…,P* and puts the port 1 on the first position in each permutation.

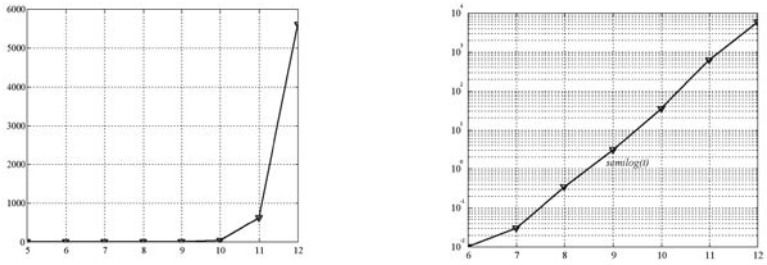


Figure 4. The exponential search space in the case of twelve nodes

Assume that we have to investigate problem of *P=5* ports, and that the total distance d_{total} is computed for the tour 1 2 3 4 5. In the symmetrical TSP case all tours with same adjacency of nodes in tour have the same distance d_{total}, so it is not necessary to compute the distance of the tour 1 5 4 3 2, since we know that the total distance of this tour is also d_{total}. This leads to improving the search space, since only *(P-1)!/2* permutations have to be examined. It is convenient to generate all permutations in lexicographical order.

There is still no algorithm, which can, in general, find the optimal solution for the TSP without suffering from exponentially growing complexity. Further researchers must examine efficient pruning methods for search tree, some kind of branch and bound method, or try to use fast algorithms for generating lexicographical permutations to cut a running time for TSP algorithm. If it is not so important to obtain a true minimal length tour, it is possible to investigate a different heuristic methods which will lead to tour that is near to the optimal one.

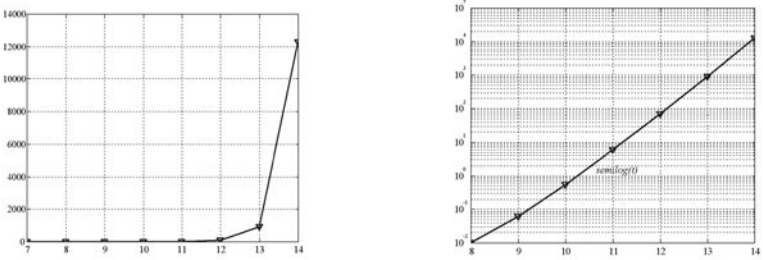


Figure 5. The exponential search space in the case of fourteen nodes

6. THE NUMERICAL EXAMPLES AND SIMULATION RESULTS

The problem being considered here is to find the exact shortest linear ship's round tour visiting fourteen arbitrary chosen ports on the Earth north-east hemisphere in accordance with previously proposed mathematical adaptation of Hopfield-Tank algorithm to the TSP. The observed ports geographical coordinates, that is their latitudes and longitudes are given in degrees and minutes in table 1.

No.	Port	Latitude φ (° ′ S)	Longitude λ (° ′ E)
1.	P_1	45° 20′	14° 20′
2.	P_2	44° 05′	15° 05′
3.	P_3	43° 25′	16° 20′
4.	P_4	42° 45′	18° 05′
5.	P_5	42° 05′	19° 10′
6.	P_6	41° 10′	16° 50′
7.	P_7	42° 35′	14° 03′
8.	P_8	43° 35′	13° 20′
9.	P_9	43° 53′	12° 55′
10.	P_{10}	44° 03′	12° 45′
11.	P_{11}	43° 40′	15° 58′
12.	P_{12}	43° 30′	16° 18′
13.	P_{13}	41° 15′	16° 35′
14.	P_{14}	42° 00′	14° 58′

Table 1. The observed ports geographical coordinates

The distances between each pair of ports can be calculated as the orthodrome one by means of the sphere trigonometry rules, that is by the equation (8):

$$d_{ort}(i,j) = \arccos\left(\sin\varphi_i\varphi_j + \cos\varphi_i\varphi_j \cos\Delta\lambda_{i,j}\right) \text{ for } (i,j) \in P \qquad \textbf{(8)}$$

where φ_i and φ_j are endpoints, i.e. endports, latitudes and $\Delta\lambda_{i,j}$ is an absolute value of the difference between endports longitudes. The problem has been treated as a symmetrical one. Namely, the distances between ports are the orthodrome in both directions while the initial orthodrome courses are different for the opposite directions. The matrix representation of the orthodrome distances between each pair in given set of ports is presented in table 2. The distances are given in nautical miles [Nm] and it is clear that it is not possible to sail from a certain port to itself, that is $d(i,j)=\infty$, if $i=j$. Since it is not possible to represent ∞ in a proper way within the algorithm, this has been achieved by replacing ∞ with zero values. Finally, by the application of in the paper proposed mathematical adaptation of the Hopfield-Tank neural algorithm, the exact optimal TSP solutions, for the cases when *1,2,…,14* ports are to be visit, have been found and given in table 3. It is to be pointed out that the number of possible combinations in the last case is even *(14-1)/2=3113510400.*

The obtained results have been presented through the optimal round tour, its total length and required time for its determination and given in table 3. While in the figure 6 are given the Hopfield-Tank neural network outputs in the case of optimal solution for fourteen ports, as well as, the scheme of linear ship's optimal route for the given ports arrangement. It is to be mentioned that all results are obtained by running executable Free Pascal programs on a 600 MHz Pentium 2 machine with 192 MB of RAM operating under

Windows 2000 system. By the obtained optimal TSP results it becomes possible to calculate Hopfield-Tank recurrent neural network energy minimum and related weight vector (table 4). This is of great importance since it allows the Hopfield-Tank neural network usage in finding the approximately optimal solutions for the similar distances between each pair of ports to those in the given example. Thus, we are in position to use this neural network for solving almost the same problem, with satisfying accuracy, when deviations and other corrections are involved in the calculations of the distances between ports.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
P_1	∞	81.53	143.45	223.97	286.20	272.79	165.45	133.39	105.97	102.41	122.00	138.56	263.94	201.87
P_2	81.53	∞	67.34	153.26	215.40	191.28	100.66	81.46	94.30	100.61	45.65	63.25	182.41	125.10
P_3	143.45	67.34	∞	86.49	148.25	136.81	111.97	130.94	150.94	159.91	21.89	5.20	130.47	104.18
P_4	223.97	153.26	86.49	∞	62.47	110.15	178.20	213.76	235.52	245.18	107.66	90.12	112.12	145.27
P_5	286.20	215.40	148.25	62.47	∞	118.21	228.88	271.91	294.72	304.90	169.75	152.15	126.11	187.19
P_6	272.79	191.28	136.81	110.15	118.21	∞	150.61	212.31	237.77	249.82	154.83	141.98	12.34	97.55
P_7	165.45	100.66	111.97	178.20	228.88	150.61	∞	67.72	92.40	104.70	106.15	112.95	138.52	53.66
P_8	133.39	81.46	130.94	213.76	271.91	212.31	67.72	∞	25.50	37.70	114.47	129.11	200.77	119.14
P_9	105.97	94.30	150.94	235.52	294.72	237.77	92.40	25.50	∞	12.32	132.76	148.56	226.26	144.47
P_{10}	102.41	100.61	159.91	245.18	304.90	249.82	104.70	37.70	12.32	∞	141.04	157.28	238.36	156.77
P_{11}	122.00	45.65	21.89	107.66	169.75	154.83	106.15	114.47	132.76	141.04	∞	17.60	147.54	109.24
P_{12}	138.56	63.25	5.20	90.12	152.15	141.98	112.95	129.11	148.56	157.28	17.60	∞	135.58	107.47
P_{13}	263.94	182.41	130.47	112.12	126.11	12.34	138.52	200.77	226.26	238.36	147.54	135.58	∞	85.33
P_{14}	201.87	125.10	104.18	145.27	187.19	97.55	53.66	119.14	144.47	156.77	109.24	107.47	85.33	∞

Table 2. The distances between ports – symmetrical problem

Number of ports	The optimal tour	Total distance [Nm]	Required time [sec]
1	-	-	-
2	1-2	81.533	0.000
3	1-2-3	148.875	0.000
4	1-4-3-2	459.335	0.000
5	1-2-3-5-4	583.572	0.000
6	1-2-6-5-4-3	683.448	0.010
7	1-2-3-4-5-6-7	732.117	0.000
8	1-2-3-4-5-6-7-8	767.781	0.010
9	1-2-3-4-5-6-7-8-9	765.859	0.060
10	1-2-3-4-5-6-7-8-9-10	774.625	0.540
11	1-2-11-3-4-5-6-7-8-9-10	774.833	5.880
12	1-2-11-12-3-4-5-6-7-8-9-10	775.750	70.620
13	1-2-11-12-3-4-5-6-13-7-8-9-10	776.010	896.130
14	**1-2-11-12-3-4-5-6-13-14-7-8-9-10**	**776.485**	**12 289.430**

Table 3. The optimal round tours and required time for their determination

Number of ports		14	
Energy minimum		-194121.25	
Weight vector			
w [1,2]	-40966.50	w [6,13]	-6371.50
w [2,11]	-23028.50	w [13,14]	-42867.50
w [11,12]	-9002.00	w [14,7]	-27034.00
w [12,3]	-2803.00	w [7,8]	-34061.00
w [3,4]	-43445.00	w [8,9]	-12950.50
w [4,5]	-31435.50	w [9,10]	-6360.50
w [5,6]	-59308.50	w [10,1]	-51408.50

Table 4. The Hopfield-Tank network energy minimum and the optimal weight vector in the case of fourteen ports

On the basis of the proposed method for solving TSP we are in position to solve successfully linear ship's routing problem. But it must be pointed out that it is still only one of many much more complex problems that must be solved previously, as well. Among these problems the most important are scheduling problems, supply and demand requirements, the optimal speed and weather routing, the optimal loading (unloading) problems, etc. Solving some of these problems separately or in combination undoubtedly requires the appropriate modifications of Hopfield-Tank TSP model. These modifications might be realized by adding for example some benefit, risk or cost coefficients to the route legs' distances in aim to emphasize how a certain route legline is convenient or not. After the proper modifications have been realized, it can be possible to apply here proposed TSP method, in the same or very similar manner, at the final stage of solving real, much more complex linear shipping problems.

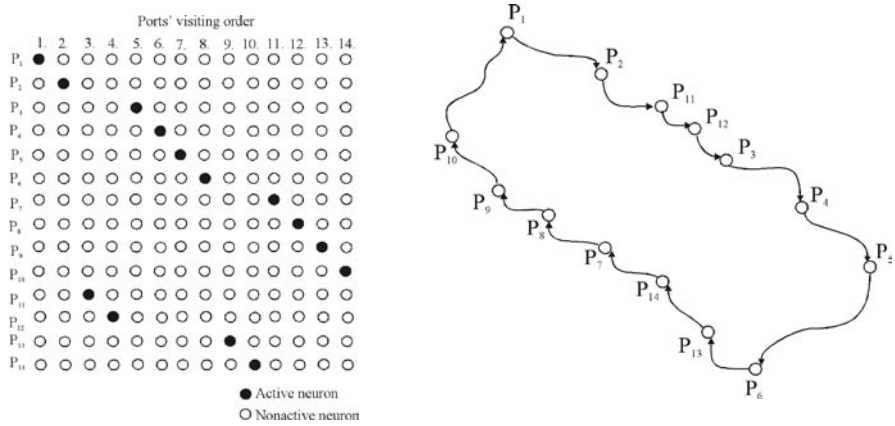


Figure 6. The TSP optimal solution in the case of fourteen ports arbitrary chosen on the Earth north-east hemisphere

CONCLUSIONS

The adaptation of the Hopfield-Tank TSP neural network algorithm to the linear ship's route modeling problem, in the mathematical sense, has been examined in some details. The main differences between classical and here presented TSP are that nodes of the network are given in sphere coordinates (by longitude and latitude) and that distances between them are not linear but nonlinear (orthodrome). The numerical results for the optimal linear ship's round tour, the energy minimum of the Hopfield-Tank neural network and the appropriate weight vector have been given for the case of enough large number of arbitrary chosen ports on the Earth north-east hemisphere. Once trained Hopfield-Tank network for a certain number of nodes, can be used for successful determination of the nearest solution to the optimal one or those that are very near to the optimal one for distances between ports being changed for the values of deviation or some other route corrections.

Although this approach is rather theoretical than practical in nature, it could be undoubtedly useful one in some kind of combination with the restrictions like demand between port pairs, weather and speed conditions are, even in the practical linear ship's route modeling. Namely, some route legs might have a certain priority over the others from some reasons. By adding to the each route legline some benefit, risk or cost coefficients, in aim to emphasize how they are convenient or not, it becomes possible to modify properly the proposed ship's routing model based upon Hopfield-Tank TSP neural algorithm.

It must be concluded, as well, that there is yet no algorithm capable of finding the optimal solution for the TSP without suffering from exponentially growing complexity. Thus, the further research work in this field, in general, independently of linear ship's route modeling, should be oriented toward efficient pruning methods for search tree, that is to some kind of branch and bound method, or to the usage of fast algorithms for generating lexicographical permutations to cut a running time for TSP algorithm.

REFERENCES

Hua-Aa Lu, Modelling Ship's Routing Bounded by the Cycle Time for Marine Liner. *Journal of Marine Science and Technology*, vol. 10, pp 61-67, 2002.

Biggs N. L., Loyd E. K., Wilson R. J., *Graph Theory 1736-1936*, Clarendon Press, Oxford, 1976.

Menger K., Botenproblem, *Ergebnisse eines Mathematischen Kolloquium*, Heft 2, pp 11-12, 1932.

Juang J., Stability Analysis of Hopfield-Type Neural Networks, *IEEE Transactions on Neural Networks*, vol. 10, pp 1366-1374, 1999.

Kakeya H., Okabe Y., Fast Combinatorial Optimization with Parallel Digital Computers, *IEEE Transactions on Neural Networks*, vol. 11, pp 1323-1331, 2000.

Gurney K., *An Introduction to Neural Networks*, London UCL Press, 1997.

Hasson H. M., *Fundamentals of Artifical Neural Networks*, London MIT Press, 1995.

Hopfield J. J., Tank D. W., Neural Computation of Decisions in Optimization Problems, *Biological Cybernetics*, vol. 52, pp 141-152, 1985.

Qiao H., Peng J., Xu Z., Nonlinear Measures: A New Approach to Exponential Stability Analysis for Hopfield-Type Neural Networks, *IEEE Transactions on Neural Networks*, vol. 12, pp 360 -369, 2001.

Bauk S., Solving TSP in Navigation by Application of Hopfield Recurrent Neural Network, *Maritime Transport II*, Barcelona, pp 425-434, 2003.

Bauk S., Avramovi Z., Hopfield Network in Solving Traveling Salesman Problem in Navigation, *Neurel 2002*, Belgrade, pp 207-210, 2002.

Christiansen M., Fagerholt K., Ronen D., *Ship Routing and Scheduling: Status and Perspectives*, Norwegian University of Science and Technology, Trondheim, Norway, 2003.

Sedgewick R., *Algorithms*, Addison-Wesley Publishing Company, Boston, 1988.

LA FORMACIÓN DE LA TRAYECTORIA DEL BUQUE A TRAVÉS DE LA ADAPTACION DEL HOPFIELD - TANK TSP ALGORITMO NEURAL

RESUMEN

En este trabajo se revisa el modo de la fijación de la trayectoria optimal del buque de la línea a través de la aplicación del Hopfield - Tank TSP algoritmo neural, ajustado a la solución del bien conocido problema del viajante de negocios (TSP). La proposición presentada en el trabajo consiste en el acceso matemático a la realización Hopfield - Tank TSP algoritmo neural. Esta principalmente basada en el modo más rápido de como engendrar zero-uno matrices de las soluciones sub-optimales.

Las palabras claves: formación de la trayectoria del buque, TSP - problema del viajante de negocios, Hopfield - Tank TSP algoritmo neural

INTRODUCCIÓN

Como es sabido, con el planeamineto de la trayectoria del buque comienzan todas las operaciones en la navegación marítima, lo que ale especialmente por la navegación de la línea. Como la trayctoria del buque de la línea es circular, se ofrece la comparación con bien estructuradoy común problema del viajante commercial - TSP (Hua - An Lu, 2002). Según este problema, el navegador tiene que completar la vuelta sobre un grupo de los puertos, visitando cada puerto exactamente una vez para disminuir la trayctoria superada. Este tipo del problema se muestra como muy complicado desde punto de la vista del cálculo. Se mostró de que el tiempo para la búsqueda de la solución optimal va creciendo con el crecimiento del número de los puertos visitados. La solución del problema se encuntra escondida; se trata de la solución donde se viene a la interesante aplicación de la inteligencia arteficial. Además de la colada simulada ("simulated annealing") y los algoritmos genéticos, la aplicación de la red Hopfield - Tank recurrente y neural en la solución del problema del viajante de negocios, una es de las metodologias más interesantes. La solución de este tipo no tiene que ser la más optima i no se atraviesa, necesariamente, en el tiempo más corto, pero con certeza penetra en la esencia del problema.

METODOLOGÍA: EL HOPFIELD - TANK TSP ALGORITMO NEURAL

Este trabajo contiene un acceso matemático a la implicación del Hopfield - Tank TSP algoritmo neural y "brute force" búsqueda en hallar de la solución TSP optima entre todas las soluciones sub- optimales posibles.

De acuerdo con Hopfield - Tank neural red para la optimización, TSP de P nudos (en este caso: puerto) puede ser codeada con la red PxP, quiere decir red de las unidades unipolares, sigmoidales y activantes, en la que cada unidad tiene salida uno si el puerto fue visitado,

en el contrario tiene salida zero. Las redes corresponden a los puertos, mientras las columnas corresponden al orden de la visita. Cada línea z cada columna tienen que tener exactamente un neutrón que en la salida da uno, en el marco del cada solución sub - optimal u optimal. La solución optimal coresponde al minimo energético de la red Hopfield - Tank.

En el trabajo se presenta la realisación del engendramiento uno - zero más rápido de la matriz para las soluciones sub optimales. Mientras la técnica "brute force" esta aprovechada en la búsqueda de la optimal, quiere decir exacta TSP solución entre las soluciones posibles sub - optimales.

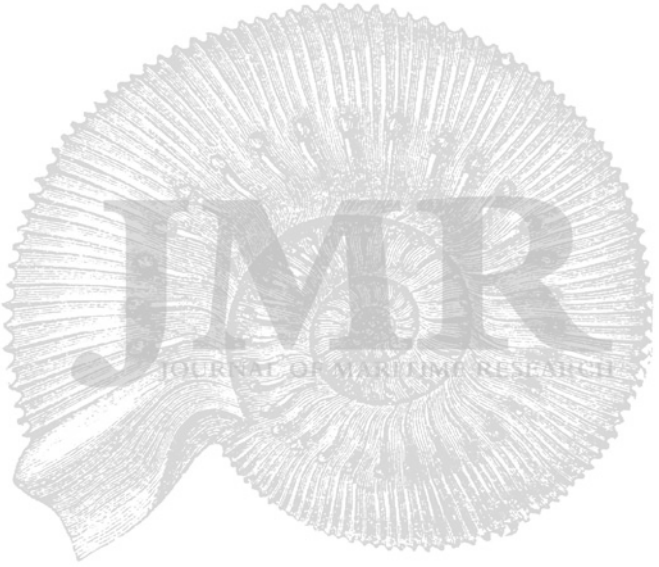
El método aprovechado esta testificado en el ejemplo numérico correspondente de los catorce puertos en el hemisferio nordeste y brinda una solución optimal del TSP en en plazo relativamente breve.

CONCLUSIONES

El ajustaje del Hopfield - Tank TSP algoritmo neural al problema de la formación de la trayectoria del buque de línea esta bien tratado e investigado en el sentido matemático. Las diferencias principales entre el problema TSP clasico y aquí presentado consiste en el hecho de que los nudos de la red son dados en las coordenadas de la esfera (longitudines y latitudines), así como en el hecho que las distancias recorridas entre ellas no son líneares, sino alíneares (ortodromas). Se presentan los resultados matemáticos para la trayectoria optimal del buque de la línea, el mínimo energético y el vector del peso correspondente para el caso del número suficiente grande de los puertos casualmente eligidos en el hemisferio nordeste. Una vez entrenada, la red Hofield - Tank para un número determinado de los nudos puede ser aprovechada con éxito para la destinación de la solución más cercana al mejor o aquellas soluciones que se encuentran cerca del óptimo para las distancias entre los puertos aplicadas para los datos de las virajes o otros cambios de la trayectoria.

Aunque se trata se un acceso en su naturaleza más teórico que práctico, este podria ser aprovechado, sin duda, en una especie de la combinación con las limitaciones como la búsqueda entre los puertos, limitaciones de tiempo i de la velocidad, incluso en la formación de la trayectoria del buque de línea. Es decir, algunos de los segmentos de la traectoria bueden ser de preferidas a los otros, por alguna razón. Si se al cada segmento de la trayectoria añade alguno *benefit*, *risk* ya que *cost* coeficiente, para que se ponga de relieve su conveniencia o inconveniencia, se puede, en modo apropiado, modificar el modelo de la trayectoria del buque basada en el Hopfield - Tank TSP algoritmo neural.

Tambien, se impone la conclusión como todavía no hay algoritmo para el hallazgode la solución del TSP sine crecimiento exponente de la complejidad del cálculo. por esta razón, la investigación siguiente de este campo, en común, independiente de la formación de la trayectoria del buque de línea, tendria que dirigirse a los mètodos de como reducir los árboles de la búsqueda loq signifia alguna especies del algoritmo de la ramificación y limitación, o sea el aprovechamiento de los algoritmos rápidos acortando, de este modo, el tiempo gastado para la solución de TSP.


JMR
JOURNAL OF MARITIME RESEARCH