# THE COMPARATIVE ANALYSIS OF TWO NEURAL NETWORKS MODELS IN THE FUNCTION OF THE LINEAR SHIP'S ROUTE COSTS MINIMIZATION

**Sanja Bauk and Nataša Kovač**
Faculty of Maritime Studies, University of Montenegro

***Abstract :*** *The paper deals with a comparison between two different methods in generating sub-optimal solutions to the Hopfield-Tank TSP (traveling salesman problem) neural algorithm. Namely, the Hopfield-Tank TSP neural algorithm has been applied in the paper to the linear ship's route, that is cycle voyage modeling, which means here finding the optimal visiting order of a given set of ports in such way as to minimize the total sailing distance and implicitly total linear ship's traveling costs. The methods being implemented here into Hopfield-Tank TSP neural structure and computationally compared are: the brute force method and the fast insertion heuristic.*
***Key words:*** *TSP (traveling salesman problem), Hopfield-Tank neural network, brute force method, fast insertion heuristic, linear ship, traveling costs.*

***Apstrakt :*** *U radu se upoređuju dva različita metoda za određivanje suboptimalnih rješenja Hopfield-Tankovog TSP (problem trgovačkog putnika) neuralnog algoritma. Naime, Hopfield-Tankov TSP neuralni algoritam je u radu korišćen u modeliranju kružnog putovanja linijskog broda, u smislu određivanja optimalnog redosleda obilaska grupe luka, na način da se minimizira ukupna dužina i posledično ukupni troškovi putovanja broda. Metodi u radu implementirani u Hopfield-Tankovu neuralnu strukturu i upoređeni u smislu brzine realizacije i tačnosti su: metod primjene čiste sile i heuristika brzog umetanja.*
***Ključne riječi:*** *TSP (problem trgovačkog putnika), Hopfield-Tankova neuralna mreža, metod primjene čiste sile, heuristika brzog umetanja, linijski brod, troškovi putovanja.*

*JEL clasification: L92;*
*Original scientific papers; Recived: May 05, 2006*

## 1. Introduction

It is well-known that the route planning is the beginning of all operations in marine shipping, particularly in the linear one [1,4]. Since the route of a linear ship provides a cycle voyage, it is naturally compared with well-defined general traveling salesman problem (TSP). According to the TSP, 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 its optimal solution grows exponentially with the number of visiting ports. Practically, navigators used to solve TSP in a way as to sail from the starting port to the nearest one, then from the second port, or port of arrival, to the next nearest one, and so on.

In the paper the Hopfield-Tank TSP neural structure has been employed in solving this problem by the implementation of two different methods in generating zero-one matrix of the neural network best or optimal solutions.

One of the proposed methods is based upon brute force algorithm, or exhaustive search, while another one is based upon fast insertion heuristic. The first one is usable, in computational sense, for relatively small number of nodes (up to 15) giving the best, exact, TSP solution; while the second one can be used for hundred of nodes, which is its great advantage, even it gives rather approximate than exact solution to the TSP.

Thus, in the paper we considered the possibilities of the previously mentioned methods implementation to the Hopfield-Tank TSP neural network and compared them computationally. In fact, these two methods have been used for fast finding Hopfield-Tank network energy minimum and the optimal weights vector upon obtaining the appropriate zero-one matrix representing active and non-active neurons of the network.

Someone may ask: why the Hopfield-Tank TSP neural structure besides these two methods? The answer is simple: once, in one of these two ways trained network, can be used later very efficiently in finding the optimal or sub-optimal solutions to the similar TSPs, that is to the TSPs with relatively small changes or deviations in orthodrome distances between some pairs of ports' in the input matrix.

The remaining parts of the paper are organized as follows:

- the second one gives some notes about the TSP;
- the third part gives some basic remarks to the functional equivalence between Hopfield-Tank neural network and TSP model;
- the fourth one considers brute force algorithm application in finding the best, that is exact TSP solution, and its implementation to the Hopfield-Tank TSP neural architecture;
- the fifth part considers heuristics strategy implementation into finding some optimal TSP solution(s) and their mapping to the Hopfield-Tank TSP neural structure;
- the sixth one comprises an appropriate example and the obtained numerical results for the TSP in the case of fourteen arbitrary chosen ports on the north-east Earth hemisphere, and
- the last one contains some conclusion remarks and directions for the forthcoming investigations in this domain.

## 2. The TSP overview

The traveling salesman problem or traveling salesperson problem (TSP) probably is the most prominent problem in combinatorial optimization. Its simple definition along with its notorious difficulty has stimulated, and still stimulates, many efforts to find an efficient algorithm [6,7,9]. Due o the non-deterministic polynomial (NP) completeness of the TSP, among others, a lot of computational 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 solving TSP. In the paper, the idea of the Hopfield-Tank TSP neural algorithm adaptation to the liner ship's route modeling problem, in a mathematical sense, has been explored experimentally by an appropriate example of fourteen nodes, i.e. ports on the north-east hemisphere. Firstly, brute force algorithm has been employed in faster finding zero-one matrices of sub-optimal solutions in the Hopfield-Tank TSP neural structure. Then, an insertion heuristic has been applied in finding the optimal solution(s) to the TSP, which is(are) later mapped onto the Hopfield-Tank neural network, as an intelligent solver to the TSP.

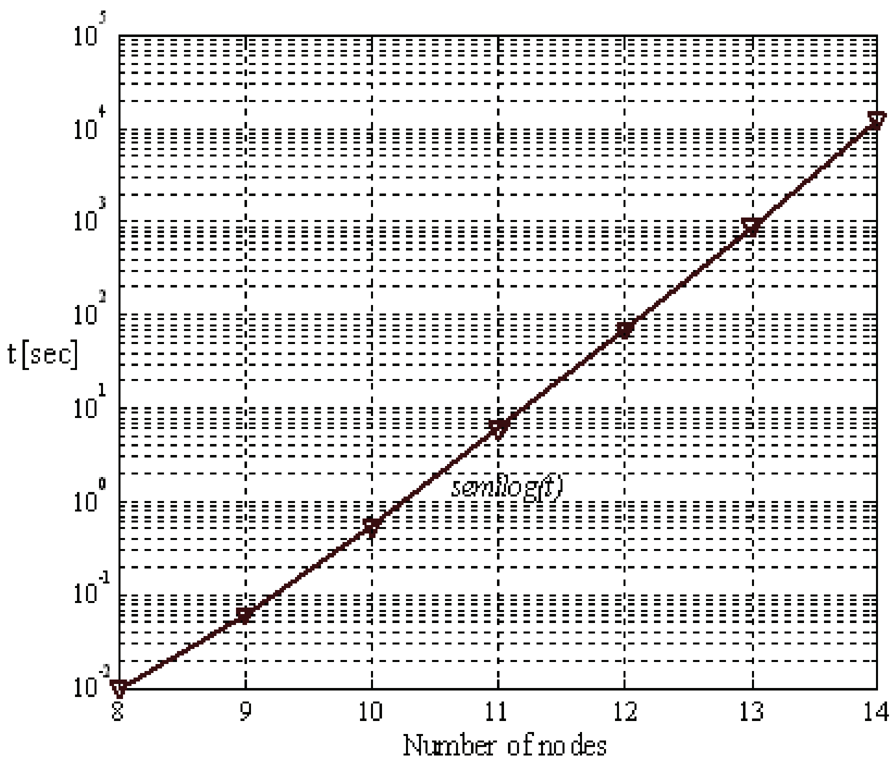
## 3. The functional equivalence between Hopfield-Tank neural network and TSP model

Hopfield and Tank have shown how the TSP can be solved by recurrent neural network. The first step was to map the problem onto the network so that solutions correspond to the states of the network. Thus, the problem for P points, here ports, may be coded into a P by P network as follows: each row of the network corresponds to a point, and the ordinal position of the point in the tour is given by the point at that place outputting a high value (nominally 1), while the rest are all at very low values (nominally 0). The functional equivalence between Hopfield-Tank neural network and TSP model has been mathematically explained in some details in the references [2,3,4,5]; thus, we are not going to explain it again here. But, we shall give some descriptive explanations about brute force and fast insertion heuristic implementation to the Hopfield-Tank TSP recurrent neural network algorithm in the following two sections.

## 4. The brute force approach to the TSP

The traveling salesman problem (TSP) is a classic example of a non-deterministic polynomial (NP) complete problem and essentially the only way known to solve it exactly, is to compute the traveling costs (or lengths) of all possible tours and to chose the best one. Exploring all tours is called a brute force approach or exhaustive search. This algorithm generates a 0 - 1 matrix so that every row and every column has exactly one - 1, and every generated matrix is treated as a sub-optimal solution to the TSP. The pseudo-code (Pascal) of the brute force algorithm is presented in table 1, while its detail description can be found in the reference [4]. After running in the reference [4] proposed program for the exhaustive search or brute force algorithm, the exponential search space can be obtain and presented like this in the figure 1 (for the case of fourteen nodes, here ports).

Figure 1. The exponential search space in the case of fourteen nodes




There is still no algorithm which can in general find the exact solution for the TSP without suffering from exponentially growing complexity, thus the 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

Table 1. The pseudo-code (Pascal) of the brute force algorithm

```
I part

const
  MaxPortNumber = 14;
  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
```

```
II part
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 current 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 columns 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;
```

algorithm. If, it is not so important to obtain a true minimal length (cost) tour, it is possible to investigate different heuristic methods which will lead to the tour that is near to the best one. Consequently, within the next section, the fast insertion heuristic strategy implementation to the Hopfield-Tank TSP neural approach shall be considered.

## 5. The fast insertion heuristic approach to the TSP

The term heuristics is commonly used for algorithms which find solutions among all possible ones, but they do not guarantee that the best one will be found, therefore they may be considered as approximately and not completely accurate algorithms. These algorithms usually find a solution close to the best one and they find it fast and easily. Sometimes they can be accurate, that is they actually find the best solution, but the algorithm is still treated as heuristic until the best solution is proven to be the best [10]. Among available heuristics, construction heuristic, nearest neighbor heuristic, insertion heuristic, saving heuristic - can be distinguished. We shall employ here the insertion heuristic. It means that starting tour is some tour on three points, in the case of symmetrical TSP, while a new point is always inserted into the tour at the place that causes the minimal increase in the length (here cost) of the tour. The detail explanation of the insertion heuristic algorithm can be found in the reference [8]. It has been used in the paper in finding the optimal Hopfield-Tank TSP neural network outputs. The first step was to map the problem onto the network so that solutions correspond to the states of the network. Through the obtained network optimal outputs by the proposed insertion heuristic, its optimal weights and energy minimum shall be easily calculated. Once in this way trained network may be used latter efficiently as a general TSP solver. The similar was also done in the case of brute force approach; with the difference that brute force algorithm needs more executing time, but it always gives the exact solution at the end.

## 6. The numerical example

The main problem being considered here is in fact finding the exact, and then optimal, linear ship's round tour visiting fourteen ports on the Earth north-east hemisphere in accordance with previously defined TSP, firstly by the brute force, and later by the insertion heuristic strategy implementation to the Hopfield-Tank neural model. The observed ports' geographical coordinates that is their latitudes and longitudes, as well as, their linear approximations in X-Y plane are given in table 2 and presented graphically in 2D space in figure 2. The TSP results obtained by the programs being realized in Pascal [4,8] are given in the next two sub-sections.

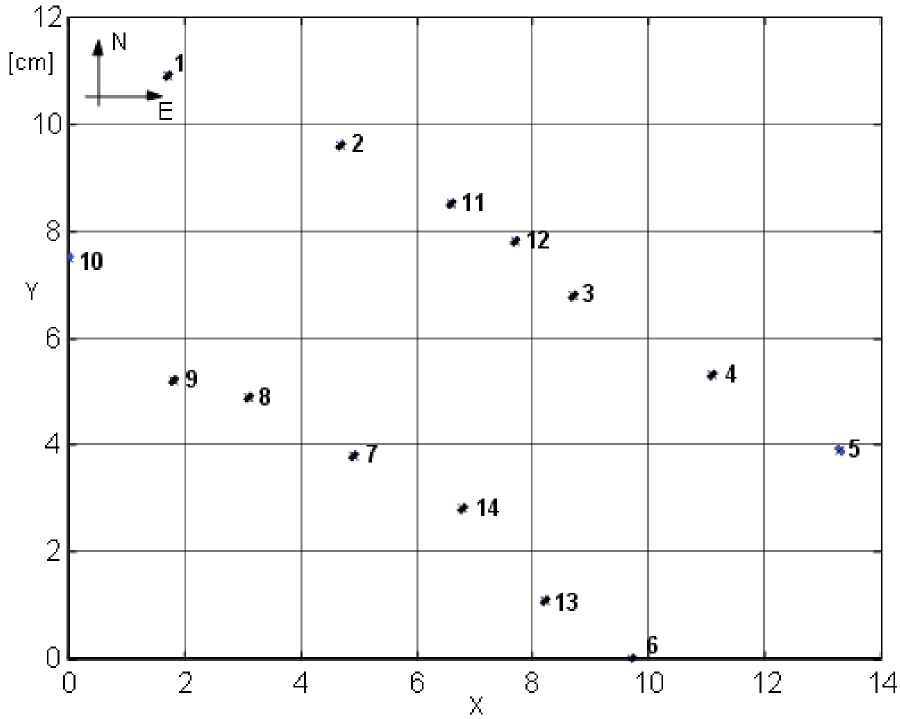
### 6.1 The original exact solution obtained by the brute force [4]

The obtained results in the case of brute force algorithm implementation to the Hopfield-Tank TSP neural network have been presented through the following best round tour: 1-2-11-12-3-4-5-6-13-14-7-8-9-10-1 (*figure 2), its total length is: 776.485 [Nm] and required time for its determination is: 12 289.438 [sec]. The results are obtained by running the Pascal program [4] on a 600 MHz Pentium 2 machine with 192 MB of RAM, operating under Windows 2000 system. The layout of the Hopfield-Tank neurons' active and non-active outputs for the best obtained solution in the examined case of fourteen ports has been given in figure 3. By the obtained exact TSP solution it becomes possible to calculate Hopfield-Tank recurrent neural network energy minimum and related weight

Table 2. The ports' sphere coordinates and their X-Y plane approximations

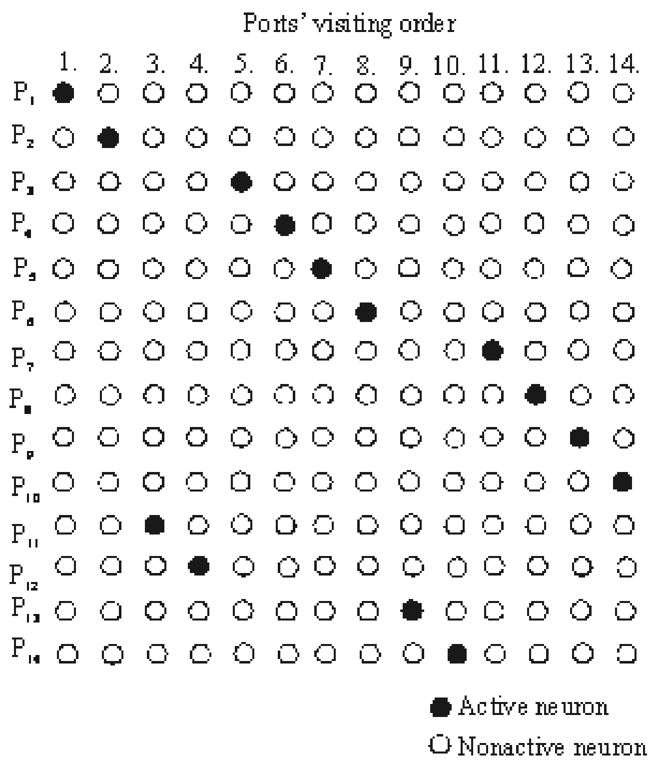
| Port no. | Latitude φ (° ′ N) | Longitude λ (° ′ E) | X [cm] | Y [cm] |
|---|---|---|---|---|
| 1 | 45° 20′ | 14° 20′ | 1.7 | 10.9 |
| 2 | 44° 05′ | 15° 05′ | 4.7 | 9.6 |
| 3 | 43° 25′ | 16° 20′ | 8.7 | 6.8 |
| 4 | 42° 45′ | 18° 05′ | 11.1 | 5.3 |
| 5 | 42° 05′ | 19° 10′ | 13.3 | 3.9 |
| 6 | 41° 10′ | 16° 50′ | 9.7 | 0.0 |
| 7 | 42° 35′ | 14° 03′ | 4.9 | 3.8 |
| 8 | 43° 35′ | 13° 20′ | 3.1 | 4.9 |
| 9 | 43° 53′ | 12° 55′ | 1.8 | 5.2 |
| 10 | 44° 03′ | 12° 45′ | 0.0 | 7.5 |
| 11 | 43° 40′ | 15° 58′ | 6.6 | 8.5 |
| 12 | 43° 30′ | 16° 18′ | 7.7 | 7.8 |
| 13 | 41° 15′ | 16° 35′ | 8.2 | 1.1 |
| 14 | 42° 00′ | 14° 58′ | 6.8 | 2.8 |

Figure 2. The ports' arrangement in X-Y plane




*The best tour is: 1-2-11-12-3-4-5-6-13-14-7-8-9-10-1

Figure 3. The TSP optimal solution in the case of fourteen ports arbitrary chosen on the Earth north-east hemisphere



vector. This is of great importance since it allows the Hopfield-Tank neural network usage in finding the optimal solutions for the similar distances between each pair of ports to those in the given example. In such way becomes possible to use this neural network structure for solving similar TSP problems, with satisfying accuracy, e.g. when the deviations and some other corrections are involved in calculating the distances between ports.

### 6.2 The optimal solution(s) obtained by the insertion heuristic [8]

The exact solution for an example of fourteen points (ports) obtained by the brute force algorithm may be used as a test example for the insertion heuristic method. The applied insertion heuristic algorithm [8] has been tested for 10000 different starting tours randomly selected from (14-1)!, i.e. 3 113 510 400 possible tours. The obtained results are extremely good.

Table 3. The numerical efficiency comparison between brute force and insertion heuristic approaches to the TSP

| | Brute force | Insertion heuristic |
|---|---|---|
| | | |
| Number of tours | 3 113 510 400 | 10 000 |
| Time | 12 289.430 [sec] | 0.05 [sec]* |
| | | *Average time for 10 000 tours |
| The best tour | 776.485 [Nm] | 776.485 [Nm]* |
| | | * Obtained in 7 331 cases, i.e. in 73.31% cases of 10 000 tested tours |

In the case of 10 000 tested different starting tours: the optimal tour length of 776.485 [Nm], with points visiting order: 1-2-11-12-3-4-5-6-13-14-7-8-9-10-1 (obtained by the "brute-force" algorithm, after 12 289.430 sec) is found in over 7 331 cases, which is 73.31% of all tested cases. The worst found tour of 1089.824 total lengths (which is 40.35% greater then the best one) was obtained only one time. The worst calculating time of 0.05 [sec] was obtained for the tour: 5-13-14-7-8-9-10-1-2-11-12-3-4-6-5. The numerical comparison of brute force and insertion heuristic has been given in table 3. Since it is obvious that the insertion heuristic algorithm gives good results for small number of visiting points, thus, we can assume that it will give good results for larger number of points (here ports), as well. But it is still rather an assumption than a guarantee.

According to the presented numerical results in table 3, it is obvious that brute force algorithm requires much more execution time than insertion heuristic and additionally, insertion heuristic gives exact solution in more than 73% within satisfying small time of execution.

## Conclusions

The TSP is well-known as a non-deterministic polynomial (NP) hard problem and it consists of determining the shortest Hamiltonian cycle in the graph. Its numerous applications arise in distribution management, scheduling, routing and manufacturing. The TSP usually appears as vehicles' or linear ships' rou-ting sub-problem. Here, it is treated as TSP with the nodes being represented by fourteen arbitrary chosen ports' coordinates on the Earth north-east hemisphere and it has been solved by the implementation of the Hopfield-Tank neural algorithm. Even the Hopfield-Tank recurrent neural network application to the TSP is an interesting methodology; the TSP has been mostly treated as linear integer or binary programming problem, so far. But, it is to be pointed that the TSP approach based upon neural networks is undoubtedly more sophisticated, since it reduces number of boundaries and enables easier sub-cycles dismissing.

I n the paper, the brute force algorithm and the fast insertion heuristic for solving TSP problem and its implementation to the Hopfild-Tank neural network as TSP solver have been considered. By the proposed methods we are in position to solve successfully linear ship's route modeling problem. It has been shown that the brute force approach gives exact solution and requires large execution time, while insertion heuristic strategy gives rather approximate solutions but in considerably smaller time interval. It must be pointed out, that TSP solving is still only one of many much more complex problems that are to be solved previously in ship's routing. Among these numerous problems, as the most important can be mentioned - scheduling problems, supply and demand requirements, the optimal speed and weather routing, the optimal loading (unloading) problems, etc. Solving some of these problems sepa-rately or in combination, primarily in the aim to minimize total costs, requires the appropriate modifications of the proposed methods. These modifications might be realized by adding for example some costs, benefit or risk coefficients (indexes) to the route legs' lengths in aim to emphasize how a certain route leg-lines are convenient or not. After realization of these modi-fications, it would be possible to apply here proposed TSP methods at the final stage of solving some real linear ship's routing problems, primarily in the function of the total traveling costs minimization, what should be the subject of our forthcoming investigations.

## References

[1] Hua-an 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.

[2] Bauk S., Avramović Z., Hopfield Network in Solving Travelling Salesman Problem in Navigation, Neurel Conference Proceedings, Belgrade, S&M, 2002. (pp 207-210)

[3] Bauk S., Kovač N., Improving Hopfield-Tank TSP Algorithm by the Insertion Heuristic, SYM-OP-IS Conference Proceedings, Vrnjačka Banja, S&M, 2005.

[4] Bauk S., Kovač N., Modeling Ship's Route by the Adaptation of Hopfield-Tank TSP Neural Algorithm, Journal of Maritime Research, Vo. 1, No. 3, 2004. (pp 51-66)

[5] Bauk S., Solving TSP in Navigation by Application of Hopfield Recurrent Neural Network, Maritime Transport 2nd Conference Proceedings, Barcelona, 2003. (pp 425-435)

[6] Christofides T., Vehicle Routing in the Traveling Salesman Problem, Lawler, Lenstra, Kan and Shmoys, eds., John Wiley and Sons, New York, 1985.

[7] Garey M.R., Johnson D.S., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.

[8] Kovač N., Bauk S., The Heuristic Strategy Implementation to the Hopfield-Tank TSP Neural Algorithm, Advancement of Modelling and Simulation Journal – section: production engineering and management, 2005. (accepted)

[9] Lawler E. L., Lenstra J. K., et al., The Traveling Salesman Problem, John Wiley and Sons, New York, 1985.

[10] Sedgewick R., Algorithms, Addison-Wesley Publishing Company, Boston, 1988.

# *KOMPARATIVNA ANALIZA DVA MODELA NEURALNIH MREŽA U FUNKCIJI MINIMIZACIJE TROŠKOVA PUTOVANJA LINIJSKOG BRODA*


## *Zaključak*

*Problem trgovačkog putnika (TSP – Traveling Salesman Problem, eng.), kao nedeterministički polinomijalan problem (NP – non-deterministic polynomial, eng.), sastoji se u određivanju najkraćeg Hamiltonovog ciklusa u grafu. TSP se obično tretira kao potproblem problema rutiranja vozila ili linijskih brodova. Ovdje se TSP rješava nad čvorovima koji predstavljaju četrnaest proizvoljno odabranih luka na sjeveroistočnoj hemisferi Zemlje, primjenom Hopfield-Tankovog neuralnog algoritma. Iako je primjena Hopfield-Tankove rekurentne neuralne mreže u rješavanju TSP-a interesantna metodologija; TSP je do sada uglavnom rješavan kao problem linearnog, odnosno, binarnog programiranja. Međutim, treba naglasiti da je pristup rješavanju TSP-a, baziran na neuralnim mrežama, nesumnjivo efikasniji, jer smanjuje broj ograničenja i omogućuje jednostavnije izbjegavanje podciklusa.*

*U radu su razmotreni metodi implicitne enumeracije (primjene čiste sile) i heuristike brzog umetanja, te njihova implementacija u Hopfild-Tankovu neuralnu mrežu kao rješavač TSP-a. Uz pomoć predloženih metoda, u poziciji smo da uspješno riješimo problem modeliranja rute linijskog broda. Pokazano je da metod primjene čiste sile daje (tačno) najbolje rješenje i zahtijeva prilično dugo vrijeme izvršenja, dok metod heuristike brzog umetanja daje uglavnom aproksimativna rješenja, ali u znatno kraćem vremenu. Treba istaći da je rješavanje TSP-a samo jedan manji dio problema koje treba riješiti prilikom modeliranja rute linijskog broda. Među brojnim problemima ovdje spadaju: problemi raspoređivanja, problemi ponude i potražnje, te problemi određivanja optimalne brzine i optimalne vremenske rute, problemi optimalnog ukrcaja i iskrcaja i sl. Rješavanje ovih problema pojedinačno ili u kombinaciji, prvenstveno u cilju minimizacije ukupnih troškova linijskog broda, zahtijeva odgovarajuće modifikacije predloženih metoda. Ove modifikacije bi se mogle npr. realizovati dodavanjem cost, benefit ili risk koeficijenata (indeksa) pojedinim segmentima rute, s ciljem naglašavanja koliko su oni pogodni ili ne. Nakon realizacije ovih modifikacija, omogućila bi se primijena predloženog TSP metoda na finalnom nivou rješavanja nekih realnih problema rutiranja linijskog broda, prvenstveno u funkciji minimizacije ukupnih troškova putovanja, što bi trebao da bude predmet budućih istraživanja.*