International Journal of Computer Science
Oddity...Probe...Reviste...

http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

Hop-Field Model Using Travelling Salesman Problem

Dr.S.S.Dhenakaran [#1], M.Saranya [*2]

[#1] Assistant professor, Department of Computer science and Engineering,
Alagappa University, Karaikudi.
India.

[*2] Department of Computer science and Engineering,
Alagappa University, Karaikudi.
India.

Abstract-*TSP is a classical example of optimization and constrain satisfaction problem which falls under the family of NP-complete of problems. I have discussed an implementation of an algorithm in neural network for an approximate solution for Travelling Salesman's Problem. Now I used Continuous Hopfield network to find the solution for the given problem. The algorithm gives near optimal result in most of the cases for up to 100 cities.*

Keywords-TSP, Hop-field, Optimization.

1. Introduction

Neural Network is the representation of brain's learning approach. This brain operates as multiprocessor and has excellent interlinked. Neural Network also can be represented as "Parallel distributed processing" planning. It is utilized in the computer applications for solving the complicated problems. There are many benefits from Neural Network such as no requirements for specifying the relevant factors, an unsophisticated model which has many factors for performance, a Straightforward model, fault tolerance and an innate synchronous. The traveling salesman problem is a classical optimization problem,

Which we can find in a lot of Algorithm's book. It is a NP-hardness (Nondeterministic Polynomial time hardness) problem, and a lot of AI's methods can find the approximate solution in this combinatorial optimization problem, ex: Genetic Algorithm (GA), Simulated annealing (SA), and Artificial Neural Network (ANN). In my classification, it is divided into the theory domain and in Optimization and Associative Memory domain (followed by IEEE Comm. 98).

Hopfield explored an innovative method to solve combinatorial optimization problems in 1986. He has implemented a neural network of the Hopfield Model (HM) into an electric circuit that produces approximate solutions to the TSP quite efficiency. Hopfield-Tank algorithm (1987) is another method, which implements TSP but has more efficiently. Then so many algorithms in ANN are proposed that also can solve TSP (Annealed NN, SOM…), and lots of papers change some steps from Hopfield to improve its drawback.

2. The Traveling Salesman Problem

Definition

Given a set of cities and the cost of travel (or distance) between each possible pairs, the TSP, is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance).

The TSP is a classical optimization problem. Let G = (V, A) be a graph where V is a set if vertices and A is a set of arcs between vertices and each arc is associated with a non negative cost. The TSP consists if finding the tour of shortest length that passes through every vertex exactly once.

There are a special case of the TSP with triangular inequality is the Euclidean traveling salesman problem (E-TSP). The E-TSP is to find a closed tour if minimum length through points that are given in two-dimensional space where the distances are computed according to the Euclidean metric. It can be defined as the same as TSP, and also is a NP-hardness problem. In an instance with N cities, there are (N! /2N) distinct tour. Thus, it is unlikely that we can find polynomial time algorithm for solving this problem exactly TSP is a representative of a large class of problems known as combinatorial optimization problems. Among them, TSP is one of the most important, since it is very easy to describe, but very difficult to solve. Actually, TSP belongs to the NP-hard class. Hence, an efficient algorithm for TSP (that is, an algorithm computing, for any TSP instance with *m* nodes, the tour of least possible cost I in polynomial time with respect to *m*) probably does not exist. More precisely, such an algorithm exists if and only if the two computational classes P and NP coincide, a very improbable hypothesis, according to the last years research developments. From a practical point of view, it means that it is quite impossible finding an exact algorithm for any TSP instance with *m* nodes, for large *m*, that has a behaviour considerably better than the algorithm which computes any of the $(m-1)!$ possible distinct tours, and the returns the least costly one.

Complexity

Given *n* is the number of cities to be visited, the total number of possible routes covering all cities can be given as a set of feasible solutions of the TSP and is given as $(n-1)!/2$.

Classification

Broadly, the TSP is classified as symmetric travelling salesman problem (sTSP), asymmetric travelling salesman problem (aTSP), and multi travelling salesman problem (mTSP).

This section presents description about these three widely studied TSP.

Symmetric TSP

Let $V = \{v1, ..., vn\}$ be a set of cities, $A = \{(r, s) : r, s \in V\}$ be the edge set, and $drs = dsr$ be a cost measure associated with edge (r,s) A .The sTSP is the problem of finding a minimal length closed tour that visits each city once. In this case cities i v V are given by their coordinates ( ) $,ix$ yi and drs is the Euclidean distance between r and s then we have an Euclidean TSP.

Asymmetric TSP

If $drs \neq dsr$ for at least one (r,s) then the TSP becomes an aTSP.

Multi TSP

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

The mTSP is defined as: In a given set of nodes, let there are m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be defined in terms of distance, time, etc. Possible variations of the problem are as follows: ***Single vs. multiple depots***: In the single depot, all salesmen finish their tours at a single point while in multiple depots the salesmen can either return to their initial depot or can return to any depot keeping the initial number of salesmen at each depot remains the same after the travel.

Number of salesmen: The number of salesman in the problem can be fixed or a bounded variable. *Cost*: When the number of salesmen is not fixed, then each salesman usually has an associated fixed cost incurring whenever this salesman is used. In this case, the minimizing the requirements of salesman also becomes an objective. *Timeframe*: Here, some nodes need to be visited in a particular time periods that are called time windows which is an extension of the mTSP, and referred as multiple traveling salesman problem with specified timeframe (mTSPTW). The application of mTSPTW can be very well seen in the aircraft scheduling problems.

Other constraints: Constraints can be on the number of nodes each salesman can visit, maximum or minimum distance a salesman travels or any other constraints. The mTSP is generally treated as a relaxed vehicle routing problems (VRP) where there is no restrictions on capacity. Hence, the formulations and solution methods for the VRP are also equally valid and true for the mTSP if a large capacity is assigned to the salesmen (or vehicles). However, when there is a single salesman, then the mTSP reduces to the TSP (Bektas, 2006).

3. Mathematical formulations of TSP

The TSP can be defined on a complete undirected graph $G = (V,E)$ if it is symmetric or on a directed graph
$G = (V,A)$ if it is asymmetric.
The set V ={1, . . . , n} is the vertex set,
$E = \{(i, j) : i, j \in V, i < j\}$ is an edge set and $A = \{(i, j) : i, j \in V, i \neq j\}$ is an arc set. A cost matrix ($C = cij$) is defined on E or on A. The cost matrix satisfies the triangle inequality whenever $cij \leq cik + ckj$, for all i , j , k . In particular, this is the case of planar problems for which the vertices are points $Pi = (Xi ,Yi)$ in the plane, and
cij= $(Xi - Xj)^2 + (Yi - Yj)^2$ is the Euclidean distance. The triangle inequality is also satisfied if cij is the length of a shortest path from I to j on G.

4. THE HOPFIELD MODEL

The HNN is a single layer feedback network consisting of a large number of neurons which are fully interconnected.
The dynamics of each neuron can be given by

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \mathbf{TV} + I_i \qquad ----1$$

where t is a time constant, ***T*** is the connection matrix, *Ii* is the bias current, and ***V*** is a vector composed of the output

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

Vi of neuron *i*. The relation between the input *ui* and the output *Vi* is characterized by a monotonically increasing function such as a sigmoid, or a piecewise linear function.

As long as the neuron has a sufficiently high gain, the first term in (1) can be neglected. In that case, the HNN has
the Lyapunov energy function

$$E = -\frac{1}{2}\mathbf{V}^T\mathbf{T}\mathbf{V} - \mathbf{V}^T\mathbf{I}, \qquad \text{----2}$$

and moreover we may note the following relations hold:

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial V_i}, \text{ and } \frac{dE}{dt} \le 0. \qquad \text{---3}$$

This means that the energy function monotonically decreases with the evolution of the network's state, and when the network reaches the final stable state, the energy function falls into a local minimum. The general method of applying
the HNN to solve optimization problems is to map the objectives and constraints involved in the problem into an energy function, and then obtain the neuron's dynamic equation by means of (3

5. THE BASIC PRINCIPLE OF STABLE STATE ANALYSIS

Before we state the basic idea of the stable state analysis technique, some terminology has to be first defined.

A. A classification of the problems solvable by the HNN

The Hopfield energy function may contain several energy terms, which may be roughly classified into constraint terms and the objective function *F(V)*, where *V* = {*V*1,*V*2, ..., *Vn*} is the set of outputs. The constraint terms may be further classified into zero-constraint terms, nonzero-constraint terms and hybrid-constraint terms.
Zero-constraint terms are used to constrain the network to converge to binary values. The general expression of these terms may

be given by $Co=\sum_{t}^{n} Vt(1\text{-}Vt)$ .If *V* can be

divided into *K* subsets *O*1, *O*2, …, *OK*, and only one nonzero output is allowed in each subset, then zero-constraint terms may be also expressed as

$$C_0 = \sum_{s}^{K} \sum_{\substack{V_i,V_j \in O_s \\ i \ne j}} V_i V_j .$$

A nonzero-constraint term is used to constrain that a subset *Ps* of *V* has *Ms* nonzero elements at the stable state.
This constraint can be given by

$$C_{P_s} = \left(\sum_{V_i \in P_s} V_i - M_s\right)^2 ,$$

, where *Ms* is a positive integer, and in the most general case *Ms* = 1. Other constraint terms are simply called hybrid-constraint terms. We use *CQs* to denote the hybrid-constraint term imposed on the subset *Qs* of *V*.
Generally, the Hopfield energy function can be expressed as a weighted sum of the above constraint terms and the
objective function, Now we

$$E = AC_0 + B\sum_{s}^{N_1} C_{P_s} + C\sum_{s}^{N_2} C_{Q_s} + D \cdot F(V). \qquad \text{---4}$$

propose a simple classification of the problems solvable by the HNN.

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

B. Experimental results and performance analysis

A large number of experiments have been done on 1000 ten-city TSPs and a 51-city TSP. In order to make a comparison, both our method and Aiyer's method are employed to process these problems. Before stating these results, some relating issues should be first clarified.

For Aiyer's method, the following dynamic equation, which is obtained from the expressions of the connection matrix and bias current provided in, is adopted:

$$\frac{du_{xi}}{dt} = -A\sum_{j\neq i} V_{xj} - B\sum_{y\neq x} V_{yi} - 2A1\cdot V_{xi} - C1\sum_{y}\sum_{j} V_{yj} - D\sum_{y\neq x} d_{xy}\left(V_{y,i-1} + V_{y,i+1}\right) + C\cdot N \quad \text{---5}$$

Where $C1 = C - \frac{2(AN - A + A1)}{N^2}$,

and N is the number of cities. A piece-wise linear function is used to characterize the neuron's input-output relation. In simulations, the parameters in (5) are set to the empirical values given in it. That is, $A = B = 8$, $A1 = 7.75$, $C = A/N$, $D = AN/80$, and the time step size is set to 0.02 for ten-city TSPs, while 0.05 for the 51-city TSP.

For our method, we use the sigmoid function to simulate the neuron's input output relation. The parameter $u0$ in this function is set to 0.01. Time step size $\Delta t = 0.00001$. The iteration number in all simulations is fixed to 2000. To judge a solution after convergence, we use the following method: Let i vary from 1 to N. If all entries in the ith column are zeros, then this solution is considered to be invalid, otherwise find the largest entry in this column, say Vxi, and x is considered as the ith city to be visited. Set all Vxj's ($j>i$) to zero, and repeat this procedure until $i = N$.

1) Ten-city TSP

1000 ten-city Euclidean TSPs are generated and the optimal solution to each problem is obtained through an exhaustive search. When using Aiyer's method and our method to solve these problems, we let the two methods use the same initial value for a specific neuron, which is equal to a deterministic value plus a random perturbation: $V\ (t\ )\ N\ R\ xi\ xi\ = 0\ = 1\ /\ + \cdot\delta$ (11) where R is a constant, and δxi is a random number between -1 and 1.

Fig. 1 shows the mean errors of the tour length. In each case, R is given a series of values. For each value of R, both Aiyer's method and our method are used to solve the 1000 10-city TSPs. Parameters A, B, C and D used in our method are set to 40, 40, 800 and 400, respectively.

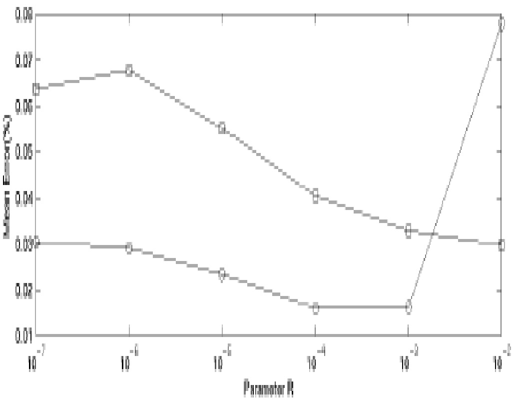

Fig 1

In each experiment, each of the two methods can produce a valid solution. From Fig. 2, we can find that both methods can generate solutions of very high quality, with our method providing slightly better results when the random number (or parameter R) is relatively small. Referring to the results reported. We may notice that the

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

performance of our method is not only better than that of the algorithm based on the polytype concept (whose mean error to 10-city TSPs is 0.73%), but also in some cases comparable to the performance of the three-opt search or simulated annealing.

2) 51-city TSP

The 51-city TSP problem used in this paper is taken from the example *eil*51.*tsp* in TSP LIB which has an optimal tour of 429.983. Again, both Aiyer's method and our method are employed to test this problem. *R* is given 1e-5. In 200 experiments, Aiyer's method can produce 176 (88%) valid solutions with the mean error of 24.29%, and the best tour has a length of 461.256 which is 7.27% longer than the optimal tour, while using our method when parameters *A*, *B*, *C* and *D* are set to 15, 15, 1000 and 500 respectively, we have obtained 100 solutions in 100 experiments with the mean error of 14.93%, and the best solution has a length of 442.824 which is 3% longer than the optimal tour.

To investigate the sensitivity of parameter values to solutions, we fix *C* and *D* to be 1000 and 500, respectively, while let *A* (and *B*) equal to a series of values from 0 to 25. For each case, we have done 100 experiments and calculated the mean error of solutions, which is shown in Fig. 3. From the results we may find that although the quality of solution is affected by the parameter values, this procedure does not change sharply. The conclusion that it is detrimental to set *A* to a too small value can be also verified here.

6. PROBLEMS FACED

- The understanding of Energy function initially was difficult. As this Hopfield network is not a usual Character recognition problem, which was solved initially. But once I went through the literature and papers, it became very clear.
- The understanding of output and activation and weight update functions also was initially difficult, but became very clear later.
- The setting for various parameter values like A1, A2, A3, A4, λ, τ, m, etc was a challenge. The best value was chosen by trial and error. Improvement is still possible for this parameters value.
- Many times the algorithm converged to local minima instead of global minimum. This problem was mostly resolved by adding a random noise to the initial inputs of the system.
- The testing of algorithm gets difficult as the number of cities increase. Though there are few software and programs available for the testing, none of them guarantees the optimal solution each time. So an approximation was made during the testing of the algorithm.

7. CONCLUSION

The optimization ability of Hopfield networks has been debated for many years.

I have discussed an implementation of an algorithm in neural network for an approximate solution for Traveling Salesman's Problem. Now I used Continuous Hope field network to find the solution for the given problem. The algorithm gives near optimal result in most of the cases for upto n cities. In this paper, we have shown the applicability of this technique to the TSP-HNN. Setting the parameters to appropriate values according to the constraints obtained by means of this technique, we can always obtain valid solutions to a 51-city TSP, which on average are about 15% longer than the optimal tour.

References

International Journal of Computer Science
Oddity...Probe...Reviste...
http://www.ijcsjournal.com Volume 1, Issue 2, No 2, 2013. ISSN: 2348-6600
Reference ID: IJCS-015 PAGE NO: 074-080

1. J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol.52, pp.141-152,1985.

2. Rajesh Matai1, Surya Prakash Singh2 and Murari Lal Mittal"Traveling SalesmanProblem: An Overview of Applications, Formulations, and Solution Approaches"

3. Gang Feng and Christos Douligeris, "Using Hopfield Networks to Solve Traveling Salesman Problems Based on Stable State Analysis Technique",Department of Electrical and Computer Engineering, University of MiamiCoral Gables, FL 33124-0640, USA.

4. Ritesh Gandhi,"Traveling Salesman's Problem's Solution using Hopfield ", Department of Electrical and ComputerEngineering,rgandhi@ece.uic.edu .

5. James R. Anderson t "Neural Networks and NP·complete Optimization Problems Complex Systems 2 (1988)" 59-89

6. Paulo Henrique Siqueira, Sérgio Scheer, and Maria Teresinha Arns Steiner "A Recurrent Neural Network to Traveling Salesman Problem",*Federal University of Paraná Curitiba, Paraná, Brazil*

7. Graupe, Daniel "Principles of Artificial Neural Networks. Advanced Series on Circuits and Systems" – Vol.3.

8. Hopfield, J.J and Tank, D.W., neural Computation of Decision in Optimizations Problems, Biol. Cybern. No. 52.

9. Ugur HALICI,"Combinatorial Optimization by Neural Networks".

10. Jean-Yves Potvin Centre de Recherche sur les Transports,"The Traveling Salesman Problem:A Neural Network Perspective"C.P. 6128, Succ. A,Canada H3C .

11. Marco Budinich and Barbara Rosario "A Neural Network for the Travelling Salesman Problem with a Well Behaved Energy Function" Dipartimento di Fisica & INFN, Via Valerio 2, 34127 Trieste, Italy.

Published by SK Research Group of Companies (SKRGC).