Pergamon

Computers ind. Engng Vol. 33, Nos 3–4, pp. 473–476, 1997

0360-8352/97 $17.00 + 0.00

PII: S0360-8352(97)00171-X

A Competitive Neural Network Algorithm for Solving vehicle Routing Problem

Abdolhamid Torki, Samerkae Somhon and Takao Enkawa

Department of Industrial Engineering & Management, Tokyo Institute of Technology, Tokyo, Japan

Abstract

This paper describes a self organization Neural Network algorithm for a class of Vehicle Routing Problems. Motivated by the outstanding performance of adaptive Neural Network approach in the Traveling Salesman Problem, we devised an algorithm to extend the domain of applicability of this approach to more complex problems. First, relevant adaptation is proposed to refine the model for the Multiple Traveling Salesman Problem. Then, an additional mechanism to satisfy further constraints are embodied into the algorithm. The effectiveness of the proposed algorithm is evaluated by considering a series of standard problems from the literature. The results show that the algorithm can yield solutions within a few percent of optimality.

keywords: Neural Network, Vehicle Routing Problem,Combinatorial Problems.

1. Introduction

Recently, Artificial Intelligence has played an active role in optimization contexts. Genetic Algorithm, Tabu Search, Simulated Annealing and Neural Network (NN) have demonstrated reliable strength to tackle different combinatorial problems. NN is attributed with several features which make it attractive in various fields of research. Among them, learning is a distinctive feature which allows the algorithm to improve its performance according to the input data. Also, inherent parallel structure is another attribute of this methodology which is progressively in demand by diffusion of the parallel hardware.

Several algorithms based on adaptive NN have been devised for TSP and researchers have reported encouraging results [6]. Since the implementation of these algorithms are highly geometrical oriented, so far there are few successful applications of adaptive NN to other combinatorial problems. The main purpose of this study is to extend the applicability of this approach. More precisely, this research is an attempt to include restriction satisfaction mechanism into the Kohonen's Self Organization Feature Map (SOFM) algorithm [2]. We selected Capacitate Vehicle Routing Problem (CVRP) as a case study to demonstrate the effectiveness of the proposed idea.

The CVRP which is one of the well known routing problems, may be described as follows. To serve N customers, each with a given demand, q_i, a fleet of M vehicles with a specified capacity, Q, are available at a central depot. Each customer should be served completely and only once. The objective is to construct the route of vehicles, originating and terminating at the central depot, with the minimum total length. Extensive research efforts are oriented toward this problem in pursuit of developing efficient algorithms which can cope with the complexity of problems when the size increases. This problem, which may be regarded as a generalization of TSP is widely exploited in logistics and distribution planning and scheduling. However, it is important to recognize that it is much more complex than TSP concerning the additional constraints. Like most important combinatorial problems, CVRP also belongs to a class of NP-hard problems.

The reminder of this paper is organized as follows. Section 2 gives an overview of NN algorithms for optimization with a focus on the Kohonen competitive learning scheme and its application to TSP. Our algorithm, which extend SOFM for CVRP, is presented in section 3. The computational experiments will be given in Section 4 and concluding remarks and directions for future research will appear subsequently.

2. NN in Combinatorial Optimization

The idea of using NN for optimization emerged from the work of Hopfield and Tank [1]. They proposed an associative NN model and utilized the notion of energy function to show convergence of the network to a steady state. It is verified that as the algorithm proceeds, the value of energy function always decreases until it converges to a local optima. To apply this model to optimization problems, the cost function as well as a penalized form of restrictions are included in the energy function. Although the algorithm has been improved by several researchers, due to the inflexibility of the model and complexity of the network, the performance of the algorithm is behind other modern heuristics. Recently, adaptive NN models which are able to learn from data are highly regarded as a promising approach for optimization. The devised algorithms based on this approach may be classified into two categories, namely Elastic net [5] and SOFM [2]. Both methods use similar schemes and the differences arise in the update mechanism of nodes. While EN takes into account the entire nodes, the SOFM yields the results through a competition mechanism. In this section, we outline the structure of the SOFM for TSP. This framework would pave the way for our description in the next section.

In this model, a simple two layer network with N output nodes can represent TSP. The weight of the nodes are initialized randomly, and updated iteratively according to the input data, which are the coordinates of cities. At each iteration a city is presented to the network and in a competition procedure the node with the closest Euclidean distance to the current city is declared as the winner. If X_i denotes the coordinate of the input city and Y_j the weight of node j, the winner node, J, is selected according to the following rule:

$$J = Argmin_j\{|X_i - Y_j|\}. \tag{1}$$

The winner node as well as the nodes that lie within its surrounding neighborhood are allowed to change their weights. The update rule may be stated as follows:

$$Y_j.^{new} = Y_j.^{old} + \mu F(d_j)(X_i - Y_j.^{old}), \tag{2}$$

where μ is a gain parameter called learning rate and d_j is a variable which represents the cardinal distance of node j from the winner node. The function $F(.)$ is selected in such a way that as time progresses the influence of the inserted city to the network on neighbor nodes of the winner shrinks. The key feature of SOFM which enables it to treat many problems including optimization, is covering several neighbor nodes besides the winner. Indeed, this strategy creates a lateral force which tightens neighbor nodes during the movement toward cities, assuring the construction of short tours.

This iterative procedure may be geometrically interpreted as follows. Initially, a flexible ring is generated randomly at the center of the cities. During the learning procedure the ring is gradually enlarged until it passes sufficiently near to cities to define a tour.

3. The Proposed Algorithm for CVRP

To be more perceivable, first, we explain the algorithm for the Multiple Traveling Salesman Problem (MTSP) which has an intermediate status between TSP and CVRP. This problem involves the distribution and routing of multiple salesmen among given cities. Imposition of capacity restriction on the vehicles would transform this problem into CVRP. It is necessary to point out here that, in the proposed algorithm the results will be constructed by considering all constraints simultaneously. Similarly, SOFM may be applied to MTSP by construction of K small petaloid paths around the central depot and iterative elongation of paths based on an active competition among nodes. Nonetheless, some modifications according to the new configuration is necessary.

One of the pitfalls of Kohonen's learning algorithm is its inability to train all of the nodes, properly [6]. This difficulty arises even in TSP and obstructs node separation at the early stages of the algorithm. For MTSP in which the separation of nodes into several paths raises the complexity of the problem, a modification to circumvent this difficulty is indispensable. In SOFM the sole mechanism that pulls the nodes toward each other is update of the winner node neighbors in accordance with the presented input data. Strengthening the lateral force between neighbors nodes would help to overcome this difficulty. To this end, we incorporated another term in the update function of the algorithm. This idea which is borrowed from the elastic net algorithm is quit effective in appropriate node separation. Thus, the modified update function may be stated as follows:

$$\Delta Y_j^r = \mu F(g,d)(X_i - Y_j^r) + \lambda(Y_{j+1}^r - 2Y_j^r + Y_{j-1}^r), \tag{3}$$

where, Y_j^r denotes the node j in route r. Observe that the second term in equation (3) creates a lateral force between neighbor nodes.

Another issue which needs adjustment is the number of nodes in the paths. Even in TSP, when the number of nodes are equal to the number of cities, assignment of one node per city is a difficult task which frequently slows down the convergence of the algorithm. To avoid this difficulty, we increased the total number of nodes to three times that of the number of demand points. This strategy builds more flexibility in the algorithm and facilitate its convergence.

To enable the algorithm to cope with CVRP, a constraint satisfaction mechanism is incorporated. However, this procedure should be deliberately implemented. Hard approaches like inhibition of overloaded vehicles would impair learning efforts and result in poor quality solutions. The mechanism should improvise the restriction into the learning procedure. To realize this idea, we turn our attention to the winner selection criterion in the learning procedure. The winner selection rule is modified by inclusion of a bias term, which reflects the load rate of each vehicle. This bias term, which is identical for all nodes in a route can be called route bias. This term will prevent the nodes from overloaded routes to be winner more frequently and facilitated acceptance of other vehicles which have not enough chances to be selected as winner. This mechanism which is interwoven with other parts of the algorithm tends to uniformly distribute loads between vehicles. Although, in this study we considered the identical capacity vehicles, the idea is easily extendible to the case of different capacity vehicles. According to this modification, the winner node, J, will be selected according to the following rule.

$$J = Argmin_j \left\{ |X_i - Y_j^r| + v.B_r \right\}, \tag{4}$$

where

$$B_r = \left(\frac{\sum_l q_l^r}{1+Q} \right)^2 . \tag{5}$$

In equation (4) the q_l^r refers to the demand l that is assigned to the route r and $|\cdot|$ represents the Euclidean distance. The first term represents the closeness of the nodes to the input city and the second term is the proposed bias which inserts the capacity restriction smoothly in the learning procedure. In order to highly penalize the overloaded vehicles, a squared form of the bias term is exploited.

The parameter v in this equation has a crucial role in reconciling between the quality and feasibility of solutions. Since the distance between the nodes and cities is rather high at the early stages of the algorithm and shrinks with time, a constant value for v would result in domination of equation (4) by the the second term which in turn leads to poor quality results. To prevent this, the value of v should iteratively shrink like lowering the temperature in the Simulated Annealing algorithm. We update this parameter according to the following simple equation:

$$v = \frac{v}{1+\beta.v}. \tag{6}$$

The proposed algorithm consists of the following, detailed steps:

Step.0 Initialization : Let N = the number of cities and M = the number of nodes on the paths. Initialize parameters and construct k petaloid paths around the central depot.

Step.1 Randomizing : Randomize the order of cities and label cities 1,...,N. Let i be the index of the city in presentation and set $i = 1$

Step.2 Winner selection : Through a competitive procedure, select the closest node to city i based on statement (4) and label the winner node as J and the corresponding path as R.

Step.3 Adaptation. Move node J and its neighbor nodes on the path R towards city i according to equation (3), using the following neighborhood function $F(g,d)$.

$$F(g,d) = \begin{cases} exp(-d^2/g^2) & d < H \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$H = 0.2 * \mathrm{M},$$

where g is the gain parameter, and d measures the cardinal distance between node j and J by

$$d = \min[\|j - J\|, M - \|j - J\|], \tag{8}$$

and $\|\cdot\|$ represents absolute value.

Step.4 Set $i = i+1$. If $i \leq$ N, go to Step 2. Otherwise, set $g = (1-\alpha)\cdot g$; $i = 1$; update the bias term for routes and the value of parameter v according to equation (5)and (6), respectively; Go to Step 5.

Step.5 Convergence test: if the position of nodes are within an acceptable distance of cities, then Stop. Otherwise, go to Step 1.

It is of interest to note that in this algorithm the effective neighborhood length, which refers to the number of nodes that are allowed to update their weights besides the winner, is considerably restricted. As equation (7) shows only 40 percent of nodes on a path can update their weights. According to our experiences, this strategy has no effect on the quality of results, while saves substantial computation time.

4. Empirical results

To demonstrate the effectiveness of the algorithm computational experiment is conducted on a series of standard data. The data set which is selected from the literature consists of 11 problems ranging in size from 22 to 200. The algorithm is coded by C language and implemented on a SUN SPARC 10 workstation. The initial values for parameters v and g is decided based on the nature of input data. The value of parameter g is decided upon the size of the problem while the initial value of v is determined proportional to the average distance of demand points from the central depot.

The simulation results are illustrated in Table 1. In this table the efficiency of the proposed algorithm is compared with the Vakhutinsky and Golden [7] algorithm, in which an adaptive NN approach is incorporated to tackle CVRP. Also, the results is evaluated by considering the optimal or best known solutions. The algorithm outperforms Vakhutinsky and Golden [7] algorithm in all of cases. Also, the deviation of results from the best known solutions is 6.5 percent in the worst case and around 4 percent on average. This comparison confirms the capability of the proposed algorithm to tackle CVRP.

TABLE 1. The of Results of the Proposed algorithm

No.	Name	Size	No. Routes	Best Known	Vakhutinsky & Golden	Proposed	
						Result	PDB*
1	eil22*	21	4	375	659	390	4.0
2	eil23*	22	5	569	950	585	2.8
3	eil30*	29	3	534	855	557	4.3
4	eil33*	32	4	835	894	889	6.5
5	C1†	50	5	521	560	537	3.1
6	C2†	75	10	838	N/A	876	4.5
7	C3†	100	8	829	N/A	863	4.1
8	C4†	150	12	1044	N/A	1082	3.5
9	C5†	199	17	1334	N/A	1386	3.9
10	C11†	120	7	1042	N/A	1066	2.3

* Refer to [3] † Refer to [4]

‡ Percentage Deviation from the Best known solution

5. Concluding Remarks

In this research we developed a new algorithm based on SOFM for solving a class of routing problems. The simulation results demonstrate the capability of the algorithm to yield favorable solutions. The proposed algorithm can be applied to other type of routing problems by a modest modification. Extension of this approach to more complex cases like assignment and scheduling problems is a promising subject for further research. Also, relying on the inherent parallel structure of NN, the proposed algorithm may be easily implemented on parallel hardware. This implementation serves drastic reduction of computation time and enable the algorithm to tackle large size cases in a reasonable time frame.

The available heuristic algorithms for combinatorial problems may be classified into three types, namely construction, improvement and composite. The composite algorithms which utilizes both constructive and improvement procedures can yield outstanding results. This fact supports the idea that a combination of the proposed algorithm with a simple improvement procedure would enhance the results. Recent studies [8] have validated the effectiveness of this proposition for the Hopefield algorithm. We plan to develop such a composite algorithm based on the proposed one. Also, another approach for extension of SOFM for complex problems is under study. In this approach the restrictions are virtually translated into a geometrical format. The results of this research will appear in subsequent papers.

References

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

2. Kohonen, T. (1984). *Self-organization and Associative Memory*, New York: Springer-Verlag.

3. Miller, D. (1995) A Matching Based Exact Algorithm for Capacitated Vehicle Routing Problem, *ORSA Journal on Computing* Vol. 7, No. 1, pp.1 – 9.

4. Osman, I.H. (1993) Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem, *Annals of the Operations Research* Vol. 41, No. 3, pp.421 – 451.

5. Potvin, J.V. (1993) The Traveling Salesman Problem: A Neural Network Perspective, *ORSA Journal on Computing* Vol. 5, No. 4, pp.328 – 348.

6. Torki, A., Somhom, S. and Enkawa, T., "A Survey of Adaptive Neural Network Models for Combinatorial Problems", Technical Report 96-8, Department of Industrial Engineering and Management, Tokyo institute of technology, 1996.

7. Vakhutinsky, A.I., Golden, B.L. (1994) Solving Vehicle Routing Problems Using Elastic Net, *IEEE International Conference on Neural Network* pp.4535 – 4540.

8. Xu, X. and Tsai, W.T. , (1991) Effective Neural Algorithms for the Traveling Salesman Problem, *Neural Network* 4, pp.193 – 205.