International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 2, Issue 5, Nov 2013, 111-118

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

OPTIMIZATION OF TRAVELING SALESMAN PROBLEM FOR AN ENTERPRISE RESOURCE PLANNING SYSTEM

CHIRAG JAIN[1], RICHA SHAH[2], ARUNA GAWADE[3] & KIRAN BHOWMICK[4]

[1,2]U.G. Student, Department of Computer Science and Engineering, D.J. Sanghvi College of Engineering, Vile-Parle West, Mumbai, Maharashtra, India

[3,4]Assistant Professor, Department of Computer Science and Engineering, D.J. Sanghvi College of Engineering, Vile-Parle West, Mumbai, Maharashtra, India

ABSTRACT

Practical applications of the classical traveling salesman problem are rare because in most real world situations there are other constraints to be considered. To increase the range of useful applications, the classical structure must be modified. In this paper, we propose a hybrid approach that consists of an Adaptive Neuro Fuzzy Inference System and the Simulated Annealing algorithm, which determine the path based on distance and other factors.

KEYWORDS: Traveling Salesman Problem, ANFIS, ERP, Simulated Annealing, Fuzzy Inference

INTRODUCTION

The Traveling Salesman Problem (TSP) is a classical combinatorial optimization problem, which is simple to state but very difficult to solve. The problem is to find the shortest possible our through a set of N vertices ('cities') so that each vertex is visited exactly once. This problem is known to be NP-complete, and cannot be solved exactly in polynomial time. [7]

Several solutions have been presented to solve the classical Traveling Salesman Problem. But, these solutions are not used in the real world scenario because distance is not the only constraint that determines the order in which the cities (referred to as "customers" from now on) are to be visited. Instead various other factors (for example, the market conditions of the product to be sold to the customer, the expenses of the salesman, etc.) are considered along with the distance.

In this paper, we propose a hybrid approach to adapt the classical TSP for an ERP System. An ERP (Enterprise Resource Planning) system covers a number of functional areas (like financial accounting, Human resources, Logistics, etc.). It provides an integrated suite of software modules that supports the basic internal business processes of a company. In an ERP system, a salesman has to visit numerous customers multiple times for various product enquiries. Thus, the path that determines the order in which the customers are to be visited can be decided using a modified solution to the classical TSP. An Adaptive Neuro-Fuzzy Inference System will be used to determine the priority of a customer using various factors. The output from the ANFIS i.e. the priority generated will be fed along with the distance of the customer to the Simulated Annealing algorithm to determine the order in which the customers must be visited. Thus the final path will be calculated considering both distance as in the classical TSP as well as various other factors.

RELATED WORK

Many exact and heuristic algorithms have been devised to solve the TSP.A few of these algorithms have been described as follows:

Exact Algorithms

Many exact approaches have been used to solve the TSP. One of them is the Brute Force algorithm, where we find all the possible solutions and then select the one with the lowest cost. Other approaches include branch and bound, progressive improvement algorithms, branch and cut, etc. The exact algorithms are designed to find the optimal solution to the TSP, that is, the tour of minimum length. The exactal gorithms are typically derived from the integer linear programming (ILP) formulation of the TSP:

$$\text{Min } \Sigma_i \Sigma_j \, d_{ij} x_{ij}$$

Subject to:

$$\Sigma_j \, x_{ij} = 1; \; i=1,...,N$$

$$\Sigma_i \, x_{ij} = 1; \; j=1,...,N$$

$$(x_{ij}) \; \varepsilon \; X$$

$$x_{ij} = 0 \text{ or } 1$$

Where d_{ij} is the distance between vertices i and j and the x_{ij}'s are the decision variables: x_{ij} is set to 1 when arc (i,j) is included in the tour, and 0 otherwise. $(x_{ij}) \; \varepsilon \; X$ denotes the set of sub tour-breaking constraints that restrict the feasible solutions to those consisting of a single tour. The exact algorithms determine the best path, given a number of cities. The computation time is negligible when the number of cities is small. But, when the number of cities increases, so does the computation time – it is of the order of n!, where n is the number of cities.

Heuristic Algorithms

Finding the optimum solution to the TSP using an exact algorithm may take less time on an expensive computer. But, it is more cost effective if we find a “near-optimal” solution on a computationally less powerful computer. Hence, heuristic or approximate algorithms are often used in place of exact algorithms for solving large TSPs.TSP heuristics can be classified as tour construction procedures, tour improvement procedures, and composite procedures, which are based on both construction and improvement techniques. [1]

- **Construction Procedures**

Procedures in this class build the path by selecting one vertex at a time and inserting it into the current path one by one. For selecting the next vertex and identifying the best place to insert it, various factors are considered like proximity of current tour and the minimum tour.

- **Improvement Procedure**

The k-opt exchange heuristics in particular, the 2-opt, 3-opt, and Lin-Kernighan heuristics are the most widely used amongst the local improvement procedures. These heuristics techniques locally modify the current solution by replacing karcs in the tour by k new arcs so as to generate a new improved tour. Typically, the exchange heuristics are applied iteratively until a local optimum is found, which cannot be improved further by using the exchange heuristic under consideration

- **Composite Procedures**

Here, the tour construction procedure is a simple greedy heuristic. In the beginning, each city is considered as a

fragment, and multiple fragments are built in parallel by iteratively connecting the closest fragments together untila single tour is generated. The solution is then processed by a 3-optexchange heuristic.

Artificial Neural Networks

A number of neural networks have been used to solve the classical TSP. Some examples have been described as follows:

- **Hopfield Tank Neural Network**

The original Hopfield neural network model is a fully interconnected network of binary units with symmetric connection weights between the units.

The connection weights are not learned but are defined a priori from problem data (the inter-city distances in a TSP context). Starting from some arbitrarily chosen initial configuration, either feasible or infeasible, the Hopfield network evolves by updating the activation of each unit in turn (i.e., an activated unit can be turned off, and an inactivated unit can be turned on). Through this update process, various configurations are explored until the network settles into a stable configuration. In this final state, all units are stable according to the update rule and do not change their activation status. [5]

Elastic Net

The elastic net algorithm is an iterative procedure where M points, with M larger than the number of cities N, are lying on a circular ring or "rubber band" originally located at the center of the cities. The rubber band is gradually elongated until it passes sufficiently near each city to define a tour. During that process two forces apply: one for minimizing the length of the ring, and the other one for minimizing the distance between the cities and the points on the ring. These forces are gradually adjusted as the procedure evolves. [6]

The disadvantage of these methods used to solve the classical TSP, is that they consider only distance as a parameter to determine the path. In the real world, a salesman decides the order in which he should visit the customers based on a number of parameters. Distance is just one of the many factors considered.

PROPOSED SOLUTION

The various methods that were studied could not be easily adapted to consider other factors along with distance. Hence, we decided to first use an Adaptive Neuro-Fuzzy Inference System where we give other factors that are considered to determine the order in which the customers are to be visited. This system will give a priority for each customer, which will be combined with distance, and given to Simulated Annealing algorithm to finally determine the path.

Thus, this paper proposes the following hybrid approach towards solving the Traveling Salesman Problem for an ERP System. This approach is composed of two phases – ANFIS, followed by Simulated Annealing.

Phase 1 (ANFIS)

Adaptive Neuro-Fuzzy inference system (ANFIS) is a kind of neural network that is based on Takagi–Sugeno fuzzy inference system. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Based on the inputs, it was easier to use an ANFIS where a set of fuzzy IF-THEN rules can be used to determine the priority of each customer.

Architecture

The ANFIS architecture consists of six layers namely, Inputs, IF-Part, Rules, Normalization, THEN-Part, and Output. The output generated by one layer is fed to next layer where each layer performs a specific task which is described below.

Layer 0: Inputs to ANFIS

The following inputs will have a numeric value that is fed into the Adaptive Neuro-Fuzzy inference system. The values can then be mapped to Low, Medium, High, Very Small, Moderate etc. depending upon the value of input. Also, the number of descriptors for a particular input can be different from other inputs. Further, the range of values (Universe of discourse) within which these inputs lie varies.

- Market Conditions for the Product
- Average Discount given for Product
- Quantity of Product to be sold
- Daily wages of Salesman
- Probability of successful deal
- Cost Price of Product

Layer 1: Evaluating Membership (IF-Part)

For all inputs Gaussian membership function is used to calculate the membership value.

The formula for Gaussian membership function is:

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

The graph of membership value against the input can be as follows

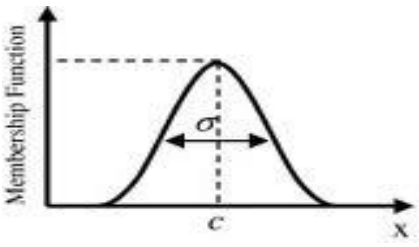

Figure 1

The output calculated can be represented as

$$O_{1i} = \mu_{A_i}(x)$$

Here x is the input value and $\mu_{A_i}(x)$ is the corresponding membership function.

Layer 2: Rules

Every node in this layer is fixed. This is where the t-norm is used to 'AND' the membership grades - for example the product:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2$$

Layer 3: Normalization

It contains fixed nodes which calculate the ratio of the firing strengths of the rules as follows:

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2 + \ldots + w_i}$$

Layer 4: Consequent Parameter Calculation (THEN-PART)

The nodes in this layer are adaptive and perform the consequent of the rules:

$$O_{4,i} = \overline{w_i} f_i$$

where f_i is a function of input values.

Layer 5: Output

There is a single node here that computes the overall output as follows:

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

The ANFIS architecture is as shown below. In our approach, there will be six inputs as described previously, along with multiple rules. The input vector is fed through the network layer by layer.

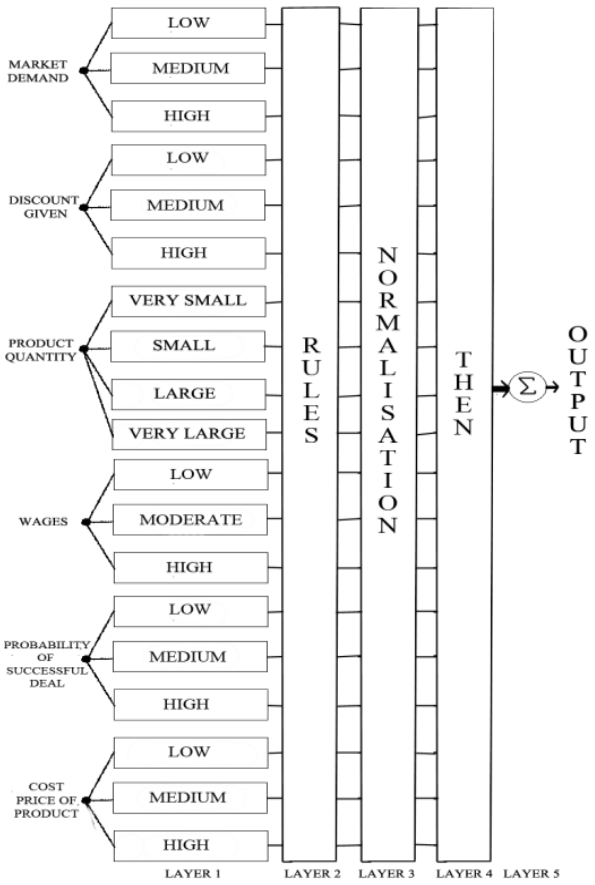

Figure 2: ANFIS Architechture

Learning

The ANFIS learns the premise and consequent parameters for the membership functions and the rules. The hybrid learning algorithm proposed by Jang, Sun and Mizutani will be used. [2]

We split the total parameter set into three:

S= set of total parameters

S_1 = set of premise (nonlinear) parameters

S_2 = set of consequent (linear) parameters

ANFIS uses a two pass learning algorithm:

- **Forward Pass:** Here $S_1$is unmodified and S_2 is computed using a LSE algorithm.
- **Backward Pass:** Here S_2 is unmodified and S_1 is computed using a gradient descent algorithm such as back propagation.

So, the hybrid learning algorithm uses a combination of steepest descent and least squares to adapt the parameters in the adaptive network.

Output

The ANFIS will give a numerical output which will represent the priority of the corresponding customer.

Phase 2 (Simulated Annealing)

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. The key idea in simulated annealing algorithm is to select an appropriate temperature schedule which needs to specify the initial, relatively large value of T and then decrease the value of T. Starting with relatively large values of T, the probability of acceptance is relatively large, which enables the search to proceed in almost all random directions. Gradually decreasing the value of T as the search proceeds gradually decreases the probability of acceptance, which emphasizes on climbing upwards.

Simulated Annealing has been used to solve the classical TSP. It has been used in this hybrid approach because it can be easily modified to consider the priority of each customer, combined with distance in order to determine the path.

Inputs to Simulated Annealing

The co-ordinates of the customers, along with their corresponding priorities will be given to the Simulated Annealing algorithm.

Acceptance

The probability of transiting from current state X_n to a current trial solution $X'_n$is specified by an acceptance probability function P(e, e', T) which depends on the energies $e = E(X_n)$ and $e' = E(X'_n)$ of the two states and a global varying parameter called the Temperature. Essential points for designing P are:

- Must be nonzero when e'> e, meaning the system might move to the new state X'_n even if it is worse than the current one. This feature prevents the method from being stuck in local minimum.
- When T goes to zero, P (e, e', T) must tend to be zero when e'> e and to a positive value if e'< e.

The flowchart for the simulated annealing algorithm is as shown below [4]

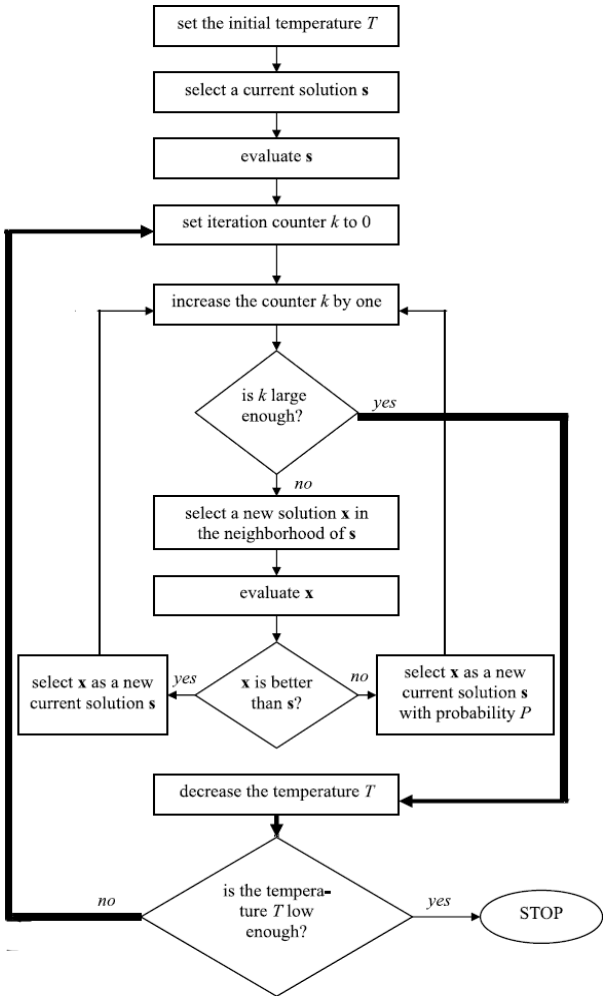

Figure 3: Simulated Annealing Flowchart

Algorithm

X_n is the solution after n iterations.

X'_n is the current trial solution.

Temperature (T) is the parameter that measures the tendency to accept the current solution as next trial solution.

The 'move selection rule' is then based on selecting which neighbor will be the next trial solution.

- Create the initial list of cities by shuffling the input list (i.e.: make the order of visit random).
- Select X'_n neighbor of X_n
- The cost value is the distance travelled by the salesman for the whole tour.
- If neighbor is 'better' in cost i.e. $f(X'_n)$ is less than $f(X_n)$, then accept with probability p_n, $X_{n+1} = X'_n$.
- If neighbor is 'worse' in cost i.e. $f(X'_n)$ is greater than $f(X_n)$, then accept with probability p_n, $X_{n+1} = X'_n$or reject with probability $(1 - p_n)$, $X_{n+1} = X'_n$. We could have

$$p_n = e^{\frac{-(f(X'_n) - f(X_n))}{T}}$$

- We update the temperature during every iteration by slowly cooling down.
- We let Temperature (T) go to zero over time.

The probability of accepting a worse state is given by the equation: [3]

$P = \exp(-c/t) > r$

Where

c = The change in the evaluation function

t = The current temperature

r = A random number between 0 and 1

The evaluation function is a function of the distance between customers and their priorities. The probability of accepting a worse move is a function of both the temperature of the system and of the change in the cost function. As the temperature of the system decreases the probability of accepting a worse move is decreased. Thus, if the temperature is zero then only better moves will be accepted.

Output

The Simulated Annealing algorithm will give the order in which the customers are to be visited as the output.

CONCLUSIONS

The model proposed in this paper aims to bridge the gap between the classical TSP and the real world scenario. Adaptive Neuro Fuzzy Inference System and Simulated Annealing have been used in order to get the near optimal solution by taking customer and product related factors also into consideration for calculations and not just distance between customers. Once implemented, the model has a huge scope as Enterprise Resource Planning systems are used in a number of industries today.

REFERENCES

1. Jean-Yves Potvin, "The Traveling Salesman Problem: A Neural Network Perspective", ORSA Journal on Computing 5, 328-348, 1993.
2. Jyh-Shing Roger Jang, Chuen-Tsai Sun and EijiMizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence"
3. S. N. Sivanandam, S. N. Deepa, "Principles of Soft Computing"
4. Zbigniew Michalewicz, Martin Schmidt, Matthew Michalewicz, Constant in Chiriac, "Adaptive Business Intelligence"
5. Jacek Ma´ndziuk, "Solving the Travelling Salesman Problem with a Hopfield - type neural network", Institute of Mathematics, Warsaw University of Technology
6. Donald Davendra, "Traveling salesman problem, theory and applications", In Tech December 2010
7. http://en.wikipedia.org/wiki/Traveling_salesman_problem