International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Solution for a Travelling Salesman Problem in Neural Networks

Sajja Tulasi Krishna

M.Tech., Assistant Professor, Computer Science and Engineering, B.V Raju Institute of Echnology, Hyderabad, India

Abstract: ***Travelling Salesman Problem is a very hard Optimization Problem in a field of Operations research. It often used as for new optimization Techniques. This paper surveys the neutrally inspired Travelling Salesman Problem using Ant Colony Optimization Algorithm. It presents Adaptive Resonance theory in combination with Ant Colony Optimization Algorithm to solve the large instance of Travelling Salesman Problem. Travelling salesman problem (TSP) consists of finding the shortest route in complete weighted graph G with n nodes and n(n-1) edges, so that the start node and the end node are identical and all other nodes in this tour are visited exactly once.***

Keywords: Travelling salesman problem, met heuristics, ant colony optimization, Adaptive Resonance Theory

1. 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 tour through a set of N vertices so that each vertex is visited exactly once. This problem is known to be NP-complete, and cannot be solved exactly in polynomial time. The Traveling Salesman Problem is a problem of a salesman who, starting from his hometown, wants to find the shortest tour that takes him through a given set of customer cities and then back home, visiting each customer city exactly once." Each city is accessible from all other cities. There are various ways to classify algorithms, each with its own merits. The basic characteristic is the ability to reach optimal solution: exact algorithms or heuristics.

The best known exact methods for solving TSP are: explicit enumeration, implicit enumeration, branch and bound method, cutting plane method and dynamic programming. These methods work solving the Travelling Salesman Problem. Heuristic methods vary from exacts methods in that they give no guarantee to find the optimal solution to the given problem (so that solution is called suboptimal), but in many cases this is the solution of good quality and we can obtain it in acceptable time. Heuristic methods are usually focused on solving the special type of problems. The significant part of heuristics comprises met heuristic methods, which differs from the classical methods in that they combined the stochastic and deterministic composition. It means that they are focused on global optimization, not only for local extremes. The big advantage of met heuristics is that they are built not only for solving a concrete type of problem, but they describe general algorithm in that, they show only the way, how to apply some procedures to become solution of the problem. This procedure is defined only descriptively, by black-box, and the implementation depends from the specific type of problem. The group of the most known met heuristics includes evolutionary algorithms, which are inspired by process in nature (for example genetic algorithms, particle swarm optimization, differential evolution, ant colony optimization, etc)

Adaptive Resonance Theory

Gail Carpenter and Stephen Gross berg (Boston University) developed the Adaptive Resonance learning model to answer how human brain works. Essentially, ART (Adaptive Resonance Theory) models incorporate new data by checking for similarity between this new data and data already learned "memory". If there is a close enough match, the new data is learned. Otherwise, this new data is stored as a "new memory".

The basic ART model is comprised of the following components:

1. The short term memory layer: F1 – Short term memory.
2. The recognition layer: F2 – Contains the long term memory of the system.
3. Vigilance Parameter: ρ – A parameter that controls the generality of the memory. Larger ρ means more detailed memories; smaller ρ produces more general memories.

Applications of ART Model

Clustering Web Users

Consider the case of having a dynamic website consisting of multiple components and each user access each component a certain number of times. How would you redesign the website to provide the most commonly used components to that specific user without being impractical in implementation?

Adaptive Neural Network Clustering of Web Users .The unsupervised variants provide a simple, effective neural clustering algorithm. The original algorithm, designed ART1, used binary input sequences. the variation developed for this solution uses two input integer sequences where the integers are the XY coordinates of a node. In this form, ART is functionally similar to K –means clustering except that K is increased dynamically as new patterns are introduced. Define:

1. M = 200: Number of unique components – F1 layer
2. N: Num. of organizational clusters that users are grouped into – F2 layer

3. H = 70: Number of users 27

2. Ant Colony Optimization Algorithm

Ant colony optimization (ACO) belongs to the group of met heuristic methods. The idea was published in the early 90s for the first time. The base of ACO is to simulate the real behavior of ants in nature. The functioning of an ant colony provides indirect communication with the help pheromones, which ants excrete. Pheromones are chemical substances which attract other ants searching for food. The attractiveness of a given path depends on the quantity of pheromones that the ant feels. Pheromones excretion is governed by some rules and has not always the same intensity. The quantity of pheromones depends on the attractiveness of the route. The use of more attractive route ensures that the ant exudes more pheromones on its way back and so that path is more also attractive for other ants. The important characteristic of pheromones is evaporation. This process depends on the time. When the way is no longer used, pheromones are more evaporated and the ants begin to use other paths. What is important for ACO algorithm the moving of ants? This motion is not deterministic, but it has stochastic character, so the ants can find the path, which is firstly unfavorable, but which is ultimately preferable for food search. The important characteristic is that a few individuals continuously use non-preferred path and look for another best way.

ACO was formulated based on experiments with double path model, where the quantification was made similar to Monte Carlo method. The base of this simulation was two artificial connections between the anthill and a food source. The simulation demonstrated that ants are able to find the shorter these two paths. A significant impact of this simulation was to quantify the behavior of ants. For practical use of ACO, it was necessary to project virtual ants. It was important to set their properties. These properties help virtual ants to scan the graph and find the shortest tour. Virtual ants do not move continuously; they move in jumps, which mean that, after a time unit, they will always be in another graph node. The absolved path is saved in ant memory. The created cycles are detected in ant memory. In the next tour, the ant decides on the base of pheromones power. Just because the property of pheromone evaporation, pheromones on shortest edges are stronger, because of the fact that the ant goes across these edges faster.

The algorithms use different rules for selection of the next city to move to, for evaporation, and for deposition of pheromone. A nightmare of each optimization algorithm is to be stuck forever in some locally optimal loop. Different techniques to escape such loops were developed for different Ant Colony Optimization algorithms. After an ant moves from one city to the next, the pheromone on the traveled path is reduced by some value. The solution construction is also different.

3. Solution Construction

The algorithm has a parameter qo €[0,1]. The parameter sets the boundary between two rules to select the next city. On each step, the algorithm generates a random number 0 ≤ q ≤ 1.If, qo < q the next city to move is selected according , else index **j** of the next city to move is:

$$j = \arg \max_{l \in T_i^k} \{\tau_{il} * \eta_{il}^{\beta}\}, \qquad (10)$$

Where

- **i** is the index of the current city,
- **j** is the index of the city to move next,
- T_I^K is a set of cities adjacent to the current city **i** at step **k**,
- **τil** is the amount of pheromone on the path **cil**,

$$\eta_{il} = \frac{Q}{d_{il}}$$

- , **Q** is a constant (in this app it is a city's extent), **dil** is a distance between the cities **i** and **l**,
- **β** - algorithm parameter After the ant moves from the city **i** to city **j**, the pheromone on the path **cij** is evaporated and updated according to

$$\tau_{ij} \leftarrow (1 - \xi) * \tau_{ij} + \xi * \tau_0 ; \qquad (11)$$

where

- ξ -the local evaporation factor,
- **τ0** – additional pheromone (an algorithm parameter).

After the first ant has made the k^{th} step and has modified a pheromone, it does not begin the next step. It waits until all other ants complete their k^{th} steps and modify the pheromones on their traveled k^{th} path. In this process, the n + 1 ant waits for the update by the ant n, ant and so on.

4. Evaporation of Pheromones

At last, the pheromones on the edges are evaporated. The evaporation helps to find the shortest path and provide that no other path will be assessed as the shortest. It is necessary to keep information about quantity of pheromones τij in memory, which has stochastic character and actually represents state of graph scan. Further on, there is a need to memorize the edge costs, or information derived from this information (ηij).Information of pheromones value τij is changing during the simulation, but values of ηij stay the same during the calculation. Virtual ants use this information during their moving across the graph. On the basis of these considerations, we can describe the ACO algorithm as system of steps:

1. Ants scan graph G. The aim of this scan is to find an optimal solution.
2. Every ant has its own memory, which is used for saving information about travelled path (for example about travelled nodes). This memory can also serve to ensure constraints or to evaluate of the solution.
3. The process begins in state xsk and has one or more ending constraints ek. Let the actual state of an ant be the state xr = (xr–1, i) and no ending constraint is complied, so the ant moves to node j in neighborhoods of the state Nk(xr) and the ant moves to the new state (xr, j) X. In case that some ending constraint is complied with, the ant ends with process of scan. The transition to a state that represents unacceptable solution is usually banned by appropriate implementation of internal ant memory.

4. The next ant motion depends on the Probability, which is calculated on the base of pheromone quantity on edges of graph, and also takes into consideration its local memory and the acceptance of this step.
5. If the ant can to add new component of graph GC, it can update the value of corresponding pheromone information (information is bound with corresponding edge, or aim node).
6. The ant can update pheromone values after reverse path construction by editing associate Pheromone values.

5. Conclusion

It can be concluded that the quality of solutions depends on the number of ants. The lower number of ants allows the individual to change the path much faster. The higher number of ants in population causes the higher accumulation of pheromone on edges, and thus an individual keeps the path with higher concentration of pheromone with a high probability .The great advantage over the use of exact methods is that ACO algorithm provides relatively good results by a comparatively low number of iterations, and is therefore able to find an acceptable solution in a comparatively short time, so it is useable for solving problems occurring in practical applications.

References

[1] Gail A. Carpenter, "Default ARTMAP", Neural Networks, July., 2003
[2] Brezina, I. (2003). Kvantitatívne metódy v logistike. Bratislava: Ekonóm.
[3] Onwubolu, G. C., & Babu, B. V. (2004). New Optimization Techniques inEngineering. Berlin-Heidelberg: Springer-Verlag.
[4] Carpenter, G., & Grossberg, S. (1988). The art of adaptive pattern recognition by a self-organizing neural network. IEEE Computer,March, 47–88
[5] Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling salesman problem. Operations Research, 21, 498–516.

Author Profile

Sajja Tulasi Krishna, received the B.Tech., degree in Computer Science and Engineering from Nirma College of Institute and Technology, JNTUK in 2009 and Post Graduation, M.Tech., in Computer Science and Engineering from Aditya College of Engineering, JNTUK in 2013. Working as Assistant Professor in Computer Science and Engineering Dept., at B.V. Raju Institute of Technology, Hyderabad, India