e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

Literature Review on Travelling Salesman Problem

Chetna Dahiya & Shabnam Sangwan

[1]Department of Computer Science Sat Kabir Institute of Technology and Management,
Maharishi Dayanand University (MDU), Rohtak
[1]chetnadahiya2008@gmail.com

[2]Department of Computer Science Sat Kabir Institute of Technology and Management (MDU),
Maharishi Dayanand University (MDU), Rohtak
[2]shabnam022@gmail.com

Abstract— 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 tour through a set of N vertices so that each vertex is visited exactly once. This problem is known to be NP-hard, and cannot be solved exactly in polynomial time. Many exact and heuristic algorithms have been developed in the field of operations research (OR) to solve this problem. In this paper we provide overview of different approaches used for solving travelling salesman problem.

Keywords— TSP, ACO, Genetic Algorithm, PSO

I. Introduction

Given a collection of cities and the distance of travel between each pair of them, the traveling salesman problem is to find the shortest way of visiting all of the cities and returning to the starting point. Though the statement is simple to state but it is more difficult to solve. Traveling Salesman Problem [1][2] is an optimization problem and has a vast search space and is said to be NP-hard, which means it cannot be solved in polynomial time. It is one of the most fundamental problems in the field computer science in today’s time. The Traveling salesman problem is being applied in many fields nowadays. Some of its applications are vehicle routing, manufacturing of microchips, packet routing in GSM, drilling in printed circuit boards etc. In simpler words, say we have a set of n number of cities, and then we can obtain (n - 1)! alternative routes for covering all the n cities. Traveling salesman problem is to procure the route which has the least distance.

Let us consider an example describing the Traveling salesman problem. We have a set of four cities A, B, C, and D. The distances between the cities are also given to us. Figure 1 illustrates the collection of the cities and their distances among each other. Here (4-1)!, that is 3! route can be generated. The tour with A→B→C→D→A will be the optimal route for given problem.

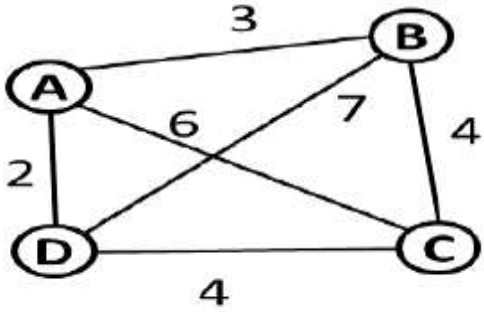

Figure 1: Traveling salesman problem

Many heuristic techniques have been used to find the efficient solution to the problem like greedy method, ant algorithms, simulated annealing, tabu search and genetic algorithms [3]. But as the number of cities increases the computation to find the solution becomes difficult. Despite the computational difficulty, we can use methods like genetic algorithms and tabu search which can give near to optimal solution for thousands of cities. In this paper we provide overview of different approaches used for solving travelling salesman problem.

II. Various Methods for solving TSP

Literature cites a whole range of TSP methods differing in various approaches to the solutions, efficiency of the procedures and also the outcomes. Let us quote brief characteristics of the most often used ones.

A. Method of total enumeration

In principle it is a combinatorial solution. The method rests in evaluation of all potential routes (sequences) in the total number of (n – 1)!. The advantage is that a global optimum is always found, however, it is not employable if higher numbers of visited places are considered. With every added element (node) the amount of possible solutions grows exponentially and not even nowadays do we have computers powerful enough for being able to provide optimum solution within reasonable time [4].

B. Method of branches and bounds

This method belongs to the oldest ones and the most often used algorithms for the TSP solutions. The merit of the method rests in a gradual decomposition of a possible solution set into a number of mutually disjunctive subsets labelled as branches. In each step the following is estimated:

● The upper limit of the objective function that is most often the value of the objective function z_H without respecting limits and

● Maximum lower bound of an objective function z_D of acceptable solutions which are known to us within the step.

Both the estimates can be employed for seeking non-prospective directions of further procedures: if for any branch $z_H < z_D$, then the given direction can be excluded. However, this method is also, especially for higher n, too laborious and does not always guarantee an optimum solution at the first attempt [5].

C. Efficient algorithm of Clarke and Wright

A significant progress in TSP solutions was provided by the Clarke's and Wright's method. The initial situation assumes that each place is supplied individually and always a return to the starting base follows. The essential idea is based on the calculation of economies achieved through integrating other places into the circular route. An indisputable asset of this algorithm is its function to respect further restrictions often generated by the practice, e.g. the need to optimize more orbital routes, to use more vehicles while respecting their various capacities etc.

Guerra, Murino and Romano worked with this algorithm for optimize the routing phase in a Location-Routing Problem (LRP) in [6]. LRP can by assimilate to a Vehicle Routing Problem (VRP) and after that they combine and balance VRP with TSP. Both problems were solved with Clarke and Wright saving algorithm and the Branch and Bound model.

D. Computer Simulation

The development of simulation models, their program support and increasing computing power brought about attempts at use of simulation techniques for solving large TSPs. Their merit rests in random PC sampling in large scale that is later evaluated according to a selected objective function. Though the solution does not guarantee the global optimum, a sufficiently large number of simulations will issue in achieving the best possible solution, the value of which will be close to the optimum [7].

E. Ant Colony Optimization Algorithm (ACO)

ACO is one of the metaheuristic methods for solving TSP. Jalali, Afshar and Marino [8] described ACO as observation of real ants, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. There is a higher probability that the trail with a higher pheromone concentration will be chosen.

The pheromone trail allows ants to find their way back to the food source and in the opposite way. The trail is used by other ants to locate food source discovered by any ant. When a number of paths available from the nest to a food source, a colony of ants may by able to exploit the pheromone trail left by individual members of the colony to discover the shortest path from the nest to the food source and back. As more ants choose a path to follow, the pheromone on the path builds up, making it more attractive to be followed by other ants.

To solve TSP, we keep the strength of pheromone trail $T_{i,j}$ for each combination of two points. The role of each ant is to find a valid solution, thus possible routes. From the starting point, the ant gradually repeats a move when it chooses a place where it has not been yet while moving to it from its current location. Once there are no more vacancies left, the ant returns to the starting location. As a result, the ant keeps its path T. If the ant k is currently at the location i, then the probability that it goes to the city j is

$$P_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}$$

Where T^{k}_{ij} is the total pheromone deposited on path ij, η^{k}_{ij} is the heuristic value of path ij according to the measure of the objective function (a priori knowledge, typically 1/cij, where cij is distance). α, β are parameters that control the relative importance of the pheromone trail versus heuristic value. When all the ants have completed a solution, the trails are updated by $\tau_{ij}^{k} = (1-\rho)\tau_{ij}^{k} + \Delta\tau_{ij}^{k}$

where p is the pheromone evaporation coefficient.

Ant colony optimization algorithms have been used to produce near-optimal solutions to the travelling salesman problem. The first ACO algorithm was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. It is able to find the global optimum in a finite time.

Rudeanu and Craus [9] presented Parallel implementation of ant colony optimization that is faster and more efficient. It's a framework based on the message-passing communication paradigm (MPI). MPI is a language-independent communications protocol used to program parallel computers. Another using of parallel algorithm using MPI was presented in [10]. ACO is one of the Swarm Intelligence systems which include many other algorithms such as Particle Swarm Optimization and River formation dynamics [11].

F. Particle Swarm Optimization Algorithm (PSO)

PSO proceed from the social behavior of organisms such as bird flocking and fishing schooling. Through cooperation between individuals, the group often can achieve their goal efficiently and effectively. PSO simulates this social behavior as an optimization tool to solve some optimization problems. Each particle flies in the search space with a velocity that is dynamically adjusted based on its own flying experience and its companions' flying experience. In other word, every particle will utilize both the present best position information of its own (*pbest*) and the global best position information (*gbest*) that swarm has searched up-to now to change its velocity and thus arrives in the new position [12].

G. Genetic algorithms

In recent years there have been attempts to use so called genetic algorithms for TSP solutions. Simply stated, genetic algorithms transfer evolution principles in living organisms into intelligent searching and model optimization in other fields. Biological terminology is applied also to this very description of the algorithm. Genetic algorithm, as well as nature, works with *population* of individuals (P) defined by one or more mathematical *genes – chromosomes* (i.e. sequences of numbers in binary notation).

The genetic algorithm (GA) uses the following steps:

1. *Generate a population*. The GA randomly samples values of the changing cells between the lower and upper bounds to generate a set of (usually at least 50) chromosomes. The initial set of chromosomes is called the *population*.

2. *Create a new generation*. In the new generation, chromosomes with a smaller fitness function (in a minimization problem) have a greater chance of surviving to the next generation. Crossover and mutation are used to generate chromosomes for the next generation.

3. *Stopping conditions*. At each generation, the best value of the fitness function in the generation is recorded, and the algorithm repeats step 2. If no improvement in the best fitness value is observed after many consecutive generations, the GA terminates.

Technically, the GA is the sort of a simulation model. For solving of GA and other specific tasks in Excel there was developed a new Evolutionary Solver. This Evolutionary Solver is available since Excel 2007 and later versions. The Evolutionary Solver is good for solving of "nonsmooth" problems with more local extremes or for solving of combinatorial models with few constrains.

Respecting the simulation background of GA, the Evolutionary Solver finds usually a very good solution, but there is no guarantee that it will find the *bes*t solution. Evolutionary Solver doesn't handle constraints well and therefore it is usually better to penalize constraints violations and include the penalties in the objective. Because of the solution process is driven by random numbers, two different runs can lead to different solutions. In spite of these "weaknesses", the Evolutionary Solver is an effective tool for solving of GA and similar problems.

More authors propose new modifications and improvements of GA. For example, an effective parallel model was presented from Bai Xiaojuan and Zhou Liang in [13]. It is based on the traditional genetic algorithm, but a new operating mechanism of GA was improved means of adaptive crossover and mutation. It uses probability of GA, which can keep the solution space effective. Further, 2-opt neighbourhood search optimization techniques are imported, which can ensure the evolution process is not stagnation, and improve the efficiency of solving. Another improving proposals of Genetic Algorithm were presented in [14] [15]. The interest of many authors about GA proved to be the promising and effective solving method for the specific group of optimization models.

III. CONCLUSION

In this paper we provide overview of different approaches used for solving travelling salesman problem. 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 tour through a set of N vertices so that each vertex is visited exactly once. This problem is known to be NP-hard, and cannot be

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 16
June 2018

solved exactly in polynomial time. Many exact and heuristic algorithms have been developed in the field of operations research (OR) to solve this problem. In this paper we study Genetic algorithm approach, Ant Colony Optimization Technique, Particle Swarm Optimization method & branch and bound algorithm to solve the travelling salesman problem.

REFERENCES

[1] Gerard Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications. Springer-Verlag, 1994.

[2] D. B. Fogel, "An Evolutionary Approach to the Traveling Salesman Problem", Biol. Cybern. 60,139-144 (1988)

[3] Kylie Bryant ,Arthur Benjamin, Advisor, "*Genetic Algorithms and the Traveling Salesman Problem*", Department of Mathematics, December 2000.

[4] Kovarik V,: *Heuristic Algorithms for decision Support*, Dissertation, Prague: VŠCHT, 2008

[5] Gros, I.: *Quantitative Methods for managerial decision making*, Prague: Grada publishing a.s., 2003, ISBN 80-247-0421-8 (In Czech)

[6] L. Guerra, T. Murino, E. Romano, *The Location - Routing Problem: an innovative approach*, 6th WSEAS International Conference on System Science and Simulation in Engineering, Venice, Italy, November 21-23, 2007

[7] Wisniewski, M.: *Methods for Managerial Decision Making*, Prague: Grada publishing, 1996, ISBN 80-7169-089-9 (In Czech)

[8] Mohammad Reza Jalali, Abbas Afshar, Miguel A. Marino: *Ant Colony Optimization Algorithm (ACO); A new heuristic approach for engineering optimization*, Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing, Lisbon, Portugal, June 16-18, 2005, pp. 188-192

[9] Laurentiu Rudeanu, Mitica Craus: *Parallel Implementation of Ant Colony Optimization for Travelling Salesman Problem*, Proceedings of the 4th WSEAS Int. Conf. on Soft Computing, Optimization, Simulation & Manufacturing Systems (SOSM 2004), Miami, Florida, April 21-23, 2004

[10] Izzatdin A. Aziz, Nazleeni Haron, Mazlina Mehat, Low Tan Jung, Aisyah Nabilah: *Solving Traveling Salesman Problem on High Performance Computing using Message Passing Interface*, Proc. of the 7th WSEAS Int. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS '08), pp. 83-87,

[11] Rabanal P., Rodríguez I., Rubio F.: *Using River Formation Dynamics to Design Heuristic Algorithms*, Springer, 2007, ISBN 978-3-540-73553-3

[12] Luping Fang, *Pan Chen, Shihua Liu: Particle Swarm Optimization with Simulated Annealing for TSP*, Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19, 2007, pp. 206-210

[13] Bai Xiaojuan, Zhou Liang: *A New Genetic Algorithm for solving Travelling Salesman Problem,* Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Science, 20, pp. 451-455

[14] Fu Yiping, Chen Guangxi, *An improved genetic algorithm for solving TSP problem*, Journal of Guilin University of Electronic Technology, Vol.27, No.4, 2007, pp.287-290

[15] Han Lixia, Wang Yuping, *A Novel Genetic Algorithm for Travelling Salesman Problem*, systems Engineering-Theory & Practice, No.12,2007, pp.145-151