

The Optimizing Multiple Travelling Salesman Problem Using Genetic Algorithm

R.Jayasutha¹ Dr. B. S. E. Zoraida²

¹Research Scholar ²Assistant Professor

^{1,2}Bharathidasan University, Trichirappalli

Abstract— The traveling salesman problem (TSP) supports the idea of a single salesperson traveling in a continuous trip visiting all n cities exactly once and returning to the starting point. The multiple traveling salesman problems (mTSP) is complex combinatorial optimization problem, which is a generalization of the well-known Travelling Salesman Problem (TSP), where one or more salesman can be used in the path. In this paper mTSP has also been studied and solved with GA in the form of the vehicle scheduling problem. The existing model is new models are compared to both mathematically and experimentally. This work presents a chromosome methodology setting up the MTSP to be solved using a GA.

Keywords: mTSP, VRP, Genetic algorithm, Chromosome method, Optimization.

I. INTRODUCTION

The work on TSP has focused on the idea of a single salesperson traveling in a continuous trip visiting all n cities exactly once and returning to the starting point [1]. There has been some work done on the idea of using multiple salesmen to visit all of the cities exactly once. In the last two decades the travelling salesman problem received quite big attention, and various approaches have been proposed for obtaining either optimal or near optimal solution for the TSP [2]. In this method used to solve the TSP range from classic methodologies based on linear programming and branch-and-bound to artificial intelligence methods such as neural networks [5]

Some of these methods are exact algorithms, while others are near-optimal or approximate algorithms. The exact algorithms use integer linear programming [3] approaches with additional constraints.

The work on solving mTSPs using GA has focused on vehicle scheduling problems [4]. The vehicle scheduling problem (VSP) consists of scheduling a fleet of m vehicles visit to n cities with each city being visited by and only one vehicle. More recently, genetic algorithms (GA) are successfully implemented to solve TSP [1]. Potvin presents a survey of GA approaches for the general TSP [2].

II. PROBLEM DESCRIPTION

The multiple TSP work has focused on vehicle scheduling problems with the goal of minimizing the total distance travelled by all the trucks (salesman) or minimizing the total number of trucks required [2].

The aim of balancing the workload (or distance travelled) among the multiple salesman that are being used to visit the required cities is a problem is very difficult. Using multiple salesmen makes the problem even harder due to the increased number of ways to visit each city once. While the

multiple TSP can be modeled as a single TSP, The result is effectively a longer tour that increases the solution space of the problem.

The multiple traveling salesman problems (MTSP) are a difficult problem that can use to significant number of practical problem. The MTSP is similar to the notoriously difficult traveling salesman problem (TSP) that seeks on optimal tour of n cities, visiting each city exactly once with no sub tours [4]. In the MTSP, the n cities are partitioned into m tours. With each assigned to a separate salesman. The MTSP is even more difficult than the TSP because it requires determining which cities to assign to each salesman as well as the optimal ordering of the cities within each salesman's tour [5].

In this paper tools developed for a modified mTSP related to the optimization of one to many distribution systems will be studied and a novel genetic algorithm based solution will be proposed.

In the case of mTSP, a set of nodes (location or cities) are given, and all of the cities must be visited exactly once by the salesman who all start and end at the single depot node. The number of cities is denoted by n and the number of salesman by m . The goal is to find tours for all salesmen, such that the total traveling cost (the cost of visiting all nodes) is minimized. The cost metric can be defined in terms of distance, times, etc., some possible variations of the problem are as follows.

- 1 *Multiple depots:* If there exist multiple depots with a number of salesmen located at each, a salesman at each depot remains the same after all the travel.
- 2 *Number of salesman:* The number of salesmen in the problem can be fixed number.
- 3 *Fixed charges:* If the number of salesmen is a bonded variable, usually the usage of each salesman in the solution has an associated fixed cost; In this case the minimization of this bounded variable may be involved in the optimization.
- 4 *Other restrictions:* These additional restrictions can consist of the maximum or minimum distance travelling duration salesman travels, or other constraints.

The main goal is to minimize the total traveling cost of the problem that is often formulated as assignment based integer linear programming [3]:

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + m c_m \quad (1.1)$$

$$\sum_{j=2}^n x_{1j} = m \quad (1.2)$$

$$\sum_{j=2}^n x_{j1} = m \quad (1.3)$$

$$\sum_{i=1}^n x_{ij=1, j=2, \dots, n} \quad (1.4)$$

$$\sum_{j=1}^n x_{ij=1, i=2, \dots, n} \quad (1.5)$$

+Sub tour elimination constraints (1.6)

$$x_{ij} \in \{0,1\}, \forall (i,j) \in A \quad (1.7)$$

Usually, mTSP is formulated by integer programming formulations; one variation is presented in equation (1.1)-(1.7). The mTSP problem is defined on a graph $G = (V, A)$, where V is the set of n nodes (Vertices) and A is the arcs (edges), Let $C = (c_{ij})$ be a cost (distance) matrix associated with A . The matrix symmetric if $c_{ij}=c_{ji}$ $(i,j) \in A$ asymmetric otherwise. Here $x_{ij} \in \{0, 1\}$ is a being variable used to represents the cost of the involvement of one salesman in the solution. Further mathematical representation can be found in [3].

III. PRELIMINARIES

A. Over View of GA

Genetic algorithms (GAs) are a relatively new optimization technique that can be applied to TSPs. The basic ideas of behind Gas evolved in the mind of John Holland at the University of Michigan in the early 1970s (Holland, 1975). Most of the work on solving mTSPs using Gas has focused on the vehicle scheduling problem (VSP) [4]. The development of effective GA operators for TSPs has led to great deal of interest and research to improve GA performance for this type of problem [5]. In a nutshell, GA work by generating a population of numeric vectors (called chromosome), each representing a possible solution to a problem. The individual components (numeric values) with in a chromosome are called genes. New chromosomes are created by crossover (the probabilistic exchange of values between vectors) or mutation (the random replacements of values in a vector). Mutation provides randomness with in the chromosomes to increase coverage of the search space and help prevent premature convergence on a local optimum. Chromosomes are evaluated according to a fitness (or objective) function, with the fittest solving and less fit being eliminated. The result is a gene pool that evolves over time to produce and better solutions to a problem. The search process typically continues until a pre-specified fitness value is reached, a set amount of computing time passes or until no significant improvement occurs in the population for a given number of iterations.

A well designed chromosome must accurately represent a solution to the problem and allow the GA operators to work effectively to generate better solutions as the iterative evolutionary process continues. Solving the TSP using Gas has generated a great deal of research focused on how to perform the action of “evolving” an optimal (or good) solution to the problem.

A new approach of chromosome representation, the so-called two-part chromosome technique can be found [7] which reduce the size of the search space by the elimination of redundant solution.

B. Fitness function

The fitness function assigns a numeric value to each individual in the population. The value define some kind of goodness, thus it determines the ranking of the individuals. The fitness function is always problem dependent.

In the case the fitness value is the total cost of the transportation, i.e. the total length of each round trip. The fitness function calculates the total length for each function calculates the total length for each chromosome, and summarizes these values for each individual. This sum is the fitness value of a solution. It is a minimization problem, thus the smallest value is the best.

C. Selection

Individuals are selected according to the fitness. The better chromosomes are more chances to be selected. The selected individuals can be presented in the new population without any changes, or can be selected to be a tournament for a crossover. We use the so-called tournament selection because of its efficiency.

Tournament selection, a few individuals are selected from the population randomly. The winner of the tournament is the individual with the best fitness value. Some of the first participants in the randomly are selected in to the new population.

D. Mutation

The mutation operator induces changes in a small number of chromosomes units. Its purpose is to maintain the population diversified enough during the optimization process. The mutation operation in GA works on a single chromosome at a time and alters the genes of the chromosomes randomly. In this TSP application using GA, the mutation plays the primary role in arriving at the solutions.

IV. GENETIC ALGORITHM IN MTSP

GA is relatively new global stochastic search algorithm which based an evolutionary biology-and computer science principles [1]. Due to the effective optimization capabilities of Ga [2], it makes these techniques suitable solving TSP and mTSP problems.

A. The Permutation representation for mTSP

If modeled properly the mTSP can be solved using any of operators developed for the TSP. Researchers have proposed two methods of modeling the mTSP search that a TSP operator can be used. One method uses two chromosomes to model the mTSB. The first chromosome provides a permutation of the cities and the second chromosome assigns a salesman to the city in the corresponding position of the first chromosome [5]. This method requires two chromosome of length n to represent a solution.

Every genetic algorithm starts with an initial solution set consist of randomly created chromosomes. This called population. The individual in the new population are generated from the previous population’s individuals by the predetermined chromosome methods. The algorithm finishes if the stop criteria is satisfied. For a specific problem it is a much more complex task, we need to define the encoding is called permutation encoding.

B. The two part technique

Fig-1 illustrates one approach for representing solution to an MTSP (where $n=15$ and $m=4$) that we refer to as the “Two chromosome” technique. as the name implies this method requires two chromosome, each of length m , to represent a solution. The first chromosome provides a permutation of the n cities while the second chromosome assigns a salesman to each of the cities in the corresponding position of the first chromosome.

1) Cities

2	5	14	6	1	11	8	13	4	10	3	12	15	9	7
---	---	----	---	---	----	---	----	---	----	---	----	----	---	---

2) Salesman

2	1	1	3	4	3	2	4	4	1	3	2	1	2	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fig. 1: Example of the two chromosome representation for a 15 city mTSP with 4salesman

In the example in Fig-1, cities 2,8,12 and 9 (in that order) are visited by salesman 2, cities 5, 14, 10 and 15 (in that order) are visited by salesman 1; and remaining cities being assigned to salesman 3 and 4. Using this chromosome design, there are $n! m^n$ possible solutions to the problem. Where n is the number of cities and m is the number of salesman. However, many of the possible solution are redundant. For example, the first two genes in each of the above chromosomes can be interchanged to create a different chromosome that results in an identical (or redundant) solution.

V. EXPERIMENTAL RESULT

Although the algorithm was tested with a big number of problems, only an illustrative result is presented here. As it was mentioned earlier the algorithm has implemented in MATLAB. The example represents a whole process of a real problem’s solution. The input data is given by distance matrix, MTSP is ($n=15, m=4$) the multiple traveling salesman problem is determined as follows: Given n cities, known as nodes, a distance matrix where, $D = [d_{ij}]$ consist of the distance between city i and city j . An attempt to finding near optimal solution for NP-hard problem

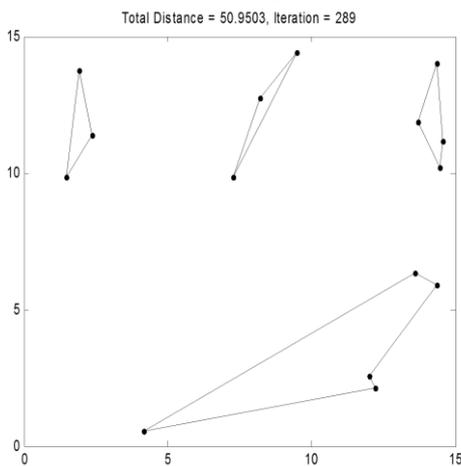


Fig. 2: The example of MTSP (initial inputs)

To analyses the new representation of genetic algorithm using this approach was developed in MATLAB. This approach is more effective of the two chromosome technique. It requires input set, coordinates of the cities and the distance table which contain the traveling distances of

cities. The fitness function simply summarizes the overall route length for salesman inside individuals.

The section is tournament selection, where tournament size i.e. the number of individuals who complete for survival is 8. The population size must be divisible by 8. The winner of the tournament is the member with the smallest fitness, the individual is selected for new individual creation, and the member will get into the new population

The winner of the tournament is the member with the smallest fitness, the individual is selected for new individual creation, and the member will get into the new population

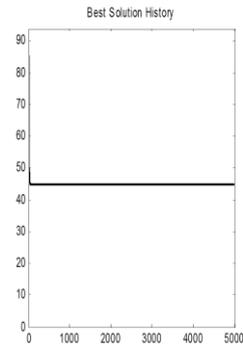


Fig. 3: mTSP Best history solution

The traveling salesman problem (TSP) is considered for combinational methods. it provides a standard test bed, to find near optimal to NP-hard problems.

Problem name	Optimal solution	Best known value
Att48	33551	42402
Dantiz17	699	770
Fri26	937	1100
Gr17	2085	2200
Ha30	Unknown	513
Kn57	Unknown	14914

Table 1: Simulation results

The test problems were selected from a standard collection of TSPs that were transformed into MTSPs by using more than one salesman (m) to complete the tour. The repeated own tests using the objective of minimizing the longest individual salesman tour. The fitness values decreases as the number of salesman increase we are dividing the cities up among more salesman. This creates a situation where each salesman visits fewer cities, so the fitness values (improve) with the addition of more salesmen. The results of these tests are presented in Fig-3

VI. CONCLUSION

This paper explored a chromosome representation for multiple traveling salesmen problem. A novel representation based genetic algorithm has been multiple depot version of MTSP. The main benefit is transparency of the representation that allows effective incorporation of heuristic and constraints and allows easy implementation of minimizing the cost of the tour. Some heuristics can be applied to improve the effectiveness of the algorithm, appropriate choice of the initial population. Moreover, the characteristics of the mTSP seem more appropriate for real-life applications, and it is also possible to extend the problem to a wide variety of vehicle routing problems (VRPs) by incorporating some additional side constraints.

REFERENCES

- [1] Goldberg, D.E., Lingle, R.J (1985). Alleles, Loci, and the Traveling Salesman Problem, Proceeding of an International Conference on Genetic algorithms, London, 154-159. Goldberg, D.E (1989) Genetic Algorithms in search, Optimization and machine Learning, Addison-Wesley.
- [2] Reading, MA. Carter AE, Ragsdale CT (2006) A new approach to solving the Multiple Traveling Salesperson using genetic algorithm, European journal of operational research 175:246-257.
- [3] Bektas T (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega, 34:209–219
- [4] Malmberg CJ (1996) A genetic algorithm for service level based vehicle scheduling. European Journal of operational research 93(1):121-134.
- [5] Kira'ly A, Abonyi J (excepted to 2010) Optimization Multiple traveling salesman problem by a novel representation based genetic algorithm. In M.Koeppean, G.Schaefer, A.Abraham, and L.Nolle, editors, Intelligent computational in Engineering: Techniques & Applications, Advances in Intelligent and soft computing. Springer.

