International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 1

ISSN: 2454-4248
25 – 29

Solving Traveling Salesman Problem for Large Spaces using Modified Meta-Optimization Genetic Algorithm

Maad M. Mijwel
Computer science, college of science,
Baghdad University
Baghdad, Iraq
maadalnaimiy@yahoo.com

Aysar Shamil[2]
Computer science, Information technology faculty,
Middle East University
Amman, Jordan
Eng.aysser83@gmail.com

Abstract**—** Traveling salesman problem also called TSP is defined to find the best shortest way between n cities such as nodes, customers, and branches etc. with known distances for traveling between each city on GPS, where the salesman leaves a location in the city, visits each of the cities just once and returns back to the origin of city where he started. The traveling salesman problem is one of the NP-hard problems (nondeterministic polynomial time) in optimization. It has a wide range of applications including distribution, planning, logistics, and it has been studied by researchers and academicians for so many years. In this paper, applied Meta-optimization genetic algorithm with neural networks is used to solve the TSP for finding all the best paths between all n cities. The meta-optimization genetic algorithm is good to find the way between all cities with low progress, less time, and compared with the TSP just using a genetic algorithm with the same parameters using the same map for the cities or nodes**.**

Keywords- *Traveling salesman problem, genetic algorithm, Meta-optimization genetic algorithm, neural networks.*

*****

I. INTRODUCTION

The TSP was studied in the eighteenth century in the book titled The Graphic Theory by an Irish mathematician named Sir William Rowam Hamilton and published by British mathematician named Thomas Penyngton Kirkman in 1976 [1]. The Travelling Salesman Problem is one of NP's best-known problems, which means that there is no exact algorithm for solving polynomial time. Minimum time is required to obtain an optimal solution [2]. The applications of the TSP are drilling of printed circuit boards (PCB), overhauling gas turbine engines, X-Ray crystallography, Computer wiring, the order-picking problem in warehouses, Vehicle routing, and Mask plotting in PCB production [3]. The problem is known to be NP-hard, therefore many heuristics have been proposed to find near-optimal solutions [4].

Genetic algorithms (GA) are types of optimization algorithms used to find the optimal solutions for a given computational problem that maximizes or reduces the function of a given function [5]. GA involves three basic steps; evaluation, crossover and mutation [6]. The GA was first proposed by Holland in 1975 and has emerged as a search and optimization method similar to the evolutionary process in biological systems [7]. Genetic algorithm and neural networks are both inspired by calculations in a biological system. Genetically, a good understanding of the biological nervous system. Artificial Neural Networks and Genetic Algorithms are two optimization and learning techniques, each with its own strengths and weaknesses [8]. There are three ways of using GA with artificial neural networks [9]

i) Weights learning: When traditional methods (e.g. backward propagation) are not feasible, optimal network weights are found by GA. It is adapted when the neuron's continuous activation function (such as sigmoid) is used so that the error function is multi-extremal and the conventional method can only find the local minimum.

ii) Architecture optimization: GA is used to find the best net architectures of network architects in certain parameterized classes.

iii) Learning procedure optimization: In this costly but effective method, GA is used to find the parameters to optimize the learning function (weight correction function). Usually, this method is used in the same way as an architectural optimization.

Meta-optimization genetic algorithm (Meta-GA) to optimize the simplicity of GA through an evolutionary process to solve problems in the GA [10]. The Meta-GA are able to find a good solution even in large search spaces with a reasonable time using neural networks [11]. The conceptual structure of a Meta-GA optimization approach is shown in Figure.1[12].

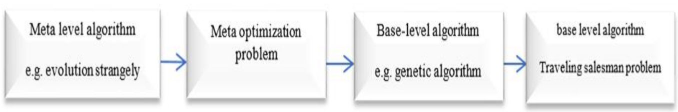

Figure.1. The conceptual structure of a Meta-GA optimization

II. PROBLEM STATEMENT

The problem with TSP is finding a complete route connecting all the nodes of a network and visiting them only once, returning to the starting point, and also reducing the total distance of the route to the minimum. This type of problem has great application in the field of logistics and distribution, as well as in the programming of the production curves. The real problem lies in the size of the large space, the number of potential paths is determined by the equation $(n-1)!$ where n is the number of points (cities) for example in the network 6 nodes the number of provable paths is equal to (7-1)! = 720 as shown in Figure.2, the problem is symmetric, i.e. the number of potential paths is halved, that is $((n-1)!)/2$, hence there are 360 different routes and do not need a computer to find the best solution for routes but when the nodes are greater than 6 nodes, for example when in network 30 cities there are more than $4*10^{31}$ possible routes. A computer that calculates one million routes per second would need 10^{18} years to solve problem, which means a significant saving in the processing time of large routers.

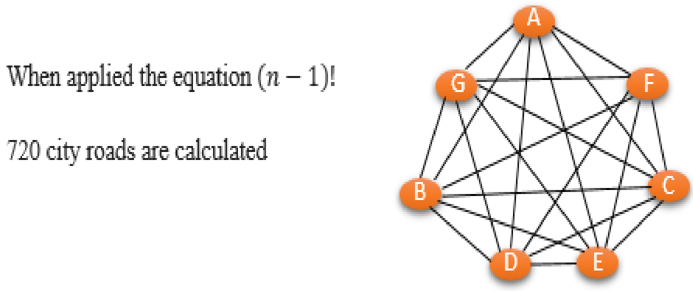

Figure.2. TSP for 6 nodes in network

III. META-OPTIMIZATION ALGORITHM

The Meta-GA belong to the family of evolutionary algorithms and the goal of Meta-GA is to obtain an approximate solution to an optimization problem, when the solution to solve it in a reasonable time is unknown. It consists of six main parts, the first is one chromosome as one of the solutions to a problem considered. It is generally encoded with a bit or character vector; the population is a set of solutions to the problem considered, Gene: part of a chromosome generally consists of one or more parts of a vector of bits or characters that encode the chromosome, fitness: assessment grade associated with a solution. Evaluation is based on a specially designed function called fitness function. Crossover means generating a new solution mixed existing solutions, and the last is mutation which is a random alteration of a solution.

Living organisms consist of cells whose nuclei contain chromosomes that are DNA chains. The basic element of these chains is a nucleotide, identified by its nitrogen base (A, T, C or G). On each of these chromosomes, a sequence of nucleotides constitutes a chain which codes the functionalities of the organism, for example, the color of the eyes. Thus, a gene is a functional phrase along the chain. The position of a gene on the chromosome is its locus. The set of genes of an individual is his genotype and the whole genetic inheritance of a species is the genome. The different versions of the same gene are called alleles.

Meta-GA method consists of a population is a set of individuals of binary bit strings, the initial values are randomly determined and evaluated. Each combination of associations and zeros is a probability that a complex field can be searched, and the relationship between them is an evaluation function that will return a capability or classification for that particular bit-string. In binary coding, each chromosome represents a sequence of 0 and 1, while in the permutation coding method, each chromosome represents the sequence of the characters that makes it up. The encoding of the chromosomes of the salesman problem is made according to the permutation method. While we are watching the points, each of the routes we watched while watching the cities are represented as chromosomes. Accordingly, for a population of 7 cities as shown in Figure.2: (ACDBFGH) -(ABFEHGD) -(AFDECHG), which represent chromosomes that represent different navigational sequences. In general, the pseudocode of Meta- optimization GA consists of the follow steps:

- Initialization: The initial population is generated randomly, which is constituted by a set of chromosomes representing problematic solutions. If it is not done randomly, it is important to ensure that within the initial population, you have the structural diversity of these solutions to have a representation of as much of the population as possible or at least avoid premature convergence.
- Evaluation: for each chromosome in this population, the capability function will be applied to "know" how the "good" solution is used.
- Term requirement: The Meta- optimization GA should be stopped when the optimal solution is obtained, but this is often unknown, so other detention criteria must be used. Normally two criteria are used: run the maximum number of iterations of Meta-optimization AG or stop the run when there are no changes in the population. While the termination requirement is not met, the following is done:
  1- Selection: After knowing the capacity of each chromosome, selection of the chromosomes to be passed on to the next generation is made. Chromosomes with better fitness are possible to be selected.

  2- crossing: The recombination is the main of Meta-optimization GA operator, represents sexual reproduction, operates on two chromosomes at once to generate two descendants where the characteristics of both parent chromosomes are combined.

  3- Mutation: It randomly changes part of the chromosome of the individuals, and allows access to the fields of the search space that were not wrapped by the individuals of the current population.

  4- Replacement: Once the Meta-optimization GA operators are applied, the best individuals are chosen to confirm the population of the next generation.

<u>Meta- optimization genetic algorithm</u>
<u>Initially</u>

ISSN: 2454-4248
25 – 29

Randomly generate population = a set of solutions to the problem at each step

- Select the best individuals in this population (the elite) according
  to the criteria, we want to optimize
- Generate a new population by crossing member of the elite
- Modify t% of the individuals of the new population by mutation

At the end

keep only the best solution for the population

IV. TSP WITH MODİFİED META-GA

The TSP can be formulated by integer linear programming. It equal [1] if there is a way to go from city i to city j, and 0 otherwise, for all cities $0,\ldots,n$. Then the integer linear programming model can be written as $x_{ij}u_{ij}c_{ij}$, in a map we have A is collection of n cities or nodes where $A=\{1,2,\ldots,n\}$ to distributed the cities or nodes in the maps based on the equations (1)(2)(3)(4) , in the equation (1) where x and y, the coordinates of the cities and i,j the cities and $x(i,j)$ denote the way from point $1,i$ to point j and the way from point $0,i$ to point j, where p represents the population

Min:

$$p=\sum_{i=1}^{n}\sum_{j=1,i\neq j}^{n} x(i,j)y(i,j) \quad (1)$$

Constraints

$$\sum_{j=1,i\neq j}^{n} x(i,j)=1, i=A \quad (2)$$

$$\sum_{j=1,i\neq j}^{n} y(i,j)=1, j=A$$

(3)

$$\sum_{j=1,i\neq j}^{n} x(i,j)\leq |S|-1AS\subset\{1,2,\ldots,n\}$$

(4)

Figure.3 shows the distribution for all cities in the map

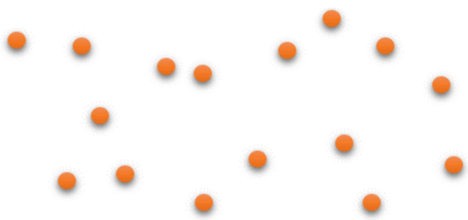

Figure.3. nodes distribution in the map

The first set of equations ensures that each city $0,\ldots,n$ of departure reaches punctually one city, and the second set of equations ensures that from every city $1,\ldots,n$ is exactly to a city (both restrictions also mean that there is punctually one exit from city 0). The last restriction requires that only one road covers all cities in the map and not two or more disjoint roads cover all cities together. To prove this, it is shown in the equation (1) that every possible solution contains only a closed sequence of cities, and in the equations (2), (3) and (4) for each of the routes that cover every city, there are values for all variables satisfying the constraints. To prove that each possible solution contains only a closed sequence of cities, it is sufficient to show that each sub-route in a feasible solution passes through city 0 (note that the equations only provide such a route). Therefore, if we add all the difference conforming to each sub-path of d steps that do not pass through the city 0.

$$x_{ij}=1nd \leq (n-1)d \quad (5)$$

Calculate the number of possible routes between all the cities by calculating the number of chromosomes using equation (6)

$$C_i=R*\sum_{i=1}^{n}\beta_i r_i(r_s-r_g) \quad (6)$$

Where C_i is the total proposed roads going from city i to city j and back to the origin point, R denotes to a nearby city. β_i denotes nearby city odds. r_s is the parant road of the big roads between cities and r_g is the chiled road the small roads between cities , and r_i is the probability of all roads between all cities. Figure.4 shows an example of the roads between all cities with distance.

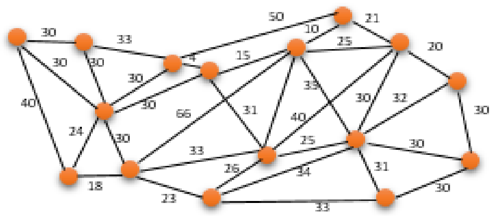

Figure.4. the distance between all cities

The number of iterations calculated by equation (7) using K_i is the iteration of salesman in the maps

$$K_i=\sum_{i=1}^{n}\sum_{j=1}^{n}\sqrt{(x_1-x_j)}-C_i \quad (7)$$

Equation (7) will number the steps proposed for moving between all cities by the salesman. Now, a show on the map all tours will cover every city, there are values of the variables that accept the constraint s. t the total time needed by the salesman to move between the cities which is calculated by the following equations

$$t_i=tu_i-tu_j\leq n-1 \quad (8)$$

$$t_j=tu_j-tu_i\leq n-1 \quad (9)$$

t_i is the total time needed by salesman to go from city i to city j, while t_j is the total time needed by salesman to return back from city i to city j. tu_i, tu_j denote the time it takes to go and return to each city $i, and\ j$. By equation (10), the total time required to reach all cities is obtained the salesman will arrive and set his course in less time

$$t_i + t_j + n_{ij} = (t) - (t-1) + n = n - 1 \qquad (10)$$

Optimizating the number of iteration for all roads in the map is achieved by using Meta- optimization GA to get path that is used by the salesman as shown, for applying the optimization, using equation (11)

$$f(K_i) = \left(\sum_{i=1}^{n-1} .0.5 + \frac{\sin^2\left(\sqrt{x_i^{\,2}+x_{i+1}^{\,2}}\right)-0.5}{\left((1+0.0001.(x_i^{\,2}+x_{i+1}^{\,2}))\right)}\right) - K_i \qquad (11)$$

Figure.5 shows the path in bold to reach all cities by the salesman and return back to the start point in shortest time and less expensive

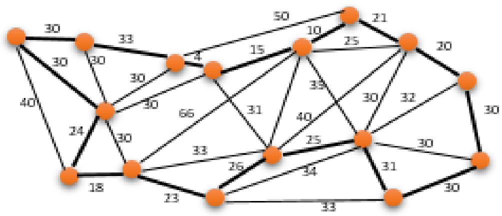

Figure.5. optimizations all roads between cities

V. Performance Evaluation

The mathematical model for TSP is solved by using visual studio C# software for optimization as it is suitable to calculates the best route between all cities. It is required to determine the cities of the user or the roots of the cities can be calculated by taking the city lists with their coordinates as XML in the program. In Meta-optimization GA, it is necessary to create an initial population in which the new generations can be derived in the first stage where the initial population is generally randomly generated. The populations consist of a collection of chromosomes (links) connecting the cities and these links are not completely random as Meta-optimization GA will choose to establish links between cities that are close together. There are 5 initial parameters to control the operation of Meta-optimization GA as shown in Table.1. All these values are used in all tests and compared with the result of TSP with genetic algorithm [13].

Table.1. the initial parameters of Meta-optimization GA

Parameters	values
Population size	**10,000**
Group Size	**5**
Mutation	**3%**
Close City	**5**
Near city probability	**90%**

Figure.6 shows the result of the Meta-optimization GA showing the number of iteration with 40 cities and determining the coordinates distribution randomly

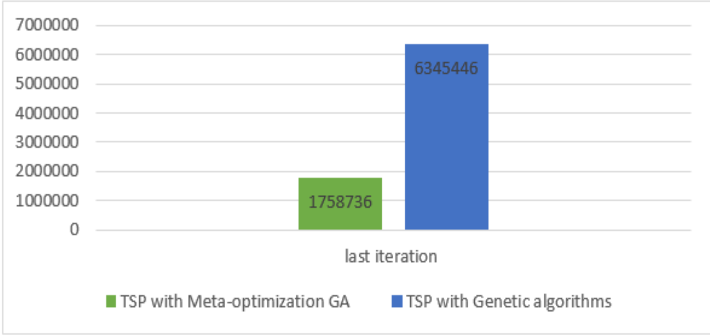

Figure.6. Comparisons between TSP Meta-GA and TSP GA by the last iteration

Figure.7 shows the result of the Meta-optimization GA showing the number of tours with 40 cities with determining the coordinates distribution randomly

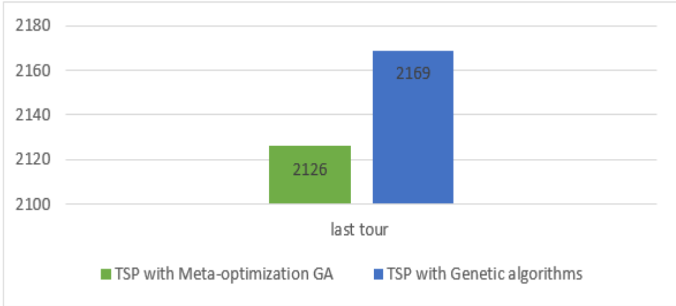

Figure.7. Comparisons between TSP Meta-GA and TSP GA by the last tour

I. Conclusions

In this paper modification, Meta-optimization GA is used for solving the TSP for large space. After implementation of the modified Meta-optimization GA achieved a high result in connected and find the best away for Traveling salesman problem TSP in order to go to all cities and return back to the city where he started with less time and no matter how many cities in the map. The lowest number of the tour it takes to find the best ways do not exceed 20 seconds. Reverse the previous algorithm it took less one minute and sometimes took more than a minute to find the best way. The TSP with modified Meta-optimization genetic algorithm rate achieved 99% to find the best path between cities with less time while the TSP with genetic algorithm achieved rate 80%. Figure.8 and Figure.9 show the accuracy of TSP with modified Meta-optimization genetic algorithm and TSP with a genetic algorithm.

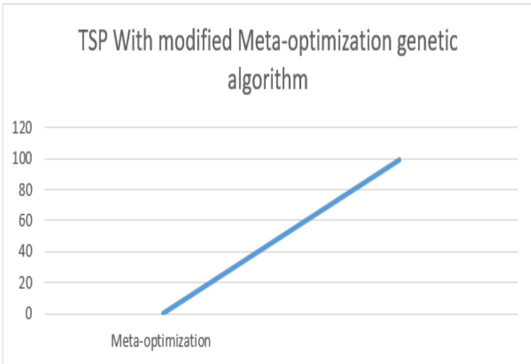

Figure.8. Completion rate of modified Meta-optimization genetic algorithm

ISSN: 2454-4248
25 – 29

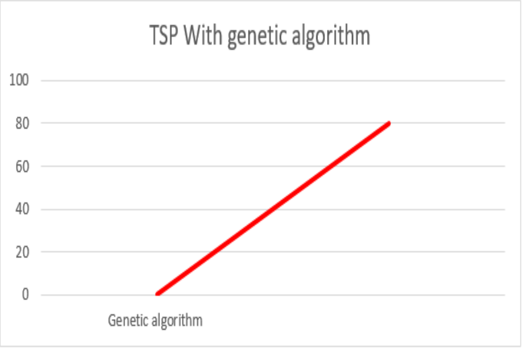

Figure.9. Completion rate of genetic algorithm

REFERENCES

[1] Bektas T. Travelling Salesman Problem Formulation and solution. Omega 2006;34: 209-219.

[2] Brezina I, Čičková Z. Solving the Travelling Salesman Problem Using the Ant Colony Optimization. Management Information Systems 2011; 4: 010-014.

[3] Arya1 V, Goyal A, Jaiswal J. An Optimal Solution to Multiple Travelling Salesperson Problem Using Modified Genetic Algorithm. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 1 January 2014; 3: 2319 – 4847.

[4] Said A, Mahmoud M, El-Horbaty M. A Comparative Study of Meta-Heuristic Algorithms for Solving Quadratic Assignment Problem. International Journal of Advanced Computer Science and Applications 2014; 5.

[5] Carr J. An Introduction to Genetic Algorithms. 2014.

[6] Wen Tsai M., Pei Hong T, Tsong Lin W. A Two-Dimensional Genetic Algorithm and Its Application to Aircraft Scheduling Problem. Hindawi Publishing Corporation Mathematical Problems in Engineering 2015.

[7] Melanie M. An Introduction to Genetic Algorithms. Massachusetts Institute of Technology 1996;2626-385.

[8] Mahajan R, Kau G. Neural Networks using Genetic Algorithms. International Journal of Computer Applications 2013; 77: 0975 – 8887.

[9] Sharapov R. Genetic Algorithms: Basic Ideas, Variants, and Analysis. Open Access Database 2007; ISBN 978-3-902613-05-9.

[10] Bajpai P. Genetic Algorithm – an Approach to Solve Global Optimization Problems. Indian Journal of Computer Science and Engineering 2010; 1: 0976-5166.

[11] Alharbi S, Venkat I. A Genetic Algorithm Based Approach for Solving the Minimum Dominating Set of Queens Problem. Hindawi Journal of Optimization Volume 2017,

[12] Díaz M, Pichler F, Arencibia Q. Computer Aided Systems Theory Euro cast 2011” 13th International Conference. Las Palmas de Gran Canaria, Spain, 2011; 2.

[13] Traveling Salesman Problem Using Genetic Algorithms (http://www.lalena.com/AI/Tsp/).

[14] Maad M Mijwel. Genetic Algorithm Optimization by Natural Selection. Augusts,2016.