

# Solving Travelling Salesman Problem and Mapping to Solve Robot Motion Planning through Genetic Algorithm Principle

K. S. Suresh<sup>1\*</sup>, V. Vaithyanathan<sup>1</sup> and S. Venugopal<sup>2</sup>

<sup>1</sup>School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India;  
kssuresh@cse.sastra.edu, vvn@it.sastra.edu

<sup>2</sup>Indira Gandhi Centre for Atomic Research, Kalpakkam - 603102, Tamil Nadu, India;  
venu@igcar.gov.in

## Abstract

Travelling Salesman Problem (TSP) is a very old problem which has been solved in so many methodologies. The solution for the Robot Path Planning (RPP) can be derived using the methodologies used for the TSP. In this paper, Genetic Algorithm (GA) principle is employed to solve the TSP and is mapped to solve the RPP with the same principle. The both problems are defined to observe similarity between these problems and enumerated the conversion phases. In these two cases how the solutions are to be derived to implement the GA technique to accomplish the optimal path in both cases and tested for different number of cities, population space and generations. The minimum cost and mean cost of the solution space proves its giving the optimal result.

**Keywords:** ASCII, Encryption, Hackers, Random Number, Symmetric Key

## 1. Introduction

TSP and RMP are broadly categorized under the optimization problem. Specifically both are NP-hard combinatorial optimization problem which gives a lot of room for the research activities. Both are constrained optimization problems those have potential to adopt any emerging methodologies tabled by the research field to test. Therefore TSP and RMP can be used in two back to back ways as:

- To test the new techniques with already existing experimented results or,
- To adopt and map into new methodology and derive the target solution.
- It may be distinctively for research purpose with simulation of real time scenario and test the accuracy and

efficiency of the methodology or may be for real time implementation.

Intentions of this paper are:

- Analyzing the nature of the TSP and RMP problems.
- Identifying the resemblance of the problems.
- Solving the TSP using Genetic Algorithm.
- Mapping the methodology to solve RMP.

Various methodologies were tested to solve the TSP such as dynamic programming by Bellman<sup>2</sup>, branch-and-bound method and the minimal spanning tree approach proposed by Lawler<sup>3</sup>. John Holland suggested the genetic algorithm principle to find the solution for optimization problem based on the heuristic approach. Heuristic approach was proposed to solve the travelling sales man

\*Author for correspondence

problem by Tadei<sup>4</sup>. TSP was solved by implementing Genetic Algorithm by Potvin<sup>5</sup>. The model for the Stochastic Travel Costs for multipath travelling salesman problem was implemented Srinivas<sup>6</sup>. The TSP was tested by Suresh<sup>7</sup> with various cross over and mutation rate for different population size and number of cities to be visited. Canny<sup>8</sup> in his “The Complexity of Robot Motion Planning” proved that all the robot path planning problems are considered as the NP-complete problem. The paper titled “Emergence of Meta Heuristic Algorithm Principles in the Field of Robot Motion Planning – A Survey of Paradigm Shift” by Suresh<sup>9</sup> classified various approaches and methodologies to attain the solution for RMP. The principle of GA was first tested for RMP by Parker<sup>10</sup>, after that GA was employed by Renner<sup>11</sup> for RMP with lot of variations. On the other hand, lot of research are going on for robot manipulator movement for kinematic solutions using different methodologies<sup>12</sup>.

## 2. Defining TSP and RMP

According to TSP, the salesman has to plan a tour starting from home city and return to home basically with three constraints. Salesman has to visit all the cities from home city, cities to be visited only once during the tour and tour to be planned with shortest path.

Similarly Robot Motion Planning (RMP) is a plan to reach the target from starting point with two essential constraints. They are avoiding obstacles in the path and taking the shortest path to reach the goal. In addition to that depends on the environment, the list of secondary constraints may be lengthened as jerks, acceleration, speed etc. and more constraints with respect to physical nature of robots and moving space configuration.

## 3. Definition Level Comparison

From the definition of TSP and RMP, it can be observed that in either case, the inputs or known data is different points or locations to take the travel. The solution space is defined as optimal path from source to destination for RMP. For the TSP, destination is nothing but the starting point. Both provide lot of feasible paths to obtain the best out of these paths. The best is defined by the constraints of the problems. Out of all feasible paths, selecting the minimum cost path to reach the destination is the common constraint. Other constraints are respective to their specific applications. It is obvious that resemblance exist in each phase.

## 4. Implementation Level Comparison

### 4.1 Mathematical Formulation of Problems

Mathematical compliance of the problem representation is the key to solve any problem. It is to be represented in a specific way to fit into the method by which the problem to be solved. Matrix form of representation is preferred as a foremost problem solving techniques. The manipulation of the matrix is so easier and implementation of the problem through computer programming also too convenient, thanks to enormous support for matrix manipulation build-in functions provided by computer languages.

Fundamentally the least objective for TSP and RMP is to develop the path traversing through the cities or points respectively for the given environment space. The environmental space can be defined through the matrix.

In case of TSP, if ‘a’ is denoted as a city, first step is determining the location of the city in the map with (x, y) coordinate system. The same may be followed in the case of RMP also for 2D path planning.

In TSP, the next step is generating the distance matrix. If there is a path between any two cities (considered as a and b), the distance has to be calculated and placed in an appropriate cell in the matrix. If there is no path exists, the cell will be specified by zero.

The distance between two cities is calculated by

$$d = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}.$$

In RMP, the equivalent matrix is formed by virtually decomposing the 2D plan into cells and for implementation, the cells occupying obstacles are placed by 0s (no path) and the cell that allows for navigation is filled by 1s.

For TSP Dist (a, b) = d (if path exists between the cities).  
0 (if no path between the cities).

For RMP Dist (a, b) = 1 (if no obstacles in the cell).  
0 (if obstacle in the cell).

### 4.2 Enumeration of Candidate Paths

From the formulated matrix, all the feasible paths are identified and enumerated. These generated feasible paths are different for these two problems.

### 4.2.1 TSP Candidate Paths

In the candidate paths of TSP, the number of elements in each path is one more than total number of cities, since tour has to end at starting point. Therefore the last element will be same as the first element. This is to be considered when generating candidate paths that the last elements of all the candidate paths are to be same element.

### 4.2.2 RMP Candidate Paths

Same as TSP, the destination of the RMP candidate path would be the target cell of the matrix form of configuration space. Hence for the RMP also the last elements of all the candidate paths are going to be the same.

### 4.2.3 Finding the Solution Path

In both problems, there is a possibility for lot of feasible paths from which the optimal path must be derived, i.e. selecting best path out of the available paths. But in the TSP case, if the number of cities increases complexity of the problem is getting increased. In RMP case, the complexity increase when the area of the robot moving space increases. In addition to that, as for as RMP is concerned, if the degree of freedom of robot increases the RMP is getting complex. But if the common point has to be taken, the increase in complexity may be viewed as the increase in number of possible paths available.

Hence selecting the best solution out of abundant paths in an efficient way is the challenge. To measure the efficiency, speed and accuracy are considered as deciding parameters. For these kinds of problems, numerous methods are available to derive the solution path. In this paper, the strategy employed to solve both problems is popular genetic algorithm principle.

## 5. Why Genetic Algorithm?

So many real time problems are categorized under NP-hard problems, may or may not reach the optimum solution. NP-hard problems can be attacked either by exact methods or heuristic approaches, considering performance parameters such as cost, accuracy and speed. Exact methods are giving assurance for accuracy, but lack in providing speed and cost factor, since it may consume lot of resources and time. On the other side heuristic methods are not guaranteed for best solution, but trying to achieve the better solution within the stipulated time frame or getting the satisfied result by considering

the chosen selection probability. The exact methods are many techniques are possible like applying deterministic optimization techniques such as linear programming, non-linear programming and dynamic programming or applying heuristic optimization techniques such as simulated annealing, genetic algorithms etc.

Therefore in TSP, if the number of cities increases, the possibility of the growth in size of the feasible paths exponentially. The RMP is alike TSP, if either resolution of the robot increases or the task space broadened, in turn the number of cells also increase exponentially. These problems can be solved by considering all candidate paths which is huge in number. When the size of the solution space grows enormously for any problem to reduce the time to converge to the optimal solution, the very suitable approach is heuristic approach.

## 6. Ideology of GA

GA is the evolutionary approach which takes encoded candidate solutions as the population in the first phase. From the derived population estimate the property or quality with fitness function to select the better population to the next generation. Applying genetic operators like cross over and mutation on the population and again do the selection process to get the better population. This will be repeated until it reaches the satisfied solution phase or time bound.

## 7. Generalized Comparison of Implementation of GA Technique to Solve TSP and RMP

### 7.1 GA for TSP

The candidate solutions for TSP are all the possible routes which are starting from the city and reaching the destination. The population space for the given starting city is generated from the already created distance matrix. The fitness function has to be formulated to estimate the quality of the population. TSP is meant for the shortest path, hence the distance to complete the tour is the criteria to evaluate the fitness of the path. After this phase, using selection method, the better paths are selected for the following phases as cross over and mutation. Then the manipulated population is scrutinized to get the better

population to supply it to the next iteration or generation. This is accomplished until the satisfied output is reached i.e. desired fitness value is emerged or number of iterations are completed.

Specific to the TSP, applying crossover and mutation operator may lead to the duplication of the cities in the each candidate path which are to be replaced by the missing cities in an arbitrary manner. Also it has to be confirmed that the first and last element of each path should be the starting city. This is legalizing the path for further estimation and manipulation.

## 7.2 GA for RMP

In RMP, the population derived from the matrix is the feasible paths comprises of cells to be traversed to reach the goal by avoiding obstacles. For all the candidate paths the last element should be the goal point. The fitness function for the RMP is also defining the shortest path to reach the goal which will be used to determine the probability of the participation of the specific path in the population for the next generation. As in TSP, after exercising the crossover and mutation operator, the RMP paths also to be altered to avoid duplicate elements and the last element should be the row index of the goal cell. Then selection process will be accomplished to continue to the next cycle.

## 8. Comparison of Cost Functions

### 8.1 Construction of Cost Function for TSP

The cost function for TSP is formulated by estimating the distance between the cities for the route to reach the city where it starts. Since the distance is directly proportional to the cost, attaching the cost per unit distance may give the cost incurred for the tour. Other conditions may be added depends on the other constraints pertain to the route to be evaluated.

### 8.2 Construction of Cost Function for RMP

The cost function for RMP is devised same as the case of TSP. The distance from the current cell to the next possible cell to be navigated is calculated. Above this, since the robot is constituted by arms and wrists which are controlled by actuators, the obstacles and the angles to be turned to reach neighboring cell also accounted for estimating the cost function for the robot path. If the next cell is away from the current cell, it is penalized depends on the number of cells distant from the current cell. This

penalty is to reduce the angle to be turned by the robot by which smoothness in movement is achieved.

## 9. Implementation of TSP

The Genetic Algorithm for TSP is tested by generating different number of path population for different number of cities to be travelled.

### 9.1 Phase 1: Distance Matrix Creation

The first step for the implementation of the GA to solve TSP is establishing the distance matrix for the given number of cities. The distance between the cities is randomly generated and formed the distance matrix by taking cities as rows and columns. Hence the diagonal elements are going to be zeros. In addition to that, the zeros will be placed where the path not exist between the cities.

### 9.2 Phase 2: Population Generation

After establishing the distance matrix, all possible routes are identified and population is generated. While generating population, starting and destination should be the same.

### 9.3 Phase 3: Duplication Avoidance

Redundancy of generated routes should be identified and removed from the population.

### 9.4 Phase 4: Crossover Implementation

Performing cross over between the population and generate off springs for the next generation. The Crossover Probability (COP) decides how many genes are going to be participating in producing next generation. The COP influences the speed of emergence of the optimal solution. According to<sup>7</sup>, it is analyzed and proved that high value of COP will lead to the possibility of emergence of result in less number of iterations.

### 9.5 Phase 5: Mutation

Mutation operator is applied on randomly selected candidate paths as per the assumed the Mutation Rate (MR). MR tells the number of paths involved in the mutation process. As suggested by<sup>7</sup>, less MR will allow settling in optimal solution in earlier iterations.

## 9.6 Phase 6: Correction and Cost Function

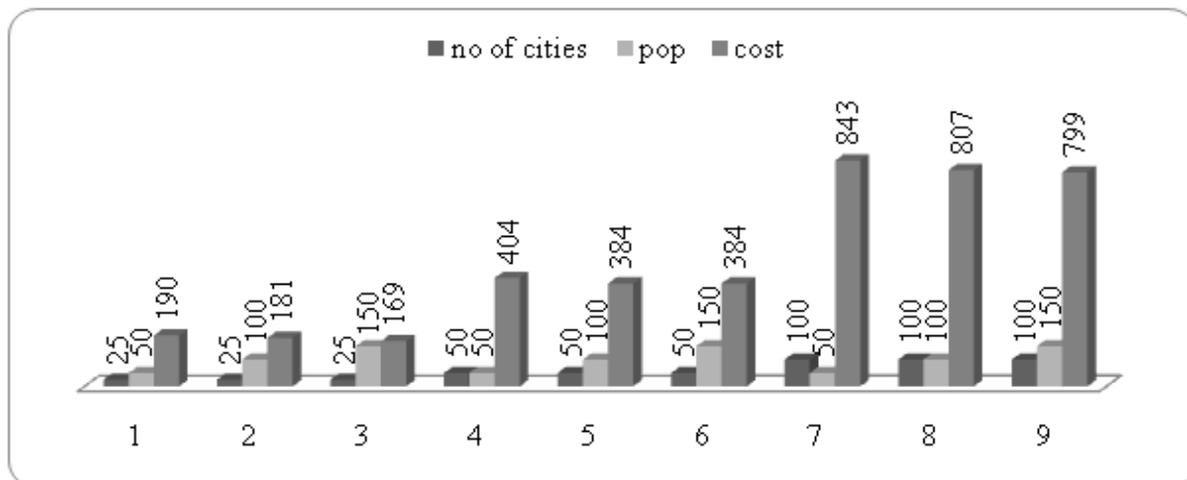
After applying the genetic operators on the population, in the offspring the duplication of cities may occur in the sequence which is to be corrected to calculate the cost for individual paths.

## 9.7 Phase 7: Evaluation and Selection

In this phase, from the cost of the individual phase, the best of 90% of the population are selected and passed to the next generation to undergo the next iteration. Again it starts the new iteration from the phase 4. In real time scenario, the iteration will be stopped either after attaining the expected minimum cost value or number of iterations. But for the research purpose, iterations were continued until it completes the desired number of iterations. The output of the analysis is depicted as chart to describe the result.

## 10. Observations from the Results

In this system, the different numbers of cities (25, 50, 100) are considered and each one is tested with different number of generated populations (50, 100, 150). The chart is plotted for each combinations ((no of cities) 3 x (no of populations) 3 = 9) with generation number at which the corresponding minimum cost occurred. From the Figure 1, it can be observed that the optimal minimum cost is achieved when the population size is high.



**Figure 1.** Optimum cost for different population, number of cities, generation achieved the optimum cost.

## 11. Mapping to RMP

To get the optimal solution for RMP problem by implementing GA, the same phases have been followed as the TSP, except the mutation phase and duplication avoidance phase. Another one is the distance matrix here appropriately named as Obstacle Matrix. Since the mutation phase is altering the sequence of the path, it may be avoided. The duplication avoidance phase is not required after the population generation phase, because in RMP, only possible paths are generated from the given obstacle matrix. Therefore there is no chance of the occurrence of duplication in path population. But it is to be added after excising crossover operator.

### 11.1 Phase 1: Obstacle Matrix Creation

The obstacle matrix is established for RMP, which is equivalent to distance matrix of the TSP. As specified earlier, the two dimension task space is partitioned into cells. If there is an obstacle the cell is represented by 1 and if there is no obstacle then the cell is represented by 0. The obstacle matrix is formed to generate paths.

### 11.2 Phase 2: Population Generation

All the possible paths are identified and enumerated which are considered as population space to implement the GA methodology. While creating population, the point to be noted is, the last element of all the paths must be the target cell.

### 11.3 Phase 3: Crossover Implementation

As in the TSP, the 90% of the population is involved to generate the next level population by implementing the single point crossover operator. The higher crossover rate leads to higher number of off spring generation for the next generation; hence the probability of the emerging to the optimal solution is high.

### 11.4 Phase 4: Duplication Avoidance

Due to high cross over rate, there is a high chance to produce the more number of paths for the next generation. This is more chance to happen in case the task space is partitioned into less number of cells with more number of possible paths. In the contrary, if number of cells increases obviously the paths generated will be more which in turn increase the complexity of computation.

### 11.5 Phase 5: Cost Evaluation

For all candidate paths, using the cost function, individual path cost is evaluated considering the distance and deviations from the current cell.

### 11.6 Phase 6: Selection

After assigning the cost for each individual path, 90% of the paths are selected for the next generation as candidate paths to get the best performance in the sense the emergence of better solution will be in the earlier generations. But the population to be processed is getting increased. In contrast, if the population space is less in size the rate of emergence to the best solution will be delayed over generations.

## 12. Analysis of the Implementation

In this paper, for the convenience of analysis, the task space is assumed to be partitioned as a square matrix. Considering the six cases such as 5 X 5, 6 X 6, 7 X 7, 8 X 8 and 9 X 9, first the obstacle matrix is formed randomly with 0s and 1s as elements of the matrix. Then all the possible paths are identified and enumerated as population and applied the GA operators. Using the cost function the next generation population, the best of 90% of the population is selected. The following are the results for the above case study.

## 13. Observations from the Results

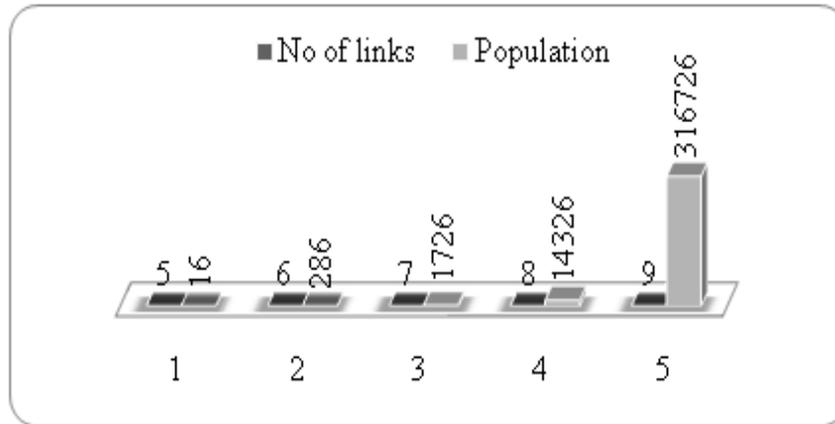
From Table 1 and Figure 2, though the matrix had been populated randomly, the results show that the number of possible paths increases many fold, even if the partition of the task space is incremented by one. This factor resists the increasing the partition of the task space. If the number of partitions of task space is increased, the computation cost will increase extremely. In contrary, the increase in the number of partitions will give lot of flexibility in terms of resolution for the movement of the robot; specifically the utilization of turning capacity can be increased by number of partitions.

## 14. Conclusion

In this paper, after performing analysis and research to find the solution for a domain specific problems, the

**Table 1.** Comparison of increase in population with increase in number of links of Robot

No of links	5	6	7	8	9
Population	16	286	1726	14326	316726
Increase in times the size of the population w.r.t. the previous links	-----	18	6	8	22
Min cost	6	7	9	13	13
Mean cost	7.986111	16.1509	19.99017	26.83452	33.52496



**Figure 2.** Comparison of different number of links of Robot with population generated.

invented general models and methodologies can be exploited positively and may be served well for the similar kind of problems arose in another domain. It is analyzed that a solution phases of one problem domain is mapped to another problem domain. It is proved that the model to solve TSP problem using GA principle can be tailored to find the solution for RMP problem.

## 15. References

- Dantzig G, Fulkerson R. Solution of a large-scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*. 1954; 2(4):393–410.
- Bellman R. dynamic programming treatment of the Travelling Salesman Problem. *JACM*.1962 Jan; 9(1):61–3.
- Lawler EL, Wood DE. Branch-and-bound methods: A survey. *Operations Research*. 1966 Aug; 14(4): 699–719.
- Maggioni F, Perboli G, Tadei R. The multi-path Traveling Salesman Problem with stochastic travel costs: Building realistic instances for city logistics applications. *Transp Res Procedia*. 2014; 3:528–36.
- Potvin JY. Genetic Algorithm for the Travelling Salesman Problem. *Annals of Operations Research*.1996; 63(3):337–70.
- Srinivas M, Patnaik L. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Man and Cybernetics*. 1994 Apr; 24(4):656–67.
- Suresh KS, Rajarajan S, Prabhu M, Vaidhyathan V. Study of the influence of Genetic Algorithm parameters of Travelling Salesman Problem. *International Journal of Applied Engineering Research*. 2013; 8(20):2667–73.
- Canny JF. *The complexity of robot motion planning*. Cambridge, Mass, MIT Press: 1988 Jun.
- Suresh KS, Vaidhyathan V, Venugopal S. Emergence of meta heuristic algorithm principles in the field of Robot Motion Planning – A survey of paradigm shift. *International Journal of Applied Engineering Research*. 2013 Oct; 8(20):2717–22.