

Travelling Salesman Problem using Dynamic Approach

Reem Al Zoubi
University of Hail
Hail
Kingdom of Saudi Arabia

Asiya Abdus Salam
University of Hail
Hail
Kingdom of Saudi Arabia

ABSTRACT

In this paper, the Travelling Salesman Problems using dynamic approach is discussed. Increasing development in our life resulted in difficulties to solve problems which can be solved in Generic Algorithm (GA) in an efficient way to reach optimal solution. Generic Algorithm follows sequence of steps to solve any problem (selection, fitness, crossover and mutation). Each of these steps has types which can be selected according to characteristics of the problem. Crossover is exchange information of offspring. So it leads us to focus on it in order to reach optimal solution in least time by changing it from generation to another. In this paper, experiment on the Travelling Salesman Problem (TSP) is applied in which time is saved and problem is solved using two point crossovers.

General Terms

Travelling Salesman Problem, Dynamic Approach

Keywords

GA, TSP, mutation, crossover, Static crossover, Dynamic crossover

1. INTRODUCTION

As the world advance, evolution becomes the notable problem solving ability and grant different range of survival strategies. The terms like technology, development and the information revolution plays vital role in our world and led to increasing the speed of delivery, performance and excellence in work. It also help to solve more problems and provide numerous methods to solve problems of decision making, classification problems and optimization problems.

The development and use of optimization model is pretty conventional. In some fields of economic analysis when the problem is huge, the use of optimization models has been restricted that also leads to large number of nonlinear influences. In order to find out the solution to the linear models approximation or simplification usage is necessary in most cases.

All these factors helped to create a science that facilitates the fastest and best solution among the set of solutions that may occur in the thinking of human which called the science of Generic Algorithm. (GA)

Generic Algorithm is a procedure based on search to compute exact or approximate solutions and search problems which is the alternative of traditional optimization technique. Generic Algorithms are most appropriate for complex nonlinear models where locating the global and best possible solution is a difficult task. Generic Algorithm is a useful approach potentially because of its techniques.

GA tag four main stages: evolution, selection, crossover and mutation. Based on the significant optimization or search criteria, the evolution procedure measures the fitness of each individual elucidation in the population and assigns it a qualified value. For development of the next generation, the course of action in selection procedure randomly selects individuals of the current population.

A range of substitute procedures have been projected but all pursue the idea that the fittest strategy has a greater chance of choice. The course of action in crossover takes two chosen individuals and combine them about a crossover point (one, two or uni form) thus creating two new individuals. The course of action in mutation method randomly transforms the genes of an individual focus to a small mutation factor, launching further arbitrariness in the population.

There are many problems solved in GA, such as Travelling Salesman, Bin Packing, 8 queens, IC board stung, Water Sprinkler, Gun firing, maze solving, cutting fabric and so on. The focus of this paper is Travelling Salesman Problem as one of the problem solved using GA.

Travelling Salesman Problem TSP is a given set of cities and distance between them which requires finding the shortest path of visiting each city only once. As mentioned earlier GA follows steps to solve any problem one of which is called crossover which has three types such as one point, two point and uniform. In TSP two points i.e. 1 order crossover is used because condition for the problem is satisfied and duplication is avoided in combination of possibilities.

A crossover point is preferred on the basis of the parents. A direct swap would introduce duplicates and remove necessary candidates from the list, given that the individual is an ordered list. So when the GA steps are followed for all generation in each experiment, the same genes of the parents is stabilized individually as named static crossover.

1.1 Background

The Traveling Salesman Problem (TSP) is generally calculated in the field of Computer Science. The TSP was first defined around 150 years ago but emerged as a complex problem in the late 1940's. Even after half a decade of research, the TSP has not been completely solved and appeared as the interests of computer scientists and mathematicians. The TSP can be applied to crack many handy problems considered day after day. Thus, a way out to the TSP would be extremely valuable. When an explanation to the TSP is conversed, it is about so many different and vibrant solutions one of which is discussed here with new approach.

Typically Travelling Salesman problem is declared as a dilemma when a salesman need a trip of some given cities one time, initiating from any city and revisiting to the original place of departure. So the question arise that what itinerary should he opt in order to decrease the total distance traveled.

The difficulty was to get the shortest possible route once when the salesmen need to stopover their clients and end where they begin. This same problem now applies to a huge number of activities and forms the basis of solution to these modern day issues. For instance distributing new stock to super marts, providing manufacturing lines, sky travel control, and even chromosome progression.

The hypothetical impact on computer science and operational investigation also require TSP solution. Complex and sophisticated computer programmers using optimization where algorithms generate the best possible consequence from several alternatives.

The time required to find an best possible solution is essential for realistic application of the TSP. Matters like time required for lorry drivers to wait for their route to be finalized to deliver the salads as the salads will only be fresh for another 24 hours, Time required for air traffic to keep controlling an airliner flying in circles around Airport is studied in the theory of computational complication. For this area of study, the TSP has vital impact. It is of more interest and benefit to the scientific community even a small incremental pace in considering the nature of this problem.

Genetic Algorithms solve TSPs and provide quick solutions as they operate strings of information. Biological and Computer fields are used to describe GAs as they were stimulated by the conduct of natural systems. In past, using GAs, optimal solutions to TSPs were first formed by Goldberg using Partial Mapped crossovers and also using Greedy Crossover by Grefenstette. Using various crossovers operation, Davis, Smith, Suh and Van Gucht solved TSPs.

Genetic Algorithms imitate the technicalities of ordinary selection by a method of randomized data substitution. They are able to seek out in a randomized, directed way to allow them to imitate some of the novel potential of natural systems.

1.2 Aim

The major concern for this paper is to introduce new technology for TSP that is based on some programming. The vision is to put up a program in shell program using two point crossover to solve TSP in new format. Experiments are performed in which static and dynamic crossover both are used in order to find optimal solution. Execution time is computed and result is shown in the last section of this paper to prove that the time for dynamic crossover is less than that of static so noticeably dynamic crossover verifies better result than the other.

This paper proposes a solution to TSP using GA, dynamic crossover that uses the two point (1 order) crossover that changes the stabilized genes in each generation for all experiments. Later compute execution time for static and dynamic crossover in order to compare the improvements in finding the best solution.

2. RELATED WORK

Previously, many researches had proposed and many algorithms and enhancement approaches for solving the TSP [2]. the crossover technique might be different form problem to another, depended on the problem itself.

To describe different types of crossover operators, let's start with the sequence described by Davis and Goldberg for the Order Crossover operator [1]. The offspring become heir to the elements linking the two crossover points that selected from parents in the same order and position. In the order in which the left over elements emerged are inherited from the alternate parent. The crossover points are determined then swapping the other elements in the parent [7]. Delete the cities that are already exit because there is no need of redundancy cities. The new string called the offspring that used in next population. Also this technique is developed by Sysweda.

Also there is Partially Mapped Crossover (PMX) that is described by Goldbreg and Lingle [9]. A parent and two crossover positions are selected randomly, cut two substrings of equal size on each parent at the same position and exchange between them to produce part of offspring, and then determine the mapping points based on selected substrings. Finally check if there redundancy cities substitute from other parent. Note that when substitution occurs, it results in a mutation where adjacency, position, or relative order is preserved by substitution [5]. Also note that PMX is dependent on crossover position.

Finally, the last type called Cycle Crossover (CX) that proposed by Oliver, it is built offspring from one parent that is distinguished from other types [4]. An element point is randomly selected and started as a cycle from a selected parent subset. The element in which the same position in other parent cannot be swapped in this position because it is produced the same subset. Follow this technique until the cycle is completed and if there is any element not presented in the offspring, selected it from unselected parent. So the offspring is always permutation [4].

3. NEW STRATEGY

The various forms of crossover and mutation can be combined to provide an assortment of genetic algorithms that can be used to crack the travelling salesman problem [6].

Greedy Algorithms are one of the methods for finding practical solution to the travelling salesman problem. The algorithm generates a list of all edges in the graph and then arrange them from smallest cost to largest cost after which it chooses the edges with smallest cost first. The greedy algorithm gives feasible services.

3.1 Method

Many methods based on crossover have been implemented for TSP [8]. The Travelling Salesman Problem is a problem in combinatorial optimization solution in short amount of time is required. With the particular set of cities and their pair-wise distances, scheme is to find a undeviating tour that each city is visited just one time with the length of the tour to be minimized.

Much less time is used to find a solution by means of a genetic algorithm. It can find a near perfect solution for a city tour in less than specific time even though it might not find the best solution. There are a few basic steps to solve the travelling salesman problem using GA [3].

Step 1: Form a crowd of many random tours that is known as population. Here use a greedy initial population that gives preference to connecting cities that are close to each other.

Step 2: Pick two better or shorter tours parent in the population and combine them to make two new child tours. Hopefully, these children tour will be better than any of the parent.

Step 3: In short amount of time, the child tours are mutated. This is done to prevent all tours in the population from looking identical.

Step 4: Keeping the size of the population same, insert new child tours into the population replacing two of the longer tours.

Step 5: Repeatedly generate new child tours until the desired goal is achieved.

Consider six cities (A, I, M, S, Q, K) and assign a number to each one. Thus a path would be represented as a sequence of integers from 1 to 6. Let's say for instance path as [1 2 3 4 5 6] which is known as permutation encoding. Then determine the fitness function which will be the total cost of the tour represented by each individual or total distance can be measured and the minimum one shell is the best solution according to cost or distance matrix as shown below:

Table 1. The Distance or Cost Matrix

	A 1	I 2	M 3	S 4	Q 5	K 6
A[1]	0	90	100	35	300	200
I[2]	90	0	60	120	400	290
M[3]	100	60	0	70	480	225
S[4]	35	120	70	0	320	150
Q[5]	300	400	480	320	0	290
K[6]	200	290	225	150	290	0

So for path [4 1 3 2 5 6], the total cost of travel or fitness will be calculated as shown below:

$$\text{Fitness} = 35+100+60+400+290+150 = 1035$$

After computing all possible combination choose the two minimum distance individual paths to make crossover and mutation. Use the two point crossovers to prevent duplicating any city in the same path as shown in figure 1.

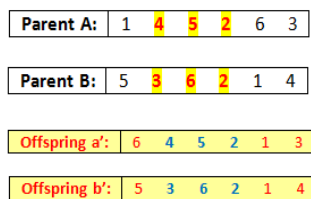


Figure 1. The Two Point Crossover (1 Order)

Final operator knows as mutation operator which means change on bit in the path to prevent local minimal problem as shown in figure 2.

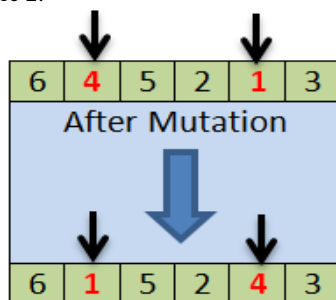


Figure 2. The Mutation Operator

All these operators applied for all generation except mutation and stop if reach the optimal solution which is needed when the number of generation is finished.

3.2 Algorithm

The proposed algorithm depends on changing the crossover points from one generation to another, that's called a dynamic crossover. Now the methods according to this algorithm is explained. There are number of cities that the tour wants to visit them that are called parents [5] .

Generation n (Parent 1, Parent 2)

Parent 1 (A B C D E F G), Parent 2 (C E A D G F B);

1. Do crossover point (i, j);
2. Offspring a' (- - i - j - -);
3. Offspring b' (- - i - j - -);
4. Compute fitness;
5. If (fitness is good or generation number is finished) end;
6. Else Change crossover point (j, k); Repeat 1;

This algorithm is working until reach to the least fitness that leads to choose the related offspring, or when the number of generation is finished which assigned before started but it is optional condition.

Now the above algorithm is applied through an example:

Generation number 2, parent A (1 2 4 6 3 5 7), parent B (2 5 7 1 3 6 4), crossover point (2, 4), new offspring (1 2 4 6 3 5 7) and (6 5 7 1 3 2 4). After change crossover point (4, 6) and apply for produced offspring that treated as parents for next generation, new offspring (1 2 4 6 3 5 7) and (6 5 7 1 3 2 4). Generation number finished, stop and calculate the fitness, then take the best one route.

4. ANALYSIS AND EXPERIMENT RESULT

The shell program is build which solved the TSP using two point crossover for many experiments. The shell is often considered a simple particular application for programming language. File manipulation, program execution, and printing text are the main operations performed by shell programming scripts.

For all generations in each experiment used the same bit to exchange between offspring which means static crossover. In each experiment number of generation change and computes execution time for each one when reach optimal solution. Let's say for instance, if there are seven cities, the two possible individuals are (1,2,4,6,3,5,7) and (2,5,7,1,3,6,4), make two point crossover for all generation if the number of city is 7 at P2 (position two) to P4 (position four) from left, so the new offspring (1,2,4,6,3,5,7) and (6,5,7,1,3,2,4).

Secondly, TSP using two point crossovers is applied for the same experiment performed above but in each generation for each experiment change the bit position which will be exchanged between each offspring that is known as dynamic crossover.

Let's say for instance, if there are seven cities, the two possible individuals are (1,2,4,6,3,5,7) and (2,5,7,1,3,6,4), two point crossover is made for the first generation if the number of city is 7 at P2 to P4 from left, so the new offspring

(1,2,4,6,3,5,7) and (6,5,7,1,3,2,4), for second generation make two point crossover from P4 to P6 so new offspring (2,1,7,6,3,5,4) and (5,6,4,1,3,2,7), for third generation make two point crossover from P6 to p1 and so on.

After running the algorithm through shell program for static and dynamic crossover, the optimum result is searched. In each case compute execution time to reach optimal solution, summarize the results for all experiments in the table below in order to compare between them in obvious way as show in table 2.

Table 2. Experimental Result

Experiment	Dynamic Crossover (Proposed)		Static Crossover	
	Generation Number	Execution Time (millisecond)	Generation Number	Execution Time (millisecond)
5	8	3775	8	3816
7	20	7722	20	8012
8	40	17956	40	18360
10	25	12388	25	12928
11	35	18606	35	20701
15	12	7992	12	8121
16	36	26169	36	27729

The graph below shows the same result comparison between execution time of each experiment for static and dynamic crossover.

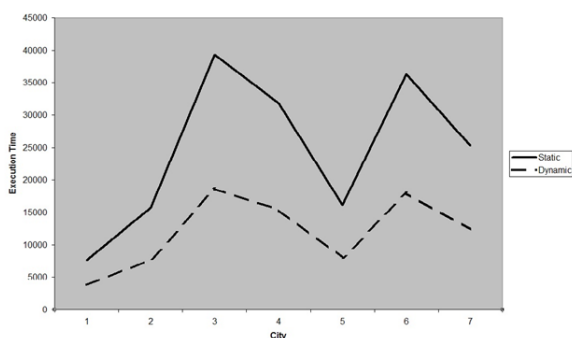


Figure 3. Graphical Result

5. CONCLUSION

Algorithms from shell program are applied on more than one experiment and compare the results. It is noticed that dynamic crossover helps us to reach the optimal solution in less time than that of static which leads to concluding that dynamic crossover is better than the static crossover.

In future each operator in GA for static and dynamic state can be check. The general rule for dynamic GA is better than static for any experiment can be analyzed as a world range.

6. REFERENCE

- [1] Pable Moscato, "On Genetic Crossover Operators for relative Order Preservation", Caltech concurrent Computation Program 158-79, California Institute of Technology, Pasadena, CA 91125, U.S.A.
- [2] Kylie Bryant, "Genetic Algorithms and the Traveling Salesman Problem", Department of Mathematics, Harvey Mudd College, December 2000.
- [3] Mouhammd Al kasassbeh, Ahmad Alabadleh, Tahsen Al-Ramadeen, "Shared Crossover Method for Solving Traveling Salesman Problem", International Journal of Information and Computer Science, IJICS Volume 1, Issue 6, PP. 153-158, September 2012.
- [4] Dr. Mrs. G. Nirmala, Mr. D. Ramprasad, "Innovative PMX – CX – Operators for GA for GA TO TSP", International Journal of Scientific and Research Publications, Volume 2, Issue 10, ISSN 2250-3153, October 2012.
- [5] Er. Ashish Gupta, Er. Shipra Khurana, "Study of Traveling Salesman Problem Using Genetic Algorithm", IJMIE, Volume 2, Issue 5, ISSN: 2249-0558, May 2012.
- [6] ABDOUN Otman, ABOUCHABAKA Jaafar, "A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem", International Journal of Computer Applications (0975 – 8887) Volume 31– No.11, October 2011.
- [7] Kusum Deep, Hadush Mebrahtu, "New Variations of Order Crossover for Travelling Salesman Problem", International Journal of Combinatorial Optimization Problems and Informatics, Vol. 2, No. 1, pp. 2-13. ISSN: 2007-1558, Jan-April 2011.
- [8] Namit Gupta and Rajeev Kumar, "Using new variation Crossover operator of Genetic Algorithm for Solving the Traveling Salesmen Problem in e-Governance", New York Science Journal 2012;5(12):164-166, (ISSN: 1554-0200).
- [9] Goldberg, D. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley