A Study of crossover operators for Genetic Algorithms to solve TSP

[1]Poonam, Student Mtech (Cse), [2]Proff. Mrs. Shakti Arora
[1,2]*Geeta Engg. College Naultha (Haryana, India)*

Abstract: *The travelling salesman problem (TSP) is the most well-known combinatorial optimization problem. TSP is used to find a routing of a salesman who starts from a home location, visits a prescribed set of cities and returns to the original location in such a way that the total distance travelled is minimized and each city 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 . TSP is solved very easily when there is less number of cities, but as the number of cities increases it is very hard to solve, as large amount of computation time is required. The numbers of fields where TSP can be used very effectively are military and traffic. Another approach is to use genetic algorithm to solve TSP because of its robustness and flexibility . Some typical applications of TSP include vehicle routing, computer wiring, cutting wallpaper and job sequencing*
 In genetic algorithms, crossovers are used as a main search operator for TSP. There were a lot attempts to discover an appropriate crossover operator. This paper presents the strategy which used to find the nearly optimized solution to these type of problems. It is the order crossover operator (OX) which was proposed by Davis, which constructs an offspring by choosing a subsequence of one parent and preserving the relative order of cities of the other parent.

I.   Introduction

Genetic algorithms are one of the best ways to solve a TSP Problem for which little is known.

They are a very general algorithm and so will work well in any search space. All you need to know is what you need the solution to be able to do well, and a genetic algorithm will be able to create a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.

Problem specification

algorithms have been developed in t h e field of operationsresearch (OR) to solve this problem. . In the sections The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem, which is simple to state but very difficult t o solve. The problem is to find the shortest possible tour through a s e t 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 that follow, we briefly introduce the OR problem-solving approaches to the TSP. Then, t h genetic algorithms are discussed.

1.2 Exact algorithms

The exact algorithms are designed to find the optimal solution to the TSP, that is, the tour of minimal length. They are computationally expensive because they must (implicitly) consider all solutions in order to identify the optimum. These exact algorithms are typically derived from the integer linear programming (ILP) formulation of the TSP, Where N is the number of vertices, d_{ij} is the distance between vertices i and j and the A_{ij}'s are the decision variables: A_{ij} is set to 1 when arc (i,j) is included in the tour, and 0 otherwise. (A_{ij}) A denotes the set of sub tour-breaking constraints

Min ij $d_{ij}A_{ij}$
Subject to:
j A_{ij} = 1 , i=1,..,N
i A_{ij} = 1 , j=1,..,N
(A_{ij}) A
A_{ij} = 0 or 1 ,

That restricts the feasible solutions to those consisting of a single tour. Although the sub tour-breaking constraints can be formulated in many different ways Without the sub tour breaking constraints, the TSP reduces to an assignment problem (AP), and a solution like the one shown in would then b e feasible. Branch and bound algorithms are commonly used to find a n optimal solution to the TSP, and the above AP-relaxation is useful to

generate good lower bounds on the optimal value. This is true in particular for asymmetric problems, where dij ☐ dji for some i,j.

II. Methodology

Ordered two-point crossover is used when the problem is of order based, for example in U-shaped assembly line balancing etc. Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and right portion.The point crossover behaves in the following way: child 1 inherits its left and right section from parent 1, and its middle section is determined

The Path matrix considered in this is drawn under topic Figures. This crossover operator extends the modified crossover of Davis by allowing two cut points to be randomly chosen on the parent chromosomes. In order to create an offspring, the string between t h e two cut points in the first parent is first copied to the offspring. Then, the remaining positions are filled by considering the sequence of cities in the second parent, starting after the second cut point (when the end of the chromosome is reached, the sequence continues a t position 1).

parent 1 : **1 2 | 5 6 4 | 3 8 7**
parent 2 : 1 4 | 2 3 6 | 5 7 8
_____
offspring
(step 1) : - - **5 6 4** - - -
(step 2) : 2 3 **5 6 4** 7 8 1
Figure The order crossover.

Clearly, OX tries to preserve the relative order of the cities in parent 2, rather than their absolute position. In **Figure of PMX** , the offspring does not preserve the position of any city in parent 2. However, city 7 still appears before city 8, and city 2 before city 3 in the resulting offspring.

2.2 Cycle crossover (CX) Oliver et al. (87)

The cycle crossover focuses on subsets of cities that occupy the same subset of positions in both parents. Then, these cities are copied from the first parent to the offspring (at the same positions), and t h e remaining positions are filled with the cities of the second parent. In this way, the position of each city is inherited from one of the two parents. However, many edges can be broken in the process, because the initial subset of cities is not necessarily located at consecutive positions in the parent tours.

In Figure , the subset of cities {3,4,6} occupies the subset of positions {2,4,5} in both parents. Hence, an offspring is created b y filling the positions 2, 4 and 5 with the cities found in parent 1, and by filling the remaining positions with the cities found in parent 2.
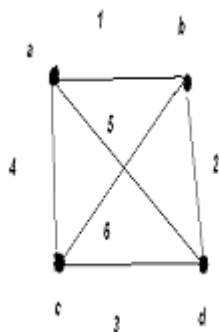
parent 1 : 1 3 5 6 4 2 8 7
parent 2 : 1 4 2 3 6 5 7 8
_____
offspring : 1 3 2 6 4 5 7 8
Figure The cycle crossover

III. Results

3.1 Tables, Figures and Equations

Figure 1- Path matrix

Sample No.	Ox,no. of itration	Shortest path	Cyclic crossover,no. of itration	Shortest path
1	1	86	1	80
2	1	348	3	330
3	1	1727	1	1500
4	1	605	2	590
5	2	2432	2	2388

From the above given table conclusion is clearly drawn out that cyclic crossover is best from the techniques that compared here. In terms of shortest path cyclic crossover(CX) provides improved results.

3.2 Equations

$F(x) = g(F(x))$ (1)

Where f objective function , g transforms the value of the objective function to a non-negative number and F- resulting relative fitness. The most fit individuals and the fitness of the others is determined by the following rules:

• MIN = 2.0 - MAX

• INC = 2.0 ×(MAX -1.0) / n

• LOW = INC / 2.0 (2)

the fitness of individuals in the population may be calculated directly as,

$f(xi) = 2- MAX + 2 (MAX -1) xi - 1/n-1$

(3)

Probability of each chromosomes selection is given by: N

$Ps(i) = f(i) / sigma f(j)$

J =1 (3)

Ps(i) and f(i) are the probability of selection and fitness

IV. Discussion

The description of the genetic algorithm included many genetic terms. In order to better understand how genetic algorithms can be applied to combinatorial optimization problems, the following equivalence will be useful. Combinatorial Optimization Genetic Algorithm, Encoded Solution , Chromosome Solution, Decoded Chromosome Set of Solutions, Population Objective function. Fitness function-

In a TSP context, each chromosome encodes a solution to the problem (i.e. a tour). The fitness of the chromosome is related to the tour length, which in turn depends on the ordering of the cities. Since the TSP is a minimization problem, the tour lengths must be transformed, so that high fitness values are associated with short tours, and conversely. A well-known approach is to subtract each tour length to the maximum tour length found in the current population. Other approaches are based on the rank of the tours in the population The genetic algorithm searches the space of solutions by combining the best features of two good tours into a single one. Since the fitness is related to the length of the edges included in the tour, it is clear that the edges represent the basic information to be transferred to the offspring. The success or failure of the approaches described in the following sections, can often be explained by their ability or inability to adequately represent and combine the edge information in the offspring. Difficulties quickly arise when the simple "pure" genetic algorithm is applied to a combinatorial optimization problem like the TSP. In particular, the encoding of a solution as a bit string is not convenient. Assuming a TSP of size N, each city would b e coded using 2 log N bits, and the whole

chromosome would encode our as a sequence of N* 2 log N☐ bits. Accordingly, most sequences in the search space would not correspond to feasible tours. For example, it would be easy to create a sequence with two occurrences of the same city, using the mutation operator. Moreover, when the number of cities is not a power of two, some bit sequences in the code would not correspond to any city. In the literature, fitness functions with penalty terms, and repair operators to transform infeasible solutions into feasible ones have been proposed to alleviate these problems However, these approaches were designed for very specific application domains, and are not always relevant in a TSP context. The preferred research avenue for the TSP is to design representational frameworks that are more sophisticated than the bit string, and to develop specialized operators to manipulate these representations and create feasible sequences. For example, applying the crossover operator at position 2 creates two infeasible offspring, as illustrated in Figure

tour (12564387) : 1 2 | 5 6 4 3 8 7
tour (14236578) : 1 4 | 2 3 6 5 7 8 _____
offspring 1 : 1 2 2 3 6 5 7 8
offspring 2 : 1 4 5 6 4 3 8 7

Figure Application of the one-point crossover on two parent tours.

None of the two offspring is a permutation of the cities. The TSP, as opposed to most problems tackled by genetic algorithms, is a pure ordering problem. Namely, all chromosomes carry exactly must be developed for this problem. In the following sections, we explain how genetic algorithms can be tailored to the TSP. The extensions proposed in theliterature will be classified according to the representational framework used t o encode a TSP tour into a chromosome, and the crossover operators used to manipulate these representations

V.   Conclusion

elite rate set beforehand is selected as the next parents. TSP is optimization problem which is used to find minimum path for salesperson. The Actual use of tsp is routing in network. Minimum path will helps to reduce the overall receiving time and improves system performance. The work proposed here intends to test the performance of different Crossover used in GA and compare the performance for each of them and compare to others. This thesis presents an investigation on different crossover techniques Gives the results of experiments comparing the proposed method with the conventional approach. In these experiments, the number of children generated by one crossover is limited because of calculation costs. Therefore, they tried generate better individuals within a limited number of children and again purposed a new crossover method that accelerates the local search efficiency. First, the children generated by the first parents are evaluated for their fitness. Then, some number of top children with an used in GA .

Since there are other methods traditionally adopted to obtain the optimum distance for TSP. This work aims at establishing the superiority of Genetic Algorithms in optimizing TSP. One of the objectives of this research work is to find a way to converge fast. Since precise minimum path remains a great challenge, the objective of this work is to develop some new and practical model with computational intelligence algorithms. As can be seen from the references, many models have been developed for TSP. From the experimental results the conclusion can be drawn thatdifferent methods might outperform the others in different situations.

Refrences

[1].   Potvin ,Jean-Yves(n.d) Genetic Algorithms for the Traveling Salesman Problem: Montréal (Québec)Canada H3C 3J7, Centre de Recherche sur les Transports Université de Montréal

[2].   Qi-yi, Zhang & Shu-chun, Chang (2009) An Improved Crossover Operator of Genetic Algorithm China: Transportation Command Department Automobile Management Institute of PLA

[3].   Rasheed, Khaled(1999)Guided Crossover:A New Operator For Genetic Algorithm Based Optimization,NewBrunswick,NJ08903,USA: Computer Science Department, Rutgers University

[4].   Shang, Yi &Li, Guo-Jie(1991) New Crossover Operators In Genetic Algorithms, P. R. China: National Research Center for Intelligent Computing Systems (NCIC).

[5].   Singh, Vijendra & Choudhary, Simran (2009) Genetic Algorithm for Traveling Salesman Problem: Using Modified Partially-Mapped Crossover Operator, sikar, Rajasthan, India: Department of Computer Science & Engineering, Faculty of Engineering & Technology, Mody Institute of Technology & Science, Lakshmangarh

[6].   Su, Fanchen et al(2009) New Crossover Operator of Genetic Algorithms for the TSP, P.R. China: Computer School of Wuhan University Wuhan.

[7].   Dwivedi,TarunaChauhan,SanuSaxena and PrincieAgrawal, "Travelling Salesman Problem using Genetic Algorithm", International Journal of Computer Applications(IJCA), 2012

[8].   Greedy randomized adaptive search procedure (GRASP) - Aiex et al. and Binato et al.

[9].   Evolutionary algorithm (EA) techniques like genetic algorithms (GA) - Aarts et al. [15], Croce et al. [17], Dorndorf et al. [18], Wang and Zheng [19], Essafi et al. [3], Gonçalves et al. [4], Hasan et al. [27], Mendes [30], Qing-dao-er-ji and Wang [38], Choi and Park [39], Chiu et al. [40, 41]);