

Genetic Operators in TSP

Mr. Akash Kadiyan¹, Mr. Sandeep Jaglan²

¹Department of Computer Science, N.C. College of Engineering, Israna
Panipat, Haryana, INDIA.

¹Department of Computer Science, N.C. College of Engineering, Israna
Panipat, Haryana, INDIA.

ABSTRACT

In this work Traveling salesperson problem is taken as Domain. TSP has long been known to be NP-complete and is a standard example of such problems. Genetic Algorithm (GA) is an approximate algorithm that doesn't always aim to find the shortest tour but to find a reasonably short tour quickly, which is a search procedure inspired by the mechanisms of biological evolution. In genetic algorithms, crossovers are used as a main search operator for TSP. Briefly speaking: the role of crossovers is to generate offspring that are better tours by preserving partial tours from the parents. 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.

Keywords: Genetic algorithm, Order Crossover, Travelling salesman problem.

1. INTRODUCTION

Genetic algorithms are one of the best ways to solve a 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.

1.1 Problem specification

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem, which is simple to state but very difficult to solve. The problem is to find the shortest possible tour through a set 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 algorithms have been developed in the field of operations

research (OR) to solve this problem. In the sections that follow, we briefly introduce the OR problem-solving approaches to the TSP. Then, the 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 x_{ij} 's are the decision variables: x_{ij} is set to 1 when arc (i,j) is included in the tour, and 0 otherwise. X denotes the set of sub tour-breaking constraints

$$\text{Min } \sum_{i,j} d_{ij} x_{ij}$$

Subject to:

$$\sum_j x_{ij} = 1, i=1, \dots, N$$

$$\sum_i x_{ij} = 1, j=1, \dots, N$$

$$(x_{ij}) \in X$$

$$x_{ij} = 0 \text{ or } 1,$$

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 be 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 $d_{ij} \neq d_{ji}$ for some i,j .

2. METHODOLOGY

2.1 Order Crossover (OX) Davis (85), Oliver et al. (87)

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 ordered two-

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 the 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 at a 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 the 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 by 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.

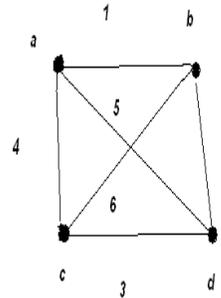
parent 1 : 1 3 5 6 4 2 8 7
parent 2 : 1 4 2 3 6 5 7 8

offspring : 1 3 2 6 4 5 7 8

Figure The cycle crossover.

3. RESULTS

3.1 Tables, Figures and Equations



	a	b	c	d
a	0	1	4	5
b	1	0	6	2
c	4	6	0	3
d	5	2	3	0

Figure 1- Path matrix

Table 1- Result of OX and CX

Sample no.	ox, No. of iteration	Shortest path	Cyclic crossover, No. of iteration	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

$$F(x) = g(F(x)) \tag{1}$$

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:

- $MIN = 2.0 - MAX$
- $INC = 2.0 \times (MAX - 1.0) / n$
- $LOW = INC / 2.0$ (2)

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

$$f(x_i) = \frac{2 - MAX + 2 (MAX - 1) x_i - 1}{n - 1} \tag{3}$$

Probability of each chromosomes selection is given by:

$$P_s(i) = \frac{f(i)}{\sum_{j=1}^N f(j)} \tag{3}$$

$P_s(i)$ and $f(i)$ are the probability of selection and fitness value

4- DISCUSSION

Genetic algorithms for the TSP-

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

the same values and differ only in the ordering of *these values*. Accordingly, *specialized permutation* operators

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 be 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 the

literature will be classified according to the representational framework used to encode a TSP tour into a chromosome, and the crossover operators used to manipulate these representations

5. CONCLUSION

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 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 help 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 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 that

different methods might outperform the others in different situations.

6. REFERENCES

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