Volume 2, No. 4, July-August 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Study of Crossover operators in Genetic Algorithm for Travelling Salesman Problem

Isha Gupta* and Anshu Parashar
Department of Computer Science & Engineering
Haryana College of Technology & Management, Kaithal,
Haryana, India
ishagupta_11@yahoo.co.in
anshulphd@gmail.com

Abstract: 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. The role of crossovers is to generate offspring that are better tours by preserving partial tours from the parents.

In this paper Travelling Salesman Problem is taken as domain. TSP has been known to be NP-complete and is a standard example of such problems. In this paper various Crossover Operators are studied like Partially-Mapped Crossover operator (PMX), Cyclic Crossover Operator (CX), Order Crossover Operator (OX), Linear Crossover Operator (LOX) and Sequential Constructive crossover operator (SCX). Brief discussion about these operators is represented.

Keywords- Cyclic crossover, Partially-mapped crossover, order crossover, Linear Crossover, Sequential crossover, Genetic algorithm, Travelling salesman problem.

I. INTRODUCTION

Genetic Algorithms (GAs) are adaptive methods which may be used to solve search and optimisation problems. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and "survival of the fittest", first clearly stated by Charles Darwin in The Origin of Species [1]. By mimicking this process, genetic algorithms are able to "evolve" solutions to real world problems, if they have been suitably encoded. Problems of these types are Travelling Salesman problem, job shop scheduling, space allocation and map coloring, shortest path problem etc. A genetic algorithm is a computer algorithm that searches for a good solution to a problem among a large number of possible solutions.

Genetic algorithms [Goldberg 1989] are inspired by the evolution in nature. A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of Evolutionary Algorithm (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. GAs attempt to find the best solution to some problem (e.g., the maximum of a function) by generating a collection ("population") of potential solutions ("Individuals") to the problem. Through mutation and recombination (crossover) operations, better solutions are hopefully generated out of the current set of potential solutions. This process continues until an acceptably good solution is found. [5]

A. Algorithm:

```
BEGIN                                   /* genetic algorithm */
        Generate initial population

        Compute fitness of each individual
WHILE NOT finished DO
  BEGIN                        /* produce new generation */
  FOR  population_size / 2 DO
BEGIN                                /* reproductive cycle */
    Select two individuals from old generation for mating
Recombine the two individuals to give two offspring
Compute fitness of the two offspring Insert offspring in new generation
END
IF population has converged THEN
Finished := TRUE
END
END
```

A Traditional Genetic Algorithm

A whole new population of possible solutions is thus produced by selecting the best individuals from the current "generation", and mating them to produce a new set of individuals. This new generation contains a higher proportion of the characteristics possessed by the good members of the previous generation [6]. In this way, over many generations, good characteristics are spread throughout the population, being mixed and exchanged with other good characteristics as they go. By favouring the mating of the more fit individuals, the most promising areas of the search space are explored. If the GA has been designed well, the population will converge to an optimal solution to the problem.

This paper is organized as follows: In Section 2, we discuss about premature convergence in GA. In section 3, the concepts of genetic algorithm for TSP followed by Chromosome representation scheme, Fitness function, selection method. And In Section 3.4 crossover operators (like PMX, CX, OX, LOX and SCX) are discussed. In Section 4 we discuss the paper and in Section 5 conclusion is there.

II. PREVENTING PREMATURE CONVERGENCE

When applying GA to solve large-scale and complex real-world problems, premature convergence [10][11] is one of the most frequently encountered difficulties. In that situation, the solving procedure is trapped in the suboptimal state and most of the operators cannot produce offspring surpassing their parents any more. If the GA has been correctly implemented, the population will evolve over successive generations so that the fitness of the best and the average individual in each generation increases towards the global optimum. Convergence is the progression towards increasing uniformity. A gene is said to have converged, when 95% of the population share the same value. The population is said to have converged when all of the genes have converged. The problem is tightly related with the loss of genetic diversity of the GA's population. It has been widely accepted that the decrease of population diversity is the direct cause of premature convergence. This fact has lead to the development of different techniques aiming to solve, or at least to minimize the problem; traditional methods usually work to maintain a certain degree of genetic diversity on the target populations, without affecting the convergence process of the GA.

Several methods have been proposed to overcome premature convergence and improve the performances of GA these include the restriction of selection procedure, the improvement of the mating procedure, the increase of mutation probability and the modification of fitness function[15] .

III. GENETIC ALGORITHM FOR TSP

The TSP is a classic, well known NP-hard problem in Combinatorial Optimization; given a set of n cities, and the costs associated with the travel between each pair, the objective is to find a roundtrip of minimal total cost (or length), visiting each city exactly once. In the travelling salesman problem, a salesman seeks the shortest tour through given cities, with each city visited exactly once before returning back to his home city. Consider G be a complete graph with n vertices. Take length (<u, v>) be the length of the edge <u, v>. A path starting at a given vertex v0, going through every other vertex exactly once and finally returning to v0 will be called a tour. The length of a tour is the sum of lengths of the edges on the path defining the tour. The problem is to find a tour of minimum length. Such problem is called the Travelling Salesman Problem [3].

For TSP solving, it is clear that the simple GA is not suitable because bit-string encoding is not suitable and simple genetic operators are not the most effective or appropriate.

For TSP solving, the genetic algorithm used will have the following chromosome representation, fitness function, selection technique and genetic operators:

A. *Chromosome Representation Scheme*

TSP tours can have different representations, such as ordinal, path, adjacency and binary matrix representation [14]. From these, the path representation is the most natural representation of a tour and we will focus on it. Path representation scheme is employed as a suitable one for representing a schedule of the original problem[1]. A Chromosome represents a tour of the salesman. A chromosome T_k ($k = 1, 2\ldots m$ is the population size) is represented as:

$T_k = (C_1\ C_2\ C_3 \ldots\ldots\ldots C_n)$

Where C_i is the i^{th} city to be visited, $i = 1, 2, \ldots\ldots\ldots, n$.

Such type of representation is called permutation representation (encoding).

All the cities are sequentially numbered starting from 1. The route between the cities is described with an array. Each element of the array represents the number of the city. The array represents the sequence in which the cities are traversed to make up a tour. Each chromosome must contain each and every city exactly once.

Example: Considering a TSP with 7 nodes, a tour:

$1 \rightarrow 3 \rightarrow 7 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 6$

can be represented using the path representing as

(1 3 7 5 4 2 6)

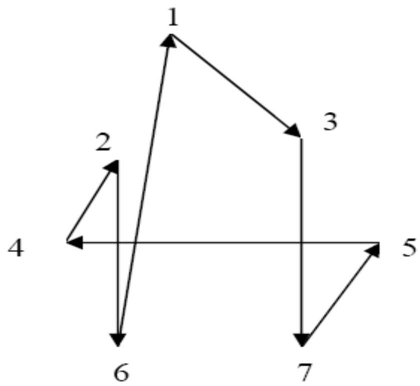

Figure 1: Shows the tour 1→3→7→5→4→2→6

For array representation of such tour, we declare an array
t[7].

1	3	7	5	4	2	6
t[0]	t[1]	t[2]	t[3]	t[4]	t[5]	t[6]

This chromosomes represents the tour stating from city 1 to city 3, city 3 to city 7, city 7 to city 5, city 5 to city 4, city 4 to city 2, city 2 to city 6 and city 6 to city 1. Chromosome describes the order of cities, in which the salesman will visit them.

B. *Fitness Function*

The objective function, the function to be optimized, provides the mechanism for evaluating the fitness of each chromosome. The fitness function [2]fit (T_k) ($k = 1, 2, \ldots, m$) is defined as:

$$fit(T_k) = \frac{1}{\sum_{i=1}^{n} d(C_i, C_{i+1}) + d(C_n, C_1)}$$

where $d(C_i, C_{i+1})$ is travelling distance from city C_i to C_{i+1}

We use computer screen as platform to describe TSP. Pixels are used to represent cities. If city C_i is represented by pixel $p(x, y)$ and city C_{i+1} is represented by pixel $q(s, t)$, then $d(C_i, C_{i+1})$ is Euclidean distance between two pixels $p(x, y)$ and $q(s, t)$ that is defined as :

$$d(C_i, C_{i+1}) = |(x-s)| + |(y-t)|$$

$$= \sqrt{(x-s)^2 + (y-t)^2}$$

Algorithm for fitness measure

Step 1: Traverse the cities according to the sequence in a tour

Step 2: Calculate $d(C_i, C_{i+1})$ using equation:

$$\sqrt{(x-s)^2 + (y-t)^2}$$

and find the total distance in the tour:

$$Total \quad dis\tan ce = \sum_{i=1}^{n} d(C_i, C_{i+1}) + (C_n, C_1)$$

Step 3: Calculate the fitness of the chromosome in the Population:

$fit\,(T_k) = 1/$Total distance

C. Selection Method

After evaluating fitness, GAs will *select* individuals for reproduction.[4]. Selection allocates more copies of those solutions with higher fitness values and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. The main idea of selection is to prefer better solutions to worse ones. To date, there have been many selection schemes developed such as roulette wheel selection, steady-state selection, stochastic universal sampling selection, tournament selection, stochastic remainder selection, and rank-based selection. The individuals selected are called *parents* and are used to produce an offspring, called a *child*, by the crossover operators.

D. Crossover Operator

Recombination combines parts of two or more parental solutions to create new, possibly better solutions The individuals selected are called *parents* and are used to produce an offspring, called a *child*, by the crossover operator. There are many types of crossover operators dependent on the problem type[6]. In terms of path representation there are two basic groups of crossover operators proposed by researchers that preserves either the relative order or the absolute position of cities/chromosomes/ from the parent chromosomes. Cycle and partially matched crossovers are operators that preserve absolute position. While order based crossovers are some examples of crossovers that preserves relative order of cities/chromosomes/. A large number of crossover have been developed for the permutation encoding such as partially mapped crossover (PMX), order crossover (OX), cycle crossover (CX), edge recombination (ERX), edge assembly crossover (EAX) etc. We will study four crossover operators i.e. PMX, CX, OX, LOX, SCX in following section:

a. Partially-Mapped Crossover Operator (PMX)

The partially-mapped crossover operator [7][12] was suggested by Goldberg and Lingle (1985). It passes on ordering and value information from the parent tours to offspring tours. A portion of one parent's string is mapped onto a portion of other parent's string and the remaining information is exchanged.

Example:

First, PMX selects uniformly at random two cut points along the tour. The symbol | shows the crossover points.

Tour 1: (1 2 | 3 4 5 | 6 7 8)
Tour 2: (6 7 | 4 2 8 | 5 3 1)

Second, PMX exchanges sub-strings between parents.

Offspring 1: (* * | 4 2 8 | * * *)
Offspring 2: (* * | 3 4 5 | * * *)

Third, PMX determines the mapping relationship as following:

$$3 \leftrightarrow 4 \leftrightarrow 2,\ 5 \leftrightarrow 8.$$

Then, the remaining cities are filled from original parent, if a city already present in the offspring it is replaced according to mapping relationship. The new tours will be:

Offspring 1: (1 3 | 4 2 8 | 6 7 5)
Offspring 2: (6 7 | 3 4 5 | 8 2 1)

b. Cycle Crossover Operator (CX)

The cyclic crossover operator [7] was proposed by Oliver et. el (1987). It attempts to create an offspring from the parents where every position is occupied by a corresponding element from one of the parents. Consider following parent tours.

Tour 1: (1 2 3 4 5 6 7 8)
Tour 2: (6 7 4 2 8 5 3 1)

First of all, find the cycle that is defined by the corresponding positions of cities between parents starting from the first city of one of the parents. how to find a circle is shown below :

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

Finding a Cycle in Cycle Crossover

The circle will be

$$1 \rightarrow 6 \rightarrow 5 \rightarrow 8 \rightarrow 1$$

Second of all, keep the cities in the cycle corresponding positions of one parent and delete other non-cycle cities and merge the tour in order to construct an offspring.

Tour 1: 1 * * * 5 6 * 8
Tour 2: * 7 4 2 * * 3 *
Offspring1: 1 7 4 2 5 6 3 8
Tour 2: 6 * * * 8 5 * 1
Tour 1: * 2 3 4 * * 7 *
Offspring2: 6 2 3 4 8 5 7 1

The CX maintains the absolute position of the elements in the parent sequence.

c. Order Crossover Operator (OX)

The order crossover operator was proposed by Davis (1985) creates new offspring by choosing a sub-tour of one parent and preserving the relative order of cities of the other parent [13]. Again, consider the following parent tours with two cut points marked by symbol|:

Tour 1: (1 2 | 3 4 5 | 6 7 8)
Tour 2: (6 7 | 4 2 8 | 5 3 1)

The offspring are constructed in the following way. First, the tour subsequences between the cut points are inherited into the offspring, which is shown below:

Offspring 1: (* * | 3 4 5 | * * *)
Offspring 2: (* * | 4 2 8 | * * *)
Second, delete the cities, which are already present in the subsequence from the other parent.
Offspring 1 : (* * 3 4 5 * * *)
Parent Tour2: (6 7 **4** 2 8 **5 3** 1)
Offspring 2: (* * 4 2 8 * * *)
Parent Tour 1: (1 **2** 3 **4** 5 6 7 **8)**

Last, starting from the second cut point of one parent, the remaining cities are copied in the order in which they appear in the other parent. When the end of the string is reached, we continue from the first place of the string. The offspring will be:
Offspring 1: (2 8 3 4 5 1 6 7)
Offspring 2: (3 5 4 2 8 6 7 1)

The OX exploits a property of the path representation where the order of cities is important.

d. Linear Order Crossover Operator (LOX)

Falkenauer and Bouffouix, (1991) proposed the LOX operator, for Linear Order Crossover [16]. It is a modified version of OX proposed to solve job-shop scheduling problems [17][18]. Recall that the order crossover operator treats the chromosome as a circular string, in which it wraps around from the end of the chromosome back to the beginning. The LOX operator treats the chromosome as a linear entity. They are "two-point" cross-over operators. As OX, LOX works on permutation encoding.

For this operator, the swap occurs in the same fashion as it occurs in the OX operator, but when sliding the parent values around to fit in the remaining open slots of the child chromosome, they are allowed to slide to the left or right. This allows the chromosome to maintain its relative ordering.

Example:

First, LOX selects uniformly at random two cut points along the tour. The symbol | shows the crossover points.
Tour 1: (3 9 | 5 4 6 2 | 7 1 8)
Tour 2: (7 4 | 3 8 9 2 | 1 5 6)
Second, values are swapped between the cut points.
Tour 1: (* * | 3 8 9 2 | * * *)
Tour 2: (* * | 5 4 6 2 | * * *)

Third, after the values are swapped, there are two open spaces in the front of the chromosome and three open spaces at the end. The algorithm then goes through Parent A and finds the first two values that were not part of the swap, in this example they are 5 and 4. These values are shifted left to fill the first two chromosome locations. The final three locations are filled in a similar manner. The offspring will be:
Offspring 1: (5 4 | 3 8 9 2 | 6 7 1)
Offspring 2: (7 3 | 5 4 6 2 | 8 9 1)

In this way LOX operator preserves the relative ordering of the parent chromosomes. By filling in the values from the beginning, the operator preserves the linear characteristics of the LOX operator.

e. Sequential Constructive Operator (SCX)

The Sequential Constructive crossover (SCX) operator [8] is proposed by Zakir H. Ahmed. This (SCX) operator constructs an offspring using better edges on the basis of their values present in the parents' structure. It also uses the better edges, which are present neither in the parents' structure. The algorithm for the SCX is as follows:

Algorithm

Step 1: - Start from 'node 1' (i.e., current node p =1).

Step 2: - Sequentially search both of the parent chromosomes and consider the first 'legitimate node' (the node that is not yet visited) appeared after 'node p' in each parent. If no 'legitimate node' after 'node p' is present in any of the parent, search sequentially the nodes {2, 3, …, n} and consider the first 'legitimate' node, and go to Step 3.

Step 3:- Suppose the 'node α' and the 'node β' are found in 1st and 2nd parent respectively, then for selecting the next node go to Step 4.

Step 4:- If $c_{p\alpha} < c_{p\beta}$, then select 'node α', otherwise, 'node β' as the next node and concatenate it to the partially constructed offspring chromosome, where $c_{p\alpha}$ is cost of edge pα and $c_{p\beta}$ is cost of edge pβ. If the offspring is a complete chromosome, then stop, otherwise, rename the present node as 'node p' and go to Step 2.

IV. DISCUSSION

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.

In genetic algorithms, crossovers are used as a main search operators for TSP. The key difference between the operators is the information which each attempts to preserve during recombination. Like PMX and CX operators preserves the absolute positions of the elements, OX and LOX preserves the relative ordering of the elements while SCX operator preserves the better edges. Our study shows role of different crossover operators in GA to generate optimization solution in genetic algorithm for TSP.

V. CONCLUSION

GAs are a very broad and deep subject area, and most of our knowledge about them is empirical. This article has described the fundamental aspects of GAs. In this paper basic genetic algorithm is considered and several operators which may be used in genetic algorithms to solve Travelling Sales Man problem.

In genetic algorithms, crossovers are used as main search operators for TSP. Briefly speaking: the role of crossovers is to generate offspring that are better tours by preserving partial tours from the parents. We have provided a survey of previous research in this area as well as provide a brief comparison of different operators used in GAs to generate optimize solutions. The researchers are doing a lot attempts to discover an appropriate crossover operator to improve the performance of genetic algorithms i.e. to find optimize solutions and to tackle premature convergence problem.

VI. REFERENCE

[1] Kester Clegg (2008) "An Introduction to Evolution for Computer Scientists"

[2] Baljit Singh, Arjan Singh and Akashdeep (2008) "Havrda and Charvat Entropy Based Genetic Algorithm for

TSP" IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.5, May 2008

[3] Sartaj Sahni, "algorithms analysis and design", Galgotia Publications Pvt. Ltd., New Delhi, 1996.

[4] Cerry M. Klein and Tony Weng (n.d.) "Operator Determination in Genetic Algorithms" Department of Industrial Engineering University of Missouri-Columbia Columbia, MO 65211 USA

[5] Beasley, Bull, D.R., and Martin, R.R., "An Overview of Genetic Algorithms: Part 1, Fundamental" University Computing, Vol. 15(2): pp. 58-69, 1993.

[6] P.Larranaga, C.M.H.Kuijpers, R.H. Murga I.Inja S. Dizdarevic(1997) ,"Genetic Algorithms for TSP:A Review of Representations and Operators", Department of computer science and Artificial Intelligence, P.O.BOX-649,University of Basque Country, E -20080 Donostia-San Sebastian, Spain.

[7] T. Starkweather, McDaniel, Mathas, Whitley(1992) "A Comparison Of Genetic sequencing operators" computer science deptt colorado state University Fort collins, CO 80523.

[8] Zakir H. Ahmed(n.d.) "Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator" Department of Computer Science, Al-Imam Muhammad Ibn Saud Islamic University, P.O. Box No. 5701, Riyadh-11432 Kingdom of Saudi Arabia.

[9] Tsai, Huai-Kuang et al.(2004) "An Evolutionary algo for Large Traveling Salesperson Problems" Taiwan: Department of computer science and Information Engineering, National Taiwan University, Taipei: Deptt of Biological Sciences and Technology and institute of BioInformatics, National Chaio tung University, Hsinchu.

[10] Liu, Juan Cai, Zixing Liu, Jianqin (2000) "Premature Convergence in Genetic Algorithm : Analysis and Prevention Based on Chaos Operator"IEEE explore, College of Information Science and Engineering ,Central South University of Technology, China.

[11] Miguel Rocha and Jose Neves (n.d.) "Preventing Premature Convergence to Local Optima in Genetic Algorithms via Random Offspring Generation" Departamento de Informatica Universidade do Minho Largo do Paco PORTUGAL

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

[13] Kusum Deep, Hadush Mebrahtu (2011)"New Variations of Order Crossover for Travelling Salesman Problem" © International Journal of Combinatorial Optimization Problems and Informatics, Vol. 2, No. 1, Jan-April,2011

[14] Amin Mohebifar(2006) "New binary representation in Genetic Algorithms for solving TSP by mapping permutations to a list of ordered numbers" Computer department of faculty of engineering University of Isfahan 81746-73441 Hezar Jerib st., Isfahan IRAN

[15] Victor M. Kureichick, Victor V. Miagkikh, Alexander P. Topchy, "Genetic Algorithm for Solution of the Traveling Salesman Problem with New Features against Premature Convergence" Taganrog State University of Radio-Engineering Nekrasovsky L. 44, GSP-17A, Taganrog, 347928, RUSSIA

[16] Keshav P. Dahal, Kay Chen Tan, Peter I. Cowling (Eds.) "Evolutionary Scheduling" Studies in Computational Intelligence, Volume 49, 2007

[17] Marie-Claude Portmann, Antony Vignier "Performances' study on crossover operators keeping good schemata for some scheduling problems" France.

[18] Serhat DUMAN, Ali ÖZTÜRK, M. Kenan DÖŞOĞLU and Salih TOSUN "Economic dispatch by using different crossover operators of genetic algorithm" Duzce University, Technical Education Faculty, Department of Electrical Education 81620, Duzce,Turkey;-2010.

