Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study of Various Crossover Operators in Genetic Algorithms

Pratibha Thakur
Student M.Tech, Dept. of Computer Science,
Himachal Pradesh University, Shimla, India

Amar Jeet Singh
Professor, Dept. of Computer Science,
Himachal Pradesh University, Shimla, India

Abstract- Genetic algorithms are search and optimization algorithms, inspired by basic principles of natural evolution and natural selection. Genetic algorithms are very prominent examples of "evolutionary algorithms". Genetic algorithms simulate natural evolution by using basic operators like selection, crossover and mutation. This paper focuses on crossover (recombination) phase of genetic algorithms. Crossover phase helps in achieving "exploration", which is one of the important issues in Genetic Algorithms. The basic crossover operators discussed in this paper are like single-point crossover, N-point crossover, uniform crossover, partially matched crossover, cycle crossover, edge crossover, arithmetic crossover.

Keywords: Genetic Algorithms, Crossover, Exploration, Evolutionary algorithms.

I. INTRODUCTION

The Genetic algorithm (GA) is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects typically fixed-length character strings patterned after chromosome strings, each with an associated fitness value, into a new population (the next generation) using operations patterned after the Darwinian principle of reproduction and survival of the fittest and after naturally occurring genetic operations [1]. Genetic Algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. Holland's GA is a method for moving from one population of "chromosomes" to a new population by using a kind of "natural selection" together with the genetics inspired operators of crossover, mutation and inversion. The Selection operators choose those chromosomes in the population that will be allowed to reproduce, and on average the fitter chromosomes produce more offspring than the less fit ones. Crossover exchanges subparts of two chromosomes, roughly mimicking biological recombination between two single chromosome ("haploid") organisms; mutation randomly changes the allele values of some locations in the chromosomes; and inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed[2].

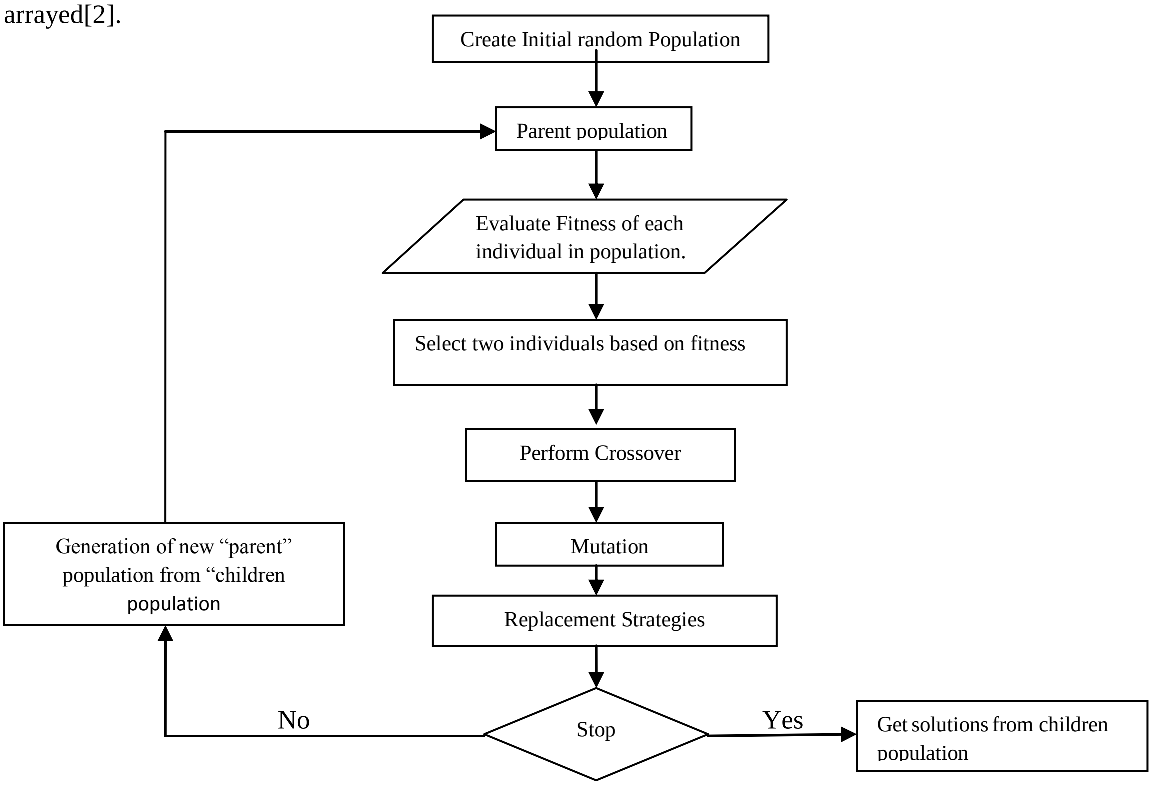

Fig.1. Flowchart of Basic GA.

In natural evolution, crossover is a complex phenomenon that combines pairs of chromosomes. The decision of which genetic crossover operators to use depends greatly on the encoding strategy. In crossover, the idea is to recombine building blocks on different strings. Two chromosomes are physically aligned, breakage occurs at one or more corresponding locations on each chromosome, and homologous chromosome fragments are exchanged before the breaks are repaired. This results in a recombination of genetic material that contributes to variability in the population. In genetic algorithm, crossover operator exchanges substrings between chromosomes represented as linear strings of symbols [3].

Crossover operation helps in achieving exploration. Exploration is the creation of population diversity by exploring the search space, and is obtained by crossover operators. Many variants of crossover operation can be made for specific problems that significantly affect the performance of GA. The main focus of this paper is to study various crossover operators.

II. TYPES OF CROSSOVER OPERATORS

A. *Single Point Crossover*

Holland's original theory of schemas assumed binary strings and single-point crossover. Single-point crossover is the simplest form: a single crossover position is chosen at random and the parts of two parents after the crossover position are exchanged to form two offspring. Single point crossover causes "positional bias" in which the schemas that can be created or destroyed by a crossover depend strongly on the location of the bits in the chromosome [2].

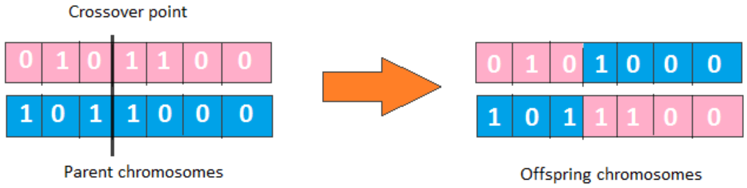

Fig. 1 One point Crossover

B. *N-point Crossover*

The concept of one-point crossover can be extended to N-point crossover, where N-crossover points are used, rather than just one or two. The N-point crossover was first implemented by De Jong in 1975. In consists of more than one crossover sites. In 2-point crossover value of crossover sites is 2. The value of N may vary from 1 to N-1. Adding of more crossover sites causes more disruption of building blocks that sometimes reduce the performance of genetic algorithm. But it allows the head and tail portion of a chromosome to be passed together in the offspring [4].

C. *Uniform Crossover*

In uniform crossover, every allele is exchanged between the pair of randomly selected chromosomes with a certain probability, known as swapping probability. Usually the swapping probability value is taken to be 0.5[5].

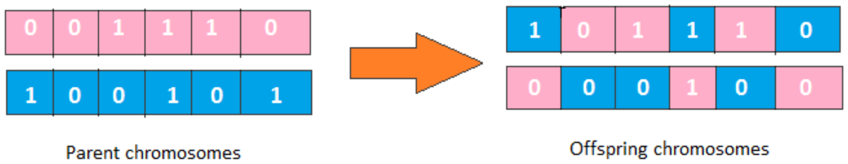

Fig. 2 Uniform Crossover

D. *Three parent Crossover*

In three parent crossover, three parents are chosen randomly. Each gene of the first parent is compared with the corresponding gene of the second parent. If both genes are same, the gene is taken for offspring otherwise the corresponding gene from the third parent is taken for the offspring [6]. Three parent crossover is mostly implemented in case of binary encoded chromosomes.

E. *Partially Matched Crossover*

Partially Matched or Mapped Crossover (PMX) was introduced by Goldberg and Lingle for Travelling Salesman Problem. PMX preserves orderings within the chromosomes. In PMX, two parents are randomly selected and two random crossover sites are generated. Alleles within the two crossover sites of a parent are exchanged with the alleles corresponding to those mapped by the other parent [5].

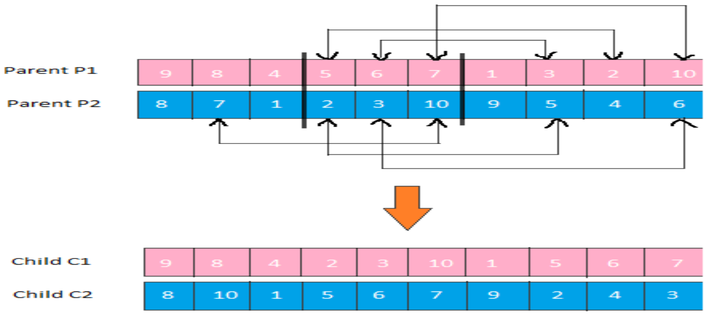

Fig. 3 Partially Matched Crossover

F. Uniform Order-Based Crossover

The N-point and uniform crossover methods described above are not well suited for search problems with permutation codes. Uniform order-based crossover is developed specifically for permutation codes. In uniform order-based crossover, two parents (say P1 and P2) are randomly selected and a random binary template is generated. Some of the genes for offspring C1 are filled by taking the genes from parent P1 where there is a one in the template. At this point we have C1 partially filed, but it has some "gaps". The genes of parent P1 in the positions corresponding to zeros in the template are taken and sorted in the same order as they appear in parent P2. The sorted list is used to fill the gaps in C1. Offspring C2 is created by using a similar process [5].

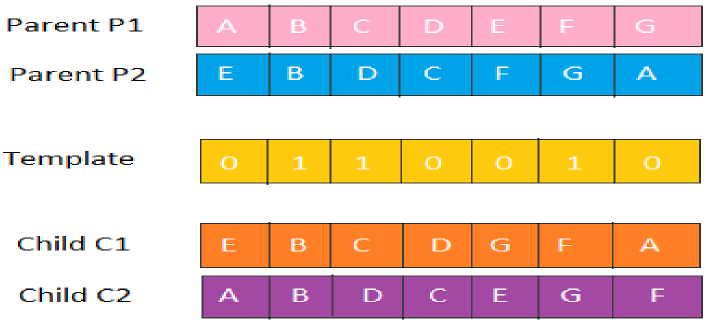

Fig. 4 Uniform Order- Based Crossover

G. Order-Based Crossover

Order-based crossover was proposed by Davis and is also used for chromosomes with permutation encoding. It is a variation of the uniform order-based crossover in which two parents are randomly selected and two random crossover sites are generated. The genes between the cut points are copied to the children. Starting from the second crossover site copy the genes that are not already present in the offspring from the alternative parent in the order they appear [5].

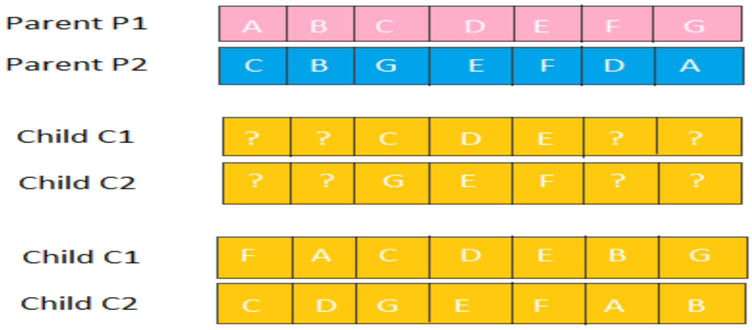

Fig. 5 Order-Based Crossover

H. Cycle Crossover

Cycle crossover is used for chromosomes with permutation encoding. During recombination in cycle crossover there is a constraint that each gene either comes from the one parent or the other [7]. The basic principle behind cycle crossover is that each allele comes from one parent together with its position. To make a cycle of alleles from parents, start with the first allele of parent1. Then look at the allele at the same position in parent2 and go to the position with the same allele in parent1. Add this allele to the cycle and repeat step until you arrive at the first allele of parent1. Put the alleles of the cycle in the first child on the positions they have in the first parent and the remaining alleles of first child come from the second parent along with their position. Generate next cycle from parent2 [4].

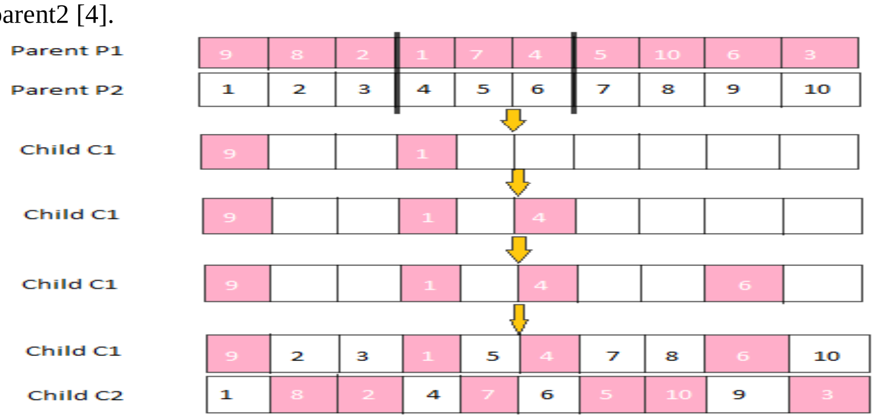

Fig. 6 Cycle Crossover

I. Edge Crossover

Edge crossover is based on the idea that an offspring should be created using only edges present in one or more parent [9, 10]. Edge table is created which for each element lists the other elements linked to it in the two parents. '+' sign in edge table indicates the presence of edge in both parents [8].

J. Arithmetic Crossover

In case of real-value encoding, we can implement arithmetic crossover. Arithmetic crossover operator linearly combines the two parent chromosomes. In arithmetic crossover, randomly two chromosomes are selected for crossover and by linear combination of these chromosomes, two offsprings are produced. This linear combination is as per the following computation:

Child1 = a.P1gene + (1-a).P2gene

Child2 = a.P2gene + (1-a).P1gene

Where Pgene represent the corresponding gene either from parent1 or parent2, and 'a' is the weight which governs dominant individual in reproduction and it is between 0 and 1.

K. Multiparent Crossover- Multiparent crossover does not exist in nature. In this case, number of parents is more than two and this scheme tests the cases which do not exist in nature. Technically, multiparent crossover will amplify the effect of recombination. Multiparent crossover operators are categorised on the basis of allele frequencies or on segmentation or numerical operations on real valued alleles [12, 13, 14].

III. CONCLUSIONS

In this paper, we have discussed nine types of crossover (recombination) strategy in the genetic algorithm procedure to create better offsprings. In single-point crossover, two parent chromosomes are crossed over only at single point. N-point crossover also works the similar way as single point crossover except for; chromosome is done at more than two points. The uniform crossover does not divide the chromosomes for crossover. Each gene of child is created by copying it, from the parent chosen according to the corresponding bit in the binary crossover mask of same length as the length of the parent chromosomes. Three parent crossover operates on three parents which are chosen randomly. Partially Matched Crossover (PMX) preserves the orderings within the chromosomes. Uniform order-based crossover, Order-based crossover and cycle crossover are used for chromosomes with permutation encoding. Edge crossover insures that edges of both parents are present in their offsprings. Arithmetic crossover is used for chromosomes with real valued encoding. Multiparent crossover does not exist in nature but amplify the effect of crossover. Many variants of crossover are found in GA Literature. Crossover operators are important as they facilitate "exploration", which means to search for new and useful adaptations. Which crossover operator to use depends in complicated ways on the particular fitness function, encoding and other details of the GA. In future, the usefulness of different types of crossover can be compared and their problem specific domains can be defined.

REFERENCES

[1] John R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, sixth edition, MIT Press, 1998.

[2] Mitchell Melanie, "An Introduction to Genetic Algorithms".

[3] C. Ryan, Automatic re-engineering of software using genetic programming, Genetic Programming Series, Kluwer Academic Publishers, ISBN 0-7923-8653-1, 2000.

[4] Sharma et al., International Journal of Advanced Research in Computer Science and Software Engineering, vol 3, No. 4, pp. 468-473, April 2013.

[5] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.

[6] S.N. Sivanandam and S.N. Deepa., Introduction to Genetic Algorithms, Springer, ISBN 9783540731894, 2007.

[7] I.M. Oliver, D.J. Smith and J.H. Holland, "A study of permutation crossover operators on the travelling salesman problem", Proceedings of the Third International Conference on Genetic Algorithms, London, Lawrence Eribaum Associates, 1987.

[8] R. Kumar et al., International Journal of Advanced Research in Computer Science and Software Engineering, vol 2, No. 8,pp. 78-82, Aug 2012.

[9] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, Heilderberg, Germany, 2003.

[10] Darrell Whitley, Functions as Permutations: Regarding No Free Lunch, Walsh Analysis and Summary Statistics. PPSN, 2000, 169-178.

[11] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 2nd edition, 1994.

[12] H. Muhlenbein, Parallel genetic algorithms, population genetics and combinatorial optimization, In Proceedings of the Third International conference on Genetic Algorithm, 1989, pp 416-421.

[13] A.E.Eiben, P.E. Raue and Z.Ruttkay, Genetic algorithms with multi-parent recombination, In Davidor et al. (ed.) Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, No. 866 in LNCS Springer Berlin Heidelberg New York, 1994, pp 78-87.

[14] S. Tsutsui, Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring, In Eiben et al. (Ed.) Proceedings of the 5th Conference on Parallel Problem Solving from Nature, 1998, pp 428-437.

