



Science

A COMPARATIVE STUDY OF CROSSOVER OPERATORS FOR GENETIC ALGORITHMS TO SOLVE TRAVELLING SALESMAN PROBLEM

Wafaa Mustafa Hameed ^{*1}, Asan Baker Kanbar ²

^{*1,2} Assistant Lecturer, Department of Computer Science, Cihan University \ Campus \
Sulaimaniya, Iraq

DOI: <https://doi.org/10.5281/zenodo.345734>

Abstract

Genetic algorithms (GAs) represent a method that mimics the process of natural evolution in effort to find good solutions. In that process, crossover operator plays an important role. To comprehend the genetic algorithms as a whole, it is necessary to understand the role of a crossover operator. Today, there are a number of different crossover operators that can be used , one of the problems in using genetic algorithms is the choice of crossover operator Many crossover operators have been proposed in literature on evolutionary algorithms, however, it is still unclear which crossover operator works best for a given optimization problem. This paper aims at studying the behavior of different types of crossover operators in the performance of genetic algorithm. These types of crossover are implemented on Traveling Salesman Problem (TSP); Whitley used the order crossover (OX) depending on specific parameters to solve the traveling salesman problem, the aim of this paper is to make a comparative study between order crossover (OX) and other types of crossover using the same parameters which was Whitley used.

Keywords: Genetic Algorithms; Crossover Operator; Traveling Salesman Problem; Order Crossover; Parameters, Natural Evaluation, Evolutionary Algorithms.

Cite This Article: Wafaa Mustafa Hameed, and Asan Baker Kanbar. (2017). “A COMPARATIVE STUDY OF CROSSOVER OPERATORS FOR GENETIC ALGORITHMS TO SOLVE TRAVELLING SALESMAN PROBLEM.” *International Journal of Research - Granthaalayah*, 5(2), 284-291. <https://doi.org/10.5281/zenodo.345734>.

1. Introduction

Genetic algorithms (GAs) are parts of the evolutionary computing, which is a rapidly growing area of artificial intelligence. (GAs) are inspired by Darwin's theory about evolution. Simply said, solution to a problem solved by genetic algorithms is evolved. Genetic Algorithms (GAs) were first suggested by John Holland and developed by him and his students and colleagues in the seventies. This leads to Holland's book “Adoption in Natural and Artificial Systems”

published in 1975 [1]. Over the last twenty years, it has been used to solve a wide range of search, optimization and machine learning problems. Thus, the genetic algorithm is an iteration procedure, which maintains a constant size population of candidate solution [1][2]. During each iteration step (generation) the structure in the current population is evaluated, and on the basis of those evaluations new populations of candidate solutions are formed. In 1992 John Koza has used the genetic algorithm to evolve programs to perform certain tasks. He called his method “genetic programming” (GP) [3]. In this paper we will try to study the effect of different types of crossover operators on certain known problem (Traveling Salesman problem), which was solved by Whitley using order crossover (OX) in genetic algorithms, This problem was chosen according to different factors such as representation of the problem (which have a great influence on genetic algorithm) can be applied more efficiently. Furthermore, this problem is chosen since, it owns a high complexity (the size and the shape of the search space), which, cannot be solved using traditional known searches, like exhaustive search method.

2. Use GA to Solve a Traveling Salesman Problem

Genetic algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search [1]. Genetic algorithm includes some parameters that should be adjusting so that the algorithm can provide positive results. Crossover operators play very important role by constructing competitive Genetic Algorithms (GAs) [4][5][6][7][8][9]. The general schema of GA may be illustrated as follows (Fig. 1).

```
Procedure GENETIC-ALGORITHM
Generate initial population  $P_0$ ;
  Evaluate population  $P_0$ ;
  Initialize generation counter  $g \rightarrow 0$ 
While stopping criteria not satisfied repeat
  Select some elements from  $P_g$  to copy into  $P_{g+1}$ ;
  Crossover some elements of  $P_g$  and put into  $P_{g+1}$ ;
  Mutate some elements of  $P_g$  and put into  $P_{g+1}$ ;
  Evaluate some elements of  $P_g$  and put into  $P_{g+1}$ ;
  Increment generation counter:  $g \rightarrow g + 1$ 
End while
```

Figure 1: Pseudocode of genetic algorithms

2.1. Problem Definition

First, we should address an important question connected with chromosome representation, should we leave a chromosome to be an integer vector? In our previous paper (optimization of function)[13] we represented a chromosome as a binary vector. This allowed us to use binary crossover and mutation operators; applying these operators we got legal offspring, i.e., offspring within the search space. This is not the case for the TSP. Clearly, if we use crossover and

mutation operators as defined earlier, we would need some sort of a “repair algorithm”; such an algorithm would “repair” a chromosome, moving it back into the search space. It seems that the integer vector (permutation) representation is better; instead of using repair algorithms, we can incorporate the knowledge of the problem into operators: in that way they would “intelligently” avoid building an illegal individual.

In this particular approach we accept path representation; a tour is described as a list of cities. The common example of (10) cities numbered from 1 to 10, which can be coded by the letters from A to J Table (1) Table (2).

Table 1: 10 cities numbered from 1 to 10

City	1	2	3	4	5	6	7	8	9	10
Code No.	A	B	C	D	E	F	G	H	I	J

Table 2: the distance between (10) cities of the TSP.

	A	B	C	D	E	F	G	H	I	J
A	0.0000	0.3361	0.3141	0.3601	0.5111	0.5176	0.2982	0.4564	0.3289	0.2842
B	0.3361	0.0000	0.1107	0.6149	0.8407	0.8083	0.5815	0.6418	0.4378	0.3934
C	0.3141	0.1107	0.0000	0.5349	0.7919	0.8207	0.5941	0.6908	0.4982	0.4501
D	0.3601	0.6149	0.5349	0.0000	0.3397	0.6528	0.5171	0.7375	0.6710	0.6323
E	0.5111	0.8407	0.7919	0.3397	0.0000	0.4579	0.4529	0.6686	0.7042	0.6857
F	0.5176	0.8083	0.8207	0.6528	0.4579	0.0000	0.2274	0.2937	0.4494	0.4654
G	0.2982	0.5815	0.5941	0.5171	0.4529	0.2274	0.0000	0.2277	0.2690	0.2674
H	0.4564	0.6418	0.6908	0.7375	0.6686	0.2937	0.2277	0.0000	0.2100	0.2492
I	0.3289	0.4378	0.4982	0.6710	0.7042	0.4494	0.2690	0.2100	0.0000	0.0498
J	0.2842	0.3934	0.4501	0.6323	0.6857	0.4654	0.2674	0.2492	0.0498	0.0000

The most prominent member of the rich set of combinatorial optimization problems is, undoubtedly, the Traveling Salesman Problem (TSP). In this problem a salesman, starting from his home city to visit each city on prescribed list exactly once and then returns home in such a way that the length of his tour is minimal. Obviously, TSP a sequencing example of NP-complete, the work area to be explored grows exponentially according with number of cities, and so does. [7][10][11][12]. In general, if n cities were must be visited by traveling salesman, then the general complexity is $n!$.

For this particular problem, the general complexity is $10! = 3628800$.

a) Initial Population

For the initialization process, we can either use some heuristics starting from different cities, or we can initialize the population by a random sample of permutation of $\{1,2,\dots,10\}$.

b) Evaluation Function

The evaluation of chromosome is straightforward, given the cost of travel between all cities n be the number of cities and $D=[d_{ij}]$ be the distance matrix, whose elements d_{ij} denote the distance between city i and city j . the problem, then, is to find the shortest tour visiting all cities exactly once.

The cost function, which is to be minimized, is chosen as:

$$f(\pi) = \sum_{i=1}^n d_i, \pi(i)$$

i.e. $f(\pi)$ gives the length of the tour corresponding to π .

In this problem, we calculate the fitness value as follows:

$$\text{Fit.} = 1/f(\pi)$$

3. Genetic Operators

3.1. Selection Operator

This method uses the roulette wheel selection method. The string with low fitness has a higher probability of contributing one or more offspring to the next generation. In roulette wheel selection, the individuals are given a probability P_i of being selected (10) that is directly proportionate to their fitness. The algorithm for a roulette wheel selection algorithm is illustrated in algorithm[3]

$$\frac{1}{N-1} * \left(1 - \frac{f_i}{\sum_{j \in \text{population}} f_j} \right) \quad (10)$$

Which f_i is value of fitness function for the individual

```
for all members of population
  sum += fitness of this individual
endfor
for all members of population
  probability = sum of probabilities + (fitness / sum)
  sum of probabilities += probability
endfor
number = Random between 0 and 1
for all members of population
  if number > probability but less than next
  probability
```

Figure 2: Roulette wheel selection algorithm

Thus, individuals who have low values of the fitness function may have a high chance of being selected among the individuals to cross

3.2.Crossover Operator

The strength of genetic algorithms arises from the structured information exchange of crossover combinations of highly fit individuals. So, what we need is a crossover-like operator that would exploit important similarities between chromosomes. For that purpose, the crossover used in this algorithm is the order (OX), as mentioned previously, given two parents, builds offspring by choosing a subsequence of a tour from one parent and preserving the relative order of cities from the other parent.[2][14].

For example, if the parents are:

$v_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10)$

$v_2 = (4\ 5\ 2\ 1\ 10\ 8\ 7\ 6\ 9\ 3)$

The resulting offspring is:

$o_1 = (1\ 10\ 8\ 4\ 5\ 6\ 7\ 9\ 3\ 2)$

$o_2 = (4\ 5\ 6\ 1\ 10\ 8\ 7\ 9\ 2\ 3)$

3.3.Mutation Operator

After the new generation has been determined, the chromosomes are subjected to a low rate mutation process. In genetic algorithms, mutation realized as a random deformation of alleles with a certain probability [15]. For this example we apply two mutation operators to introduce genetic diversity into the evolving population of permutation. The first operator is a simple two point mutation, which randomly selects two elements in the chromosome and swap them (1 10 8 4 5 6 7 9 3 2) becomes (1 10 3 4 5 6 7 9 8 2). The second operator is a shuffle mutation, which shunts the permutations forward by a random number of places; thus (1 10 3 4 5 6 7 9 8 2) shuffled forward six places becomes (6 7 9 8 2 1 10 3 4 5).

3.4.Genetic Parameters

For this particular problem, Whitley [15] used the following parameters: population size $pop_size=20$, probability of crossover $P_c=0.7$, probability of mutation $P_m = 0.005$.

4. Experimental Results

In table (3) we provide the generation number for which we noted improvement in the evaluation function, together with the value of the function. The best chromosome after (300) generations was: $v_{min} = (4\ 5\ 6\ 7\ 8\ 10\ 9\ 3\ 2\ 1)$ Which is slightly less than 2.8568.

Table 3: Results of 300 generations for TSP

Generation number	Evolution function	Fitness
0	4.6921	0.213
5	4.6617	0.215
13	4.5788	0.218
24	4.3250	0.231
39	4.0350	0.248
55	3.7874	0.264
91	3.7757	0.265
137	3.6613	0.273
195	3.6351	0.275
278	3.3576	0.298
290	2.8568	0.350

For this problem, a simulation has been constructed in order to apply the GA, when using the crossover parameters mentioned above, the following results are be obtained from [15]:

$$v_{min} = (1 \ 2 \ 3 \ 7 \ 6 \ 8 \ 9 \ 10 \ 5 \ 4)$$

Which corresponds to $f(v_{min}) = 2.6908$, then the fit. = $1/2.6908 = 0.3716$.

5. The Effect of Different Types of Crossover on Aveling Salesman Problem

In this part, we will try to study the effect of applying different types of crossover on the reported algorithms, on their performance, speed, and ability to find the solution .To see the effect of using different types of crossover operators on this problem, Whitley [16] used the Order crossover (OX) depending on the following parameters: Pc=0.7, Pm=0.005, Pop size=20 NG=1000. Table (3) describes the comparative study of the iterations results between the above crossover and the other kinds of crossovers (Partially-Mapped Crossover (PMX) , Ordered Crossover (OX), Alternating Edges crossover(AL), Cycle Crossover (CX), Simple Inversion crossover (SI), uniform Order-based Crossover(UOX),Point Crossover (PO)) [7][17][18] [19][20] [21][22][23] which are implemented on TSP problem,In addition, the table (4) shows average of iterations results for (10) runs.

Table 4: Comparison study of OX crossover and other kinds in TSP

CEOSSOVER	NG	FITNESS
PMX	597	0.295
OX	479	0.330
CX	546	0.312
UOB	564	0.300
SI	616	0.294
PO	498	0.325
AL	327	0.350

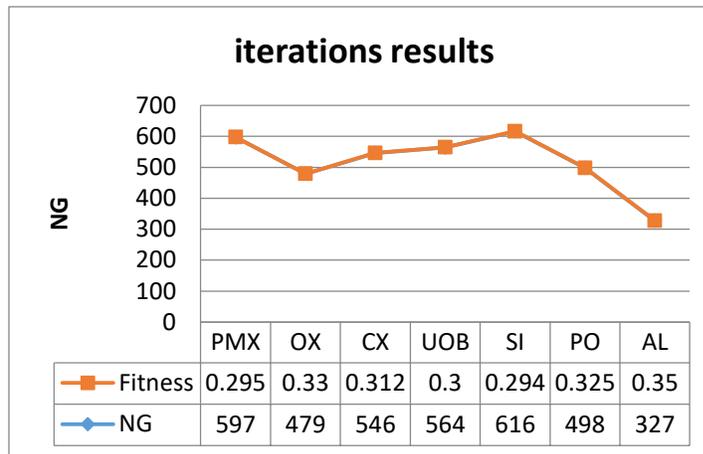


Figure 3: the average of iterations results for (10) runs

6. Conclusions & Future Scope

In this paper we introduce a Comparison study of OX crossover and other kinds of crossover (Partially-Mapped Crossover (PMX) , Ordered Crossover (OX), Alternating Edges crossover(AL), Cycle Crossover (CX), Simple Inversion crossover(SI), uniform Order-based Crossover (UOX),Point crossover (PO)) which implemented on TSP problem , the average iteration results show that the Alternating Edges crossover(AL) ,was the best to be applied for the TSP because it chooses a random edge from the second parent, which does not introduce cycle. The Simple Inversion crossover is the worst because the reversing of the chosen substring of chromosome may destroy the quality of the fitness value. Our future work will extend to the study for other kinds of crossover operators also studying of several selection methods for GA's such as ranking, tournament and proportional on effectiveness of results of role crossover.

References

- [1] Syswerda, G., "Uniform Crossover in Genetic Algorithms", Proceedings of the 3th International Conference on Genetic Algorithms, San Mateo, 1989.
- [2] Jorge magalhaes-mendes rua Dr.Antonio."A Comparative Study of Crossover Operators For Genetic Algorithms To Solve Job Scheduling Problem", West transaction on computers, ISSN 4,volum 12 April 2013.
- [3] Eshelman, L. J., Carnana, R. A., and Schaffer, J. D. "Biases in the Crossover Landscape", Proceedings of the 3th International Conference on Genetic Algorithms, San Mateo, 1989.
- [4] Koza J. R., "Genetic Programming", 2nd Edition, 1992.
- [5] Grant K., "An Introduction to Genetic Algorithm", C/C++ Users journal , 1995.
- [6] Abdoun Otman, Abouchbaka Jaffar, "A comparative Study of Adaptive Opaerators For Genetic Algorithms To Resolve The Traviling Salesman Problem", intrnational journal of computer application, volum 31, No 11, october 2011.
- [7] Jhon Grefenstelle, Rajev Gopal, "Genetic Algorithms for the Traveling salesman problems", Computer Science dep. Veaderbilt University.
- [8] Elaoud S. Loukilt, Teghmj, "Apareto Fitness Genetic Algorithms , Test Function study". European Journal Operational Research", 177(3), 1703-1719. 2007.
- [9] Lust T. Teghem ,J. memots "Amemetic Algorithms integrating tabusarch for Combinatorial Mutiobjective Optimization, RAIRO, 42, 3-33.

- [10] Speras, W.M, De Jong, K.A, "On The Virues if Paramitriize Uniform Crossover", Proceeding of the 4th International Conference on Genetic Algorithms, San Mateu 1991.
- [11] Imtiaz Hussain Khan, "Assesing Different Crossover Operators For Travelling Salesman Problem", I.J.Intelligent Systems and Applications, 2015, 11, 19-25.
- [12] J. Lysgaard, "Cluster Based Branching For the Asymmetric Traveling Salesman Problem", European Journal of Operational Research 119 (1999) 314-325
- [13] Wafaa Mustafa Hameed, "The Role of Crossover on Optimization of a Function Problem Using Genetic Algorithms", IJCSMC, Vol. 5, Issue. 7, July 2016, pg.425 - 429 .
- [14] Dorigo, M., & Gambardella, L. M. "Ant Colonies For the Traveling Salesman Problem. BioSystems, 43, 73-81. (1997).
- [15] R.R.Sharapov, "Genetic Algorithms : Basic idea Variants and Analysis, vision Sysdtems: Segmantation and Pattern Recognition ", Goro Obinata and Ashish Dutta (ED), 2007. PP.407-422, ISbN: 978-3-902613-05-9.
- [16] Whitley D., Starkweather, T., and Fuquay, D. A., "Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator", 1989.
- [17] D. Goldberg and R. Lingle. Alleles, loci, and the travelling salesman problem. In proceedings of the first international conference on genetic algorithms and their applications, 1985, 154-159.
- [18] I. Oliver, D. Smith and J. Holland. A study of permutation crossover operators on the travelling salesman problem. In proceedings of the second international conference on genetic algorithms and their applications, 1987, 224-230.
- [19] M. Yamamura, T. Ono and S. Kobayashi. Characterpreserving genetic algorithms for travelling salesman problem. Journal of the Japanese society for artificial intelligence, 1992: 7(6), 1049-1059.
- [20] Gridhar Gopal, Rakesh Kumar, Ishan Jawa, Nveen Kumar, "Enhanced Order Crossover For Permutation Problems", IJRSET, Vol.4, issue 2. Feb.2015.
- [21] Michalewicz, Z., "Genetic Algorithm + Data Structure = Evolution Programs", 3rd Revised and Extended Edition, Springer- Verlag Berlin Heidelberg, New York, 1996.
- [22] Zakir H. Ahmed. Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator . IJBB 3(6).2010.
- [23] D. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning. Addison Wesley, 1989.

*Corresponding author.

E-mail address: afafa_ju223@yahoo.com