International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:6, No:3, 2012

Two Individual Genetic Algorithm

Younis R. Elhaddad, Aiman S.Gannous

***Abstract*—**The particular interests of this paper is to explore if the simple Genetic Algorithms (GA) starts with population of only two individuals and applying different crossover technique over these parents to produced 104 children, each one has different attributes inherited from their parents; is better than starting with population of 100 individuals; and using only one type crossover (order crossover OX). For this reason we implement GA with 52 different crossover techniques; each one produce two children; which means 104 different children will be produced and this may discover more search space, also we implement classic GA with order crossover and many experiments were done over 3 Travel Salesman Problem (TSP) to find out which method is better, and according to the results we can say that GA with Multi-crossovers is much better.

***Keywords*—**Artificial Intelligence; Genetic Algorithm; order crossover; Travel Salesman Problem

I. Introduction

FOR the class of problems which is known to have non polynomial time solution, one usually resorts to heuristics solution. Artificial Intelligent (AI) is known to be an effective framework for solving such problems; some of the technique used in AI includes Genetic Algorithms (GA), Simulated Annealing (SA), and Ant Colony Optimization (ACO). Some of them are time consuming, while others could not find the optimal solution. Because of this many researchers tried to improve GAs using different methods and operations in order to improve solutions quality and reduce execution time [1]. Crossover is the most important operation of GA. This is because in this operation characteristics are exchanged between the individuals of the population. Accordingly this paper is concerned to explore experimentally the effect of reducing population size and increasing crossover operations during the process of GA; thus two GAs are implemented first one uses order crossover with population size of 100 individuals, while other uses multi crossover with population size of two individuals only.

Three instances from TSPLIB [2] are chosen for these experiments: KroA100, d198, and pcb442.The experimental results show that using two individual population with 50 crossover techniques have a much better performance than the classic techniques. It took less time and it obtained much better results for all problems used in these experiments. Moreover according to standard deviation, it is clear that the GA with Multi-crossover is more stable than the GA with one crossover technique.

Younis R. Elhaddadw is with Faculty of information Technology, University of Benghazi ,Libya (younis.haddad@benghazi.edu.ly)
Aiman S.Gannous is with Faculty of Public Health University of Benghazi, Libya (gannous@yahoo.com)

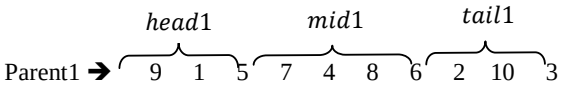
II. Genetic Algorithm

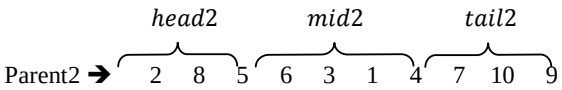
From the study of biological evolution Holland in 1975 introduced new optimization and search technique called Genetic Algorithm (GA), where each potential solution is represented as chromosome which contained the set of parameters for a given problem[3], GA started with population of chromosomes and according to a fitness function the best solutions are selected for recombination operation which uses both crossover and mutation operators to produces new improved generation of solutions, the best new solutions take place of poor old solutions, as this process works as we become closer to optimal solution, termination condition to be set to stop this process which can be time, reaching desired solution or number of iterations.

III. Multi-crossover technique

Multi-crossovers are applied to two individuals to produce 104 children with different characteristics inherited from their parents. Unlike the classic GA which used a large population size and mostly one type of crossover, the multi-crossover technique used only two individuals as a population, and applied 50 different crossover techniques over these parents to produce 104 children. Each one inherited a completely different attribute from their parents. The basic principle of this crossover is two random cut points (p_1 and p_2), a head, containing $(1,2 \ldots\ldots, p_1 - 1)$, the middle containing $(p_1, p_1 + 1, \ldots\ldots, p_2)$, and the tail containing $(p_2 + 1, p_2 + 2, \ldots\ldots, n)$.

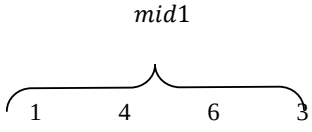
For example two selected random crossover points are:
$p_1 = 4$ and $p_2 = 7$; and the two parents' tours are:





A. Two points multi-crossover

For valid tours the elements of head2 and tail2 are removed from the parent1 to get mid1.



In the same way elements of head1 and tail1 are removed from the parent2 to get mid2.

$mid2$

8 6 4 7

Step 1: If the parts (head2; mid1; tail2) are reconnected using all possible permutations, six different children can be obtained (3!)

$head2$ $mid1$ $tail2$

$child1 \rightarrow$ 2 8 5 1 4 6 3 7 10 9

In the same way for (head1, mid2, tail1), six more children are produced.

$head1$ $mid2$ $tail1$

$child2 \rightarrow$ 9 1 5 8 6 4 7 2 10 3

Step 2: If the two heads are flipped and treated as in step 1 twelve new are produced.

$child3 \rightarrow$ 5 8 2 1 4 6 3 7 10 9

$child4 \rightarrow$ 5 1 9 8 6 4 7 2 10 3

Step 3: If the two tails are flipped and treated as in step 1; twelve new children are produced.

$child5 \rightarrow$ 2 8 5 1 4 6 3 9 10 7

$child6 \rightarrow$ 9 1 5 8 6 4 7 3 10 2

Step 4: If the two mids are flipped and treated as in step 1; twelve new children are produced.

$child7 \rightarrow$ 2 8 5 3 6 4 1 7 10 9

$child8 \rightarrow$ 9 1 5 7 4 6 8 2 10 3

Step 5: If the two heads and tails are flipped and treated as in step1; twelve new children are produced.

$child9 \rightarrow$ 5 8 2 1 4 6 3 9 10 7

$child10 \rightarrow$ 5 1 9 8 6 4 7 3 10 2

Step 6: If the two mids and tails are flipped and treated as in step1; twelve new children are produced.

$child11 \rightarrow$ 2 8 5 3 6 4 1 9 10 7

$child12 \rightarrow$ 9 1 5 7 4 6 8 3 10 2

At each step 12 children are produced. Therefore, finally $6 \times (3!) \times 2 = 72$ completely different children produced from only two parents.

B. One point Multi-crossover

If p2 was used as one crossover point,

$head1$ $tail1$

Parent1 -> 9 1 5 7 4 8 6 2 10 3

$head2$ $tail2$

Parent2 -> 2 8 5 6 3 1 4 7 10 9

After removing the elements of tail2 from parent1 the remaining elements formulate head1.

$head1$

1 5 4 8 6 2 3

After removing the elements of tail1 from parent 2 the remaining elements formulate head2.

$head2$

8 5 6 1 4 7 9

Step 7: If head1 and tail2 connect $(2!)$, two new children are produced. Also if head2 and tail1 connect $(2!)$, two new children are produced. The total number of children produced by this step is $2 \times (2!) = 4$ new children.

Step 8: If head1 and the flipped tail2 connect $(2!)$, two new children are produced. Also if head2 and the flipped tail1 connect $(2!)$, two new children are produced. The total number of children produced at this step is $2 \times (2!) = 4$ new children.

Step 9: If the flipped head1 and tail2 connect $(2!)$, two new children are produced. Also if the flipped head2 and tail1 connect $(2!)$, two new children are produced. The total number of children produced at this step is $2 \times (2!) = 4$ new children.

Step 10: If the flipped head1 and flipped tail2 connect (2!), two new children are produced. Also if the flipped head2 and flipped tail1 connect (2!), two new children are produced. The total number of children produced at this step is $2 \times (2!) = 4$ new children.

Thus from steps7, 8, 9 and 10 the number of children produced is $4 \times 2 \times (2!) = 16$.

If p1 is used as one crossover point,

$head1$ $tail1$

Parent1 -> 9 1 5 7 4 8 6 2 10 3

$head2$ $tail2$

Parent2 -> 2 8 5 6 3 1 4 7 10 9

Following the same procedure as in steps 6, 7, 8, and 9, 16 new children are produced.

Finally, the total number of produced children is $72 + 16 + 16 = 104$

IV. Order Crossover (OX)

The order crossover operator was developed by Davis in 1985 [4]. If we use this crossover technique for TSP and we

have 10 cities, two individuals (parents) of the population are selected for the crossover operation, if these parents are:

Parent 1 ← (0 8 4 5 6 7 1 2 3 9)
Parent 2 ← (6 7 1 |2 4 8 3 5 9 0)

and assuming that (4 and 7) are the crossover points selected randomly

Parent 1 ← (0 8 4 | 5 6 7 | 1 2 3 9)
Parent 2 ← (6 7 1 | 2 4 8 | 3 5 9 0)

to create the offspring, first the subtour between the two cut points are copied into the offspring, which gives:

Child 1 ← (x x x | 2 4 8 | x x x x)
Child 2 ← (x x x | 5 6 7 | x x x x)

Next, starting from the city located next to the second cut point of parent2, the rest of the cities, (after omitting the cities that are already present in child 1) are copied in the order in which they appear (1 3 9 0 5 6 7), starting from the position next to the subtour which placed offspring1. When the end of the offspring1 string is reached, it continues from its first position. In the example this gives the following child 1:

Child 1 ← (5 6 7 | 2 4 8 | 1 3 9 0)

Child 2 is generated in a similar manner for parent1:

Child 2 ← (2 4 8 | 5 6 7 | 3 9 0 1)

V. Experiments and Results

In this work many experiments have been done to explore the difference between classic GA with 100 individuals population using order crossover and GA with two individuals population multi-crossover. Three instances from TSPLIB are chosen for these experiments: KroA100, d198, and pcb442.The optimal solutions for the three instances are 21282, 15780 and 50778 respectively. In these experiments, order crossover (OX) in population size 100 and inversion mutation operation are used. No additional operations are performed. Using the same parameters the multi-crossover is implemented. Only the population size and type of crossover are changed. Both GA with OX and GA with multi-crossover are performed. Ten independent runs are performed respectively to all three problems. The termination condition is set at 20000 iterations for each run. The results of the ten runs for each instance are written in separated tables where tables 1 and 2 show the results obtained by GA with order crossover (OX) for the kroA100 instances from TSPLIB.

TABLE I
Result of kroA100 with OX

Exp. No.	Result	time
1	23502	1661
2	23234	1816
3	29324	1905
4	22090	1768
5	24146	1861
6	31044	1821
7	23929	1634
8	22892	1490
9	22314	1586
10	24119	1667
Avg.	24659.4	1720.9
St.dev	3022.3	133.48

TABLE II
Result of kroA100 with two individual

Exp. No.	Result	time
1	22791	1010
2	22780	935
3	23146	930
4	22996	943
5	22578	942
6	23188	929
7	23187	933
8	23154	956
9	23040	1006
10	22374	936
Avg.	22923.4	952
St.dev	282.9	30.5

According to the results of the experiment which is summarized in Tables 3 and 4, one can say that the new GA with two individual and multi-crossover techniques have a much better performance than the classic techniques. It took less time to finish the 20000 generations and it obtained much better results for all three problems. Moreover according to standard deviation, it is clear that the GA with Multi-crossover is more stable than the GA with order crossover.

TABLE III
Results of GA with two individual

Problem	Best result	Average result	St.dev.	Avg. time
KroA 100	22374	22923.4	282.9	952
d198	17276	17642	197.2	997.2
Pcb 442	94041	95684.6	1156.5	1313.9

TABLE IV
Results of GA with Order crossover (OX)

Problem	Best result	Average result	St.dev.	Avg. time
KroA 100	22090	24659.4	3022.3	1720.9
d198	19794	20281.6	345.2	2574.2
Pcb 442	106867	113741.8	282.9	5506.2

Figure 1 shows the comparison between the convergence rates during 20000 iterations, for the two crossover operations over the pcb442 problem. It clearly demonstrates that GA with two individual and multi-crossover has better convergence.

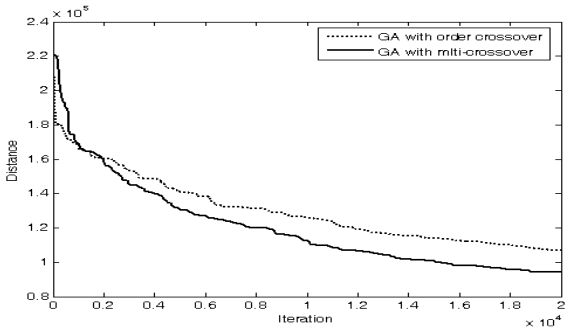


Fig. 1 convergence of GA with OX and GA with Multi-crossover over the pcb442 problem

Figure 2 shows the comparison of execution times during 20,000 iterations of the two crossover operations over the pcb442 problem. It demonstrates that GA with Multi crossover is faster and finished the 20,000 iterations in less time.

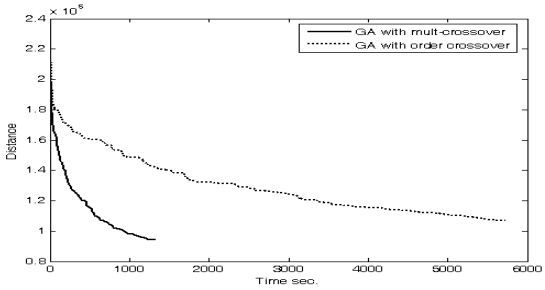


Fig. 2 time comparison of GA with OX and GA with two individual over the pcb442 problem

REFERENCES

[1] *Artificial Intelligence: A Modern Approach.* Russell, S and Norvig, P. New Jersey : Prentice Hall, 1995.
[2] http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/.*Heidelberg* University
[3] Beasley, D, Bull, D R and Martin, R. An Overview of Genetic Algorithms :. Part 1, Fundamentals. Norwegian University of Science and Technology.
[4] P. Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza and S. Dizdarevic. Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators