Journal of Education and Vocational Research
Vol. 4, No. 1, pp. 1-4, Jan 2013 (ISSN 2221-2590)

An Experimental Study of the Genetic Algorithm Convergence

Younis Elhaddad
University of Benghazi, Benghazi, Libya
younis.haddad@benghazi.edu.ly

Abstract: Genetic algorithm is a well-known heuristic search algorithm, typically used to generate valuable solutions to optimization and search problems. The most important operation in a genetic algorithm is crossover, as it has the greatest effect on its convergence rate. Therefore, in order to achieve the most optimal results in a reasonable time, one has to decide on the crossover type, as well as make a selection of a crossover point. In order to explore the effect of the crossover point selection methods on the convergence rate, we conducted experiments based on different crossover point selection criteria, whereby the results indicate the high importance of controlling the randomization of the crossover point selection range.

Keywords: *Crossover, Convergence rate, Genetic algorithm, Optimization problems*

1. Introduction

Genetic algorithms (GAs) have been previously found suitable as an approach to solving the optimization problems for which the exact solutions cannot be easily found by conventional methods (Goldberg, 1989). Thus, in solving an optimization or search problem, when implementing an efficient genetic algorithm, some points have to be taken into consideration, such as chromosome encoding, choosing the crossover type, mutation, and selection operations suitable for the specific problem. Moreover, if necessary, additional operations can be innovated. In this approach, crossover points are randomly chosen for two randomly selected individuals, dividing the chromosome to *n+1* segments, where *n* is the number of crossover points. Moreover, some of these segments can be swapped between the two selected individuals to produce new individuals in the next generation. In this work, we conduct experiments with the aim of exploring the effect of controlling the randomization of the crossover point selection methods. In other words, we attempt to limit the range from which the potential crossover points can be selected. To address this issue, we implemented standard GA with order crossover, using TSP instances rat783 and pcb442 defined in TSPLIB (Moscato, 1989). This arrangement is used as a test bench to illustrate how the range of the crossover points affects the GA convergence. The experimental results indicate that the choice to limit the crossover point range should be aligned with the problem being solved and the length of the chromosome.

2. Genetic Algorithm

Genetic Algorithm is adaptive heuristic search algorithm inspired by the evolutionary ideas of natural selection, the concept of the survival of the *fittest* and genetic research (Moscato, 1989). The main goal of the GAs is to simulate the mechanism of natural selection and survival of the *fittest theory* applicable to all living beings. Using a random population of individuals (chromosomes) that represent candidate solutions to a given problem, the process starts with the evaluation of each individual based on a predefined fitness function. Next, according to the natural selection and survival of the *fittest theory, only the* individuals of the highest quality are selected for the recombination procedure, whereby crossover operation is applied to two parent chromosomes to produce the new generation of individuals. The mutation operation is applied on a selected number of individuals, according to a predefined mutation rate. These steps are repeated until previously determined stop condition is met.

3. Crossover points

As the crossover operation plays an important role on GA implementation with the aim of solving the optimization problems (TSPLIB95), many researchers have attempted to improve the crossover techniques in

order to improve the GA convergence rate. In this work, we will address this issue from a different perspective, as the goal is to experimentally examine the effects of the crossover point values on the convergence rate. To achieve the goal of this research, a traditional GA with order crossover is implemented, using four different crossover point selection methods. Each experiment based on one of the four methods is run five times and the *average* of thus obtained results is used in the final comparison. The TSP instances rat783 and pcb442 defined in TSPLIB were used as a test bench for comparing the results (Moscato, 1989). It is important to mention that, as finding the optimal solution is not the aim of this work, each experiment is run for five minutes for rat783 instance and three minutes for pcb442 instance, as this is deemed sufficient for the comparison purposes.

4. Results

Experiment 1: In this experiment, the crossover points are selected randomly, provided that their values do not exceed the first quarter of the chromosome length.

$$1 < Crossover\ point1\ value \leq 0.25 \times n$$
$$1 < Crossover\ point2\ value \leq 0.25 \times n$$

Where $Crossover\ point1\ value < Crossover\ point2\ value$, and n is the chromosome length. The results of five experimental runs based on the above conditions, using rat783 instance, and each lasting for five minutes, are presented in Table 1.

Table 1: Crossover Points≤

No.	Iterations	Result
1	4517	32151
2	4521	2046
3	4544	33578
4	4519	33834
5	4544	32480
Average	4529	32817.8

Experiment 2: This experiment differs from the previous one only in that the range from which the crossover points could be selected did not exceed half of the chromosome length, as noted in the expressions below.

$$1 < Crossover1\ point\ value \leq 0.5 \times n$$
$$1 < Crossover2\ point\ value \leq 0.5 \times n$$

Where $Crossover\ point1\ value < Crossover\ point2\ value$, and n is the chromosome length. Table 2 shows the results of Experiment 2.

Table 2: Crossover Points≤ 0.5 n

No.	Iterations	Result
1	3841	41692
2	3840	40823
3	3845	40249
4	3692	39524
5	3711	40974
Average	3785.8	40652.4

Similarly, the range from which the crossover points can be selected in the third experiment is defined as:

$$1 < Crossover1\ point\ value \leq 0.75 \times n$$
$$1 < Crossover2\ point\ value \leq 0.75 \times n$$

With the results of five experimental runs given in Table 3.

Table 3: Crossover Points≤ 0.75

No.	Iterations	Result
1	3203	43386
2	3180	44306
3	3221	46331
4	3190	47444
5	3192	46678
Average	3197.2	45629

Finally, in line with the above, the conditions for the crossover point selection in the fourth experiment are given as:

$$1 < Crossover1\ point\ value \leq n$$
$$1 < Crossover2\ point\ value \leq n$$

And the results of five repeated experimental runs are given in Table 4.

Table 4: crossover Points ≤ n

No.	Iterations	Result
1	2777	48553
2	2743	44057
3	2741	46774
4	2749	43211
5	2803	47600
Average	2762.6	46039

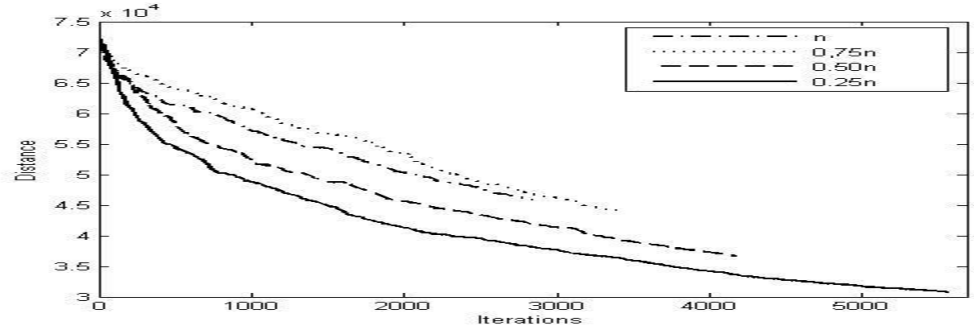
After all experiments have been performed, an average result obtained for each range setting is calculated. Table 5 summarizes the average results of the four experiments for the rat783 instance.

Table 5: Results for Rat 783 Instance

Limitations	Iterations	Result
n	2762.6	46039
0.75n	3197.2	45629
0.50n	3785.8	40652.4
0.25n	4529	32817.8

Figure 1 shows the comparison between the convergence rates for the four experiments run using the rat783 instance. As previously noted, the termination time was set to five minutes.

Figure 1: Comparison between convergence rates for rat783 instance



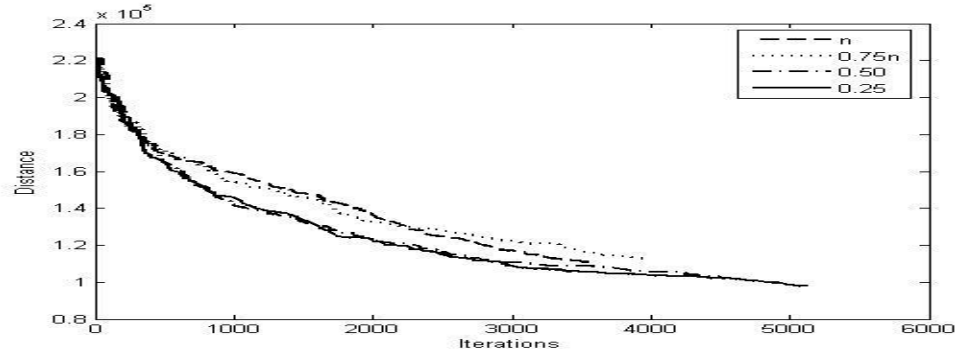
The above experiments were repeated (five runs for each range setting) using pcb442 instance, and the average results were calculated, as previously described. Table 6 summarizes the average results from five runs of four experiments for the pcb442 instance, with each run lasting three minutes.

Table 6: Results for pcb442 instance

Limitations	Iterations	Result
n	3232.2	110136.2
0.75n	3707.4	108650
0.50n	4201.2	105788
0.25n	4915.8	99179.2

In line with the presentation of the experimental results pertaining to rat783 instance, Figure 2 depicts the comparison between the convergence rates for the four experiments run using the pcb442 instance, with termination time set to three minutes.

Figure 2: Comparison between convergence rates for pcb442 instance



The above findings clearly demonstrate the importance of the position of crossover points in determining the GA convergence rate. More specifically, when the range from which the points can be randomly selected is set to be less than or equal to first quarter of the chromosome length, the GA exhibits the best convergence rate and the algorithm using rat783 instance is performed considerably faster. The same is true for pcb442 instance, albeit with less significant improvements, indicating that the optimal crossover point position also depends on the chromosome length. In the first case, where the length of the chromosome was 783, the difference between the results of the four experiments was significant, whereas, when the length of the chromosome was nearly halved (442 in the second experimental arrangement using pcb442 instance), the difference between the results is not that apparent.

5. Conclusion and future work

This paper presented the results of an experimental study of the effect of crossover point position in the chromosome, using two well known TSP instances as a test bench. According to the reported results, it can be concluded that the most optimal range from which the crossover points should be selected lies in the first quarter of the chromosome, as this leads to the marked improvement in the convergence rate. Moreover, the algorithm runs faster, indicating that limiting the range of potential crossover point values can strongly affect the overall GA performance. This work can be improved upon by testing other types of limitations or by examining the effect of setting the crossover points to constant values.

References

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wiley Publishing Company, Massachusetts, Mass, USA.

http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/. Heidelberg University.

Moscato, P. (1989). On genetic crossover operators for relative order preservation, Caltech Concurrent Computation program, Report C3P 778.