IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. I (Sep.- Oct. 2017), PP 31-36
www.iosrjournals.org

Performance of Arithmetic Crossover and Heuristic Crossover in Genetic Algorithm Based on Alpha Parameter

*Mhd. Furqan[1], Hartono[2], Erianto Ongko[3], Muhammad Ikhsan[4]
[1,4]Ilmu Komputer UINSU Medan, Indonesia)
[2]Program Doktor Ilmu Komputer USU Medan, Indonesia)
[3]Teknik Informatika ATI Immanuel Medan, Indonesia)
Corresponding Author: Mhd. Furqan

Abstract: *Genetic algorithm (GA) is a heuristic search algorithm based on the idea of natural selection that occurs in the process of evolution and genetic operations. One of the critical stages in the genetic algorithm is a crossover process. In the crossover, phase occurs the gene mix between the parent that it will determine the diversity in the population. This paper will describe the effects of the alpha parameter in the crossover process which includes arithmetic crossover and heuristic crossover. The Case studies that used in this study is the Traveling Salesman Problem (TSP). The influence of parameters on the performance of genetic algorithm alpha is associated with gene diversity resulting from the crossover. The results showed that in the arithmetic crossover, the best alpha value is 0.5, becoming the best alpha value because of a balanced genes combination from both parents. On Heuristic Crossover obtained different results where the alpha value which gives the best performance is 0.9. This method slightly different from the Arithmetic Crossover caused on Heuristic Crossover; the alpha parameter used as multiplier factor after the subtraction process of genes from both parents.*

Keywords : *Genetic Algorithm, Alpha Parameter, Arithmetic Crossover, Heuristic Crossover*

Date of Submission: 25-08-2017 Date of acceptance: 08-09-2017

I. Introduction

Genetic algorithms (GAs) are a class of evolutionary algorithms made famous by John Holland and his colleagues during the 1970s (Holland 1975). GA is a searching method used for choosing the best solution of the different problems, based on the mechanism of natural selection.

Lozano et al. (2008), states that if the diversity of the population becomes increased and so that the quality of the solutions gets better, thus preserving high levels of useful diversity. Pasquier and Erdogan (2010) in their research about Genetic Algorithm Optimization in Maze Solving Problem states that Alpha value as the multiplier factor of the Improved Segment Crossover has direct influence over the diversity.

There are 16 crossover operators includes: Discrete crossover, simple arithmetic crossover, single arithmetic crossover, whole arithmetic crossover, Local Crossover, SBX Crossover, BLX-Alpha Crossover, BLX-Alpha-Beta Crossover, Flat Crossover, BGA Crossover, Heuristic Crossover, Average Crossover, One Point Crossover, No Crossover, Combination Crossover, and Random Crossover (Picek et al.2013).

Ongko (2015) in his research on Performance Analysis of the Method Arithmetic Crossover in Genetic Algorithm, obtain the result that the whole arithmetic crossover has the best average fitness values are better than simple arithmetic crossover and simple arithmetic mean crossover that has the best fitness value is better than a single arithmetic crossover.

Gupta and Ghafir (2012) proposed a research about the method for maintaining diversity in a genetic algorithm. In this study, they determine that the difference in genetic algorithm do a population adapt quickly to changes in the environment and it allows the community to continue searching for productive niches, thus avoiding becoming trapped at local optima. Thus improving diversity in GAs makes GA more useful efficient way to solve problems.

Malik and Wadhwa (2014) proposed a Preventing Premature Convergence in Genetic Algorithm using DGCA and Elitist Technique. In this study, they conclude that This factor will reduce the chance of premature convergence and therefore reduced the chance that GA will be trapped in a local optimum.

We have already made a previous research about the influence of Alpha value as multiplier factor on the arithmetic crossover and this research give a result that diversity in genes of chromosomes as a result of arithmetic crossover give the influence to performance of the genetic algorithm.

This study will analyze the performance of arithmetic crossover and heuristic crossover in the genetic algorithm based on the alpha parameter.

II. Genetic Algorithm (Ga)

This algorithm is adaptive heuristic based on ideas of natural selection and genetics. A genetic algorithm is one of the most known categories of the evolutionary algorithm. The mine of concept these evolutionary algorithms did stimulate process in natural system necessary for evolution. GA's are used for numerical and computational optimization and based on studying the evolutionary aspects of models of social systems. The GA performs a balanced search on various nodes, and there is a need to retain population diversity exploration so that any relevant information cannot lose because there is a great need to focus on relevant portions of the population. The basic GA operators are Crossover, mutation, selection (Malik et al.2014).
Selection is used to select the chromosome whose fitness value is small (Gupta et al.2013). We have used the roulette wheel selection.

Crossover is a genetic parameter which will combine two chromosomes (can also be called as parents) to produce a new chromosome (also called as offspring). The result of the crossover will give the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. The new offspring will have some properties from one parent and some properties from another parent. For example, if the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two single−chromosome (haploid) organisms (Hole et al.2013).

Mutation can take place after the crossover gets performed. This case is to prevent falling all solutions in population into a local optimum of the solved problem. The mutation depends on the encoding as well as the crossover. For example, the string 00000100 might be mutated in its second position to yield 01000100. Variation can occur at each bit position in a chain with some probability, usually very small (e.g., 0.001) (Hole et al.2013). The process of genetic algorithm can be seen (Figure 1)

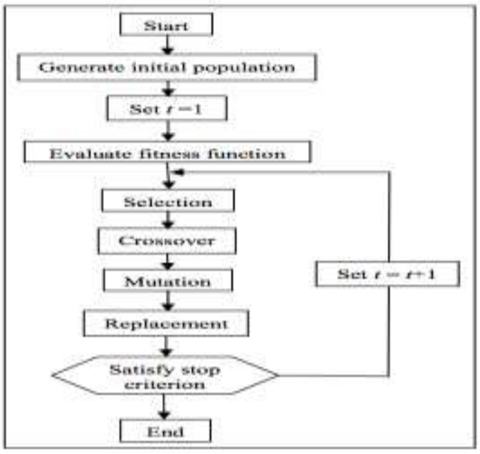

Figure 1. Flowchart of Genetic Algorithm (Kasim et al.2012)

In roulette wheel selection, the individuals are mapped to contiguous segments of a line, such that each segments is equally sized to its fitness. A random number is generated, and the individual whose section spans the random number is selected. The process repeats until the desired number of people is obtained (called mating population). This technique is analogous to a roulette wheel with each slice proportionally sized to the fitness (Pencheva et al.2009).

2.2. Arithmetic Crossover

In case of real-value encoding, we can implement arithmetic crossover. Arithmetic crossover operator linearly combines the two parent chromosomes. In an arithmetic crossover, randomly two chromosomes are selected for crossover, and by a linear combination of these chromosomes, two off springs are produced. This linear combination is as per the following computation: (Thakur,2014).

Child1=a.P1gene + (1-a).P2gene
Child2=a.P2gene+ (1-a).P1gene (1)

There are three types of arithmetic crossover, which is as follows. (Picek et al.2013).

1. Single Arithmetic Crossover

First, a random allele k is chosen. At that position take the arithmetic average of two parents. Other points are copied from the parents (Picek et al.2013). In the area specified gene, the gene value will be determined through an arithmetic operation on the value of the parent genes according to equation[2] (Eiben,2007). The arithmetic operations on single arithmetic crossover can be seen in Equation[2] and Figure 3.

$$\text{Child} = \left\langle x_1, \ldots, x_k, \alpha . y_k + (1-\alpha) . x_k, \ldots, x_n \right\rangle \qquad (2)$$

Where:
α = variable multiplier whose value ranges from 0-1

1. Simple Arithmetic Crossover

In simple arithmetic crossover, Determine the random number as a point of intersection between 0 and along the chromosomes of each parent. For gene on chromosome offspring to the limit before the crossover point is copied from a gene on chromosome parent (Picek et al.2013). For gene after the point of intersection, the existing gene formed of arithmetic operations in the genes of chromosomes as the parent with equation[3] (Eiben et al.2007). Illustration of simple arithmetic crossover process can be seen in Figure 4.

$$\text{Child} = \left\langle x_1, \ldots, x_k, \alpha . y_{k+1} + (1-\alpha) . x_{k+1}, \ldots, \alpha . y_n + (1-\alpha) . x_n \right\rangle \qquad (3)$$

Where:
α = variable multiplier whose value ranges from 0-1

2. Whole Arithmetic Crossover

On the whole arithmetic crossover, gene on chromosome offspring obtained from the results of arithmetic operations gene on chromosome parent [8], where the arithmetic process that is carried out in accordance with the equation[4] (Eiben et al.2007). Illustration of arithmetic crossover whole process can be seen in Figure 5.

$$\text{Child} = \left\langle \alpha . \bar{x} + (1-\alpha) . \bar{y} \right\rangle \qquad (4)$$

2.3. Heuristic Crossover

First, take two parents β^{f1} and β^{f2} and assume that the first parent (β^{f1}) has smaller value on each allele (Wright, A.H.1991). Then the offspring β^{S} is created as

$\beta_i^{S} = \alpha . (\beta_i^{f1} - \beta_i^{f2}) + \beta_i^{f1}$ (5)

III. Travelling Salesman Problem

Mathematical problems of the Traveling Salesman Problem proposed in 1800 by the Irish mathematician William Rowan Hamilton and the British mathematician Thomas Penyngton. TSP problems this is an issue where a salesman must visit all the cities in which each city is visited only once, and he had to start and return to the city of origin. The goal on the TSP problem is to find the shortest route for a salesman (Biggs et al.1976).

IV. Application Of Genetic Algorithm

Moon et al. (2002), using a genetic algorithm for solving the TSP problem with precedence constraints.Research results, Moon et al. (2002) gives the result that genetic algorithm produced an optimal solution and showed superior performance compared to the traditional algorithm. Several other studies have been done about the genetic algorithm. Roeva et al. (2013), proposed research about the influence of the population size on the genetic algorithm performance. The result of the study was the increase of population size enhances the accuracy of the solution.

V. Genetic Algorithm Model For Travelling Salesman Problem

Benchmark data used in this study is berlin52.tsp.the distance between the two cities can be calculated using the euclidean equation.

$$d_{(i,j)} = \sqrt{(x_i - x_j) + (y_i - y_j)^2} \qquad (5)$$

Specification:
x_i = x coordinates of the city i
x_j = x coordinates of the city j
y_i = y coordinates of the city i
y_j = y coordinates of the city j
the fitness value can be calculated using the equation 6.
Fitness = 1 / Total of Distance (6)

Number of generation in this study was 100. This study uses 10 chromosomes and 51 genes. Probability of crossover used is 0.5 and the mutation rate is 0.1.

VI. Results And Discussion

In this study the performance assessment results will be displayed for each arithmetic crossover method and heuristic crossover in solving the problems of TSP. Performance measures will be based on the average fitness value being produced in each generation. The higher of the average fitness value means the better the results obtained. Measurements will be performed by using the Alpha value of 0.1, 0.3, 0.5, 0.7, and 0.9.

6.1. Testing Results on Whole Arithmetic Crossover

Tests performed a total of 100 generations with crossover probability value of 0.5 and mutation rate values of 0.1 and α valueare 0.1, 0.3, 0.5, 0.7, and 0.9.

Table 1. Testing Results on Whole Arithmetic Crossover

Number of Testing	Alpha Value				
	0.1	0.3	0.5	0.7	0.9
1	0.00003339	0.00003479	0.00003451	0.00003449	0.00003347
2	0.00003608	0.00003466	0.00003494	0.00003473	0.00003454
3	0.00003384	0.00003504	0.00003493	0.00003420	0.00003409
4	0.00003377	0.00003499	0.00003487	0.00003405	0.00003411
5	0.00003501	0.00003403	0.00003533	0.00003436	0.00003417
6	0.00003597	0.00003465	0.00003450	0.00003455	0.00003424
7	0.00003444	0.00003471	0.00003546	0.00003401	0.00003334
8	0.00003491	0.00003550	0.00003441	0.00003503	0.00003409
9	0.00003393	0.00003541	0.00003542	0.00003440	0.00003320
10	0.00003481	0.00003428	0.00003460	0.00003562	0.00003405
Average Fitness	0.00003459	0.00003480	0.00003489	0.00003454	0.00003393

6.2. Testing Results on Simple Arithmetic Crossover

Tests performed a total of 100 generations with crossover probability value of 0.5 and mutation rate values of 0.1 and α valueare 0.1, 0.3, 0.5, 0.7, and 0.9.

Table 2. Testing Results on Simple Arithmetic Crossover

Number of Testing	Alpha Value				
	0.1	0.3	0.5	0.7	0.9
1	0.00003384	0.00003497	0.00003412	0.00003632	0.00003399
2	0.00003501	0.00003386	0.00003443	0.00003513	0.00003428
3	0.00003353	0.00003381	0.00003498	0.00003385	0.00003386
4	0.00003297	0.00003358	0.00003597	0.00003279	0.00003424
5	0.00003457	0.00003416	0.00003412	0.00003429	0.00003419
6	0.00003343	0.00003500	0.00003495	0.00003359	0.00003409
7	0.00003358	0.00003536	0.00003351	0.00003243	0.00003383
8	0.00003451	0.00003536	0.00003568	0.00003545	0.00003368
9	0.00003326	0.00003591	0.00003402	0.00003376	0.00003404
10	0.00003509	0.00003396	0.00003533	0.00003429	0.00003330
Average Fitness	0.00003396	0.00003458	0.00003469	0.00003415	0.00003395

6.3. Testing Results on Single Arithmetic Crossover

Tests performed a total of 100 generations with crossover probability value of 0.5 and mutation rate values of 0.1 and α valueare 0.1, 0.3, 0.5, 0.7, and 0.9.

Table 3. Testing Results on Single Arithmetic Crossover

Number of Testing	Alpha Value				
	0.1	0.3	0.5	0.7	0.9
1	0.00003546	0.00003345	0.00003379	0.00003467	0.00003356
2	0.00003431	0.00003265	0.00003679	0.00003501	0.00003296
3	0.00003331	0.00003386	0.00003495	0.00003479	0.00003368
4	0.00003456	0.00003456	0.00003521	0.00003533	0.00003416
5	0.00003447	0.00003320	0.00003464	0.00003418	0.00003435
6	0.00003415	0.00003510	0.00003506	0.00003270	0.00003327
7	0.00003419	0.00003504	0.00003296	0.00003297	0.00003406
8	0.00003435	0.00003473	0.00003614	0.00003540	0.00003268
9	0.00003285	0.00003511	0.00003286	0.00003446	0.00003333
10	0.00003383	0.00003278	0.00003583	0.00003199	0.00003420
Average Fitness	0.00003413	0.00003402	0.00003478	0.00003411	0.00003362

6.4. Testing Results on Heuristic Crossover

Tests performed a total of 100 generations with crossover probability value of 0.5 and mutation rate values of 0.1 and α value are 0.1, 0.3, 0.5, 0.7, and 0.9. The result of the testing can be shown in Table 4.

Table 4. Testing Results on Heuristic Crossover

Number of Testing	Alpha Value				
	0.1	0.3	0.5	0.7	0.9
1	0.00003411	0.00003411	0.00003474	0.00003503	0.00003567
2	0.00003382	0.00003397	0.00003467	0.00003487	0.00003520
3	0.00003364	0.00003406	0.00003465	0.00003499	0.00003511
4	0.00003346	0.00003423	0.00003484	0.00003509	0.00003509
5	0.00003340	0.00003421	0.00003453	0.00003481	0.00003582
6	0.00003371	0.00003398	0.00003443	0.00003465	0.00003527
7	0.00003383	0.00003415	0.00003445	0.00003506	0.00003514
8	0.00003316	0.00003410	0.00003473	0.00003486	0.00003546
9	0.00003351	0.00003412	0.00003436	0.00003464	0.00003589
10	0.00003396	0.00003421	0.00003490	0.00003451	0.00003571
Average Fitness	0.00003366	0.00003411	0.00003463	0.00003485	0.00003544

Then, the conclusion of the testing results can be seen in Table 5

Table 5. Conclusion of the Testing Results

Methods	Alpha Value				
	0.1	0.3	0.5	0.7	0.9
Whole Arithmetic	0.00003459	0.00003480	0.00003489	0.00003454	0.00003393
Simple Arithmetic	0.00003396	0.00003458	0.00003469	0.00003415	0.00003395
Single Arithmetic	0.00003413	0.00003402	0.00003478	0.00003411	0.00003362
Heuristic	0.00003366	0.00003411	0.00003463	0.00003485	0.00003544

Test results are presented in graphical form as shown in Figure 2.

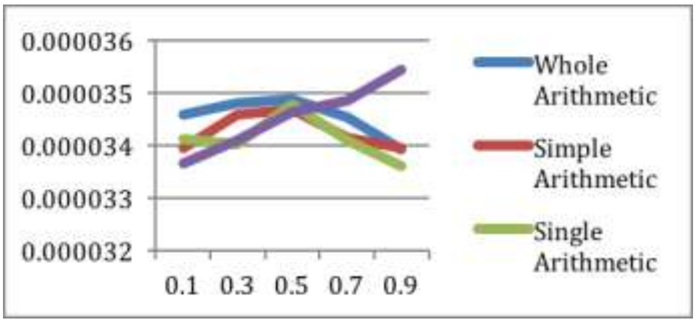

Figure 2. Testing Result in Graphical Form

We see that after the first hundred generations, in the arithmetic crossover, an alpha parameter of 0.5 seems to offer the highest diversity over the whole time period. This condition applies to the whole arithmetic, simple arithmetic and single arithmetic. Increasing diversity is demonstrated by the increase in performance that occurs along with an increase in the value of alpha.

Heuristic crossover seems slightly different from the Arithmetic Crossover caused on Heuristic Crossover, the alpha parameter is used as multiplier factor after the substraction process of genes from both parents. In the Heuristic Crossover, an alpha parameter of 0.9 seems to offer the highest diversity over the whole time period.

At the whole arithmetic method, the increase in performance occurs on increasing the alpha value of 0.1 to 0.3, from 0.3 to 0.5. From 0.5 to 0.7 occurred performance decreases, and from 0.7 to 0.9 also performance decreases. So it can be concluded that, at the whole arithmetic, Alpha value = 0.5 is best, because it produces a nicely balanced incorporation of genes from parent 1 and parent 2 so that diversity increases. It is also supported by the results of the two best in the ranking is occupied by 0.3 compared with the results of 0.1

and 0.9. Both 0.1 and 0.9 occupy the worst result since diversity is low as a result of the dominance of one parent gene.

In simple arithmetic method, the increase in performance occurs on increasing the alpha value of 0.1 to 0.3, from 0.3 to 0.5. From 0.5 to 0.7 occurred performance decreases, and from 0.7 to 0.9 also performance decreases. So it can be concluded that, at the simple arithmetic, Alpha value = 0.5 is best, because it produces a nicely balanced incorporation of genes from parent 1 and parent 2 so that diversity increases. It is also supported by the results of the two best in the ranking is occupied by 0.3 and 0.7 compared with the results of 0.1 and 0.9. Both 0.1 and 0.9 occupy the worst result since diversity is low as a result of the dominance of one parent gene.

The same conditions occur on a single arithmetic, where the increase in performance occurs on increasing the alpha value of 0.1 to 0.3, from 0.3 to 0.5. From 0.5 to 0.7 occurred performance decreases, and from 0.7 to 0.9 also performance decreases. So it can be concluded that, at the whole arithmetic, Alpha value = 0.5 is best, because it produces a nicely balanced incorporation of genes from parent 1 and parent 2 so that diversity increases. in the case of arithmetic crossover singles are exceptions where the performance of the value of alpha = 0.1 better than 0.3 or 0.7.

In Heuristic Crossover, the increase in performance occurs on increasing the alpha value of 0.1 to 0.3, 0.3 to 0.5, 0.7 to 0.9. Alpha value = 0.9 give the best performance because the bigger alpha value as the multiplier factor after the substraction process of genes will give the higher diversity.

VII. Conclusions

a. The value of alpha affects the diversity in arithmetic crossover, which it give affects to the performance of the genetic algorithm, in which the best value of alpha in arithmetic crossover must shows a balanced mix of genes from each parent.

b. The value of alpha also affects the diversity in a heuristic crossover, which the bigger alpha value as the multiplier factor after the subtraction process of genes will give the higher diversity.

c. An alpha value of 0.5 is the alpha value which gives the highest performance results both for the whole arithmetic crossover method, simple arithmetic crossover, and a single arithmetic crossover.

d. An alpha value of 0.9 is the alpha value which gives the highest performance results for the heuristic crossover.

References

[1] Biggs, N.L., Lloyd, E.K. and Wilson, R.J. 1976. Graph Theory 1736-1936. Clarendon Press: Oxford Eiben, A.E. & Smith, J.E. 2007. Introduction to Evolutionary Computing Genetic Algorithms. Springer: New York

[2] Gupta, Deepti, Ghafir, Shabina. 2012. An Overview of Methods Maintaining Diversity in Genetic Algorithms.International Journal of Emerging Technology and Advanced Engineering (IJETAE)**2**(5): 56-60

[3] Gupta, Saloni, Panwar, Poonam. 2013. Solving Travelling Salesman Problem Using Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering **3**(6): 376-380

[4] Hole, Komal R, Gulhane, Vijay S, Shellokar, Nitin D. 2013. Application of Genetic Algorithm for Image Enhancement and Segmentation. International Journal of Advanced Research in Computer Engineering & Technology **2**(4): 1342-1346

[5] Holland JH. 1975. Adaptation in Natural and Artificial Systems. The University of MichiganPress: Ann Arbor.

[6] Kasim, Nurhayati, Majid, Hairudin Abdul, Samah, Azurah A. 2012. Parameter Estimation of Warranty Cost Model Using Genetic Algorithm. International Journal of Soft Computing and Engineering**2**(5): 163-166

[7] Lozano, Manuel, Herrera, Francisco and Cano, Jose Ramon. 2008, Replacement Strategies to Preserver Useful Diversity in Steady-State Genetic Algorithm.Information Sciences**178**: 4421-4433

[8] Malik, Shikha, Wadhwa, Sumit. 2014. Preventing Premature Convergence in Genetic Algorithm Using DGCA and Elitst Technique. International Journal of Advanced Research in Computer Science and Software Engineering **4**(6): 410-418

[9] Moon, Chiung, Kim, Jongsoo, Choi, Gyunghyun and Seo, Yoonho. 2002. An Efficient Genetic Algorithm for The Travelling Salesman Problem with Precedence Constraints. European Journal of Operational Research**140**: 606-617

[10] Ongko, Erianto. 2015. Performance Analysis of the Method Arithmetic Crossover in Genetic Algorihtm. Thesis. University of Sumatera Utara

[11] Pasquier, Thomas, and Erdogan, Julien. 2010, Genetic Algorithm Optimization. InstitutSuperieurd'Electronique de Paris

[12] Pencheva, T., Atanassov, K., Shannon, A. 2009. Modelling of a Roulette Wheel Selection Operator in Genetic Algorithm Using Generalized Nets. Journal of BioAutomation**13**(4): 257-264

[13] Picek, Stjepan, Jakobovic, Domagoj and Gloub, Marin. 2013. On the Recombination Operator in The Real-Code Genetic Algorithms, 2013 IEEE Congress on Evolutionary Computation, pp. 3103-3110

[14] Roeva, Olympia, Fidanova, Stefka and Paprzycki, Marcin. 2013. Influence of The Population Size on The Genetic Algorithm Performance in Case of Cultivation Process Modelling. Proceedings of The 2013 Federated Conference on Computer Science and Information Systems, pp. 371 – 376.

[15] Thakur, Pratibha and Singh, Amar Jeet. 2014. Study of Various Crossover Operators in Genetic Algorihtms. International Journal of Advanced Research in Computer Science and Software Engineering **4** (3): 320-323.

[16] Wright, A.H. 1991. Genetic Algorihtm for Real Parameter Optimization. Morgan Kaufmann: Massachusetts

Mhd. Furqan. "Performance of Arithmetic Crossover and Heuristic Crossover in Genetic Algorithm Based on Alpha Parameter." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 5, 2017, pp. 31–36.