DOI: 10.14421/ijid.2022.3743 IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 11, No. 2, 2022, Pp. 222-231

Data Search Process Optimization using Brute Force and Genetic Algorithm Hybrid Method

Yudha Riwanto[1], Muhammad Taufiq Nuruzzaman[2,*], Shofwatul 'Uyun[3], Bambang Sugiantoro[4]
Department of Informatics
UIN Sunan Kalijaga
Yogyakarta, Indonesia
yudha.6357@gmail.com[1], m.taufiq@uin-suka.ac.id[2], shofwatul.uyun@uin-suka.ac.id[3], bambang.sugiantoro@uin-suka.ac.id[4]

Article History
Received November 8th, 2022
Revised January 7th, 2023
Accepted January 7th, 2023
Published January 2023

Abstract— High accuracy and speed in data search, which are aims at finding the best solution to a problem, are essential. This study examines the brute force method, genetic algorithm, and two proposed algorithms which are the hybrid of the brute force algorithm and genetic algorithm, namely Multiple Crossover Genetic, and Genetics with increments values. Brute force is a method with a direct approach to solving a problem based on the formulation of the problem and the definition of the concepts involved. A genetic algorithm is a search algorithm that uses genetic evolution that occurs in living things as its basis. This research selected the case of determining the pin series by looking for a match between the target and the search result. To test the suitability of the method, 100-time tests were conducted for each algorithm. The results of this study indicated that brute force has the highest average generation rate of 737146.3469 and an average time of 1960.4296, and the latter algorithm gets the best score with an average generation rate of 36.78 and an average time of 0.0642.

Keywords— search technique; direct approach; multiple crossover genetics; genetics with increments value; data search

1 INTRODUCTION

Optimization in finding solutions to problems is required in all sectors. Oftentimes, solutions to complex problems are needed immediately. There are many search methods in Artificial Intelligence (AI), but each algorithm has its own advantages and disadvantages. Thus, it is necessary to develop a method to get maximum results. The problem often occurs in finding solutions to problems is the speed needed to find the best solution. A good algorithm is an algorithm that effectively uses time and memory space so that it can be analyzed for efficiency and complexity [1].

Some of the algorithms used in the search include genetic algorithms and brute force algorithms. Genetic algorithms are algorithms that can be used to solve complex optimization problems that are difficult to do with conventional methods [2]. Genetic algorithms by combining each allele into an n-point crossover can speed up the process of space and time [3]. The research conducted by Wayan et al concluded that the genetic algorithm with the n-point crossover model uses a value of 0.8 which requires 5.38 minutes of computing time, while the cycle crossover model with a crossover probability value uses the appropriate value of 0.6 which requires time. computing. of 9.27 minutes. Another algorithm that can also be used to find the best solution is the brute force algorithm, which is an algorithm that looks for solutions by testing all possibilities to find the best solution. The essence of brute force is that patterns and text are compared character by character by shifting the search pattern from one position to the right and the comparison is repeated until a match is found or the end of the text is reached [4].

In its implementation, optimization is required because it often takes a long time to find the best solution. Even though the results of the search process are often close to the maximum value, because there is a random process, the previous value often decreases, so the processing time increases. With the constraints on the two methods, the researcher proposes a new flowchart of the data search process based on the genetic algorithm and the brute force method by limiting random values so that the process will be more structured.

2 METHOD

This study examines the brute force method, genetic algorithm and 2 proposed algorithms which are the development of the brute force algorithm and genetic algorithm, namely Multiple Crossover Genetic (MCG), and Genetics with increments value (MWIV). This test selected the case of determining the pin series by looking for a match between the target and the search result. To test the suitability of the method, 100 times testing was conducted for each algorithm. In the test, the initial state (initial population) was defined. The function of the definition was to ensure fairness in all methods so that other factors outside the research concept that could affect research results could be minimized. The initial state was defined by generating random values, which would then be used to conduct research.

In conducting the research, the researcher also used the same device specifications for all methods so that the hardware factor would not make a comparison of the results between methods. The following devices were used in this study. The devices used in this study are described in Table 1.

Table 1. Table of Tools and Materials

Type	*Detail Type*
Hardware	Processor Intel® CoreTM i7-860
	Ram 8 Gb
	SSD 750 Gb
Software	Linux Ubuntu 18.04 LTS
	IBM SPSS Statistics 22
	Python 3.7

2.1 *Research and Development (R&D)*

The type of research used in this research is Research and Development (R&D). R&D is a strategy or research method that is powerful enough to improve practice [5]. The flow carried out in this study includes several stages which are explained through a flowchart as shown in the diagram in Figure 1.

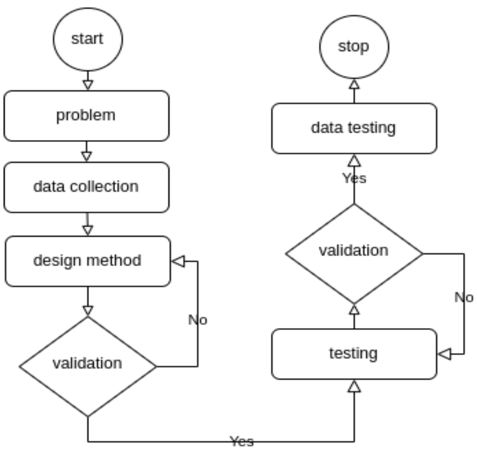

Figure 1. Research flowchart

2.1.1 Potential and problems: R&D starts from the potential and problems found by researchers. Potential and problem data obtained from other people's research and their own research.

2.1.2 Data collection: Once the potentials and problems are studied, greater data collection is needed as a tool for development planning. Data is obtained from various sources such as journals, books, the internet, etc.

2.1.3 Planning: In planning this research, the R&D method includes formulating research objectives, conducting needs management, and determining the qualifications of researchers and their participation in research.

2.1.4 Method Design: The final result of the initial research series can be in the form of a work plan for a new method or a new method that has been identified and the best approximate solution is found.

2.1.5 Method validation: Stages to evaluate the design of a new method that is rationally appropriate for testing.

2.1.6 Method design revision: After validating the method that has been designed, it is revised after the method's weaknesses are known.

2.1.7 Method trial: Conducting limited trials on methods that have been revised. This trial is a small-scale trial with temporary data in accordance with the problem to be studied. At this stage product testing is carried out more broadly, the tests carried out include several aspects, namely:

- Product design effectiveness test.
- Design testing using model-based experimental techniques.
- The results of the test will become an effective design reference in terms of substance and methodology.

2.1.8 Method revision: The method's revision is based on a limited trial's results. Improvements at this stage include bug fixes to ensure all possibilities that are out of control or outside the scheme can be overcome. This step will further refine the product being developed. The refinement stage needs to be done to determine the accuracy of the developed product. At this stage, the level of effectiveness can be accounted for.

2.1.9 Trials: tested in real conditions to find out the results of the products that have been studied.

2.2 Testing

In this study, the test was carried out using the whiteboard method with a loop testing technique to validate various iterations and loop end conditions in the control structure to obtain appropriate results [6]. This method was chosen because, in this study, there were many iterations that needed to be considered in the testing phase. Meanwhile, the white box method was chosen in this study because it did not have a GUI display that could be used by users.

This method is used to perform testing and analysis of program code without looking at the interface of the system [7]. The steps taken are as follows:

2.2.1 Mapping source code for each function: At this stage, mapping is carried out on each existing function so that it will be easier to check the system.

2.2.2 Create flow graphs from source code mapping: The flow graph is created as an illustration to see the flow's iterations on the internal system.

2.2.3 Test scenario: Tests are carried out to ensure that all systems run as expected, and all loops can run and can handle all possibilities that occur.

2.3 Comparison Test

In the testing stage, the Independent Sample T-test method is used. This stage is used to compare the results obtained. The tests carried out are divided into 3 tests, namely testing time, generation and method complexity. In each test, a comparison test of the results obtained by comparing the results of research between one method with another method is carried out, so that the average difference in the results obtained is known [8]. This test is carried out with SPSS as a tool for testing the Independent Sample T-test.

2.4 Brute Force

Brute force (Figure 2) is a search method with a direct approach to solving problems[9]. In general, this algorithm looks for a solution based on the problem statement and the definition of the concepts involved [10]. Brute force algorithms can solve problems in a very simple, direct and clear way[11]. There are several characteristics of the brute force algorithm, including:

- The brute force algorithm works in an unintelligent and inefficient way.
- The brute force algorithm is more suitable to be used to solve problems that require a small size. Because the flow is simple, and the implementation is easy.
- The brute force algorithm is generally used as a comparison algorithm for other search algorithms.
- Brute force Algorithms can search for models with regularity, or special tricks that can be used as a reference to find efficient or smarter algorithms.
- Brute force Algorithms can almost solve all problems. In fact, there were problems that could only be solved by the Brute Force method.
- The brute force algorithm is an algorithm that is not preferred because of its inability, but the search process with basic patterns, regularities, or special tricks, can help find algorithms that are smarter, more efficient and more effective.
- For problems that are small in scale, simple and basic are usually more taken into account than incompetence.

The brute force algorithm has several advantages from its use, including:

- Can solve big problems.
- Has a simple and easy-to-understand flow.
- Has developed into an algorithm for solving several problems such as searching, sorting, string matching, or matrix multiplication.
- Produces a standard algorithm for calculating the multiplication of n numbers, determining the minimum or maximum value in data.

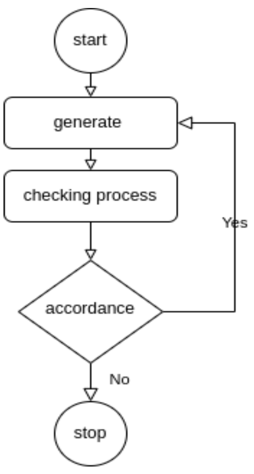

Figure 2. Flowchart of brute force algorithm

In addition to the advantages it has, there are also disadvantages to the brute force algorithm:

- Rarely produce effective algorithms.
- Some are slow, so it is unacceptable.
- Not constructive, creative other problem-solving techniques

There are several examples of calculations using the brute force algorithm, including:

1. Calculates an n (a > 0, n is a non-negative integer) as shown in Equation 1:

$$a^n = a \times a \times \ldots \times a \text{ (n times)}, \text{ if } n > 0 \quad (1)$$
$$= 1, \text{ if } n = 0$$

   Algorithm: multiply 1 by an n time.

2. Multiply two n × n matrices:

   Let C = A × B (Equation 2) and the elements of the matrix are expressed as c_{ij}, a_{ij}, and b_{ij}

$$C = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{n2} b_{nj} = \sum a_{ik} b_{kj} \quad (2)$$

   Algorithm: calculate each result element individually, by multiplying two vectors by length n.

2.5 *Genetic Algorithm*

In finding a solution to a problem, sometimes complex mathematical formulas are needed to help find a definite solution. The optimal solution can be obtained, but it often requires long and complicated processes and calculations. These problems can be solved using heuristic methods, namely search methods based on intuition or empirical rules to find the best solution [12].

The heuristic method does not always produce the best solution, but if the design is done well, the algorithm with the maximum solution will be obtained in a short time. Genetic Algorithm (GA) (Figure 3) is a branch of evolutionary algorithm which is an optimization technique based on natural genetics. The process of finding the genetic algorithm solution is finding the optimal solution by finding the optimal point based on the probability function [13].

The genetic algorithm takes the principles of genetics and natural selection described in Darwin's theory discovered by John Holland of the University of Michigan, United States. David Goldberg, who is a student of John Holland, conducted research and introduced it more broadly in a book entitled "Adaptation of Natural and Artificial Systems" published in 1975 [14]. Genetic algorithms are designed to simulate processes in natural systems that occur in the evolutionary process, especially in the theory of natural evolution put forward by Charles Darwin, namely survival of the fittest [15].

The flow of the genetic algorithm itself starts from initializing the initial solution which is then referred to as the population. The solution for the best population is taken and used to form a new population. This is so that the newly formed population is better than the old population [16]. The selection process begins with choosing the best individual from several populations. The best individuals are expected to produce offspring that inherit the traits of their parents and will be added to the next generation. If the parent has good traits, it is expected that the offspring will have better traits and a chance to survive. This process is carried out iteratively and ultimately produces the generation with the best individuals. The solution chosen to form a new solution is selected by testing the fitness value of each. The solution with the highest fitness value will be the solution used.

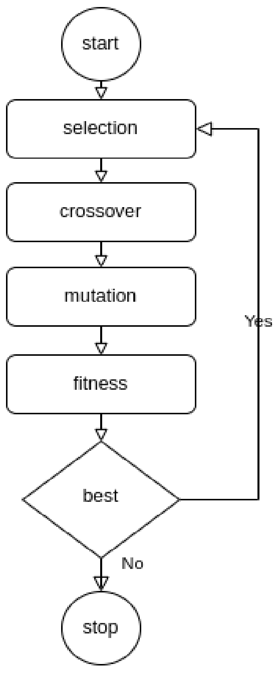

Figure 3. Flowchart of genetic algorithm

Stages of Genetic Algorithm:

2.5.1 *Gen and Individual Encoding/Decoding Techniques:* Encoding is used to encode the value of the genes that make up an individual. The value of this gene is obtained randomly [17]. There are 3 coding methods most commonly used, namely coding for real numbers: gene values are in the interval [0 1]
Example:

There are three variables (x1, x2, x3) which are coded into individuals consisting of three gens.
Decoding is used to encode the genes that make up the individual so that the value does not exceed a predetermined range and becomes the value of the variable that will be sought as a solution to the problem. If the value of the variable x that has been coded is changed to [ra rb], i.e., rb=lower limit, ra=upper limit, then the way to change the value of the variable above until it is in a new range [rb ra], is called Decoding.

- Decoding Real Numbers: X = rb + (r a - r b) g
- Decimal Discrete Decoding: X = rb + (r a - r b) (g 1 x 10 -1 + g 2 x 10 -2 +.......+ g n x 10 -n)
- Binary Encoding: X = rb + (r a - r b) (g 1 x 2 -1 + g 2 x 2 -2 +.......+ g n x 2 -n)

2.5.2 *Population Initialization:* This process begins with collecting individuals, which is referred to as a population. Each individual is a solution to the problem to be solved. An individual is identified by a set of parameters known as genes. The genes themselves are then combined into strings in the form of chromosomes (solution). In Genetic Algorithm, the gene set of an individual is described using strings in numeric/alphabetical form, or in general binary values can also be used (strings 0 and 1). Then the genes in the chromosomes are coded. Before generating the initial population, first the number of individuals in the population is determined. For example, the number of individuals is N. After that, we generate an initial population that has N individuals at random.

2.5.3 *Fitness Function*: The fitness function is a technique to measure the level of conformity with the problems of each individual in order to survive and compete with other individuals. The probability that an individual will be selected for reproduction is determined based on the fitness value. The fitness value can also be said as a statement of the value of the objective function. The purpose of the genetic algorithm is to maximize the fitness value. If the maximum value is sought, then the fitness value is the value of the function itself. But if what is needed is a minimum value, then the fitness value is the inverse of the value of the function itself. The reverse process can be done using Equation 3.

$$\text{Fitness} = C - f(x) \text{ atau Fitness} = \frac{C}{f(x)\in} \quad (3)$$

C is a constant and ∈ is a small number which is determined to avoid division by zero and x is an individual.

2.5.4 *Selection:* The function of the selection stage is the process of selecting the best individuals and making existing individuals can be processed for the next stage. At this stage, 2 individuals are selected who will come from parents who have the highest fitness value to carry out the mutation process. This is chosen because the individual is considered to be the best individual so it is worthy to mutate or produce another individual whose hope is a new individual with better quality.
However, sometimes there is a thing or x-factor that causes individuals to have fitness values that are similar to each other. This will be an obstacle when the selection process is carried out because errors can occur when choosing parents, such as leading to a pseudo-optimum value.
Problems like this often arise in genetic algorithms. To avoid this, a mechanism called Linear Fitness Ranking (LFR) was created. The purpose and mechanism of this are actually to scale the fitness values using Equation 4.

$$\text{LFR(i)} = F_{\max} - (F_{\max} - F_{\min})\left(\frac{R(i)-1}{N-1}\right) \quad (4)$$

Information:

LFR(i)	= i individual LFR value
N	= number of individuals in the population
R(i)	= the rank of the i-th individual after being sorted and the fitness value from largest to smallest
f_{max}	= highest fitness value
f_{min}	= lowest fitness value

2.5.5 *Crossover:* This stage is the most significant process in the genetic algorithm. At this stage, each pair of parents is mated, and then the crossover point is chosen randomly among the genes. Offspring is obtained from the process of exchanging gene values from parents among themselves until the crossover point is met. Then the results of the new offspring are entered into the population.
An individual that leads to the optimal solution can be obtained through a crossover process, but the crossover process can only be done if the random number r in the generated interval [0 1] is smaller than a certain probability value, or it can be said that the value of r < problem. In general, the prob value is determined to be close to 1. The simplest way to do a crossover is to do a one-point crossover method, namely the position of a single point of intersection, which is done randomly.

2.5.6 *Mutation:* In the results of new offspring formed from several genes can sometimes experience mutations with a small random probability. This is because the implication of some of the bits in the bit string is reversible.
This mutation stage aims to maintain diversity between populations and prevent premature convergence (overestimation). Mutations are carried out on all genes contained in the individual if the resulting random number is smaller and the mutation probability p is determined. Then it is done by setting

p = 1/N, where N is the number of genes in the individual. For binary code, the mutation is done by inverting the value of bit 1 to bit 0, or vice versa.

2.5.7 *Population replacement:* Population replacement (generational replacement) means that all initial individuals and one generation are replaced by temporary individuals resulting from the process of crossing over and mutations.

2.5.8 *Algorithm Termination:* The search on the genetic algorithm stops if the target population has been found or does not find a better solution than before. Then it can be said that the genetic algorithm has found a solution to the problem.

2.6 Multiple Crossover Genetics (MCG)

MCG is based on a genetic algorithm but uses three cross-processes. This is because the results of observations of the genetic algorithm process at this crossover stage use random values that make it possible to reduce the previous fitness value. The process begins by selecting a predetermined initial population and then proceeding with the best crossover chromosome (Fig. 4). In the crossover process, the value used as the cutoff value is not only a random value but there are three values used, namely a quarter of n, half of n and a random value. Therefore, an additional process is carried out to check the fitness value of each crossover. The values of the 3 crossovers are compared and the best results from the crossover are then followed by the mutation function. The fitness calculation function will get the best results.

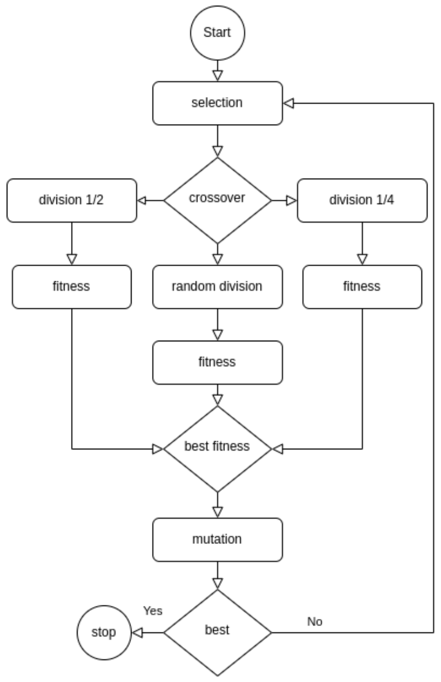

Figure 4. Flowchart of Multiple Crossover Genetics

2.7 Genetics With Increments Value (GWIV)

Genetics with increments value (GWIV) uses a genetic pattern that changes the trajectory when the fitness value hits a predetermined threshold. It is intended that when the pin combination has touched the specified fitness number, the random value in the genetic algorithm process is no longer used. Before running the process, first, determine the baseline to ensure the settings for the transition process. This baseline value is randomly determined with a minimum of > 75%.

The process starts with selecting the initial population (Fig. 5). After the selection process enters the crossover stage, then the entry process enters the mutation process. After getting the mutation results, the process will run by calculating the value. The fitness value obtained is checked against a predetermined baseline. If the value has touched the baseline, then it is continued, but if not, a re-selection process is carried out and continued until the fitness value touches the baseline.

A checking process is also carried out, to check whether the value matches the target or not. If the value matches the target, then the process stops. If the value does not match the target, then changes are made to each gene, one by one after the calculation is carried out. If the fitness value does not change, then the gene will continue to be changed until it gets a better fitness value. If the fitness value increases, then changes will be made to the gene afterwards. This process is carried out until the best solution is found.

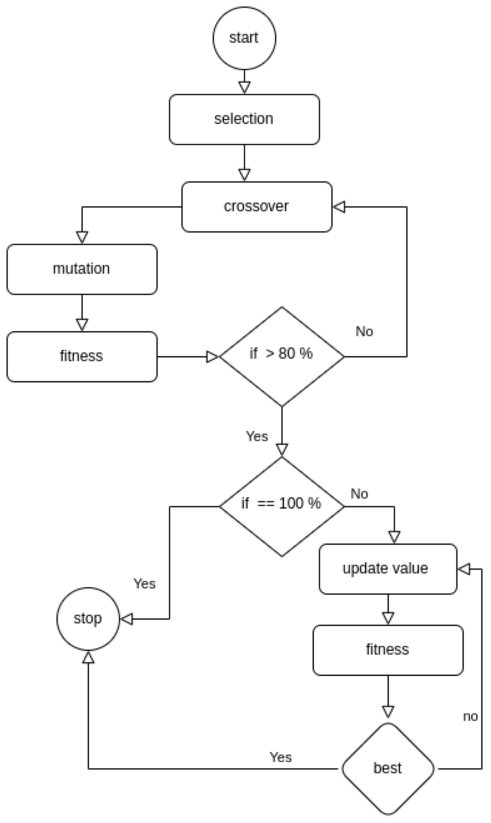

Figure 5. Flowchart of Genetics with Increments Value

2.8 Hybrid recommender system

A hybrid recommender system is a technique of combining several methods to reduce weaknesses in a method. It is a method that combines two or more recommendation techniques to improve the performance of recommendations, which are usually used to solve problems that exist in each of the methods [18]. In the development of a hybrid system, there are seven techniques: weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level [19]. The hybrid method itself is divided into 3 types, namely the first monolithic, which includes combination and meta-level, and both ensemble designs, which include augmentation, cascade, weighting and switching features, the three mixed systems stand alone and are not considered monolithic or ensemble [19].

3 RESULT AND DISCUSSION

The results are the comparative data for each method which was calculated on the average of the total number of trials. They are the number of generations of each method and the time needed for each algorithm to get the best results in searching for a series of pins. In this study, the target pin to look for was 403401. To equalize the initial conditions of all algorithms before performing the computation, the initial population data were determined, namely gene 1 695649, gene 2 565760, gene 3 021394, and gene 4 456228. These were data obtained from generated before testing. In the brute force method, the value of the 4th gene was taken as the initial population because it got the value closest to the goal.

3.1 Brute Force

This technique began with a specified point and then continued after getting the pin combination, by checking each number/gene and trying each pattern and text and then comparing each character one by one. The pattern was then shifted from one position to the right and compared. If a mismatch was found in a pin combination, the system would regenerate the combination until it matched.

In the research conducted 100 times with the same initial data, an average generation value of 671283.4747 times was obtained with a travel time of 1782.4656 seconds and an algorithm complexity value of 1345599.8 for each process. Figure 6-8 shows this result.

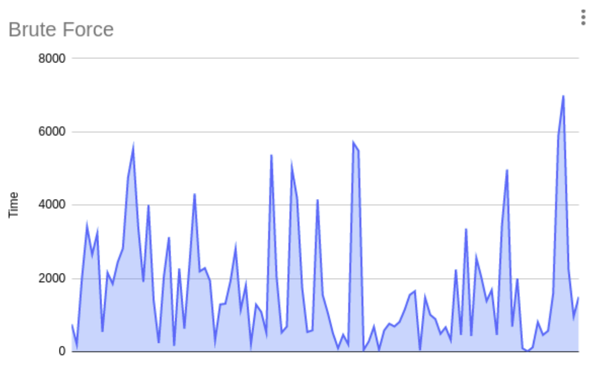

Figure 6. Result time brute force

3.2 Genetic Algorithm

The search process began with selecting a predetermined population because the initial value was a predetermined value to ensure the similarity of the initial conditions of the study. The process continued with crossover and proceeded to the mutation function to obtain a new population. The function to calculate the fitness value would be obtained as a benchmark with the best solution. If the solution obtained was still bad, then it was repeated or returned to the selection process to get the best solution. This process was also repeated 100 times to ensure accurate results.

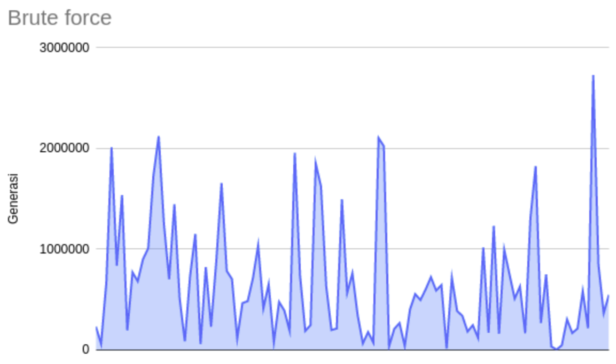

Figure 7. Result generation brute force

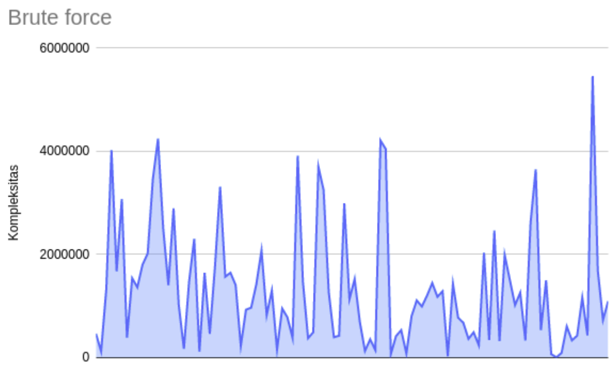

Figure 8. Result complexity brute force

With the GA, the average generation found was 153.52, and it took 0.4621 seconds to find the best solution and the total complexity was 1371.83. See Figure 9-11 for visualization.

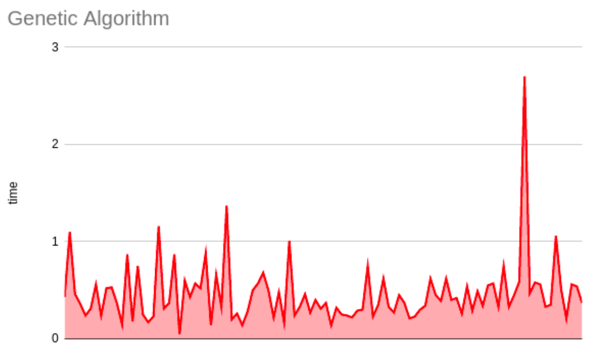

Figure 9. Result time genetic algorithm

3.3 Multiple Crossover Genetics

In the development of MCG which crossed with n quarter, n half and n random intersection points, the outcomes obtained were compared to take the best result which would be used as the cross value. In the crossover process, random values were avoided because they could reduce the fitness value later. By using MCG which changed the crossover process, it obtained an average generation value of 72.14 times, the average time needed for one process was 15.52 seconds, and a complexity value of 1085.55. Figure 12-14 shows the results.

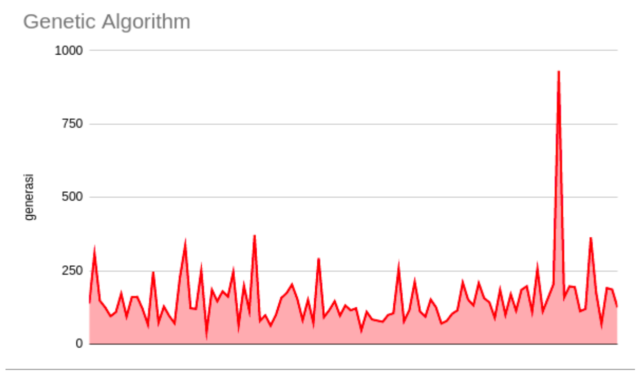

Figure 10. Result generation genetic algorithm

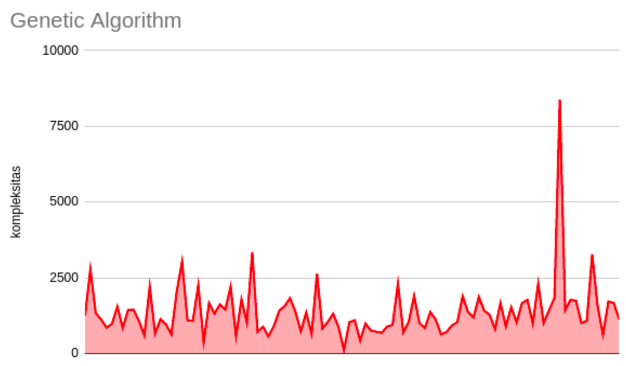

Figure 11. Result complexity genetic algorithm

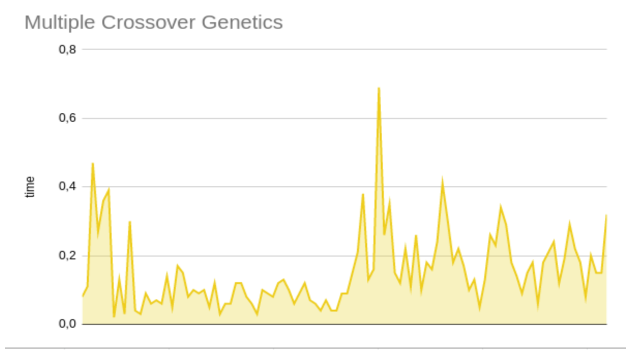

Figure 12. Result time Multiple Crossover Genetics

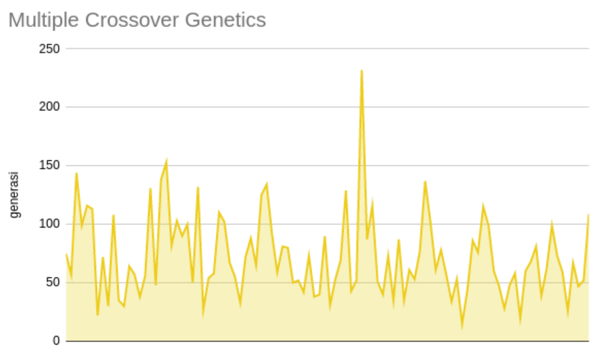

Figure 13. Result generation Multiple Crossover Genetics

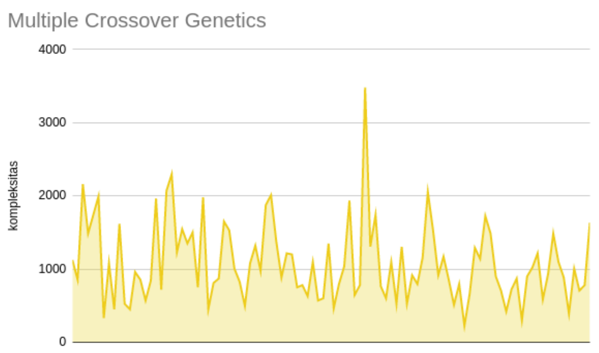

Figure 14. Result complexity Multiple Crossover Genetics

3.4 Genetics with increments value

Genetics with incremental values is a genetic algorithm process that when it reaches a certain fitness value, an incremental value system will be carried out by checking each existing value as is done in the brute force algorithm process. However, if the value matches, then it will move to the right. This process adopts validation valid value done using the brute force method. With this pattern, the number of iterations and the complexity of the method becomes faster in pin code search because the search is more structured.

A better value for this method is obtained because when the fitness value approaches the specified limit, the search is no longer carried out randomly. This allows a very low fitness value reduction. This statement is also proven in this research which found a solution search time of 0.0576 minutes, a generation value of 43.38, and a method complexity of 269.73.

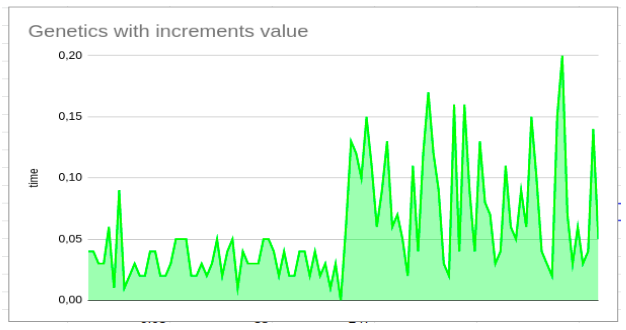

Figure 15. Result time Genetics with Increments Value

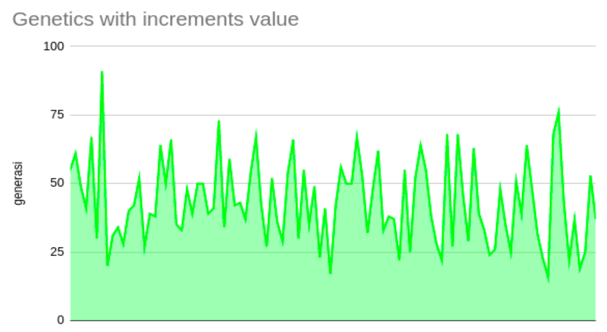

Figure 16. Result generation Genetics with increments value

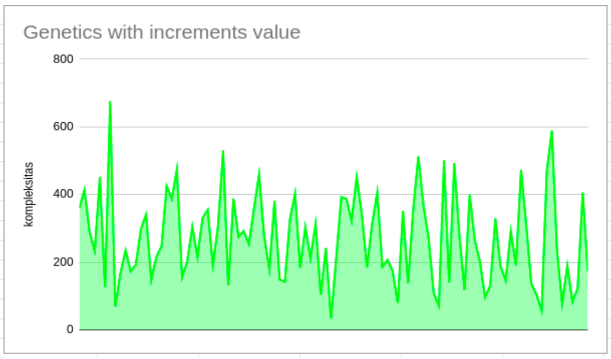

Figure 17. Result complexity genetics with increments value

3.5 *Independent Sample T Test*

In this study, the data obtained were then tested by the independent sample T Test which was carried out on each method. The tests carried out are divided into 3, namely generation comparison testing, time comparison testing and method complexity. The results of the tests carried out on the generation comparison test carried are shown in Table 2-4:

Table 2. Table Test sample T test time

Method 1	*Method 2*	*Equal variances assumed*	*Equal variances not assumed*
bf	ga	0,000	0,000
bf	mcg	0,000	0,000
bf	mwiv	0,000	0,000
ga	mcg	0,000	0,000
ga	mwiv	0,000	0,000
mcg	mwiv	0,000	0,000

Table 3. Table Test sample T-test generation

Method 1	*Method 2*	*Equal variances assumed*	*Equal variances not assumed*
bf	ga	0,000	0,000
bf	mcg	0,000	0,000
bf	mwiv	0,000	0,000
ga	mcg	0,000	0,000
ga	mwiv	0,000	0,000
mcg	mwiv	0,000	0,000

Table 4. Table Test sample T-test complexity

Method 1	*Method 2*	*Equal variances assumed*	*Equal variances not assumed*
bf	ga	0,002	0,003
bf	mcg	0,000	0,000
bf	mwiv	0,000	0,000
ga	mcg	0,009	0,009
ga	mwiv	0,000	0,000
mcg	mwiv	0,000	0,000

From the data, it can be concluded that there are differences in the average results between methods, but very slight differences occur in the genetic algorithm and the MCG at the stage of testing the complexity of the method. This happens because in this study only changes were made to the crossover process by dividing the crossover point into 3 parts. Although the genetic algorithm and MCG get the lowest score in the test sample testing in the calculation of complexity, this does not rule out the possibility of differences in results between the two.

4 CONCLUSION

Based on the results, it was concluded that the brute force algorithm is the algorithm that takes the longest time to find a solution. This happens because brute force performs processing on all eventualities. In this study, it was found that the generation was 737146.3469 times, and it took 1960.4296 seconds for each process. Meanwhile, the genetic algorithm gets a faster time than brute force. However, when the fitness value is close to 1 (100%), there is often a decrease, so the processing time increases. The average generation obtained is 107.72 times, and the time is 0.2924 seconds. The MCG algorithm in finding a solution is faster than the genetic algorithm because, at the time of crossover with several intersection points, it is possible to find a solution that is faster because it can minimize the decrease in fitness value. In research using MCG to get 67.1 generations and the time for one process is 0.1842 seconds. In the GWIV study, this process obtained the fastest time compared to other processes, namely 36.78 generation times and only required an average time of 0.0642 seconds. This study also obtained the highest time complexity score using the brute force method with an average value of 1345599.8. This happened because the search process was more structured and reduces the decrease in the fitness value obtained before. In the T-Test Sample Test, the methods were divided into generation results testing and yield time testing. In the results of this test, it was found that there is a significant average difference in each method.

AUTHOR'S CONTRIBUTION

Yudha Riwanto is the first author to conduct a literature review on previous research, data collection, idea research, testing, analysis, and implementation, while Muhammad Taufiq Nuruzzaman as the second author provides suggestions and input on the research concept. Shofwatul Uyun and Bambang Sugiantoro provided input in writing the draft.

COMPETING INTEREST

Complying with the publication ethics of this journal, Yudha Riwanto, Muhammad Taufiq Nuruzzaman, Shofwatul Uyun and Bambang Sugiantoro as the authors of this article declare that this article is free from conflict of interest (COI) or competing interest (CI).

ACKNOWLEDGMENT

Thank you to all the family and lecturers for their guidance and support, as well as all parties involved in this research until the completion of this research.

REFERENCES

[1] Beki Subaeki and Asep Muhammad Indra Purnama, "Optimalisasi Perbandingan Algoritma Brute Force dan Knuth-Morris-Pratt untuk Meningkatkan Kecepatan Pencarian Data pada Aplikasi Mobile Tentang Hewan Vertebrata," Techno Socio Ekonomika, vol. 10, pp. 224–333, 2017.

[2] V. Handrianus Pranatawijaya and P. Bagus Adidyana Anugrah Putra, "Implementasi Algoritma Genetika Pada Penjadwalan Program Profesional Jurusan Teknik Informatika Universitas Palangka Raya," vol. 5, pp. 90–98, 2019, doi: 10.22216/jsi.v5i2.4659.

[3] W. Supriana et al., "Implementasi Dua Model Crossover Pada Algoritma Genetika untuk Optimasi Penggunaan Ruang Perkuliahan," 167 Jurnal Resistor, vol. 4, no. 2, pp. 167–177, 2021, [Online]. Available: http://jurnal.stiki-indonesia.ac.id/index.php/jurnalresistor

[4] A. Mohammad, O. Saleh, and R. A. Abdeen, "Occurrences Algorithm for String Searching Based on Brute-force Algorithm," Journal of Computer Science, vol. 2, no. 1, pp. 82–85, Jan. 2006, doi: 10.3844/jcssp.2006.82.85.

[5] N. S. Sukmadinata, "Metode Penelitian Pendidikan," Bandung, 2016.

[6] M. Kumar, A. Professor, S. Kumar Singh, R. K. Dwivedi, and A. Professor, "A Comparative Study of Black Box Testing and White Box Testing Techniques," International Journal of Advance Research in Computer Science and Management Studies, vol. 3, no. 10, 2015, [Online]. Available: www.ijarcsms.com

[7] N. Golian, V. Golian, and I. Afanasieva, "Black and White-Box Unit Testing for Web Applications," Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies, no. 1 (7), pp. 79–83, Jul. 2022, doi: 10.20998/2079-0023.2022.01.13.

[8] Nurul Fuadi, Muh. Arif, And Sri Zelviani, "Pengaruh Kebisingan Terhadap Frekuensi Denyut Nadi dan Kelelahan Kerja Menggunakan Uji Statistik Spss pada Uji Paired Sampel T-Test," Jurnal Insitek, vol. 7, pp. 325–333, 2022.

[9] R. Imanuel and N. #1, "Studi Komparatif Algoritma Fisher Yates dengan Brute Force pada Permainan Kartu 24," 2020.

[10] G. H. Fandi Nainggolan et al., "Pencarian Berita Pada Web Portal Menggunakan Algoritma Brute Force String Matching," JIPI, vol. 6, pp. 1–10.

[11] Bayu Widia Santoso, Firdiansyah Sundawa, and Muhammad Azhari, "Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database," Sisfotek Global, vol. 6, pp. 1–8, 2016.

[12] H. A. Taha, Operations Research: An Introduction, 4th ed., vol. 40. United State of America: Preason Education, 2007.

[13] Z. Michalewicz., "Artificial Intelligence in Medicine Book reviews," 1997.

[14] Xin-She Yang, Nature-Inspired Optimization Algorithms. London: mara corner.

[15] Try Feby Ramadonna, Ade Sivia, and Ciksadan, "Perbandingan Algoritma Genetika dan TSP Untuk Optimalisasi Jaringan Akses Fiber To The Home," 2017.

[16] L. P. S. Ardiyani, "Perbandingan Algoritma Genetika dengan Algoritma Steepest Ascent Hill Climbing untuk Optimasi Penjadwalan Kuliah," Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI), vol. 11, no. 1, p. 63, Apr. 2022, doi: 10.23887/janapati.v11i1.43172.

[17] A. B. Hassanat, V. B. S. Prasath, M. A. Abbadi, S. A. Abu-Qdari, and H. Faris, "An improved Genetic Algorithm with a new initialization mechanism based on Regression techniques," Information (Switzerland), vol. 9, no. 7, Jul. 2018, doi: 10.3390/info9070167.

[18] P. Brusilovsky, "Hybrid Web Recommender Systems," 2007. [Online]. Available: http://www.google.com

[19] B. U. Tri Wahyo and A. Widya Anggriawan, "Sekolah Tinggi Manajemen Informatika dan Komputer ASIA Malang 6 Sistem Rekomendasi Paket Wisata Se-Malang Raya Menggunakan Metode Hybrid Content Based dan Collaborative," 2015.