1. [Initialize parameters such as population size (n), mutation: crossover probability, and number of iterations.
2. [Initial Population] Generate an initial population of n chromosomes.
3. [New Population] Repeats the following steps to create a new population.
   a. [Fitness] Calculates the fitness value of each chromosome in the population.
   b. [Selection] Selects the parent chromosome for reproduction.
   c. [Regeneration] Perform crossover or mutation process
   d. [Offspring evaluation] Placing a new offspring into the population if the fitness value of the offspring is greater than the fitness in the current population.
4. [Population switch] Using the new population for the next genetic process.
5. [Stop check] If the stop condition is reached, the genetic process ends.
6. [Recursion] If the condition in step 5 is not met, then go back to step 3.

B. *Hybrid Genetic Algorithm*

Hybridization is an effective way to improve the performance and efficiency of genetic algorithms. One form of hybridization that is often used is the combination of genetic algorithms with local search algorithms. Local search is the exchange of information carried by genes on child chromosomes. In a hybrid genetic algorithm, the main role of the genetic algorithm is to explore the search space to find the global optimum, while the local search algorithm uses the information gathered by the genetic algorithm to solve the local optimum [9].

The following are the steps for the hybrid genetic algorithm proposed by Wan and Birch [10]:

1. Define the fitness function and set the GA operators (such as population size, number of generations, selection method, mutation probability, crossover).
2. Randomly generate an initial population and make it the current parent population.
3. Evaluate the objective function for each individual in the initial population.
4. Generate the child population using GA operators (such as parent selection, crossover, and mutation).
5. Evaluate the fitness of the children.
6. Is the child the best among the other children and the best of the current parent population?
7. If not, do step 9. No local search occurs in this generation.
8. If yes, then perform local search on the best child chromosome. Find the best locally improved solution and replace the best offspring with it.
9. Decide which chromosome to include in the next population. This step is referred to as "replacement" where the chromosomes of the current parent population are "replaced" by a new population consisting of the chromosomes of the child and/or parent populations.
10. If the number of stopping conditions is met, then the looping process is stopped. Otherwise, proceed to step 3.

III. METHODS

A. *Research Data*

Data was obtained from the Magelang District Education Office in the form of teacher data and elementary school data. Teacher data consists of 636 data consisting of id_teacher, name, and coordinates of the house where they live. Meanwhile, school data includes id_school, school name and coordinates. It is assumed that each school has 6 study groups. From the data above, the distance between the teacher's house and school is calculated using the Haversine formula. [6]

B. *Chromosome Representation*

The chromosome expresses a solution, namely how to place teachers in schools so that the total distance is minimal. In order for the chromosome to be subjected to the genetic algorithm operator, in this case the chromosome is expressed in the form of an array of 636 length where the contents are in the form of characters representing the identity of the teacher. while the array represents school classes, where each school consists of 6 classes. This can be described as shown in Fig. 1

S1	S2	S3	S4	S5	S6				Sm
T1	T2	T3	T4	T5	T6	…	…	…	Tn

Figure 1. Representation of chromosome

C. *Fitness Function*

The goodness of a chromosome or solution is determined by calculating its fitness function. In this case the fitness function is the total distance from each teacher to their respective schools. This fitness function can be calculated because all distances between teachers to schools are known using the Haversine formula.

D. *Selection Method*

The process of selecting chromosomes for crossover or mutation is done using the *roulette wheel method*, where the probability of a chromosome being selected for crossover or mutation is proportional to its own fitness value. The higher the fitness score, the higher the probability of selection. [11]

E. *Crossover*

In this research, the crossovering method used is single-point crossover. In the single point crossover method (SPX), there is only one crossing point with a randomly generated position. The crossing point is determined by generating a random number in the range of 1 to the length of the chromosome. For example, as shown in Fig. 2., there are two chromosomes, parent 1 and parent 2 with a chromosome length of 6, and the random number as a crossing point generated is 3, so genes 1 to 3 will be cut with

4 to 6. Furthermore, genes 1 to 3 in the first chromosome will be crossed with genes 4 to 6 in the second chromosome, to become the children and vice versa. [12], [13]

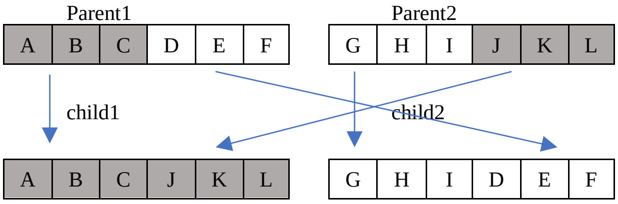


Figure 2. Single Point X-Over

F. *Mutation*

In this research, the mutation operator used is swap mutation as seen on Fig 3. The first step is to randomly select two gene positions in the parent chromosome. Then the two genes are swapped to become new generation. [14]

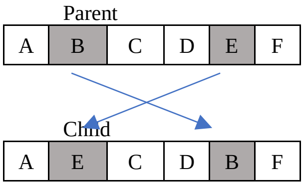


Figure 3. Swap Mutation

G. *Local Search*

The exploration capability of genetic algorithms can be further enhanced by utilizing local search. Local search is performed by changing the genetic makeup of the child chromosomes without affecting the quality of the previous chromosomes. Local search is performed to find better solutions in the neighbourhood of offspring, and increase population diversity, especially when the resulting offspring are similar to the population. Basically, local search is expected to provide another better solution around the local optima given by the genetic algorithm. The local search operator used here is the insert operator. This insert local search procedure is performed by randomly selecting two gene positions, namely positions 1 and 2 in the parent chromosome, where position 1 cannot be the same as position 2, then insert the gene at position 2 in position 1 and shift to the right all genes to the right of the insertion location, as shown in Fig.4. [15]

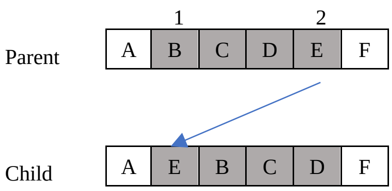


Figure 4. Insertion

IV. Result

In this study, the program was created using the Python programming language, then the program was run many times by varying the number of chromosomes in the initial population, the mutation/crossover probability (pm) while the number of iterations was limited to a maximum of 1,000 iterations. The input of this program is data obtained from the Magelang District Education Office, in the form of distance data between schools and teachers, where the number of schools is 106 while the number of teachers is 636 people. From the experiments conducted, the following results were obtained as shown in Table 1 and Table 2:

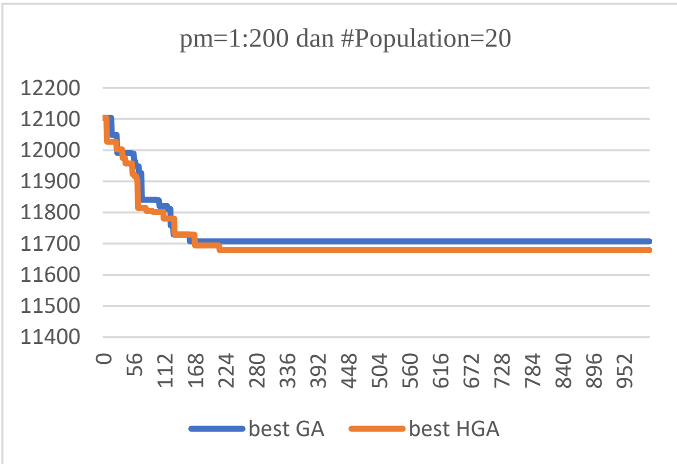
Table 1. Result of Pure GA

#Population	pm	Total Distance
20	1:100	11,802.2091 km
20	1:200	11,707.2259 km
40	1:100	11,563.0078 km
40	1:200	11,563.8300 km

Table 1. Result of Hybrid GA

#Population	pm	Total Distance
20	1:100	11,654.55845 km
20	1:200	11,679.21755 km
40	1:100	11,292.26081 km
40	1:200	11,286.90802 km

The graph below (Fig. 4) shows the change in fitness value of the pure genetic algorithm (GA) and the hybrid genetic algorithm (HGA), where it is visually apparent that the hybrid genetic algorithm is always better than the pure genetic algorithm.



V. Discussion

From the experiments conducted, it can be seen that changing the mutation/crossover probability has no effect on the results obtained. On the other hand, increasing the number of initial populations has a significant effect on the results obtained, namely better or shorter results, both for pure genetic algorithms and for hybrid genetic algorithms. When these two algorithms are compared, overall for each experimental parameter, it turns out that the hybrid genetic algorithm is able to produce results that are always better than the pure genetic algorithm. This shows that hybridization of genetic algorithms can indeed produce new offspring that are better than pure genetic algorithms. This is because the hybrid genetic algorithm can produce a variety of new offspring from the results of the pure genetic algorithm, and the probability of getting better offspring can arise from there. From the graph of experimental results, it can be seen that for pm = 1:200 and population of 20, starting from the 165th iteration, the results of the hybrid genetic algorithm always outperform the results of the pure genetic algorithm. Genetic algorithm hybridization cannot only use the insertion method, there are other methods that can be used such as the swap or scramble method or others. These other hybrid methods are worth trying as they also give hope for more varied offspring.

VI. Conclusion

The findings of this study can be summarized as follows:

1. Hybrid Genetic Algorithm using Local Search shows better performance than Pure Genetic Algorithm.
2. The shortest total distance produced by the pure Genetic Algorithm is 11,562.83011 km while the Hybrid Genetic Algorithm produces 11,286.90802 km.
3. At the 165th iteration the Hybrid Genetic Algorithm has started to outperform the Pure Genetic Algorithm, so there is no need to do large iterations.

References

[1] H. Sriwindono, P.H.P. Rosa, A.M. Polina, and R.A. Nugroho, “The Model of Elementary Teacher School Placement in Magelang District by Using Genetic Algorithm,” *In Proc. CSAI’17*, 2017, p. 143.

[2] P.H.P. Rosa, H. Sriwindono, R.A. Nugroho, A.M. Polina, and K. Pinaryanto, “Comparison of Crossover and Mutation Operators to Solve Teachers Placement Problem by Using Genetic Algorithm,” *Journal of Physics: Conference Series*, vol. 1566, no. 1, pp. 012021, May. 2020.

[3] H. Sriwindono, P.H.P. Rosa, and K. Pinaryanto, “The Teacher Placement using K-Means Clustering and Genetic Algorithm,” *In Proc. CORIS’22*, 2022, vol. 4, No. 1. p.43-51.

[4] M. Gen, and R. Cheng, *Genetic Algorithm and Engineering Design*, Toronto, Canada: John Wiley & Sons, Inc. 1997.

[5] Y. Yun, C. Moon, and D. Kim, “Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems, ”*Computers & Industrial Engineering*, vol. 56, Issue 3, pages 821-838, 2009.

[6] G.V. Brummelen, *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*, Princeton, USA: Princeton University Press, 2013.

[7] J.H. Holland, *Adaptation in Natural and Artificial Systems*, Cambridge, USA: MIT Press, 1992.

[8] D.E. Goldberg. *Genetic Algorithms: in Search, Optimization, and Machine Learning*, Boston, USA: Addison Wesley Professional, 1989.

[9] T. A. El-Mihoub, A.A. Hopgood, L. Nolle, and A. Battersby. “Hybrid genetic algorithms: A review,” *Engineering Letter*, 13:2, 2006.

[10] W. Wan and J.B. Birch, “An Improved Hybrid Genetic Algorithm with a New Local Search Procedure,” *Journal of Applied Mathematics*, vol. 2013, Article ID103591, 10 pages, 2013.

[11] S.N. Sivanandam and S.N. Deepa, *Principles of soft computing*, Wiley India Pvt. Limited, 2007.

[12] S.N. Sivanandam and S.N. Deepa, *Introduction to Genetic Algorithms*, Heidelberg, Germany: Springer Berlin, 2007.

[13] A.E. Eiben and J.E. Smith, *Introduction to evolutionary computing*, Springer-Verlag Berlin Heidelberg, 2015.

[14] Obitko, M. Introduction to Genetic Algorithms; Czech Technical University: Prague, Czech Republic, 1998.

[15] L. Gharsalli and Y. Guerin. Composite structures optimization using hybrid genetic algorithm for space applications. *In Proc. ICCMSE-2019.* Volume 2186, Issue 1. 2019.