Hybridization of Genetic Algorithm and Artificial Immune System for Assignment Problem

La Ode Muhammad Farhan¹, Zainudin Zukhri²
Department of Informatics, Islamic University of Indonesia

Email: ¹ldmuhfarhanf@gmail.com, ²zainudin@uii.ac.id

Abstract. Genetic Algorithms (GA) have proven its reliability to obtain near optimum for complex problems. However, to optimize solutions and prevent premature convergence, operators on the GA were highly dependent on large population sizes, causing the computational speed to be slower than newer algorithms. Therefore, in this study, we propose a way to cause a significant improvement in GA for a higher quality of solutions and lower computing costs yet easy to implement. Hence, we analyzed the clonal selection algorithm from AIS to inspire genetic processes in GA. We then compared the performance of the proposed algorithm called HGA as with the GA to solve the assignment problem. Simulation results showed that HGA performed better in case of preventing on being stuck with the local optima as it accurately obtains an average of the optimum solution up to 100% for every given dataset while also reducing computational costs as it has less generation.

1. Introduction
Genetic Algorithm (GA) is one of the evolutionary computation methods that mostly known and used in a wide range of problem areas. The application of GA has been tested in various fields, one of which is the assignment problem using the Traveling Salesman Problem (TSP) approach as in [1]. In that research, the use of GA operators has proven to solve assignment problems and provide optimal solutions. Even though it can find effective global optimum values, GA cannot guarantee that the solutions provided are not stuck at the local optimum. This can occur depending on the scale of the problem at hand. In this case, GA and other similar algorithms are very dependent on population size so that they are vulnerable to the problem of premature convergence that triggers stagnation, causes the search process to stop and leads to wasteful computing. Although these problems can be overcome with large populations, computing time will increase dramatically. Coupled with the structural complexity and complex genetic operators of GA, this can cause computational speed becomes slower compared to newer algorithms [2].

Artificial Immune System (AIS) is a bio-inspired computing area, but it is relatively new compared to GA. AIS adopts the principle of immunity or the body's defense system from attacks by harmful microorganisms such as viruses and bacteria (pathogens). To overcome the pathogen attack, AIS carries out a learning process or an adaptation of the attacking pathogen so that an optimal immune response is achieved. In its application, AIS shows that the algorithm used can be applied to solve complex problems that can be done by GA in general as in [3] and [7]. AIS and GA have a similar algorithm. This is inseparable from the principle of evolution that exists in each of these algorithms. However, the most striking difference between AIS and GA is the absence of crossover operators on
the AIS because normally cells only need to duplicate themselves to increase diversity in the population. Even so, AIS is still able to produce optimal solutions and reduce computing costs. With the assumption that AIS can solve problems that can also be achieved by GA, this algorithm can be the key to solving the complexity of algorithms that GA has in general.

Several studies have been carried out to improve GA performance, including GA hybridization as in [5] and [6], and simplification of GA as in [2] and [7] but few have tried to hybridize GA and AIS. In a recent study, simplified GA was inspired by AIS to solve MST problems by removing the crossover operator [7]. The results of the study stated that the proposed method can solve the problem as GA did even without a crossover and discussed that the research needs to be done by solving other complex problems. Therefore, in this study, we conduct a hybridization of GA and AIS to solve the assignment problem. We then carried out the analysis with a comparative study between three algorithms. The first algorithm is the real coded genetic algorithm (RGA) which was used as a benchmark for comparison against the proposed algorithms. The second algorithm that we proposed is the simplification of genetic algorithms by ignoring the crossover operator (SGA). Meanwhile, the third algorithm that we proposed is the hybridization of GA and AIS (HGA) with the ideas to extract some prominent feats from the clonal selection algorithm to the GA. This comparative study is carried out by comparing the computational time and the quality of the solutions obtained from each algorithm. The data that we tested is in the form of an assignment matrix based on 10 data sets that have tested before with the assumption that many resources and activities are the same for each data set. The results of this trial are expected to contribute to the application of bio-inspired algorithms, especially GA and AIS so that the development areas in this field which cover the theoretical scope and application of theories to overcome various optimization problems can be more advanced.

2. Literature Review

In this section, we presented several works related to this study that has tried to improve GA performance by hybridizing GA with AIS and also discussed the clonal selection algorithm as one of the AIS methods that inspire the proposed method which is HGA. We then explored the nature of AIS particularly the clonal selection and analyze the prominent feats of its algorithm.

2.1. Related Works

The research on the hybridization of GA-AIS was carried out by Ali et al. as in [8]. The performances of the Hybrid method is tested based on eight popular mathematical benchmark functions. The Hybrid includes two processes; firstly running the AIS to earn the desired solution. After that, GA is run with the initial population taken from the solution from the last generation of AIS with the purpose that Hybrid makes GA enters the stage of standard solutions rapidly and accurately higher compared with GA initialized population at random. Moreover, the results after 20 times trials based on the minimum value of objective function and time for each method indicate that Hybrid GA-AIS is performs better compared with GA and AIS separately as it obtained the optimum solution with acceptable time. Although the proposed method has better performance, it is not simple to apply given the complexity of the algorithm and the number of parameters that need to be determined.

In another study, Zukhri [7] proposed a simplified genetic algorithm (GA\textsubscript{0}) to solve the MST problem by adopting AIS operator. In this case, the complexity of real coded GA (GA\textsubscript{1}) is simplified by setting aside crossover operators based on the similarity between the operators of GA and AIS. Moreover, in AIS, the cloning operator can be said to belong to the selection operator so that there are only two operators, namely selection and mutation. Hence, the difference in GA leaves only the crossover operator. Assuming that AIS can solve complex problems even without crossover operators, this certainly can be applied to GA. By consider this analogy, a trial to compare the performance of GA\textsubscript{0} and GA\textsubscript{1} were performed on 3 data sets with consecutive nodes of 6, 8, and 13. The trial took random parameter values that normally used with the condition that both algorithms were tested using the same parameter values. From the results of the two-sample t-test, it can be said that there is no difference in the quality of the solution produced between the two algorithms so it can be concluded
that GA\textsubscript{0} considered being well implemented in other complex problems and also brings convenience when determining parameters. Thus, in this study, we tested this approach and also implemented it on the HGA that we are proposing to solve complex assignment problems.

2.2. Clonal selection algorithm
There are three main theories on the immune system that underlie the birth of the field of Artificial Immune Systems, namely the Immune Network Theory, the Negative Selection theory, and the Clonal Selection theory [9]. With the similar structure of the algorithm between GA and clonal selection algorithm, the clonal selection is easy to implement and uses for hybridization with GA.

2.2.1. Biological explanation. Clone selection is a term used to describe the basic response of the adaptive immune system to antigenic stimuli. The idea of this theory states that only cells that can recognize antigens with high affinity can reproduce by multiplying themselves (proliferation) while other cells that cannot recognize antigens will not be preserved. In this case, they will destroy (apoptosis). Furthermore, after proliferation, the formed cells will be subject to mutations with a fairly high probability (hypermutation). Cells that have an increased affinity will be differentiated to form an immune response while cells that have decreased affinity will be destroyed [10]. Thus, unlike the simple genetic algorithm which generally avoids high mutation rates, the mutation in clone selection is the main operators that forming population. These characteristics will be further explored in the proposed HGA. Meanwhile, the clonal selection process, in general, can be seen in Figure 1 below.

2.2.2. Analysis of algorithm. In order to solve global optimization problems, the hybrid algorithm should strike a balance in exploration (global search) and exploitation (local search) [11]. Several studies support this approach and apply the GA hybridization method with local search [6, 8, 12]. Although GA can find optimum global, in many occasions, it has a proclivity to being stuck on local optima. Several reasons might explain this behavior such as weak initial population, highly dependent on population size, and lack of local search. Meanwhile, the clonal selection in AIS has local search attributes that can be used in GA to balance the global search. Therefore, we are using this capability of AIS to inspire GA in certain parts of genetic operators.

In the clonal selection algorithm, there are three main operators used, namely selection, cloning, and mutation. The immune response obtained in the form of a group of antibodies represents a solution. Meanwhile, the antigen acts as a fitness function so that the strength of the receptor binding (affinity) to the antigen represents the quality of the solution. Furthermore, clonal selection has the capability of local search, it can be seen in the cloning process (clone pool). In this process, the algorithm exploits its antibodies whereas the best clone from a particular antibody selected from the
mutated clone pool and had a chance to replace the current antibody if the affinity is higher. The clone pool selection has a similar function in the mutation of GA. The difference is GA uses the low probability of mutation and no selection thus allowing it to move further into other search areas while keep maintaining diversity at an acceptable rate. However, mutations in GA often lead to unfavorable search points, causing slow convergence. Therefore, in this study, we modified the mutation in GA to mimic the local search as in clone pool selection. This approach is used on HGA as shown in Figure 4.

3. Methods

To complete this research various stages are carried out as follows:

- Conduct a comparative study between GA and AIS.
- Model RGA, SGA, and HGA algorithms to solve assignment problems.
- Collect datasets in the form of assignment matrices of varying sizes.
- Implement model design in the form of simulation software.
- Conduct experimental testing of simulation software that has been developed to given datasets and performs results analysis.
- Conduct a comparative study of the effect of the proposed model on the performance of GA.

3.1. RGA for solving assignment problem

Assignment problems are complex problems that can be used in experiments to compare performance between optimization algorithms. The type of assignment problem taken in this study is a one-to-one assignment problem. In this type, there are n resources and n activities. To solve this problem, a real coded genetic algorithm (RGA) is modeled from chromosome representation to genetic operators. The RGA model is used as a basis for modeling the proposed algorithm such as SGA and HGA.

3.1.1. Chromosome Representation. To solve the one-to-one assignment problem, this study proposes a combinatorial chromosome representation in the form of positive integers whose values range from 1 to n (the total number of genes). Each gene (g_i) on a chromosome represents a resource while the gene index represents an activity. The chromosome representation can be seen in Figure 2 below.

```
resources-i: g_1 g_2 g_3 g_4 g_5 ... g_n
activities-j:  1  2  3  4  5  ...  n
```

**Figure 2. Proposed chromosome representation**

3.1.2. Fitness function. The objective function is used to calculate the total cost obtained from all assignments of resources to activities. If the costs required by the resources in i-th gene (g_i) to carry out the i-th activity are represented as C_{ig_i}, then the objective function to determine the total cost can be formulated as in equation (1) below.

\[ f(x) = \sum_{i=1}^{n} C_{ig_i}(x) \]  

The fitness function can be determined based on the optimization problem. If the assignment problem uses a function for maximization, then the fitness function used can be defined directly from the objective function as in equation (1). Whereas if the assignment problem uses a function for minimization, the fitness function described as eval(x) used is the opposite of the objective function as shown in equation (2) below.
3.1.3. RGA operators. In this study, any method will be chosen to determine the effect of simplification and modification made on SGA and HGA operators. However, the chosen method is the same for all three algorithms tested. The methods to be used are roulette wheel selection combined with elitism, order based crossover using absolute replacement, and order based mutation.

3.2. SGA
In this study, a simplified genetic algorithm (SGA) is proposed and tested for its performances. SGA previously proposed and tested by Zukhri to solve the minimum spanning tree problem (MST) [6]. This algorithm is tested again for a different problem, namely the assignment problem. The algorithm is modeled by eliminating the crossover operators, so SGA only uses the selection and mutation operators to manipulate the population.

3.3. HGA
HGA is modeled by removing the crossover operators and mimicking local search as in clone pool selection. Moreover, a fairly high probability of mutation is applied in HGA. Since local search is applied to mutation, more selection is used causing vast convergence. To balance the global search with local search, reinitiate population is activated when the average fitness of the population is below the certain threshold. Therefore, the entire operation on the HGA can be seen in Figure 3 below.

\[
eval(x) = \frac{1}{f(x)} = \frac{1}{\sum_{i=1}^{n} C_{ig_i}(x)}
\]

\[\text{(2)}\]

\[\text{3.1.3. RGA operators.} \]

\text{In this study, any method will be chosen to determine the effect of simplification and modification made on SGA and HGA operators. However, the chosen method is the same for all three algorithms tested. The methods to be used are roulette wheel selection combined with elitism, order based crossover using absolute replacement, and order based mutation.}

\text{3.2. SGA}

\text{In this study, a simplified genetic algorithm (SGA) is proposed and tested for its performances. SGA previously proposed and tested by Zukhri to solve the minimum spanning tree problem (MST) [6]. This algorithm is tested again for a different problem, namely the assignment problem. The algorithm is modeled by eliminating the crossover operators, so SGA only uses the selection and mutation operators to manipulate the population.}

\text{3.3. HGA}

\text{HGA is modeled by removing the crossover operators and mimicking local search as in clone pool selection. Moreover, a fairly high probability of mutation is applied in HGA. Since local search is applied to mutation, more selection is used causing vast convergence. To balance the global search with local search, reinitiate population is activated when the average fitness of the population is below the certain threshold. Therefore, the entire operation on the HGA can be seen in Figure 3 below.}

\text{Pseudocode of HGA}

\begin{verbatim}
1. Encode solution space
2. Set pop_size, chrom_size, max_gen, reinitiate threshold, mutation_rate, Gen = 0
3. Initialize population P randomly
4. For each individual i ∈ P: calculate fitness (i)
5. While Gen < max_gen
   * do selection
   * do mutation
   * Test: Test whether the average fitness of population P below the reinitiate threshold. If so, reinitiate P. If not, continue using current P.
   * do replacement
   * save current best solution
6. Return the best solution
\end{verbatim}

\text{Figure 3. Pseudocode of HGA}

\text{Henceforth, local searches based on clone pool selection applied in genetic operator. The algorithm is added to mutation after the clone is mutated. The pseudocode of an additional algorithm in HGA mutation can be seen in Figure 4 below.}
4. Test Cases
The analysis to compare the three algorithms starts with determining the data sets and parameters to be used. Testing is done to get results in the form of solution quality and computational costs. The testing is done using java programming language. Meanwhile, all the executions were progressed using a computer with an Intel (R) HD Core ™ i5-5200U CPU that works to a frequency of clock of 2.20 GHz, 4 GB of RAM Memory with Windows 8 operating system.

4.1. Data sets
The data sets that will be used in this study are one-to-one assignment problem data sets with a matrix size from 6x6 to 15x15. The 6x6 up to 10x10 matrix are primary data sets that have previously been tested in various studies. While 11x11 up to 15x15 matrix are engineered data sets (dummy) in which the values are randomly generated from 40 to 60.

Table 1. Special parameters of algorithms

| Method | Parameters |
|--------|------------|
|        | mr | cr | rθ |
| RGA    | 0.05 | 0.5 | - |
| SGA    | 0.05 | -  | - |
| HGA    | 0.75 | -  | 0.01 |

Table 2. General parameters of algorithms

| Matrix size | Opt. solution (99.5 %) | Population size | Maximum of generation | Number of elites (RGA & SGA) | Number of elites (HGA) | Number of trial |
|-------------|------------------------|-----------------|-----------------------|-----------------------------|------------------------|-----------------|
| 6x6         | 105,21                 | 20              | 1000                  | 4                           | 1                      | 349             |
| 7x7         | 95,11                  | 20              | 2000                  | 4                           | 1                      | 349             |
| 8x8         | 33,06                  | 25              | 3000                  | 4                           | 1                      | 349             |
| 9x9         | 625,32                 | 40              | 4000                  | 6                           | 1                      | 349             |
| 10x10       | 387,45                 | 60              | 5000                  | 8                           | 1                      | 349             |
| 11x11       | 471,84                 | 75              | 6000                  | 10                          | 1                      | 349             |
| 12x12       | 519,87                 | 80              | 7000                  | 12                          | 1                      | 349             |
| 13x13       | 543,08                 | 100             | 8000                  | 12                          | 1                      | 349             |
| 14x14       | 589,17                 | 150             | 9000                  | 15                          | 1                      | 349             |
| 15x15       | 616,32                 | 200             | 10000                 | 15                          | 1                      | 349             |

4.2. Parameters of each algorithm
In this study, the value of each parameter is taken arbitrarily. There are two types of parameters that will be used, namely special parameters and general parameters. In the special parameters as shown in Table 1, each algorithm has its value. In this case, the operator used determines what parameters are...
used. Whereas in the general parameters as shown in Table 2, all algorithms assign the same value but will change as the size of the assignment matrix increases.

4.3. Experiment and results
In this study, each assignment matrix will be subjected to 349 trials for performances testing. The number of trials is based on the Isaac and Michael Tables. First, the average computational time will be measured and then normalized as in equation (3) whilst given 1000 ms as the estimated time to get the optimum value.

$$\text{Normalized exec. time} = \frac{\text{Average execution time}}{1000} \times 100\%$$

Second, the average quality of the solution will be calculated based on the maximum generation that has been set. Furthermore, the average quality of the solution for each assignment matrix will be measured for its accuracy to the global optimum value as in equation (4) below.

$$\text{Accuracy of the solution} = \frac{\text{Max. solution - Solution}}{\text{Max. solution - Min. solution}} \times 100\%$$

The results of experiments to measure computational time and the quality of solutions can be seen as in Table 3 below.

| Matrix size | Average accuracy of the solution (%) | Average normalized execution time per 1000 ms (%) |
|-------------|--------------------------------------|-----------------------------------------------|
|             | RGA   | SGA   | HGA   | RGA   | SGA   | HGA   |
| 6x6         | 100,00| 100,00| 100,00| 0,77  | 0,60  | 0,42  |
| 7x7         | 100,00| 100,00| 100,00| 0,71  | 0,61  | 0,49  |
| 8x8         | 100,00| 100,00| 100,00| 0,98  | 0,82  | 0,55  |
| 9x9         | 100,00| 100,00| 100,00| 0,86  | 0,72  | 0,58  |
| 10x10       | 100,00| 100,00| 100,00| 3,78  | 3,07  | 1,41  |
| 11x11       | 100,00| 100,00| 100,00| 7,54  | 5,82  | 1,90  |
| 12x12       | 99,80 | 99,78 | 100,00| 49,95 | 28,15 | 3,15  |
| 13x13       | 100,00| 100,00| 100,00| 13,72 | 10,29 | 1,44  |
| 14x14       | 99,96 | 99,86 | 100,00| 33,72 | 25,92 | 2,43  |
| 15x15       | 99,92 | 99,85 | 100,00| 38,34 | 36,35 | 1,97  |

4.4. Comparative analysis
After the experimental results are obtained, a statistical test of two-sample t-tests is then performed to determine the effect of the removal of the crossing and hybridization operators on the quality of the RGA solution. In this case, given a hypothesis (H0) based on statistical data, there is an assumption that the quality of the solution produced by SGA and HGA is the same as the quality of the solution produced by RGA. Meanwhile, the samples to be compared with two-sample t-tests are the 12x12, 14x14, and 15x15 matrices. This is occurred because the other data samples have obtained an average solution quality of up to 100%. The results based on the H0 can be seen in Table 4 below.
| Method Compared | Dataset | Two sample t test (α = 0.01) | Two sample t test results |
|----------------|--------|-----------------------------|--------------------------|
|                |        | Statistical t value | 2-tailed critical t value |                      |
| SGA and RGA    | 12x12  | 0.099845             | 2.582912                 | H0: µ1 = µ2 accepted |
|                | 14x14  | 3.055085             | 2.582912                 | H0: µ1 ≠ µ2 rejected  |
|                | 15x15  | 5.112955             | 2.582912                 | H0: µ1 ≠ µ2 rejected  |
| HGA and RGA    | 12x12  | -8.49995             | 2.582912                 | H0: µ1 ≠ µ2 rejected  |
|                | 14x14  | -2.66886             | 2.582912                 | H0: µ1 ≠ µ2 rejected  |
|                | 15x15  | -9.67696             | 2.582912                 | H0: µ1 ≠ µ2 rejected  |

5. Discussions
The results of experiments and analysis of the three algorithms namely RGA, SGA, and HGA can picture the performance of each algorithm in solving a variety of complex problems, especially the assignment problem that was tested in this study. From the analysis result of the quality of the solution, there is no significant difference between RGA, SGA, and HGA over 1000 generations although HGA provides the best average solution quality. Furthermore, based on the results of the two-sample t-test conducted to prove the H0 hypothesis, it can be assumed that for relatively small-sized datasets there are no significant differences whereas for relatively large-sized datasets it is estimated that there will be differences in the quality of the resulting solutions. Henceforth, conducting test at larger data sizes with greater complexity of problems is also necessary.

In another experiment, HGA provides a significant reduction in terms of time complexity. In this test, an estimated maximum of 1 second is given to find the global optimum solution. As a result, HGA obtained the lowest percentage in approaching 1 second compared to the other two algorithms, followed by SGA and then RGA. In this case, HGA can reduce the complexity of RGA time significantly because it has the lowest percentage of time usage and also the lowest average optimum generation, while SGA can also reduce the time complexity of RGA even though the average optimum generation owned by SGA is greater than RGA.

The excellent performance of HGA depends on the mutation process where the clone pool selection is added to the mutation operator to mimic local search. Also, mutation on HGA used fairly high probability on mutation rate. Therefore, it can be assumed that the higher the mutation probability, the faster the convergence will occur which will affect the decrease in time complexity. For further research, a simplification can also be made on the HGA by eliminating the selection operator so that only mutation remains.

6. Conclusions
The Hybridization of Genetic Algorithm and Artificial Immune System can be effectively carried out and successfully tested to solve complex problems such as one-to-one assignment problems. The solution obtained for all datasets with matrix sizes ranges from 6x6 to 15x15 can reach global optimum values. Meanwhile, GA-AIS hybridization (HGA) can also simplify the process of RGA while providing a fairly high increase in performance compared to the RGA or SGA methods. HGA can significantly reduce the time complexity, has the highest quality of the solution and has the lowest optimum generation. Also, HGA is easy to implement because it doesn’t need the crossing operator and the whole process of AIS to hybridize.

References
[1] Zukhari Z 2004 Penyelesaian Masalah Penugasan dengan Algoritme Genetika Seminar Nasional Aplikasi Teknologi Informasi pp 51-8
[2] Ngamtawee R and Wardkein P 2014 Simplify and Improve RGA for Parameter Optimizations Adv. in Electrical and Computer Eng. vol 14 no 4 pp 55-64
[3] Coello C C, Rivera D C and Cortes N C 2003 Use of an Artificial Immune System for Job Scheduling Int. Conf. on Artificial Immune Systems

[4] Idris I 2012 Model and Algorithm in Artificial Immune System for Spam Detection Int. J. of Artificial Intelligence & Applications vol 3 no 1 pp 83-94

[5] Ahmed Z and Habeebvulla M 2014 A Hybrid Genetic Algorithm for the Quadratic Assignment Problem Proc. of the 2nd Int. Conf. on Emerging Research on Computing, Information, Communication and Applications (Amsterdam: Elsevier) pp 916-22

[6] Ariani A K and Wayan F M 2017 Hibridisasi Algoritme Genetika dan Simulated Annealing untuk Optimasi Multi-Trip Vehicle Routing Problem with Time Windows Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer vol 1 no 12 pp 1608-15

[7] Zulkri Z 2017 Penyelesaian Masalah Pohon Rentang Minimum Dengan Algoritme Genetika yang Disederhanakan Suatu Adopsi Operator Dari Sistem Kekebalan Buatan Prosiding Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri

[8] Ali M O, Koh S P, Chong K H and Yap D F 2010 Hybrid AIS-GA based on mathematical test function Proc. of 2010 IEEE Student Conf. on Research and Development pp 256-61

[9] Purbasari A and Supriana I 2011 Clonal Selection Algorithm: Bio Inspired Algorithms Sebagai Solusi Persoalan Kompleks Konferensi Nasional Sistem dan Informatika pp 329-34

[10] Timmis J, Hone A, Stibor T and Clark E 2008 Theoretical Advances in Artificial Immune Systems Theoretical Computer Sci. (Amsterdam: Elsevier) pp 11-32

[11] Anu et al 2013 Hybridizing GA with Hill Climbing in Replacement Operator Int. J. of Adv. Research in Computer Sci. and Software Eng. vol 3 issue 12 pp 270-8

[12] Rahmi A, Mahmudy W F and Anam S 2017 Hibridisasi Algen dan Variabel Neighborhood Search Pada Optimasi Biaya Distribusi Jurnal Teknologi Informasi dan Ilmu Komputer vol 4 no 2 pp 87-96

[13] Hongtao L and Fengju K 2016 Adaptive Chaos Parallel Clonal Selection Algorithm for Objective Optimization in WTA Application Optik (Amsterdam: Elsevier) pp 3459-65