# Development of Artificial Immune System in Multi-Objective Vehicle Routing Problem with Time Windows

Iris Martin[1] and Eric Wibisono[2(✉)]

[1, 2] Dept. of Industrial Engineering, University of Surabaya, Surabaya 60293, Indonesia
ewibisono@staff.ubaya.ac.id

**Abstract.** Setting logistics routes and product distribution in everyday problems, such as delivery of fresh products, requires an algorithm that can produce decisions in a short time. This type of problem belongs to a methodology popularly known as the vehicle routing problem (VRP). VRP is NP-Hard, and its complexity increases with additional settings such as time windows and multiple objectives (MOVRPTW). One popular metaheuristic for MOVRPTW is genetic algorithm, but the literature suggests that the algorithm's running time is usually too long, making it prohibitive for daily logistics applications. In this paper, we proposed a modified Artificial Immune System (AIS) for MOVRPTW by hybridizing it with chromosome splitting procedure called Split and nine-step local search mutation. The objective functions are minimum total distance and minimum number of vehicles. Based on the experimental results against Solomon data set c104, although the proposed algorithm still cannot beat the best-known solution, it is able to find solutions in a very short computation time under one minute in all scenarios.

**Keywords:** Artificial Immune System, Multi-Objective, Vehicle Routing Problem, Time Windows.

## 1 Introduction

The Vehicle Routing Problem (VRP) is a methodology widely applied for setting logistics routes. Its applications cover strategic themes such as the construction of maritime shipping network as well as tactical decisions such as daily distribution of consumer products. VRP is known to be NP-Hard, therefore additional settings will further increase its complexity. Possible additional settings include time windows and multiple objectives. Under this class, the model is called multi-objective vehicle routing problem with time windows or abbreviated MOVRPTW. Given the complexity, metaheuristic approaches are more commonly used to solve the problems than exact optimization models. Research conducted in [1] demonstrated a hybridized genetic algorithm (GA) in a maritime logistics application with two objectives. The proposed algorithm in that research took 5-6 hours of computing time. Such a long computing time can still be accepted for strategic decisions such as determining routes in an industry that involves large operational costs. However, for everyday problems such as delivery of fresh products, a faster algorithm is needed.


M. Hartono et al. (eds.), *Proceedings of the 4th International Conference on Informatics, Technology and Engineering 2023 (InCITE 2023)*, Atlantis Highlights in Engineering 21,
https://doi.org/10.2991/978-94-6463-288-0_44

One metaheuristic that has not been much explored for VRP is the artificial immune system (AIS). In [2], AIS was tested for fresh product delivery with a single objective in cost minimization. To the best of our knowledge, there has never been an application of AIS for MOVRPTW. This becomes a research gap as well as the background of this research.

Given the above background, the objective of this research is to develop an AIS algorithm to solve routing problems with two objectives. The two objective functions are to (1) minimize the total distance and (2) minimize the number of vehicles. The algorithm is coded in Python and tested against Solomon benchmark instance data set c104. Short computation time is expected from the proposed algorithm to pave future applications in logistics cases that require fast decisions such as delivery and distribution of fresh products.

## 2 Literature Review

One famous VRP variant is the vehicle routing problem with time windows (VRPTW). These time windows can be formulated as hard time windows (must be satisfied) or soft time windows. Soft time windows means that vehicles can make deliveries faster or slower than the specified time as long as they are within the given time limits [3]. This flexibility allows logistics companies to save on distribution costs incurred to meet customer satisfaction. It also invites many researchers to continue developing methods for VRPSTW in order to produce the best solution.

VRP with single objective only optimizes one objective function, whereas multi-objective vehicle routing problem (MOVRP) optimizes two or more conflicting objective functions [4]. In the multi-objective case, usually there are a number of feasible solutions referred to as a set of non-dominated solutions or also called the Pareto front. In [4], the two objective functions are to minimize the total distance and minimize the number of vehicles used for delivery.

VRP for multi-objective cases, especially those using an *a priori* approach to seek non-dominated sets of solutions, gained researchers' attention in around 2000. As suggested in [5], a popular metaheuristic algorithm used in MOVRP research is an elitist non-dominated genetic sorting algorithm or NSGA-II based algorithm. However, from a review conducted in [6], there are other alternative approaches. For example, a metaheuristic scatter search algorithm was used in [7]. Another example is [8] where the authors combined a multi-objective neighborhood dominance-based algorithm and an e-constraint meta-heuristic algorithm. In [9], multi-objective particle swarm optimization (MOPSO) and multi-objective ant colony optimization (MOACO) were compared to the non-dominated sorting genetic algorithm (NSGA) and the results showed that NSGA produced better performance than the other two algorithms.

This research developed an algorithm for MOVRPTW based on the artificial immune system (AIS). AIS is inspired by the immune system in the human body. AIS can be categorized into either population based or network based [10]. Examples of the population-based AIS include clonal selection theory and negative selection theory. The clonal selection theory is used to explain the basic response of the adaptive immune system to antigenic stimuli [11]. Several AIS algorithms have been developed by imitating the clonal selection theory. A clonal selection algorithm called CLONALG forms a population of N antibodies, with each antibody having a specific random solution for the optimization process. In each iteration, some of the best antibodies will be selected as the clone, followed by a mutation process to form a new candidate population. The new antibody will be evaluated, and a certain percentage of the new antibody will be introduced into the population. Antibodies that are not selected will be replaced with new antibodies that are randomly generated [12].

The basic CLONALG algorithm as described above serves as the main algorithm in this research. However, we replaced the mutation process with the nine-step mutation suggested in [13]. This mutation process is basically a local search procedure with the aim of finding a wider distribution of solutions. This local search procedure relies on four points, $u$, $v$, $x$, and $y$, obtained randomly from a chromosome. The nine mutation steps are as follows:

M1: If $u$ is not a depot, move $u$ behind $v$
M2: If $u$ and $x$ are not depots, move $(u, x)$ behind $v$
M3: If $u$ and $x$ are not depots, move $(x, u)$ behind $v$
M4: If $u$ and $v$ are not depots, swap positions between them
M5: If $u$, $x$ and $v$ are not depots, swap positions between $(u, x)$ and $v$
M6: If $(u, x)$ and $(v, y)$ are not depots, swap positions between $(u, x)$ and $(v, y)$
M7: If $u$ and $v$ are on the same route, swap $(u, x)$ and $(v, y)$ for $(u, v)$ and $(x, y)$
M8: If $u$ and $v$ are on different routes, swap $(u, x)$ and $(v, y)$ for $(u, v)$ and $(x, y)$
M9: If $u$ and $v$ are on different routes, swap $(u, x)$ and $(v, y)$ for $(u, y)$ and $(x, v)$

## 3 Methodology

After finding the research gap, defining the research objectives, and conducting literature review, the next step is to build the algorithm. The proposed algorithm combines the following principles:

1. Iterative process with the principle of population-based evolutionary algorithm;
2. Nearest neighbor (NN) and Clarke-Wright (CW) heuristics to produce the first two chromosomes;
3. Split procedure to transform the chromosome to VRP routes;
4. Clonal selection algorithm (CLONALG) to duplicate chromosomes (clones);
5. Mutation of nine steps (exhaustive neighborhood search) in each clone.

The modified AIS algorithm for MOVRPTW above is used to solve the first objective function which is to minimize the total distance. The NN and CW heuristics were used to generate two initial chromosomes with a fairly good fitness value (FV), i.e. the total distance. This aims so that not all chromosomes in the initial population are random, to accelerate the search process. Other chromosomes of *popsize* – 2 (*popsize* = population size) were formed randomly. The chromosome is a giant tour consisting of nodes without depot representation and trip delimiters. In this research, each chromosome from the Solomon c104 data set will contain 100 customer points. An example of a chromosome representation with sequence of nodes 1, 2, 3, …, 100 is given in Figure 1. To form VRP routes from this chromosome, the Split procedure is used. We refer readers to [13] for detailed information of the procedure.


Fig. 1. An example of a chromosome structure with ordered nodes 1 to 100.

Chromosomes will be accepted as members of the population if the spaced condition is met, namely each chromosome must have a different FV from the other chromosomes at least with a value of $\Delta > 0$ (1). Based on [14], a fairly good $\Delta$ is 1.

$$|FV(P1) - FV(P2)| \geq \Delta \quad \forall P1, P2 \in \pi : P1 \neq P \tag{1}$$

$$Clone\ rate = \frac{Affinity}{FV_{max}} \times 10 \tag{2}$$

The flow of the algorithm is shown in Figure 2 with an example of an initial population of five chromosomes. Then, using the clonal selection algorithm (CLONALG), the five initial chromosomes will be duplicated based on the affinity value of each chromosome (2). Chromosomes with better FV will produce more clones. The initial number of chromosomes to be duplicated can be one of the parameters in the algorithm. In cases where the initial number of chromosomes is large enough, it is not necessary to duplicate all the chromosomes and only the top percentile (e.g. 80% of the chromosomes with the best FV) of the population is duplicated. In cases where the initial number of chromosomes is not too large, it is possible for all of them to be duplicated and this will result in a population size of the clone that is larger than the size of the base population, as shown in Figure 2.

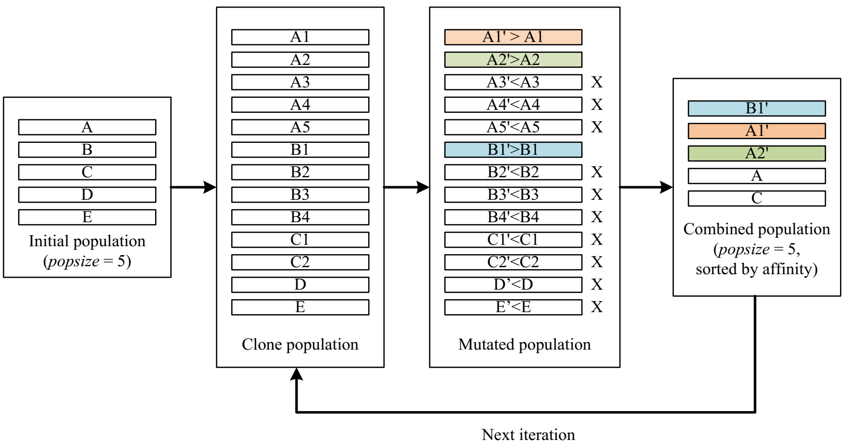


Fig. 2. The proposed modified AIS algorithm.

In the next stage, the chromosome will be mutated based on the principle of neighborhood search. There is a random factor in play in this process due to the selection of 2–4 cutting points in the chromosome. Because chromosomes with better FV have a greater number of clones, these chromosomes have a greater probability to produce better results. After that, the population resulting from the mutation is sorted from the best to the worst FV and the best chromosomes will become the new population. This process runs iteratively until the stopping criteria are met.

The algorithm is coded in Python. The code is then run on Solomon c104 benchmark instance. This data set was chosen because it contains data on time windows hence suitable for VRPTW. The results are reported in the next section.

## 4 Results and Discussion

The following parameters were used in the experiment. Ten replications are required due to the randomness in the selection of cross points during the run of the algorithm.

- population size: 100
- number of iterations: 50
- clone rate: 50%
- number of replications: 10

**Table 1.** Results of the first-run experiment.

| Replication no. | Total distance | Num. of vehicles | Run time (sec.) |
|---|---|---|---|
| 1 | 1258.07 | 15 | 34.98 |
| 2 | 1272.68 | 16 | 32.24 |
| 3 | 1203.60 | 15 | 29.19 |
| 4 | 1269.75 | 15 | 29.80 |
| 5 | 1275.37 | 15 | 31.96 |
| 6 | 1259.65 | 16 | 31.00 |
| 7 | 1270.84 | 16 | 32.54 |
| 8 | 1254.95 | 16 | 32.79 |
| 9 | 1268.73 | 15 | 31.98 |
| 10 | 1276.54 | 16 | 31.19 |

The AIS solution to minimize total distance is given in Table 1. The best solution is obtained in the third replication with a total distance of 1203.60 and 15 vehicles. The routes of those 15 vehicles are as follows.

Route 1: [0, 57, 55, 54, 53, 56, 58, 60, 0]
Route 2: [0, 81, 78, 76, 71, 70, 73, 77, 79, 80, 0]

Route 3: [0, 32, 33, 35, 37, 38, 39, 36, 59, 0]
Route 4: [0, 18, 19, 15, 16, 14, 12, 0]
Route 5: [0, 13, 99, 0]
Route 6: [0, 87, 98, 96, 95, 94, 97, 17, 100, 0]
Route 7: [0, 90, 89, 88, 85, 84, 83, 82, 86, 91, 0]
Route 8: [0, 43, 42, 41, 40, 44, 45, 46, 48, 50, 51, 52, 49, 47, 0]
Route 9: [0, 92, 93, 9, 7, 75, 1, 2, 0]
Route 10: [0, 6, 8, 11, 10, 0]
Route 11: [0, 68, 64, 61, 72, 5, 3, 4, 0]
Route 12: [0, 67, 65, 63, 62, 66, 69, 0]
Route 13: [0, 27, 25, 24, 29, 30, 34, 0]
Route 14: [0, 74, 0]
Route 15: [0, 20, 21, 23, 26, 22, 28, 31, 0]

From this solution, a series of trial-and-error was then applied to reduce the number of vehicles. The results are given in Table 2. Note that the solution with 12 vehicles dominates the solutions with 14 and 13 vehicles.

**Table 2.** Results of trial-and-error.

| Num. of vehicles | Total distance |
|---|---|
| 14 | 1389.90 |
| 13 | 1355.76 |
| 12 | 1344.13 |

**Sensitivity Analyses.** Sensitivity analyses were conducted by changing the model parameters to evaluate their impact on the solutions. The studied parameters include the number of iterations, the population size, and the clone rate.

Two different iteration numbers were tested in addition to the original 50, i.e. 100 and 150. The results are summarized in Table 3. From Table 3, it can be concluded that increasing the number of iterations has no impact on improving the best objective function. With 50 iterations, the total distance is minimal (1203.60) with the same number of vehicles (15) compared to other scenarios. However, it is important to note that on average, a higher number of iterations results in a better average FV, although not that significant.

**Table 3.** Sensitivity analysis on the number of iterations.

| Number of iterations | 50 | 100 | 150 |
|---|---|---|---|
| Best FV | 1203.60 | 1245.08 | 1246.10 |
| Number of vehicles | 15 | 15 | 15 |
| Average FV | 1261.02 | 1258.45 | 1256.64 |
| Average run time (sec.) | 31.77 | 56.26 | 51.02 |

The second sensitivity analysis was done by increasing the population size from 100 to 200. The number of iterations in the two experiments is maintained at 50 to save computation time. The experimental results are summarized in Table 4. With 200 members of the population, the best FV was 1245.68, or worse than the FV of the initial experiment with 100 population members (1203.60). However, the number of vehicles in the solution with 200 population members is 14, which is less than that in the solution with 100 population members (15). This means that the two solutions do not dominate each other and can be accepted as two solutions in a multi-objective problem. The 14-vehicle solution with a total distance of 1245.68 is also better than the 14-vehicle solution with a total distance of 1389.90 obtained previously from trial-and-error. Another interesting result from this sensitivity analysis is that the average computation time is significantly lower than other scenarios with an average of under 19 seconds.

**Table 4.** Sensitivity analysis on the population size.

| Population size | 100 | 200 |
|---|---|---|
| Best FV | 1203.60 | 1245.68 |
| Number of vehicles | 15 | 14 |
| Average FV | 1261.02 | 1262.24 |
| Average run time (sec.) | 31.77 | 18.21 |

Lastly, the third sensitivity analysis was conducted by changing the clone rate from 50% to 80%. The number of iterations and population sizes are the same as in the initial experiments, 50 and 100, respectively. The experimental results are reported in Table 5. With a clone rate of 80%, the best FV is 1219.72, worse than the results from the initial experiment with a 50% clone rate. The number of vehicles in both clone rates are the same 15, which means that the best solution at the 80% clone rate is dominated by the best solution at the 50% clone rate. However, on average, the FV at the 80% clone rate is slightly better than that at the 50% clone rate. In addition, the average computation time at the 80% clone rate is significantly lower than the other scenarios with an average of under 18 seconds.

**Table 5.** Sensitivity analysis on the clone rate.

| Clone rate | 50% | 80% |
|---|---|---|
| Best FV | 1203.60 | 1219.72 |
| Number of vehicles | 15 | 15 |
| Average FV | 1261.02 | 1255.57 |
| Average run time (sec.) | 31.77 | 17.19 |

From all of the above results, three non-dominated solutions are obtained below:

1. 15 vehicles, total distance 1203.60
2. 14 vehicles, total distance 1245.68

3. 12 vehicles, total distance 1344.13

These solutions, marked in red, are plotted in Figure 3 together with the other solutions which are marked in black.

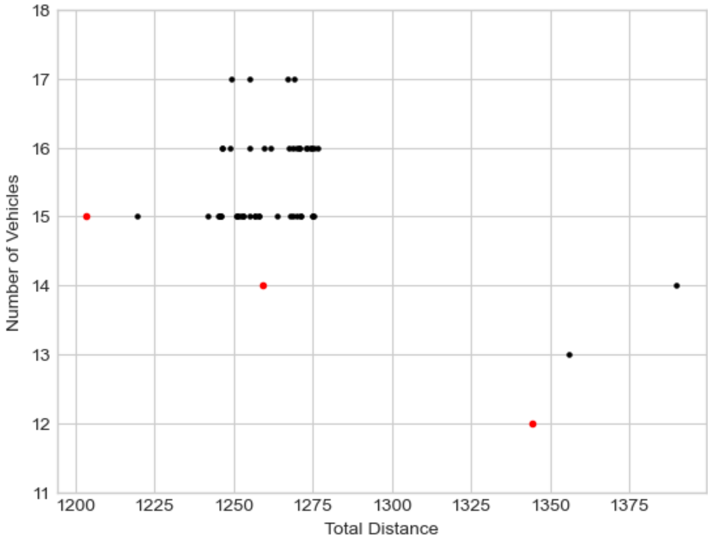


Fig. 3. Scatter plot of solutions.

## 5 Conclusion and Further Remarks

This research succeeded in developing an AIS-based algorithm to solve the VRP problem with two objective functions, namely minimizing the total distance and the number of vehicles. The developed algorithm is an integration of several existing algorithms such as NN and CW heuristics, Split procedure, and mutation based on local (neighborhood) search. A computer program in Python was coded to run the algorithm.

Initial experiments were conducted with certain parameters, followed by trial-and-error and sensitivity analysis. From these extensive experiments, three non-dominated solutions were obtained along with the other dominated solutions. The obtained solutions, however, are still dominated by the best-known solutions (BKS) for Solomon c104 that can achieve 822.9 of total distance with 10 vehicles using the exact approach, and 824.78 with 10 vehicles using a heuristic approach. The performance of the proposed algorithm is the limitation of this research if the two objective functions are considered as the primary goals without observing the run times.

Information on the computing times for the above BKS was not found, but the exact methods and heuristics generally require significant run time. On the other hand, the strength of the proposed algorithm lies in its computing time, where none of the runs exceeded one minute in many different scenarios. With a very short computation time, this algorithm is suitable for use in logistics cases that call for fast routing solutions.

The state of the research above indicates there is still room for model development. The direction of development needed to be sought is at improving the total distance without too much worsening the computation time. In addition, VRP with heterogeneous fleet can be studied, especially if the algorithm on homogeneous fleets has achieved satisfactory results.

## References

1. Wibisono, E., Jittamai, P.: Multi-objective evolutionary algorithm for a ship routing problem in maritime logistics collaboration. International Journal of Logistics Systems and Management 28(2), 225–252 (2017).
2. Shukla, M., Jharkharia, S.: Artificial Immune System-based algorithm for vehicle routing problem with time window constraint for the delivery of agri-fresh produce. Journal of Decision Systems 22(3), 224–247 (2013).
3. Zhang, H., Zhang, Q., Ma, L., Zhang, Z., Liu, Y.: A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. Information Sciences 490(6), 166–190 (2019).
4. Khoo, T. S., Mohammad, B. B.: The parallelization of a two-phase distributed hybrid ruin-and-recreate genetic algorithm for solving multi-objective vehicle routing problemwith time windows. Expert Systems with Applications 168, 114408 (2021).
5. Deb, K., Agrawa, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI), pp. 849–858 (2000).
6. Martin, I., Wibisono, E.: A Proposed Taxonomy for Literature Review in Multi-Objective Vehicle Routing Problems. In: 8th International Conference on Engineering, Technology, and Industrial Application. Universitas Muhammadiyah Surakarta, Surakarta (2021).
7. Melián-Batista, B., De Santiago, A., AngelBello, F., Alvarez, A.: A bi-objective vehicle routing problem with time windows: A real case in Tenerife. Applied Soft Computing 17, 140–152 (2014).
8. Bula, G. A., Afsar, H. M., González, F. A., Prodhon, C., Velasco, N.: Bi-objective vehicle routing problem for hazardous materials transportation. Journal of Cleaner Production 206, 976–986 (2019).
9. Ganji, M., Kazemipoor, H., Molana, S. M. H., Sajadi, S. M.: A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. Journal of Cleaner Production 259, 120824 (2020).
10. Talbi, E. G.: Metaheuristics: From Design to Implementation. Wiley, NJ (2009).
11. Al-Enezi, J. R., Abbod, M. F., Alsharhan, S.: Artificial immune systems-models, algorithms and applications. International Journal of Research and Reviews in Applied Sciences 3(2), 118–131 (2010).
12. De Castro, L. N., Von Zuben, F. J.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002).
13. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Computers and Operations Research 31(12), 1985–2002 (2004).
14. Wibisono, E., Martin, I., Prayogo, D. N.: Genetic Algorithm with Adaptive Diversification and Intensification for the Vehicle Routing Problem. In: AIP Conference Proceedings 2470(1), 050012 (2022).