

Optimizing a Transportation System Using Metaheuristics Approaches (EGD/GA/ACO): A Forest Vehicle Routing Case Study

Hossein Havaeji*, Thien-My Dao, Tony Wong

Mechanical Engineering Department, École de technologie supérieure, Montreal, Canada

Email: *SeyedHossein.Havaeji.1@ens.etsmtl.ca, Thien-My.Dao@etsmtl.ca, Tony.Wong@etsmtl.ca

How to cite this paper: Havaeji, H., Dao, T.-M. and Wong, T. (2024) Optimizing a Transportation System Using Metaheuristics Approaches (EGD/GA/ACO): A Forest Vehicle Routing Case Study. *World Journal of Engineering and Technology*, 12, 141-157. <https://doi.org/10.4236/wjet.2024.121009>

Received: November 12, 2023

Accepted: February 5, 2024

Published: February 8, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The large-scale optimization problem requires some optimization techniques, and the Metaheuristics approach is highly useful for solving difficult optimization problems in practice. The purpose of the research is to optimize the transportation system with the help of this approach. We selected forest vehicle routing data as the case study to minimize the total cost and the distance of the forest transportation system. Matlab software helps us find the best solution for this case by applying three algorithms of Metaheuristics: Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Extended Great Deluge (EGD). The results show that GA, compared to ACO and EGD, provides the best solution for the cost and the length of our case study. EGD is the second preferred approach, and ACO offers the last solution.

Keywords

Metaheuristics Algorithms, Transportation Costs, Optimization Approach, Cost Minimisation

1. Introduction

In recent years, many studies have been carried out on the development of heuristics and Metaheuristics. The large-scale optimization problem needs some optimization techniques to search the problem space profoundly and to optimize the problem efficiently and effectively [1]. Metaheuristics solve difficult optimization problems in practice. The process of getting forest goods from suppliers to consumers is known as the distribution of forest products. Transportation expenses usually account for 10% - 20% of the overall cost of items sold according to [2].

The first point is that the forest sector accounts for a considerable share of exports from nations like Chile, Canada, Sweden, Finland, and New Zealand, according to [2]. At almost \$26 billion in 2011, Canada ranked as the world's top exporter of forest products, according to Natural Resources Canada. Thus, this industry ranks second in terms of exports, behind the oil and gas industry. The second point is that a large portion of the total operating costs of a forest may be attributed to transportation [2]. Thirdly, for the forest industry to remain competitive, it must be guaranteed to retain or increase the efficacy of all of its activities [2]. In the last 20 years, metaheuristics have proven particularly effective in solving the vehicle routing problem. Every route begins with the first client, and further customers are added one at a time until the capacity or time limit is reached. Next, utilizing local search heuristics tailored to routing issues, this preliminary solution is enhanced.

The first problem is related to the total costs of the forest transportation system by a significant wooden company that distributes wood products from the depot of the company to its customers in different cities. The second problem refers to the total distance of this forest transportation system. The purpose of this study is to minimize the total cost and the total length of this transportation system and then compare the cost and the length of this case to find the best solution. This is based on three algorithms (EGD/GA/ACO) releasing a good solution of forest routing optimization independently of parameters. The rest of the paper is organised as follows. First, the paper provides an overview of three metaheuristic algorithms in Section 2. Next, we discuss the research methods used to optimize the model in Section 3. Then, the mathematical model and case study is outlined in Section 4. All results are described in Section 5. Then, the results are discussed in Section 6, which is called Discussion. Finally, we briefly present the conclusions in Section 7.

2. Research Background

2.1. The Use of Metaheuristics

Techniques for solving combinatorial problems fall into two broad groups – traditional optimization techniques (exact algorithms) and nontraditional optimization techniques (approximate algorithms) [3]. This author states traditional optimization technique guarantees that the optimal solution to the problem will be found. This author continues the traditional optimization techniques like branch and bound, simplex method, brute force search algorithm etc. methodology is very inefficient for solving combinatorial problems because of their prohibitive complexity (time and memory requirement). Nontraditional optimization techniques are employed in an attempt to find an adequate solution to the problem [3]. A nontraditional optimization technique—Memetic algorithm, genetic algorithm, simulated annealing and tabu search were developed to provide a robust and efficient methodology for cryptanalysis [3]. The aim of these techniques is to find a sufficient “good” solution efficiently with the characteris-

tics of the problem, instead of the global optimum solution, and thus it also provides an attractive alternative for large-scale applications, according to [3].

Metaheuristics have been demonstrated by the scientific community to be a viable, and often superior, alternative to more traditional (exact) methods of mixed-integer optimization such as branch and bound and dynamic programming [4]. The use of Metaheuristic techniques allows us to obtain reasonably good solutions without having to explore the whole solution space [5]. Especially for complicated problems or large problem instances, Metaheuristics are often able to offer a better trade-off between solution quality and computing time [4]. Moreover, they assert Metaheuristics are more flexible than exact methods in two important ways. First, because Metaheuristics frameworks are defined in general terms, Metaheuristics algorithms can be adapted to fit the needs of most real-life optimization problems in terms of expected solution quality and allowed computing time, which can vary greatly across different problems and different situations [4]. Secondly, Metaheuristics do not put any demands on the formulation of the optimization problem (like requiring constraints or objective functions to be expressed as linear functions of the decision variables), according to them. However, they highlight this flexibility comes at the cost of requiring considerable problem-specific adaptation to achieve good performance.

2.2. Genetic Algorithm (GA)

Genetic algorithms are population-based Meta heuristics. GA is an evolutionary computation technique that replicates the mechanism of biological evolution and natural selection process [6]. They continue GAs follow Charles Darwin's principles, based on which the fittest individuals have the highest chance of survival. The algorithm, according to these authors, works with a population of potential solutions, represented by chromosomes, and each chromosome (solution) carries encoded information represented by genes. GA works iteratively and modifies the population of solutions [6]. These are the steps they explain for GA process: a) at each step, they explain it selects individuals from the current population to play the role of parents and produce the children for the next generation; b) the selection process gives preference to the fittest individuals to let them pass the quality genes to the next generation; c) a fitness function is used to evaluate the potential solutions and a fitter solution is the one with a better fitness value; d) this fitness function can be identical to the objective function; e) a new population of solutions is created using genetic operators. The most popular genetic operators are 1) the crossover operator that generates new solutions by combining pairs of existing solutions and 2) the mutation operator that maintains diversity in the population by making small changes to the genes of individual solutions [6]. This evolutionary process, according to them, is stopped when a termination criterion or one of the termination criteria is met; can be a certain number of iterations, a specific runtime, a certain fitness value, or when no more improvement occurs in the fitness value in consecutive iterations.

2.3. Extended Great Deluge (EGD)

The great deluge algorithm is a local search procedure that was introduced by Dueck in 1993. The idea of the great deluge comes from the analogy that a person climbs a hill and tries to move in any direction of finding a way up to keep his feet dry as the water level rises during a great deluge [7]. Inserting a great deluge algorithm within a genetic algorithm is considered an effective way to produce a high-quality solution rather than using a genetic algorithm alone [7]. The study [7] explains that like other local search methods, the extended great deluge iteratively repeats the replacement of a current solution by a new one, until some stopping condition has been satisfied. They also say a new solution is selected from a neighborhood. The mechanism of accepting or rejecting the candidate solution from the neighborhood is different from other methods, according to them. In the extended great deluge approach, the algorithm accepts a solution whose objective function value is more than or equal (for the maximization problems) to the upper limit [8].

2.4. Ant Colony Optimization (ACO)

The idea of employing a colony of simple cooperating agents to solve combinatorial optimization problems was proposed by [9]. The ant algorithm was inspired by the collective performance of real-life ant colonies [10]. Ant colony optimization shows very good results in each applied area. The ant colony has also been adapted with success to other combinatorial optimization problems such as the vehicle routing problem, telecommunication networks management, graph coloring, constraint satisfaction, and Hamiltonian graphs [10]. The authors [10] explain how the ants act to find the best route solutions when they search for food. Ants lay down in some quantity an aromatic substance, known as pheromone, in their way from food [10]. They assert the pheromone quantity depends on the length of the path and the quality of the discovered food source. An ant chooses a specific path in correlation with the intensity of the pheromone; and the pheromone trail evaporates over time if no more pheromone is laid down, according to them. Other ants can observe the pheromone trail and are attracted to follow it; thus, the path will be marked again and will therefore attract more ants [10]. They define the pheromone trail on paths leading to rich food sources close to the nest will be more frequented and will therefore grow faster; in that way, the best solution has more intensive pheromone and a higher probability to be chosen. The described behavior of real ant colonies can be used to solve combinatorial optimization problems by simulation: artificial ants searching the solution space simulate real ants searching their environment, according to them. They state the objective values correspond to the quality of the food sources. The authors in [10] explain that the ACO approach associates pheromone trails to features of the solutions of a combinatorial problem, which can be seen as a kind of adaptive memory of the previous solutions; in addition, the artificial ants are equipped with a heuristic function to guide their search through the set of feasible solutions. Solutions are iteratively constructed in a

randomized heuristic fashion biased by the pheromone trails left by the previous ants. The pheromone trails are updated after the construction of a solution, enforcing that the best features will have a more intensive pheromone, according to them.

3. Research Methods

To achieve our objectives, we reviewed literature by reading academic articles and books to learn about Metaheuristics techniques and select the right Metaheuristics algorithms for the case study. Therefore, the authors applied three algorithms (GA, ACO, and EGD) to minimize the total cost and the total length of the forest transportation system.

Matlab software was used to find the best solutions for these algorithms. Validating the obtained results to compare the results with the previous studies came two important to make to the last step.

Metaheuristic techniques prepare reasonably good solutions without exploring the space of the whole solution [5]. Metaheuristics often can offer a better trade-off between the solution quality and computing time, especially for complicated or large-scale problems [4]. Moreover, they assert two important ways to make Metaheuristics more flexible than other methods. First, Metaheuristics algorithms can fulfill the needs of most real-life optimization problems regarding the quality of the expected solution and allow computing time, which are different in problems and situations [4]. Secondly, there are no demands on the formulation of the optimization problem in Metaheuristics (for instance, requiring limitations or objective functions to be stated as linear functions of the decision variables), according to them. However, they highlight that his flexibility comes at the cost of requiring significant problem-specific adaptation to attain good performance.

The most usage of Metaheuristic strategy applied to the variable selection problem is the Genetic Algorithm (GA) [5]. GA is a population-based Metaheuristics and is an evolutionary computation technique replicating the mechanism of biological evolution and the process of natural selection [6]. They continue GAs follow Charles Darwin's principles, based on which the fittest individuals have the highest chance of survival. According to these authors, the algorithm works with a potential solutions population represented by chromosomes, and each chromosome (solution) carries encoded information represented by genes.

The Extended Great Deluge (EGD) algorithm is a local search procedure introduced by [11]. The idea of this algorithm comes from the analogy that an individual climbs a hill and tries to move in any direction to find a way up to keep his feet dry as the water level rises during a great deluge [7].

[9] and [12] proposed the idea of Ant Colony Optimization (ACO) to solve combinatorial optimization problems. The ant algorithm was inspired by the collective performance of real-life colonies [10]. ACO shows particularly good results in an applied area. The ant colony uses other combinatorial optimization

problems such as the problem of vehicle routing, telecommunication networks management, graph coloring, constraint satisfaction, and Hamiltonian graphs [10].

4. Mathematical Model and Case Study

The heuristic solutions quality differs and depends on the methods used [5]. Metaheuristics techniques have proved to be superior methodologies in various optimization problems [5]. The success of methods (such as GA, SA, TS, SS, AS, and so forth) depends on some factors such as their ability to consider specific constraints arising in practical applications, their ease of implementation, and the quality of the solutions they produce [13].

This research refers to a real-life location routing problem encountered by a major wooden company distributing wood products from the company's depot to its customers located in various cities. The data of 50 customers comes from [14] paper (Table 1). According to the literature review, we decided to use three algorithms to solve the factory's localization problem, the dimensioning problem, and the forest products vehicle problem: GA, ACO, and EGD.

The other important information used in MATLAB software for three different algorithms (GA, ACO, and EGD) is as follows:

Population size: 50

Number of generations: 50

Crossover probability: 0.8

Mutation probability: 0.25

Three different threshold values ϵ_1 : 1%, 3% and 5%

Size of the Restricted Candidate List: 20

The location: Kilometer

The central depot of all customers is started to serve in the location: (0, 0)

The central depot is open: 0 hr to 40 hr

The fleet is: homogeneous.

The average speed of vehicles: 65 Km/h

The cost of transport: 2.8 \$

The vehicles used have a load capacity Q: 50 cubic meters

Total number of vehicles: 9

The total cost of transport is a mathematical model shown in Equation (1). This model is adapted from [4].

$$\begin{aligned} \text{The Min } C_t = & MC_f + \sum_{i=1}^{L+1} \sum_{j=1}^{L+1} \sum_{k=1}^m C_{ijk} D_{ij} X_{ijk} \\ & + (C_{vt} + C_{dt}) \sum_{i=1}^{L+1} \sum_{j=1}^{L+1} \sum_{k=1}^m X_{ijk} (T_{ij} + W_{jk} + S_j) \end{aligned} \quad (1)$$

The elements of this function are as follows:

MC_f = the total vehicles fixed cost

C_f = the unit vehicle fixed cost, covering loading and unloading

$\sum_{i=1}^{L+1} \sum_{j=1}^{L+1} \sum_{k=1}^m C_{ijk} D_{ij} X_{ijk}$ = the total distance cost summation

C_{ijk} = the unit cost of transport per kilometer of vehicle k from i to j

Table 1. Costumers' demand [14].

| Customer | Location | | Demand (m ³) | Start time | Finish time |
|----------|----------|--------|--------------------------|------------|-------------|
| 1 | 50 | -290 | 29.379 | 0 | 40 |
| 2 | -120 | -250 | 3.711 | 0 | 40 |
| 3 | -90 | -250 | 23.664 | 0 | 40 |
| 4 | -145 | -320 | 4.79 | 0 | 40 |
| 5 | -100 | -390 | 4.517 | 0 | 40 |
| 6 | -80 | -390 | 2.018 | 0 | 40 |
| 7 | -82 | -337 | 1.57 | 0 | 40 |
| 8 | -40 | -332 | 5.317 | 0 | 40 |
| 9 | -46 | -324.5 | 13.403 | 0 | 40 |
| 10 | -42 | -286 | 3.725 | 0 | 40 |
| 11 | -63 | -200.5 | 6.676 | 0 | 40 |
| 12 | -112.5 | -199 | 6.78 | 0 | 40 |
| 13 | -38 | -242 | 4.3 | 0 | 40 |
| 14 | 9 | -251 | 2.065 | 0 | 40 |
| 15 | 40 | -230 | 5.335 | 0 | 40 |
| 16 | -159 | -108 | 3.797 | 0 | 40 |
| 17 | -211 | -126 | 0.856 | 0 | 40 |
| 18 | -142 | -163 | 5.631 | 0 | 40 |
| 19 | -121 | -242 | 1.496 | 0 | 40 |
| 20 | -124 | -213 | 24.741 | 0 | 40 |
| 21 | -61 | -6 | 3.464 | 0 | 40 |
| 22 | -79 | -11 | 3.691 | 0 | 40 |
| 23 | -51 | 14 | 1.293 | 0 | 40 |
| 24 | -117 | 18 | 7.899 | 0 | 40 |
| 25 | -126 | -8 | 2.255 | 0 | 40 |
| 26 | -88 | -28 | 2.664 | 0 | 40 |
| 27 | -95 | -9 | 1.504 | 0 | 40 |
| 28 | -112 | -54 | 0.859 | 0 | 40 |
| 29 | 3 | -3 | 8.319 | 0 | 40 |
| 30 | 38 | -29 | 1.299 | 0 | 40 |
| 31 | -35 | -41 | 1.715 | 0 | 40 |
| 32 | -5 | 39 | 1.535 | 0 | 40 |
| 33 | 46 | 48 | 6.198 | 0 | 40 |
| 34 | 91 | 52 | 18.079 | 0 | 40 |
| 35 | 187 | 47 | 2.008 | 0 | 40 |
| 36 | 225 | 19 | 1.692 | 0 | 40 |

Continued

| | | | | | |
|----|-------|-------|--------|---|----|
| 37 | -40.5 | -97.5 | 44.018 | 0 | 40 |
| 38 | -90 | -100 | 6.067 | 0 | 40 |
| 39 | -83 | -118 | 0.539 | 0 | 40 |
| 40 | 0.5 | -117 | 2.432 | 0 | 40 |
| 41 | -180 | 468 | 1.978 | 0 | 40 |
| 42 | -106 | 126 | 7.446 | 0 | 40 |
| 43 | -269 | 81 | 11.174 | 0 | 40 |
| 44 | 34.5 | 237.5 | 43.413 | 0 | 40 |
| 45 | 149.5 | 45 | 0.744 | 0 | 40 |
| 46 | -195 | -198 | 2.103 | 0 | 40 |
| 47 | -200 | -207 | 1.829 | 0 | 40 |
| 48 | -52 | -231 | 0.676 | 0 | 40 |
| 49 | 5 | -446 | 3.978 | 0 | 40 |
| 50 | 109.5 | 38 | 3.116 | 0 | 40 |

D_{ij} = the distance between two locations i and j

X_{ijk} = vehicle k go from i to j

$(C_{vt} + C_{dt}) \sum_{i=1}^{L+1} \sum_{j=1}^{L+1} \sum_{k=1}^m X_{ijk} (T_{ij} + W_{jk} + S_j)$ = the total vehicles route time cost and total drivers work time cost summations

C_{vt} = the unit vehicle route time cost

C_{dt} = the unit driver's work time cost

W_{jk} = the vehicle k waiting time at customer j

S_j = the customer j service time

Equation (2) shows the time spent from i to j .

$$T_{ij} = D_{ij} / V \quad i \in [1, L+1]; j \in [1, L+1] \tag{2}$$

$$W_{ik} = \max(T_i^s - T_{ik}, 0) \quad i \in [1, L+1]; k \in [1, m] \tag{3}$$

The definitions of elements are:

T_{ij} = vehicle k arrival time to customer i

W_{ik} = vehicle k waiting time at customer i

S_i = customer i service time

There are nine constraints restrictions:

The vehicle k total load cannot exceed the maximum vehicle load Q :

$$\sum_{j=2}^{L+1} Q_j Y_{kj} \leq Q \quad k \in [1, m] \tag{4}$$

Every customer is served only by one vehicle:

$$\sum_{k=1}^m Y_{kj} = 1 \quad j \in [1, L+1] \tag{5}$$

Customer j is served by vehicle k passing through i , for every customer:

$$\sum_{i=1}^{L+1} X_{ijk} = Y_{kj} \quad j \in [1, L+1]; k \in [1, m] \tag{6}$$

The variable Y_{kj} is binary:

$$Y_{kj} = \begin{cases} 1, & \text{the customer } j \text{ is served by the vehicle } k \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The variable X_{ijk} is binary:

$$X_{ijk} = \begin{cases} 1, & \text{vehicle } k \text{ goes from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The bond between vehicle k arrival times to customers i and j :

$$X_{ijk} = (T_{ik} + W_{ik} + S_i + T_{ij}) = T_{jk} \quad i, j \in [1, L+1]; k \in [1, m] \quad (9)$$

The service at customer j must be completed before T_k , the end of the vehicle k time:

$$\sum_{i=1}^{L+1} \sum_{j=1}^{L+1} X_{ijk} (T_{ij} + W_{jk} + S_j) \leq T_k \quad k \in [1, m] \quad (10)$$

No customer can be served before his time of beginning:

$$T_j \geq T_j^s \quad j \in [1, L+1] \quad (11)$$

No customer can be served after his time of the end:

$$T_j \leq T_j^e \quad j \in [1, L+1] \quad (12)$$

The distance between two locations x_i and x_j is calculated with Equation (13).

$$D_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad i, j \in [1, L+1] \quad (13)$$

We used the cost calculation data of [2] in which the unit vehicle fixed cost is $c_f = 160$ \$, the unit vehicle route time cost is $c_{vt} = 120$ \$, the unit driver work time cost is $c_{dt} = 18$ \$ and all service times are $s_j = 0$.

5. Results

People, in real-world applications, are more interested in finding good solutions in a reasonable amount of time instead of being obsessed with optimal solutions [5]. As mentioned before, this study refers to the problem of real-life location routing. A major wooden company distributes wood products from the company's depot to its customers located in various cities. The data comes from the study of [14]. Three algorithms (GA, ACO, and EGD) were selected to solve the cost and length problems of the forest transportation system. We used MATLAB software to find the best solutions for these algorithms. The results of each algorithm are illustrated in different following sections.

5.1. Results Obtained by Genetic Algorithm

The most popular genetic operators are (1) crossover operators that generate new solutions by combining some of the existing solutions and (2) mutation operators that maintain diversity in the population by making small changes to the genes of individual solutions [6]. For our case study, the result obtained by the GA shows the best cost is 19901.1417 \$ and the total distance is 1093.46 (km).

This came from MATLAB software in 1540.7213 seconds. **Figure 1** shows the best cost of the GA forest vehicle routing convergence graphic. **Figure 2** illustrates the GA forest vehicle routing optimization graphic in which the vehicles' routes have different colors. There are nine vehicles and 50 customers. The central depot is also shown with a blue dot in the center. In this figure, each vehicle passed the specific route to some customers; each vehicle route is shown by one color. Therefore, nine different colors show nine vehicle routes. **Table 2** shows the GA route for each vehicle, the length of each route, and the capacity used by each vehicle. The maximum capacity of each vehicle is 50 m³, the number of customers is 50, and the number of vehicles is 9.

5.2. Results Obtained by Ant Colony Optimization

ACO algorithms are population-based optimization approaches that have been applied with success to solve various combinatorial optimization problems, such as the traveling salesman problem [10]. We also used this algorithm for our case study to solve the problem with MATLAB software. In 1889.60 seconds, the software shows the best cost of 42345519.77 \$ and the total distance of 1679.81 (km) for this case. **Figure 3** and **Figure 4** show the cost of the ACO forest vehicle routing convergence graphic and the ACO forest vehicle routing optimization graphic, respectively. **Figure 4** shows various routes passed by nine vehicles; the specific route of each vehicle allocates one color. The central depot is blue in the figure. **Table 3** illustrates the routes of nine vehicles, the length of each route, and the capacity used by each vehicle (m³). There are nine vehicles, 50 customers, and a maximum vehicle capacity of 50 m³ for each.

5.3. Results Obtained by Extended Great Deluge

Using a mathematical model in the Extended Great Deluge (EGD) algorithm seems useful as we aim to minimize the total transport cost as well as the total distance that each vehicle should pass.

The EGD method converges in 6000 iterations, with the best distance of 4383.3053 (km), the best cost of 48930.6899 \$, and 3110.01 seconds. **Figure 5** shows the EGD forest vehicle routing cost convergence graphic. It illustrates that the best cost converges into approximately 50,000 \$ after 6000 iterations. **Figure 6** shows the EGD forest vehicle routing optimization graphic. In this figure, each color clarifies the route of each vehicle that passes its direction to different cost customers blue point in the center is the central depot. **Table 4** illustrates each vehicle that passes the route to some customers (EGD Rout), the length of each route, and the vehicle capacity. In this case, there are nine vehicles with a maximum capacity of 50 m³.

6. Discussion: Comparison of GA, ACO, and EGD Results

Table 5 clearly shows that in comparing three algorithms, GA proposes the best cost of 19901.1417 \$, the minimum distance (1093.46 km), and the computation

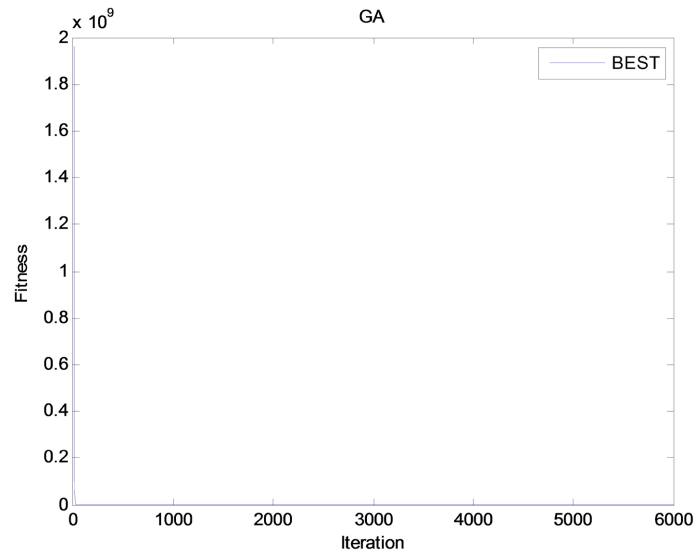


Figure 1. Cost of GA forest vehicle routing convergence graphic.

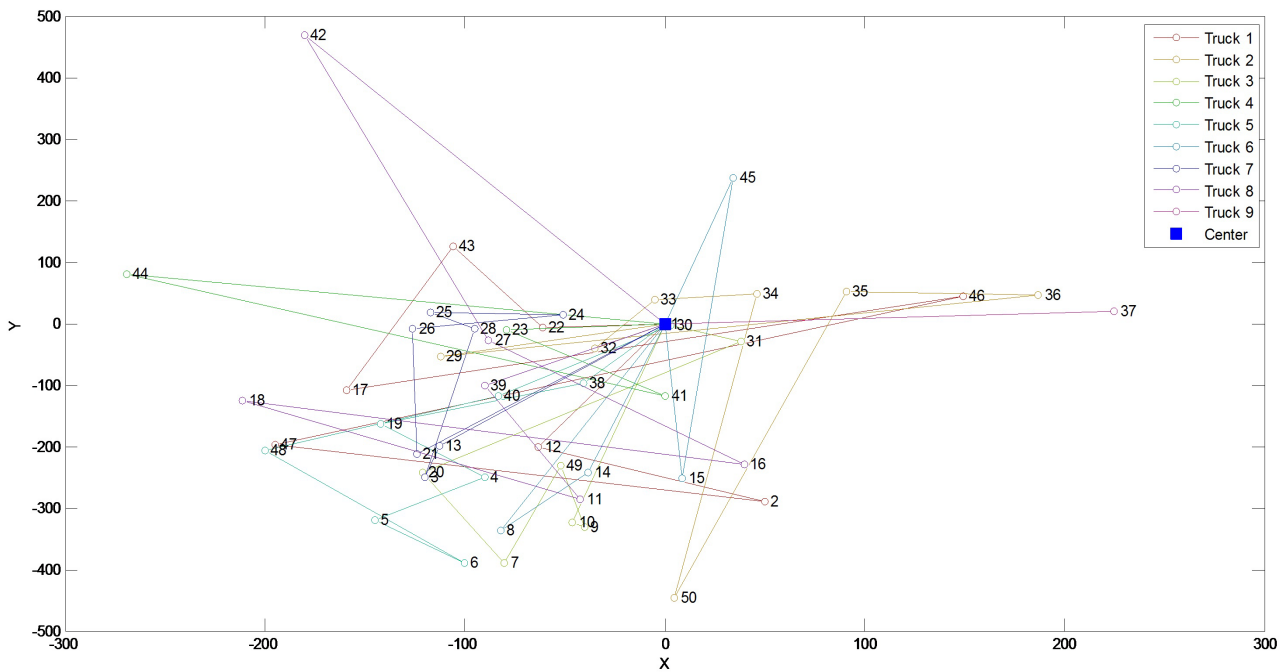


Figure 2. GA forest vehicle routing optimization graphic.

Table 2. Results for GA.

| No. of truck | GA Route | Length (km) | Capacity used by each vehicle (m ³) |
|--------------|-------------------------------|-------------|---|
| 1 | [22, 43, 17, 46, 47, 2, 12] | 147.82 | 30.144 |
| 2 | [29, 36, 35, 50, 34, 33, 32] | 78.77 | 40.947 |
| 3 | [31, 20, 7, 49, 9, 10] | 150.38 | 49.132 |
| 4 | [23, 41, 44] | 166.11 | 46.684 |
| 5 | [38, 19, 4, 5, 6, 48, 40, 30] | 137.71 | 23.295 |

Continued

| | | | |
|-------------------------|-----------------------------|--------|--------|
| 6 | [45, 15, 1, 8, 14] | 147.39 | 42.84 |
| 7 | [13, 3, 28, 25, 24, 26, 21] | 110.90 | 45.105 |
| 8 | [39, 11, 18, 16, 27, 42] | 121.87 | 25.593 |
| 9 | [37] | 32.48 | 44.018 |
| Total Distance: 1093.46 | | | |

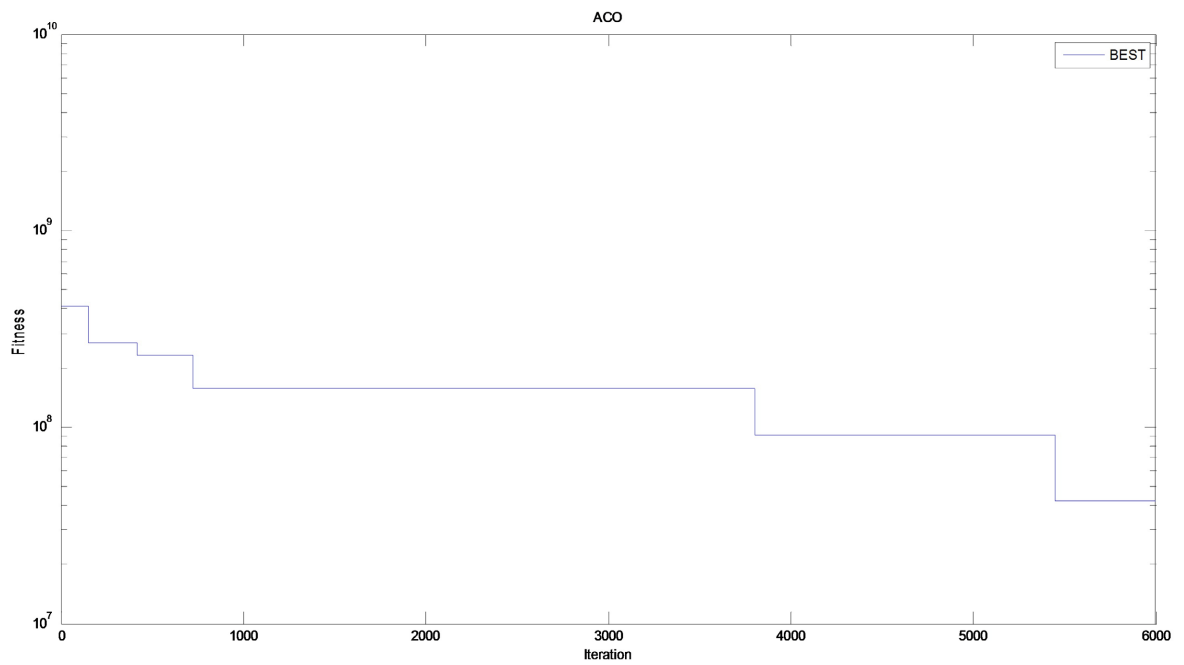


Figure 3. Cost of ACO forest vehicle routing convergence graphic.

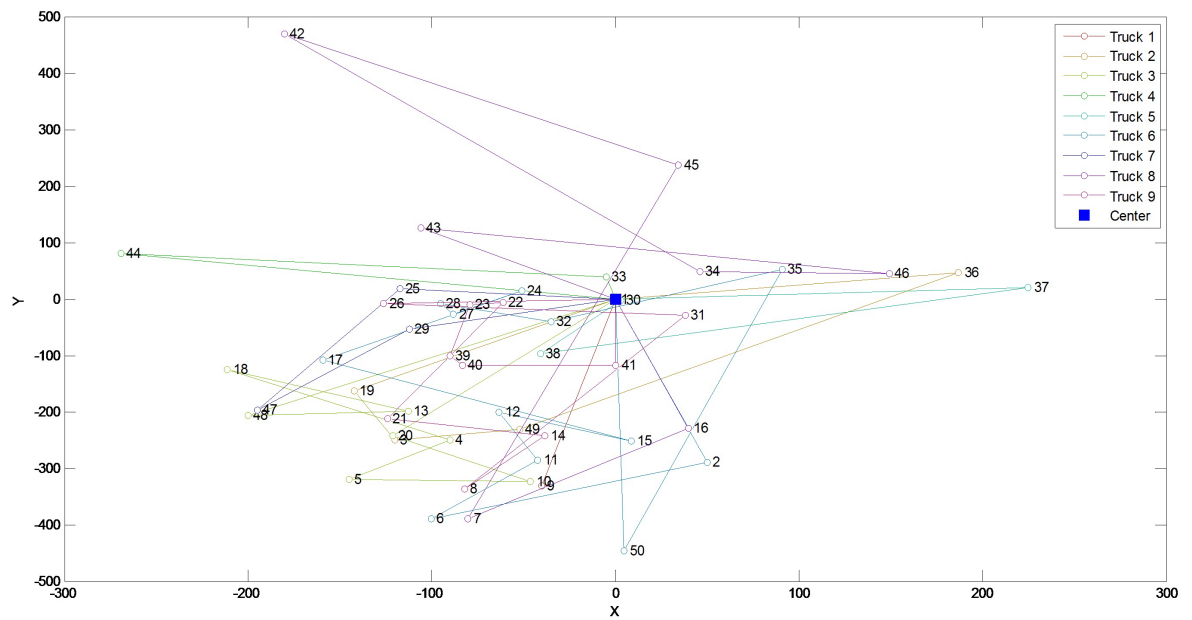


Figure 4. ACO forest vehicle routing optimization graphic.

Table 3. Results for ACO.

| No. of vehicle | ACO Route | Length (km) | Capacity used by each vehicle (m ³) |
|-------------------------|--|-------------|---|
| 1 | [9, 1] | 111.38 | 42.782 |
| 2 | [19, 3, 49, 36] | 193.93 | 30.83 |
| 3 | [20, 10, 5, 4, 18, 13, 48] | 169.3 | 48.38 |
| 4 | [33,44] | 76.35 | 49.611 |
| 5 | [30, 38, 37] | 53.74 | 51.384 |
| 6 | [2, 6, 11, 12, 15, 17, 24, 27, 28, 32, 35] | 288.11 | 42.297 |
| 7 | [29, 47, 25] | 97.01 | 12.403 |
| 8 | [43, 46, 34, 42, 45, 7, 16] | 355.13 | 44.913 |
| 9 | [26, 31, 8, 14, 21, 22, 23, 39, 40, 41] | 334.86 | 25.158 |
| Total Distance: 1679.81 | | | |

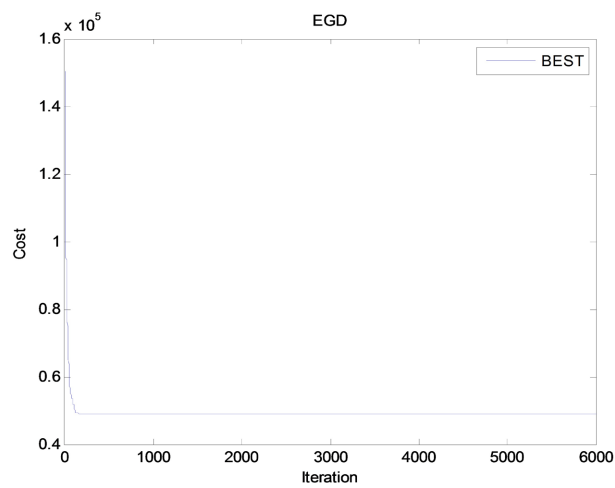


Figure 5. Cost of EGD forest vehicle routing convergence graphic.

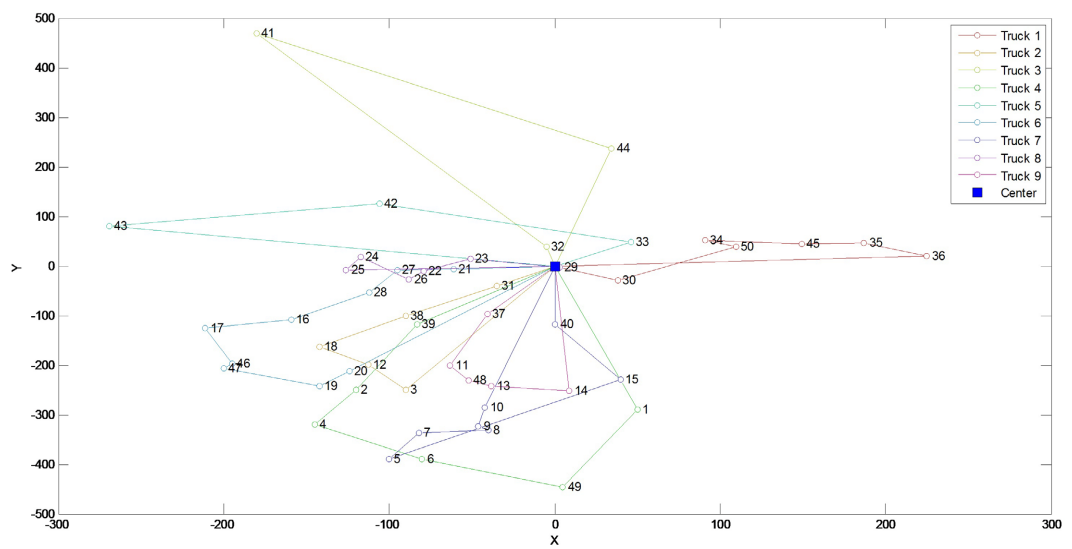


Figure 6. EGD forest vehicle routing optimization graphic.

Table 4. Results for EGD.

| No. of truck | EGD Route | Length (km) | Capacity used by each vehicle (m ³) |
|-------------------------|--------------------------------------|-------------|---|
| 1 | [30, 50, 34, 45, 35, 36] | 538.46 | 26.938 |
| 2 | [31, 38, 18, 12, 3] | 584.24 | 43.857 |
| 3 | [32, 41, 44] | 1057.49 | 46.926 |
| 4 | [39, 2, 4, 6, 49, 1] | 1009.63 | 44.415 |
| 5 | [33, 42, 43] | 687.35 | 24.818 |
| 6 | [21, 27, 28, 16, 17, 46, 47, 19, 20] | 702.53 | 40.649 |
| 7 | [10, 9, 8, 7, 5, 15, 40] | 884.95 | 36.299 |
| 8 | [29, 23, 22, 26, 24, 25] | 325.77 | 26.121 |
| 9 | [37, 11, 48, 13, 14] | 560.24 | 57.735 |
| Total Distance: 6350.66 | | | |

Table 5. Compares the best solutions obtained by GA, ACO, and EGD.

| No. of truck | GA | | ACO | | EGD | |
|---|-------------------------------|----------------|--|-----------------|--------------------------------------|-----------------|
| | GA Route | GA Length (km) | ACO Route | ACO Length (km) | EGD Route | EGD Length (km) |
| 1 | [22, 43, 17, 46, 47, 2 12] | 147.82 | [9, 1] | 111.38 | [30, 50, 34, 45, 35, 36] | 538.46 |
| 2 | [29, 36, 35, 50, 34, 33, 32] | 78.77 | [19, 3, 49, 36] | 193.93 | [31, 38, 18, 12, 3] | 584.24 |
| 3 | [31, 20, 7, 49, 9, 10] | 150.38 | [20, 10, 5, 4, 18, 13, 48] | 169.3 | [32, 41, 44] | 1057.49 |
| 4 | [23, 41, 44] | 166.11 | [33, 44] | 76.35 | [39, 2, 4, 6, 49, 1] | 1009.63 |
| 5 | [38, 19, 4, 5, 6, 48, 40, 30] | 137.71 | [30, 38, 37] | 53.74 | [33, 42, 43] | 687.35 |
| 6 | [45, 15, 1, 8, 14] | 147.39 | [2, 6, 11, 12, 15, 17, 24, 27, 28, 32, 35] | 288.11 | [21, 27, 28, 16, 17, 46, 47, 19, 20] | 702.53 |
| 7 | [13, 3, 28, 25, 24, 26, 21] | 110.90 | [29, 47, 25] | 97.01 | [10, 9, 8, 7, 5, 15, 40] | 884.95 |
| 8 | [39, 11, 18, 16, 27, 42] | 121.87 | [43, 46, 34, 42, 45, 7, 16] | 355.13 | [29, 23, 22, 26, 24, 25] | 325.77 |
| 9 | [37] | 32.48 | [26, 31, 8, 14, 21, 22, 23, 39, 40, 41] | 334.86 | [37, 11, 48, 13, 14] | 560.24 |
| Total Distance | | 1093.46 | | 1679.81 | | 6350.66 |
| Computation Time by Computer (sec) | | 1540.7213 | | 1889.60 | | 3110.01 |
| Best Cost (\$) | | 19901.1417 | | 42345519.77 | | 48930.6899 |

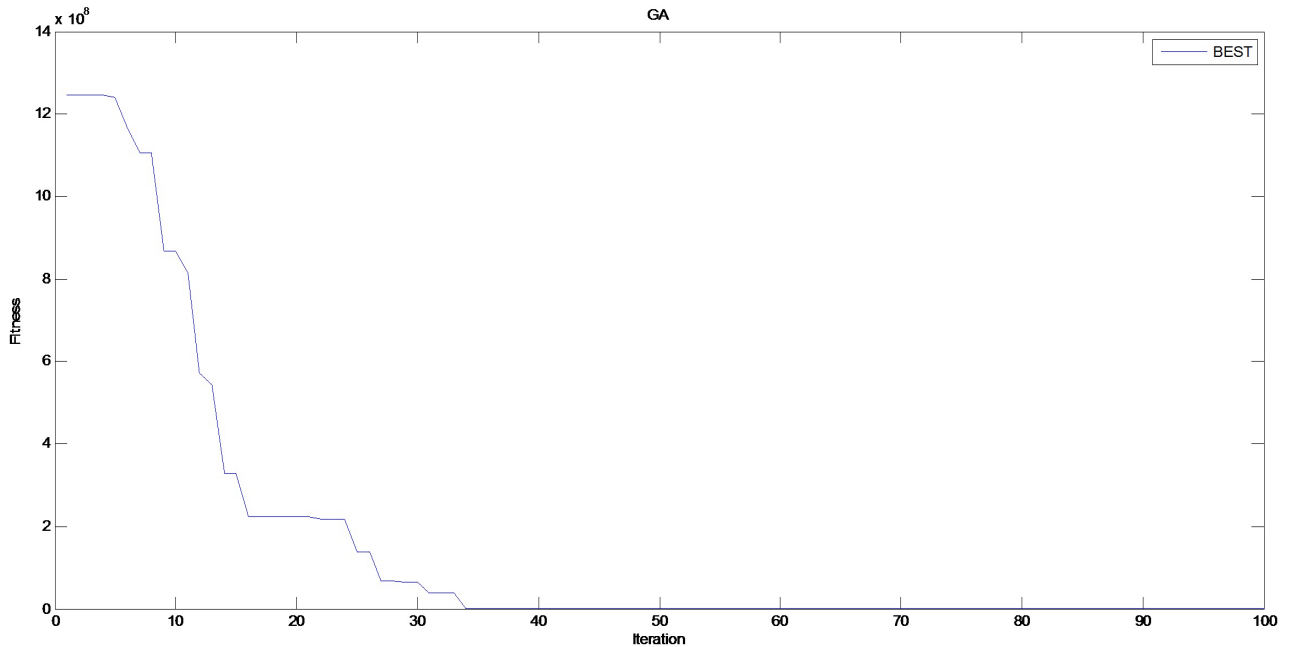


Figure 7. Cost of GA forest vehicle routing convergence graphic after 100 iterations.

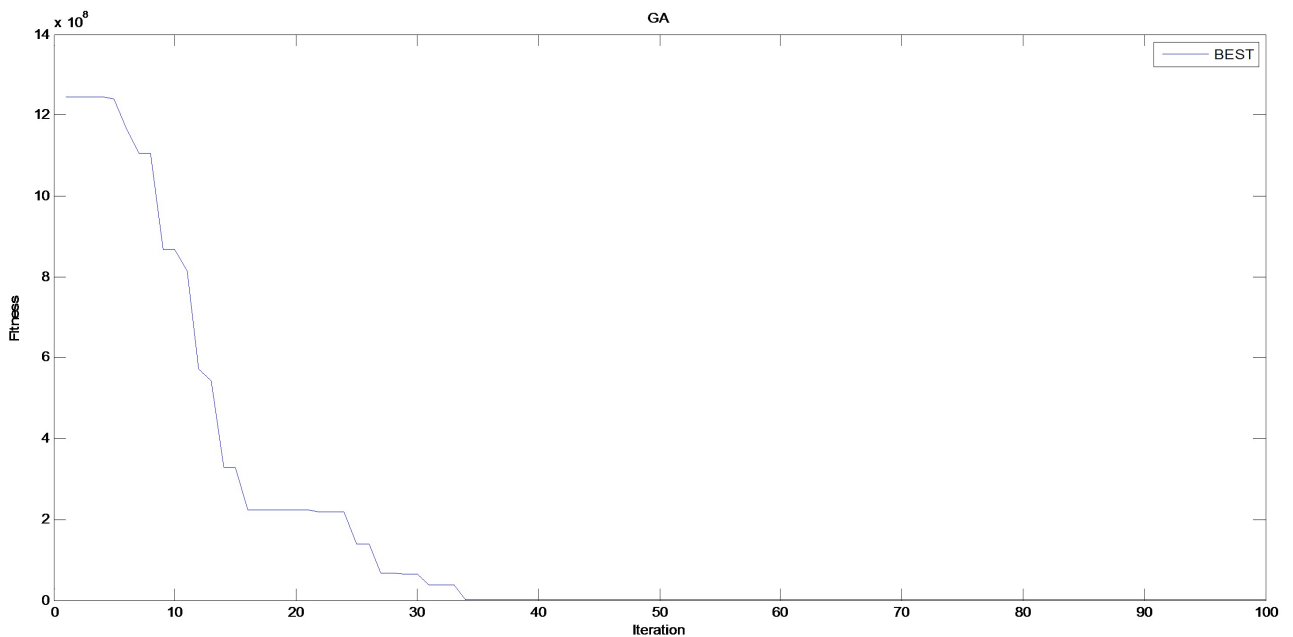


Figure 8. Cost of ACO forest vehicle routing convergence graphic after 6000 iterations.

time (1540.7213 sec). On the other hand, EGD took the computation time with 3110.01 (sec), the best cost of 48930.6899 \$, and the total distance of 6350.66 (km), and stands in the second position. ACO is the one that proposes the highest cost of these three algorithms.

Looking carefully at **Figure 1**, **Figure 3**, and **Figure 5**, we find out the cost of the forest vehicle routing for GA, ACO, and EGD converge into the best cost in different iterations: GA after around 100 iterations (see **Figure 7**); ACO after

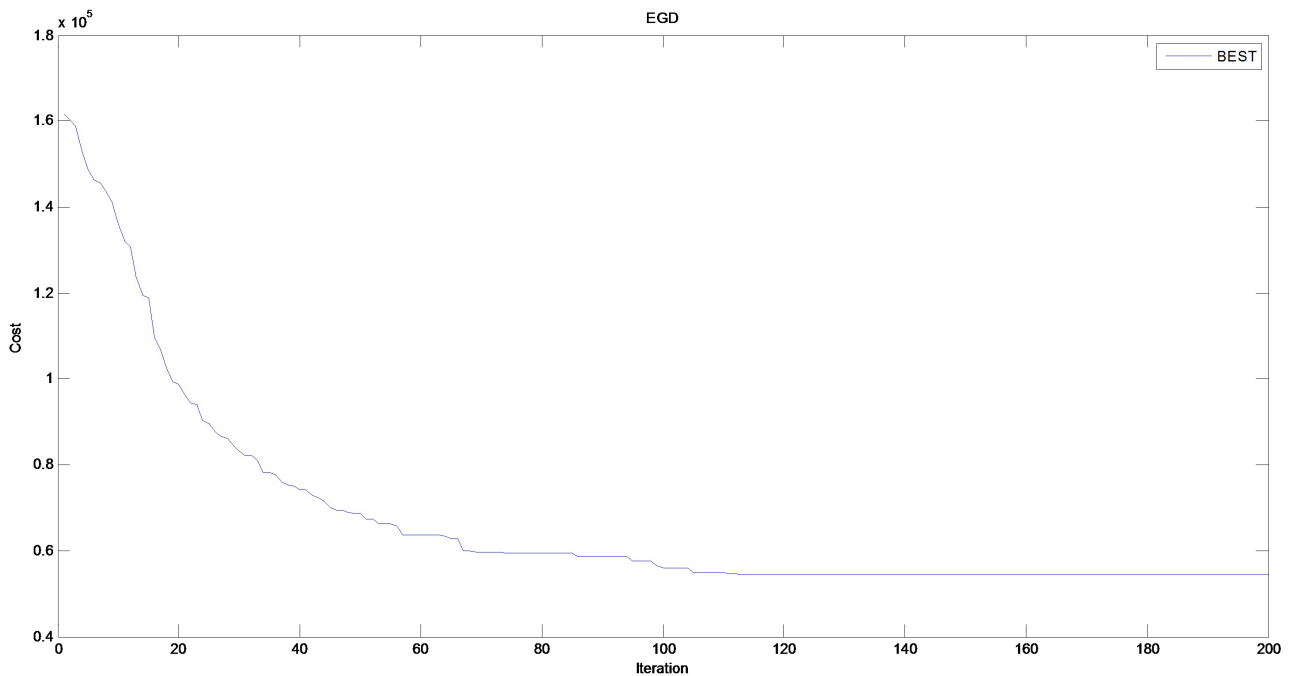


Figure 9. Cost of EGD forest vehicle routing convergence graphic after 200 iterations.

around 6000 iterations (see **Figure 8**); and EGD after around 200 iterations (see **Figure 9**). Therefore, we run MATLAB for these new iterations and compare them in **Figure 7**, **Figure 8**, and **Figure 9**.

7. Conclusion

Transportation cost is a noticeable issue for each company to reduce cost and distance. This study uses the Metaheuristics approach to solve this problem for the transportation system in Forest Vehicle Routing adopted. We applied three algorithms of Metaheuristics for this case study: Genetic Algorithm, Ant Colony Optimization, and Extended Great Deluge. The authors then compare the best cost, the total distance, and the computation time for these three algorithms. The results show that the best solution for all three items (cost, distance, and computation time) is for the GA. EGD stands in the second step for this case, and ACO comes in the following. In some related studies, Tabu Search and EGD were applied for the same case (with some minor differences), and the results of the total cost and the total length for these two algorithms are higher than GA. Therefore, GA is the preferred algorithm in this case study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Devika, K., Jafarian, A. and Nourbakhsh, V. (2014) Designing a Sustainable Closed-

- Loop Supply Chain Network Based on Triple Bottom Line Approach: A Comparison of Metaheuristics Hybridization Techniques. *European Journal of Operational Research*, **235**, 594-615. <https://doi.org/10.1016/j.ejor.2013.12.032>
- [2] Bagayoko, M., Dao, T.-M. and Ateme-Nguema, B.H. (2013) Optimization of Forest Vehicle Routing Using the Metaheuristics: Reactive Tabu Search and Extended Great Deluge. *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Agdal, 28-30 October 2013, 1-7.
- [3] Garg, P. (2009) A Comparison between Memetic Algorithm and Genetic Algorithm for the Cryptanalysis of Simplified Data Encryption Standard Algorithm. *International Journal of Network Security & Its Applications*, **1**, 34-42.
- [4] Glover, F. and Sörensen, K. (2015) Metaheuristics. *Scholarpedia*, **10**, No. 4. <https://doi.org/10.4249/scholarpedia.6532>
- [5] Yusta, S.C. (2009) Different Metaheuristic Strategies to Solve the Feature Selection Problem. *Pattern Recognition Letters*, **30**, 525-534. <https://doi.org/10.1016/j.patrec.2008.11.012>
- [6] Fahimnia, B., Davarzani, H. and Eshragh, A. (2018) Planning of Complex Supply Chains: A Performance Comparison of Three Meta-Heuristic Algorithms. *Computers and Operations Research*, **89**, 241-252. <https://doi.org/10.1016/j.cor.2015.10.008>
- [7] Badawi, U.A. and Alsmadi, M.K.S. (2013) A Hybrid Memetic Algorithm (Genetic Algorithm and Great Deluge Local Search) with Back-Propagation Classifier for Fish Recognition. *International Journal of Computer Science Issues*, **10**, 348-356.
- [8] Nahas, N., Khatab, A., Ait-Kadi, D. and Nourelfath, M. (2018) Extended Great Deluge Algorithm for the Imperfect Preventive Maintenance Optimization of Multi-State Systems. *Reliability Engineering and System Safety*, **93**, 1658-1672. <https://doi.org/10.1016/j.res.2008.01.006>
- [9] Colorni, A., Dorigo, M., Maniezzo, V. and Trubian, M. (1994) Ant System for Job-Shop Scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science*, **34**, 39-53.
- [10] Nourelfath, M., Nahas, N. and Montreuil, B. (2007) Coupling ant Colony Optimization and the Extended Great Deluge Algorithm for the Discrete Facility Layout Problem. *Engineering Optimization*, **39**, 953-968. <https://doi.org/10.1080/03052150701551461>
- [11] Dueck, G. (1993) New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *Journal of Computational Physics*, **104**, 86-92. <https://doi.org/10.1006/jcph.1993.1010>
- [12] Dorigo, M. and Gambardella, L. (1997) Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53-66. <https://doi.org/10.1109/4235.585892>
- [13] Taillard, É.D., Gambardella, L.M., Gendreau, M. and Potvin, J.-Y. (2001) Adaptive Memory Programming: A Unified View of Metaheuristics. *European Journal of Operational Research*, **135**, 1-16. [https://doi.org/10.1016/S0377-2217\(00\)00268-X](https://doi.org/10.1016/S0377-2217(00)00268-X)
- [14] Marinakis, Y. and Marinaki, M. (2007) A Bilevel Genetic Algorithm for a Real Life Location Routing Problem. *International Journal of Logistics: Research and Applications*, **11**, 49-65. <https://doi.org/10.1080/13675560701410144>