Comparison of GA-Shift Neighbourhood Mutation and GA-Pairs Exchange Mutation with Multi Cut Point Crossover in Solving RICH-VRP

Ismail Yusuf Panessai*, Sanusi2), Pratiwi Hendro Wahyudiono3), Nurdiana Daeng Pawawo4)
E-mail: ismail.lamintang@yahoo.com, sanusi@stt-ibnusina.ac.id

Abstract

This research was focused on a heterogeneous fleet of passenger ships to solve multi depot by using genetic algorithm (GA) to solve combinatorial problem i.e. vehicle routing problem (VRP). The objective of this study is to compare the roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation with roulette wheel selection, multi cut point crossover, and pairs exchange mutation to minimize the sum of the fuel consumption travelled, the cost for violations of the ship draft and sea depth, and penalty cost for violations of the load factor; maximize number port of call; and maximize load factor. Problem solving in this study is how to generate feasible route combinations for rich VRP that meets all the requirements with optimum solution. Route generated by roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation could decrease fuel consumption about 19.4350% compared to roulette wheel selection, multi cut point crossover, and pairs exchange mutation about 18.6738%.

Keywords— Vehicle Routing Problem; Genetic Algorithms; Multi Cut Point; Shift Neighbourhood Mutation; Pairs Exchange Mutation

1. INTRODUCTION

Vehicle routing problem (VRP) is a classical combinatorial optimization problem. It is a key component of transportation management. It was first introduced to determine vehicle routes with minimum cost to serve a set of customers whose geographical coordinates and demands are known in advance [1]. A vehicle is required to visit each customer only once. Typically, vehicles are homogeneous and have the same capacity restriction.

VRP can be represented as the following graph-theoretic problem. Let \( G = (P, A) \) be a complete graph where \( P = \{0, 1, ..., n\} \) is the vertex set and representing customers with the depot located at vertex 0; \( A \) is the arc set. Vertices \( j = \{1, 2, ..., n\} \) correspond to the customers, each with a known non-negative demand, \( d_j \). A non-negative cost, \( c_{ij} \), is associated with each arc \((i, j) \in A\) and represents the cost of travelling from vertex \( i \) to vertex \( j \). If the cost values satisfy \( c_{ij} \neq c_{ji} \) for all \( i, j \in P \), then the problem is said to be an asymmetric VRP; otherwise, it is called a symmetric VRP. In some contexts, \( c_{ij} \) can be interpreted as a travel time or travel cost.

The VRP consists of designing a set vehicle routes with minimum cost, defined as the sum of the costs of the routes’ arcs such that:
- all vehicle routes start and end at the same depot
- each customers in \( P \) is visited exactly once by exactly one vehicle
- some side soft and hard constraints are satisfied

MDVRP has been propose as follow each depot stores and supplies various products, and has a number of identical vehicles with the same capacity to serve customers who demand different quantities of various products [2]. Each vehicle starts the tour from its resided depot, delivers products to a number of customers, and returns to the same depot. One variant of the CVRP is the heterogeneous fleet vehicle routing problem (HVRP). In HVRP, the fleet is composed of a fixed number of vehicles with different in their equipment, capacity, age or cost
which the number of available vehicles is fixed as a priori [3]. The decision is how to best utilize the existing fleet to serve customer demands.

Forward six formulations used by Yaman [4] which are enhanced by valid inequalities and lifting; Choi & Tcha [5] present a linear programming relaxation of which is solved by the column generation technique and used column generation technique which is enhanced by dynamic programming schemes; a branch-cut-and-price algorithm over an extended formulation that capable for solving HVVRP proposed by Pessoa, et.al. [6] and a tabu search used approach using GENIUS for HVVRP [7].

Developing an algorithm based on heuristics and followed by a local search procedure based on the steepest descent local search and tabu search [8] while three phase heuristic developed by Dondo, et.al. [9] and an iterated local search base on heuristic proposed by Penna, et.al. [10]. Hybrid algorithm that composed by an iterated local search based on heuristic and a set partitioning formulation proposed by Subramanian, et.al. [11]. The set partitioning model was solved by means of a mixed integer programming solver that interactively calls the iterated local search heuristic during its execution.

An evolutionary hybrid meta-heuristic which combines a parallel genetic algorithm with scatter search presented by Ochi, et.al. [12], while a record-to-record travel metaheuristic published by Li, et.al. [13]. In addition, memetic algorithm to solve HVVRP is used by Prins [14].

HVVRP has been solved by implementing a threshold accepting procedure where a worse solution is only accepted if it is within a given threshold [15]; and provided an improved version in Tarantilis, et.al. [16]. A memory programming metaheuristic proposed by Li, et.al. [17] While tabu search algorithm to solve HFVRP used by Brandão [18].

Several simple heuristic has been developed by Nag, et.al. [19] and more advanced heuristic proposed by Chao, et.al. [20], tabu search by Cordeau & Laporte [21] while memetic algorithm to solve SDCVRP used by Nagata & Bräysy [22].

AVRP is related to Asymmetric Travelling Salesman Problem (ATSP). It is a generalized travelling salesman problem in which distances between a pair of cities need not equal in the opposite direction. The ATSP is an NP-hard problem, thus many meta-heuristic algorithms have been proposed to solve this problem, such as hybrid genetic algorithm by Choi, et.al. [23] and tabu search proposed by Basu, et.al. [24].

2. VEHICLE ROUTING PROBLEM MODEL

This study is on a heterogeneous fleet of passenger ships to solve multi depot. The objective of our problem is to consist of:

i. Minimum fuel consumption

The fuel consumption of each vehicle depends on the type is related with the type of engine used proposed by Ismail et. al. [25]:

\[ f^k = \eta \times P^k \times \Phi \times T^k \times \mu \]  

\[ f^k \] = Total fuel consumption served by ship \( k \)
\[ T^k \] = Total voyage time by ship \( k \)
\[ L^k_r \] = Total distance travelled for route \( r \) served by ship \( k \)
\[ v^k \] = Speed of ship \( k \)
\[ \eta \] = High Speed Diesel constant (0.16)
\[ P^k \] = Engine power of ship \( k \) (HP)
\[ \Phi \] = Number of engine
\[ M \] = Efficiency (0.8)

ii. Maximum number port of call

Number port of call of the route \( r \) that served by ship \( k \) donated by

\[ \xi^k_r \]
iii. Maximum load factor
Load factor of ship $k$ in each path calculated by:

$$b_{ij}^k = \frac{\gamma_{ij}^k}{q_k}$$  \hspace{1cm} (2)

- $b_{ij}^k$ = Load factor of ship $k$ sailing from port $i$ to port $j$ in route $r$
- $\gamma_{ij}^k$ = Number of passenger in ship $k$ sailing from port $i$ to port $j$ in route $r$
- $q_k$ = Capacity of ship $k$

Hard constraints are dealt with by removing unfeasible route. Hard constraints in this study include:

i. Fuel Port
A route must include at least one fuel port.

ii. Travel Time
The maximum duration for each tour is called commission days, $T$ which is 14 days for this case. Hence, ship must return to the depot within $T$. If $T^k$ is the ship’s voyage time, then $T^k \leq T$.

iii. Travel Distance
Since each ship has a different fuel tank size. Hence, total distance travelled for route $r$ served by ship $k$, $L^k_r$ that it can travel is different. If $L^k$ is maximum allowed routing distance for ship $k$, then $L^k_r \leq L^k$. $L^k$ calculated by:

$$L^k = \frac{\theta^k \cdot \nu^k}{\eta \cdot P^k \cdot \Phi^k \cdot \mu} - (v^k \cdot 24)$$  \hspace{1cm} (3)

Where,
- $L^k$ = Maximum allowed routing distance for ship $k$
- $\theta^k$ = Maximum capacity tank of the ship $k$
- $\nu^k$ = Speed of ship $k$
- $\eta$ = High Speed Diesel constant (0.16)
- $P^k$ = Engine power of ship $k$ (HP)
- $\Phi^k$ = Number of engine used in ship $k$
- $\mu$ = Efficiency (0.8)

A. Mathematical model
Let, $G = (P, A)$ be a graph, where $P$ is the set of all ports, denoted by the nodes $C$ (customer ports) and $D$ (fuel ports) at which $K$ is a set mix vehicles with capacity $q_k$ are based. $A = \{(i, j) \mid i, j; i < j\}$ is the set of arcs. Every arc $(i, j)$ is associated with a non-negative distance matrix $L= l_{ij}^k$, which represents the asymmetric travel distance from port $i$ to port $j$, i.e., $l_{ij}$ may be different from $l_{ji}$; $i, j \in P$.

In order to present the mathematical formulation of the models, we use the following:

B. Notation
- $C = \{1, 2, ..., m\}$ is a set of customer ports
- $D = \{(m+1), (m+2), ..., (m+n)\}$ is a set of fuel ports
- $P = C \cup D = \{1, 2, ..., m, (m+1), (m+2), ..., (m+n)\}$ is the set of all ports; $n(P) = \text{number of the ports}$
- $K = \{1, 2, ..., k\}$ is a set of ship; $n(K) = \text{number of the ships}$

C. Parameter
- $h_i = $ Sea depth of port $i, i \in \{1, 2, ..., m+n\}$
\[ v^k = \text{Speed of ship } i \]
\[ \delta^k = \text{Ship draft of ship } i; \; i \in \{1, 2... n \} \]
\[ r^k_i = \text{Route } i \text{ for ship } k \]
\[ f_{ij}^k = \text{Fuel consumption for ship } k \text{ to sail from port } i \text{ to port } j \]
\[ f_r^k = \text{Fuel consumption for ship } k \text{ to serve route } r \]
\[ t_{ij}^k = \text{Travel time for ship } k \text{ sailing from port } i \text{ to port } j \]
\[ T_{ij}^k = \text{Travel time for ship } k \text{ sailing from port } i \text{ to port } j \text{ and stay in port } i \]
\[ T^k = \text{Total voyage time by ship } k \]
\[ T = \text{Maximum allowed routing time (commission days)} \]
\[ l_{ij}^k = \text{Distance travelled for ship } k \text{ sailing from port } i \text{ to port } j; \; l_{ij} \text{ may be different from } l_{ji} \]
\[ L_{ij}^k = \text{Distance travelled for ship } k \text{ sailing from port } i \text{ to port } j \text{ and back to port } i \]
\[ L_r^k = \text{Total distance travelled for route } r \text{ served by ship } k \]
\[ L^k = \text{Maximum allowed routing distance for ship } k \]
\[ b_{ij}^k = \text{Load factor for ship } k \text{ sailing from port } i \text{ to port } j \]
\[ B_{ij}^k = \text{Average load factor for ship } k \text{ sailing from port } i \text{ to port } j \text{ and back to port } i \]
\[ b_r^k = \text{Average load factor for route } r \text{ served by ship } k \]
\[ q_{ij}^k = \text{Available seat capacity of the ship } k \text{ travel from ports } i \text{ to } j \]
\[ \gamma_{ij}^k = \text{Number of passenger on board, travel from ports } i \text{ to } j \]
\[ \alpha = \text{Penalty cost for violations of the ship draft and sea depth} \]
\[ \beta = \text{Penalty cost for violations of the load factor} \]
\[ \xi = \text{Number port of call} \]

D. Decision variables

\[ u_{ki} = \begin{cases} 1 & \text{if ship } k \text{ is used for serving route } i \\ 0 & \text{otherwise} \end{cases} \]

\[ w_{r,i} = \begin{cases} 1 & \text{if port } i \text{ is served by ship } k \text{ in route } r \\ 0 & \text{otherwise} \end{cases} \]

\[ \alpha = \begin{cases} 500 & \text{if ship } k \text{ with ship draft } \delta_k \text{ sailing from port } i \text{ with sea depth } h_i, \; \text{where } \delta_k \geq h_i \\ 0 & \text{otherwise} \end{cases} \]

\[ \beta = \begin{cases} 1000 & B_{ij}^k < 50 \\ 500 & 50 \leq B_{ij}^k \leq 75 \\ 0 & \text{otherwise} \end{cases} \]

The problem is to construct route with minimum fuel consumption in feasible set of routes for each vehicle. The feasible route for ship \( k \) is to serve ports without exceeding the constraints:

1. Total travel time \( T^k \) for any vehicle is no longer than \( T \)
2. Total travel distance \( L^k \) for any vehicle is no longer than \( L^k \)
3. The feasible route must include at least one fuel port

The mathematical formulation is given in:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} f_{ij}^k u_{ij}^k + \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} \alpha . u_{ij}^k + \sum_{k \in K} \sum_{i \in P} \beta . u_{ij}^k \\
\text{maximize} & \quad \sum_{i \in P} \frac{x_i}{5} \\
\text{maximize} & \quad \sum_{i \in P} B_{ij}^k \\
\end{align*}
\]

1. All ports (customer and fuel port) \( i \) are serviced by ship \( k \) minimum at once

\( \sum_{k \in K} \sum_{i \in P} u_{ij}^k \geq 1, \quad \forall i \in P, \forall k \in K \) (7)

2. Travel time of the ship \( k \) is no longer than the maximum allowed routing time \( T \), \( T = 14 \) days.

\( \sum_{k \in K} T_i^k \leq T \) (8)

3. Total distance travelled for route \( i \) served by ship \( k \) is no longer than the maximum allowed routing distance of the ship \( k \), then \( L_i^k \leq L_k \).

\( \sum_{k \in K} L_i^k \leq L_k \) (9)

4. Travel time of ship \( k \) equals to the distance travelled and divided by running speed \( v^k \).

\( T_i^k = \frac{L_i^k}{v^k} \) (10)

5. The vehicle capacity constraint

\( \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} q_i \cdot u_{ij}^k \leq q_j^k \) (11)

6. Ship \( k \) with ship draft \( \delta_k \) sailing from port \( i \) with sea depth \( h_i \) and it is equal to \( \alpha \).

\( \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} x^k_{ij} = \alpha \) (12)

7. Ship \( k \) sailing from port \( i \) to port \( j \) and back to port \( i \) by average load factor \( B_{ij}^k \) and it is equal to \( \beta \).

\( \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} B_{ij}^k = \beta \) (13)

8. Route \( r \) served by ships \( k \) should possess a fuel-port

\( \sum_{k \in K} \sum_{i \in P} \sum_{j \in P} p \cdot u_{ij}^k \geq 1 \) (14)

Three objectives in our study are minimum fuel consumption, maximum port of call and maximum load factor. All objective tested in different scenarios.

- Minimum fuel consumption

In this case the fitness value is the total fuel consumption of each ship. Obviously it is a minimization problem, thus the smallest value is the best. The fitness function represents as Eq. (15):

\[
f = \frac{1}{\sum_{i \in P} f_i^k + 1} * 1,000,000
\]

\( f_r^k = \) Fuel consumption for ship \( k \) to serve route \( r \)
• **Maximum number port of call**

In this case the fitness value is the total port of call of each route. Obviously it is a maximization problem, thus the largest value is the best. The fitness function represents as Eq. (16):

$$
    f = \sum \xi^k_r 
$$

$\xi^k_r$ = Number port of call of the route $r$ that served by ship $k$

• **Maximum load factor**

In this case the fitness value is the average of load factor of each route. Obviously it is a maximization problem, thus the largest value is the best. The fitness function represents as Eq. (17):

$$
    f = \sum B^k_r 
$$

$B^k_r$ = Average load factor for route $r$ served by ship $k$

In this research, a classical selection method is used for solving routing problems, namely the roulette wheel selection. Selection process begins by spinning the roulette wheel $n$ times; each time, a single chromosome is selected for a new population in the following 2 steps:

**Step 1:** Generate a random number $r$ in a range $[0, 1]$.

**Step 2:** If $r \leq q_s$, then select the first chromosome $s_1$ otherwise, select the $s$-th chromosome $(1 \leq s \leq n)$ such that $q_{s-1} < r \leq q_s$.

---

**E. Roulette wheel selection**

Roulette wheel selection was selecting a new population with respect to the probability distribution based on their fitness values. The roulette wheel selection can be constructed as follows:

The roulette wheel selection can be constructed as follows:

- Calculate the fitness value $f_s$ of each chromosome $s$:

  $$
  f_s = f(x) 
  $$

- Calculate the total fitness of population:

  $$
  F = \sum_{s=1}^{n} f_s 
  $$

- Calculate the selection probability $p_s$ of each chromosome:

  $$
  p_s = \frac{f_s}{F} 
  $$

- Calculate the cumulative probability $q_s$ of each chromosome $s$:

  $$
  q_s = \sum_{i=1}^{s} p_i 
  $$

$n = \text{Number of population}$

$s = 1, 2, 3 \ldots n$

---

**F. Mutation**

There are two important things in mutation i.e determine which chromosome for mutation and mutation processes.

To determine a chromosome in-mutation is started by generate random number as many as the number of population and multiple number of P-arm. Random number is generated and compared with the value of mutation rate. If the random number is less than the mutation rate, then the chromosome is selected for mutation process. The process of mutation is to exchange portions of a chromosome in the same chromosome are eligible for in-mutation. There are two mutation methods used i.e. pairs exchange and shift neighbourhood mutation. Both of the mutation applied for comparing the best fitness in the end of generation. Fig.1 and Fig.2 are the description of the two mutations used for this research.
Fig. 1. Pairs exchange mutation

Fig. 2. Shift neighbourhood mutation

Fig. 3. Multi cut point crossover

G. Crossover

The type of crossover method used is multi cut point crossover. Fig. 3 is the description of the multi cut point crossover.
In order to show the effectiveness of GA, simulations were carried out. The algorithm proposes coded in java and, using a Intel(R) Core(TM) i5 CPU M430 @ 2.27GHz. As methods compared have a stochastic behaviour, they have been tested 50 times on each benchmark for every GA operator.

**TABLE 1. BENCHMARK**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Number of</th>
<th>Benchmark</th>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Port</td>
<td>Vehicle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td>Fuel</td>
<td></td>
</tr>
<tr>
<td>40c-9d-8k</td>
<td>40</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>28c-9d-9k</td>
<td>28</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>45c-11d-11k</td>
<td>45</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>32c-4d-8k</td>
<td>32</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>34c-11d-11k</td>
<td>34</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>63c-14d-11k</td>
<td>63</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>18c-6d-8k</td>
<td>18</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>28c-6d-11k</td>
<td>28</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>12c-4d-8k</td>
<td>12</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>53c-12d-11k</td>
<td>53</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>24c-5d-10k</td>
<td>24</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Determination of combinations of parameter values: the probability of crossover, the probability of mutation, and the number of populations, is generally determined by "random" through experimental methods throughout the study Kumar & Panneerselvam [26] and by Bae & Moon [27]. The parameters of the genetic algorithm at one particular value were also determined by Kocha, et.al. [28]. The probability of crossover is set to 0.7; 0.8; 0.9 and 0.95 (high), and the mutation probability value set at 0.05; 0.2 and 0.3 (low), as well as with a population of 50 & 100.

The mutation probability values to vary by 0; 0.1; 0.2 so that 1 (from the lowest to the highest), and the number of populations varies from 8 to 100. Then, the combination of the mutation probability values and the number population set one by one, while the value for crossover probability is set to 0 by Mungwattana [29] and also proposed by Volna [30]. While Ghani, et.al. [31] use very small mutation probability values (ie below 0.1), but with a large probability of crossover ranges from 0.9).

In this research, GA parameters used are population size: 100, maximum generation: 1000, crossover rate: 0.7, mutation rate: 0.5 and selection rate: 0.5. In addition, roulette wheel selection, multi cut point crossover and two type of mutation used, namely pairs exchange mutation and shift neighbourhood mutation.

4. RESULTS AND DISCUSSION

To check the quality of solution obtained over algorithms in 11 benchmarks then check efficiency was done. In this section, a computational study is carried out to study about performance of GA (GA Var.3 and GA Var.4) compared to the best know result and heuristic for solving our problem.
TABLE 2. FUEL CONSUMPTION OVER 11 BENCHMARKS

<table>
<thead>
<tr>
<th>Code</th>
<th>Benchmark</th>
<th>Fuel Consumption</th>
<th>Genetic Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best Known (real life)</td>
<td>Heuristic</td>
</tr>
<tr>
<td>a</td>
<td>40c-9d-8k</td>
<td>1,275,883</td>
<td>1,122,712</td>
</tr>
<tr>
<td>b</td>
<td>28c-9d-9k</td>
<td>2,375,323</td>
<td>2,064,836</td>
</tr>
<tr>
<td>d</td>
<td>32c-4d-8k</td>
<td>1,036,758</td>
<td>919,118</td>
</tr>
<tr>
<td>e</td>
<td>34c-11d-11k</td>
<td>2,743,105</td>
<td>2,377,556</td>
</tr>
<tr>
<td>f</td>
<td>63c-14d-11k</td>
<td>4,755,085</td>
<td>4,095,004</td>
</tr>
<tr>
<td>h</td>
<td>18c-6d-8k</td>
<td>1,491,149</td>
<td>1,308,901</td>
</tr>
<tr>
<td>j</td>
<td>28c-6d-11k</td>
<td>2,134,324</td>
<td>1,858,045</td>
</tr>
<tr>
<td>k</td>
<td>12c-4d-8k</td>
<td>2,945,322</td>
<td>2,549,070</td>
</tr>
<tr>
<td>l</td>
<td>53c-12d-11k</td>
<td>1,267,387</td>
<td>1,116,445</td>
</tr>
<tr>
<td>m</td>
<td>24c-5d-10k</td>
<td>1,267,387</td>
<td>1,116,445</td>
</tr>
</tbody>
</table>

GA Var.3 = Shift Neighbourhood Mutation
GA Var.4 = Pairs Exchange Mutation

The quality of solution obtained by GA in 11 benchmarks was checked by:

\[
\text{Algorithm Efficiency} = \frac{\text{Alg. proposed - Best known solution}}{\text{Best know solution}} \times 100\% \quad (22)
\]

Based on the Table 3, the average of efficiency algorithm over 11 benchmarks between roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation (GA Var.3) is about 19.4350%. and roulette wheel selection, multi cut point crossover, and pairs exchange mutation (GA Var.4) is about 18.6738%. It seems the best performance of GA algorithm by roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation (GA Var.3).

TABLE.3. PERCENTAGE OF EFFICIENCY FOR FUEL CONSUMPTION OVER 11 BENCHMARKS

<table>
<thead>
<tr>
<th>Code</th>
<th>Benchmark</th>
<th>Fuel Consumption</th>
<th>Genetic Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Best Known (real life)</td>
<td>Heuristic</td>
</tr>
<tr>
<td>a</td>
<td>40c-9d-8k</td>
<td>0</td>
<td>12.0051</td>
</tr>
<tr>
<td>b</td>
<td>28c-9d-9k</td>
<td>0</td>
<td>13.0714</td>
</tr>
<tr>
<td>c</td>
<td>45c-11d-11k</td>
<td>0</td>
<td>13.6628</td>
</tr>
<tr>
<td>d</td>
<td>32c-4d-8k</td>
<td>0</td>
<td>11.3469</td>
</tr>
<tr>
<td>e</td>
<td>34c-11d-11k</td>
<td>0</td>
<td>13.3261</td>
</tr>
<tr>
<td>f</td>
<td>63c-14d-11k</td>
<td>0</td>
<td>13.8816</td>
</tr>
<tr>
<td>h</td>
<td>18c-6d-8k</td>
<td>0</td>
<td>12.2220</td>
</tr>
<tr>
<td>j</td>
<td>28c-6d-11k</td>
<td>0</td>
<td>12.9446</td>
</tr>
<tr>
<td>k</td>
<td>12c-4d-8k</td>
<td>0</td>
<td>11.8293</td>
</tr>
<tr>
<td>l</td>
<td>53c-12d-11k</td>
<td>0</td>
<td>13.4536</td>
</tr>
<tr>
<td>m</td>
<td>24c-5d-10k</td>
<td>0</td>
<td>11.9097</td>
</tr>
<tr>
<td>Average</td>
<td>0</td>
<td>12.6957</td>
<td>19.4350</td>
</tr>
</tbody>
</table>

GA Var.3 = Shift Neighbourhood Mutation
GA Var.4 = Pairs Exchange Mutation
5. CONCLUSIONS

In this paper, the MDVRP, HFMVRP, SDCVRP and AVRP were studied and it is combined to solve ship routing. The best routing is minimum fuel consumption, maximum number port of call and maximum load factor. In order to validate our algorithms, the mathematical programming model was applied to 11 benchmarks. A computational study is carried out to compare of fuel consumption between roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation to roulette wheel selection, multi cut point crossover, and pairs exchange mutation.

Result shows the best performance algorithm is GA Var.3. Route generated by GA Var.3 could decrease fuel consumption about 19.4350% compared to GA Var.2 about 18.6738%.

This phenomenon proves that the GA proposed effectively used for solve our problem. Which the effective operator used are roulette wheel selection, multi cut point crossover, and shift neighbourhood mutation.

ACKNOWLEDGMENT

This work was supported in part by Grant RG078-11ICT.

REFERENCES