DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

PENINGKATAN OPTIMASI RUTE KENDARAAN: KOMBINASI ALGORITMA GENETIKA DENGAN PENDEKATAN *FIXED RADIUS NEAR NEIGHBOR*

ENHANCING VEHICLE ROUTING OPTIMIZATION: HYBRID GENETIC ALGORITHMS WITH FIXED RADIUS NEAR NEIGHBOR APPROACH

Nur Iksan[1], Ismail Yusuf Panessai[2], Pamor Gunoto[3], Anton Viantika[4], Hidayah Novitasari[5]
[1,3-5]*(Teknik Elektro, Fakultas Teknik, Universitas Riau Kepulauan, Indonesia)*
[2]*(Faculty of Computing & Meta-Technology, Universiti Pendidikan Sultan Idris, Malaysia)*
[1]*nur.iksan@ft.unrika.ac.id*

Abstrak

Artikel ini menyelidiki optimasi Vehicle Routing Problem menggunakan dua pendekatan berbeda: Genetic Algorithm (GA) dan Fixed Radius Near Neighbour-Genetic Algorithm (FRNN-GA). Masalah yang dihadapi melibatkan optimalisasi kendaraan multi-depot, heterogen, dan Vehicle Routing Problem (VRP) asimetris, yang merupakan tugas penting dalam logistik dan manajemen operasi. Studi ini mengevaluasi kinerja kedua algoritma dengan menganalisis nilai yang diperoleh dari generasi ke generasi dan membandingkan kualitas solusinya dalam hal pengurangan konsumsi bahan bakar. Hasilnya menunjukkan bahwa GA hibrid secara konsisten mengungguli GA umum, mencapai nilai fungsi tujuan yang jauh lebih rendah dan menunjukkan konvergensi yang lebih efisien terhadap solusi kompetitif. Temuan ini menggarisbawahi efektivitas hibridisasi algoritma genetika dengan Fixed Radius Near Neighbor (FRNN) dalam meningkatkan kualitas solusi dan efisiensi konvergensi. Studi ini memberikan kontribusi wawasan berharga ke dalam metodologi optimasi dan memberikan landasan untuk penelitian lebih lanjut dalam desain dan penerapan algoritma.

Kata Kunci; Optimization; Rich VRP; Heuristic

Abstract

This article investigates the optimization of a Vehicle Routing Problem using two distinct approaches: a general genetic algorithm (GA) and a Fixed Radius Near Neighbor - Genetic Algorithm (FRNN-GA). The problem at hand involves optimizing a multi-depot, heterogeneous vehicle, and asymmetric Vehicle Routing Problem (VRP), a crucial task in logistics and operations management. The study evaluates the performance of both algorithms by analyzing the values obtained across successive generations and comparing their solution qualities in terms of fuel consumption reduction. Results indicate that the hybrid GA consistently outperforms the general GA, achieving significantly lower objective function values and demonstrating more efficient convergence to competitive solutions. The findings underscore the effectiveness of hybridizing genetic algorithms with Fixed Radius Near Neighbor (FRNN) in enhancing solution quality and convergence efficiency. This study contributes valuable insights into optimization methodologies and provides a foundation for further research in algorithm design and application.

Keywords; *Optimization; Rich VRP; Heuristic*

INTRODUCTION

The Rich Vehicle Routing Problem (RVRP), which encompasses multidepot, asymmetric, and heterogeneous characteristics, has been the subject of extensive research in recent years. (Vidal et al., 2012) introduced a hybrid genetic algorithm for multidepot and periodic vehicle routing problems, demonstrating impressive computational efficiency and solution quality. Similarly, (Yldrm & Çatay, 2015) highlighted the increasing trend towards matheuristics in recent literature, emphasizing their effectiveness in providing rapid and high-quality solutions for vehicle routing

problems. (Shi et al., 2020) developed a heuristic solution method for multi-depot vehicle routing-based waste collection problems, incorporating adaptive large neighborhood search and variable neighborhood search. These recent studies underscore the diverse and evolving nature of algorithmic approaches to address the complexities of the RVRP.

Furthermore, (Perboli et al., 2011) emphasized the significance of addressing flow movement issues in the literature, shedding light on the broader context of the RVRP and its implications for logistics and operations. Additionally, recent works by (Keçeci et al., 2021) and (Miao et al., 2012) have focused on heuristic approaches and hybrid algorithms to tackle specific variations of the vehicle routing problem, including heterogeneous fleet configurations and three-dimensional loading constraints. These contributions reflect ongoing efforts to develop tailored solutions for the intricate challenges posed by the RVRP.

Moreover, the literature offers insights into specific aspects of the RVRP. For instance, (Masmoudi et al., 2018) highlighted the importance of addressing CO_2 emissions through the use of alternative fuel vehicles, which is particularly relevant in the context of the RVRP's requirement for a heterogeneous fleet. Additionally, recent studies by (Yuan et al., 2021) and (Agany Manyiel et al., 2021) have focused on enhancing genetic algorithms and addressing issues such as premature convergence, further advancing the state-of-the-art in algorithmic approaches for the RVRP.

In summary, recent research in the field of vehicle routing problems, particularly the rich vehicle routing problem, has witnessed significant advancements in algorithmic approaches, heuristic methods, and the consideration of specific problem variations. These contributions underscore the dynamic and evolving nature of research in addressing the complexities of the RVRP, paving the way for further advancements in optimization and logistics.

To investigate the impact of using a fixed radius neighbor in generating the initial population of a genetic algorithm for the rich vehicle routing problem, it is essential to review the existing literature on vehicle routing problems (VRPs) and genetic algorithms. The VRP is a complex combinatorial optimization problem, and the use of metaheuristic algorithms, such as genetic algorithms, has been widely explored in the literature (Tao et al., 2022). The study by demonstrated the effectiveness of a genetic algorithm in solving a vehicle routing problem, indicating the potential for improvement by incorporating fixed radius neighbor selection (Li et al., 2020). Additionally, the work by presented a general heuristic for vehicle routing problems, providing insights into the development of heuristic methods for addressing VRPs (Pisinger & Ropke, 2007). Furthermore, the review by highlighted the significance of considering variants of the vehicle routing problem, such as those with time windows and capacitated constraints, which are relevant to the rich VRP (Kumar & Panneerselvam, 2012).

Moreover, the study by emphasized the integration of genetic algorithms with neighborhood selection mechanisms for managing diversity in vehicle routing problems, indicating the potential for enhancing the performance of genetic algorithms through neighborhood-based approaches (Shi et al., 2020). Additionally, the research by extended the classic vehicle routing problem by introducing new precedence constraints, demonstrating the continuous evolution of problem formulations and solution approaches in the field of VRPs (Bahalke et al., 2022). Furthermore, the work by addressed a multi-depot heterogeneous vehicle routing problem, which aligns with the

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

characteristics of the rich VRP, emphasizing the importance of considering heterogeneous fleets in routing optimization (Panessai et al., 2019) and (Zhou et al., 2021).

In the context of genetic algorithms, the study by provided a comprehensive classification and review of VRPs, highlighting the diverse problem characteristics and solution methodologies, which can inform the development of tailored genetic algorithm-based approaches for specific VRP variants (Braekers et al., 2016). Additionally, the research by surveyed recent works on VRPs and metaheuristic methods, providing insights into the application of metaheuristics in addressing complex routing problems, including the rich VRP (Tao et al., 2022).

By synthesizing the insights from these references, it is evident that the use of a fixed radius neighbor in generating the initial population of a genetic algorithm has the potential to significantly enhance the performance of the algorithm for addressing the rich vehicle routing problem. The literature review provides a foundation for further research and development of genetic algorithm-based approaches tailored to the specific characteristics of the rich VRP, with the objective of minimizing fuel consumption.

PROBLEM DESCRIPTION

A fleet of ships pick up and drop passenger at designated ports. The ships are of different types. They can only refuel in certain ports, called fuel ports. So in a route, the ships must visit at least one fuel port. In a VRP, the fuel ports are similar to depots. There are distance and time constraints of ships in serving a route.

The objective of this VRP is to minimize the number of transits. Suppose s is the total number of ports in route k, d_{ij} is the mile distance from port i to port j and, f^k is number of fuel consumption of ship k. Then the total fuel consumption of routes is:

$$\sum_1^k \sum_{i=1,j=i+1}^{s} (d_{ij}.f^k) \quad (1)$$

Constraints

The constraints for the problem are maximum voyage time, maximum mileage of voyage, and port served. The maximum voyage time of ships is 336 hours, determined by the company. Suppose d_{ij} is the distance between port i to port j, and v^k is the speed of the ship k, then voyage time of ship k from port i to port j is:

$$T_{ij}^k = \frac{d_{ij}}{v^k} \quad (2)$$

Route r consists of s port, and then voyage time of ship k in serving the route r is:

$$T_r^k = \sum_{i=1,j=i+1}^{s} T_{ij}^k \quad (3)$$

If maximum voyage time is T, then:

$$T_r^k \leq T \quad (4)$$

So that the maximum voyage time of ship k to serve route r is calculated by:

$$\sum_{i=1,j=i+1}^{s} \frac{d_{ij}^k}{v^k} \leq T \quad (5)$$

Maximum mileage of voyage is the maximum mileage that a ship can take after refuelling. It is calculated from a fuel port to another fuel port in a route. Suppose maximum fuel tank capacity of ship k is q^k. Speed of the ship is v^k and power of its engine is p^k. The number of engine operated is n^k with μ is efficiency of the engine and η is factor constant. Then maximum mileage of ship k is given in (6) (Pertambangan Minyak Nasional, 2008).

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

$$L^k \leq \frac{q^k.v^k}{\eta.p^k.n^k.\mu} \quad (6)$$

A route must include at least one fuel port.

$$\sum_{i=1}^{s} f_i \geq 1 \quad (7)$$

where,f_i are coefficients with the following property:

$$f_i = \begin{cases} 1, & \text{if port } i \text{ is a fuel port} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Each port must be served at least once. For given p, where p is number of port must satisfy the condition

$$\sum_{i=1}^{p} a_i = p \quad (9)$$

where, a_i are coefficients with the following property:

$$a_i = \begin{cases} 1, & \text{if port } i \text{ is served} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Mathematical Model

Let, $G = (P, A)$ be a graph, where $P = \{1, 2, ..., p\}$ is the index set of ports (nodes) and $A = \{(i, j) \mid i < j\}$ is the set of arcs (links). Every arc (i, j) is associated with a distance matrix $D = d_{ij}$, which represents the asymmetric travel distance from port i to port j, i.e., d_{ij} is not necessarily equals to d_{ji}.

In order to present the mathematical formulation of the models, we use the following nomenclature and decision variables.

P = the index of a set of all ports, $P = \{1, 2, \ldots, p\}$
K = the index of a set of ships, $K = \{1, 2, \ldots, k\}$
S = the index of a set of ports in a route, $S = \{1, 2, \ldots, s\}$
v^k = speed of ship k
q^k = maximum fuel tank capacity of ship k
p^k = power of the ship k engine
n^k = the number of engine operating on ship k
μ = efficiency of engine
η = constant factor
T = maximum allowed routing time (*commission days*)
t_i = spent time at port
d_{ij} = distance port i to port j; d_{ij} is not necessary equals to d_{ji}
o_{ij} = number of passengers travelling from port i to port j
x = the total number of passengers
x_n = the number of not transit passengers
x_t = the number of transit passengers
e = effectiveness of routes
n_{ij} = binary variable denoting that segment from i to j is served once or more (1), or otherwise (0)
a_i = binary variable denoting that port i is served (1), or otherwise (0)
f_i = binary variable denoting that port i to j is a fuel port (1), or otherwise (0)

To construct a route with minimum passenger transit, we have to find a feasible set of routes for each ship. A feasible route for ship k is to serve ports must satisfy the following constraints:

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

1. Total travel time T_r^k for any ship is at most T.
2. Total travel distance from a fuel port to another is at most L^k.
3. The feasible route must include at least one fuel port.

The mathematical formulation of the objective of minimizing fuel consumption is:

$$min \sum_1^k \sum_{I=1,j=i+1}^{s} (d_{ij}.f^k) \quad (11)$$

Subject to:

1. All ports i are served at least once.
$$\sum_{i=1}^{p} a_i = p \quad (12)$$
2. Travel time of ship k is not longer than the maximum allowed routing time (T).
$$\sum_{I=1,j=i+1}^{s} \frac{d_{ij}^k}{v^k} + \sum_{i=1,j=i+1}^{s} \frac{d_{ji}^k}{v^k} + \sum_{i=1}^{s} t_i \leq T \quad (13)$$
3. The distance between fuel ports is no longer than the maximum mileage of ship k (L^k).
4. Route r served by ship k should possess a fuel-port.
$$\sum_{i=1}^{s} f \geq 1 \quad (14)$$

SOLUTION TO THE SHIP ROUTING

In solving this problem, a fixed-radius near neighbor method is adapted. The idea is to find the next port within a predetermined radius (time travel) so that the ships are travelling from a port to another within the reasonable time travel of the case study. Suppose there are k ships available to serve the routes. Each ship serves a route. Each route consists of s ports. There are two fuel ports in each route. The first fuel port in route k is port $\overset{k}{x}$, where x is a quarter of the length of sub-chromosomes, and the second is port $\overset{k}{y}$, where y is three-quarters of the length of sub-chromosomes. The form of each route is shown in Fig. 1.

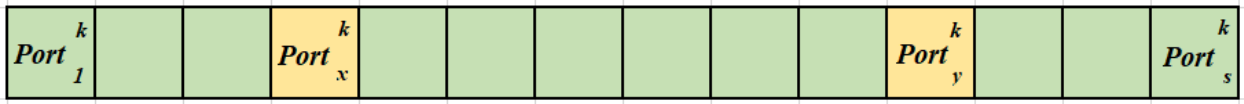

Fig. 1. Form of route *k*

General GA Method

In this problem, chromosomes represent combination of routes. Number of populations in one generation is n chromosomes. The chromosomes are divided into k sub-chromosomes. Each sub-chromosome represents a route and consists of s ports. The ports are represented as genes. There are two fuel ports in each route. The first fuel port in route k is gene $\overset{k}{x}$, where x is a quarter of the length of sub-chromosomes, and the second is gene $\overset{k}{y}$, where y is three-quarters of the length of sub-chromosomes. The form of the chromosome is shown in Fig. 2.

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

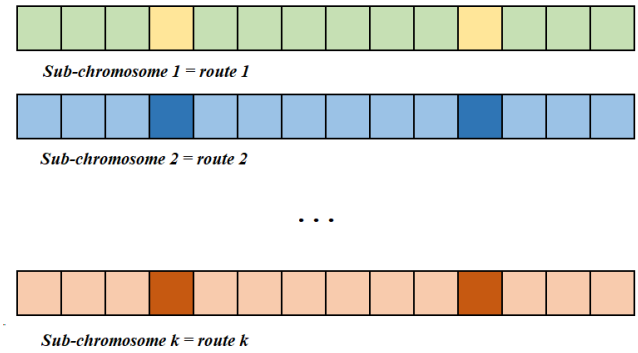

Fig. 2. Form of chromosome

The algorithm used to solve this problem is organized as following stages:

1. Generating *n* chromosome as initial population.
2. Roulette wheel selection.
   In this stage, the generated chromosomes are then selected considering its fitness values. The higher of the fitness value, the higher probability of the chromosome to be selected.
3. Crossover.
   In this stage, certain genes of chromosome are exchanged with the same genes of another chromosome.
4. Mutation.
   In this stage, certain genes of sub-chromosome are exchanged with the same genes of another sub-chromosome in the same chromosome.
5. Breeder GA's.
   This stage will ensure that the best chromosomes are given bigger opportunity to survive and pass to next generation.

Stages 2, 3, 4, and 5 are repeated several times until the predetermined number of iterations is reached.

The GA parameters used and the process of generating initial population, roulette wheel selection, crossover method, mutation and breeder GA's are as follows.

In this project we use the following GA parameters:

- Number of generations: 500
- Crossover rate: 0.25
- Mutation rate: 0.02

The first stage is generating initial population. In this stage, *n* chromosomes are generated. Each chromosome is generated using following steps:

1. If gene *s* is for fuel ports (gene $\frac{k}{x}$or gene $\frac{k}{y}$), then pick up a fuel port randomly.
2. If gene *s* is another port (all genes except gene $\frac{k}{x}$or gene $\frac{k}{y}$), then pick up a port randomly. It doesn't matter whether the port is customer port or fuel port.
3. Calculate time travel and maximum distance constraints using the same formulation that is used in heuristic.
4. If the constraints are satisfied, then generate the port for the next route using steps 1, 2, and 3 until all of the routes are generated

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

5. If not all of the ports are served, do repairing procedures using the same repairing procedures on heuristic method process. If still not satisfied, redo steps 1, 2, 3 and 4.
6. If all of the ports are served, then proceed with the next chromosome.

Roulette Wheel Selection

Roulette wheel selection is applied to select the initial population to become parent chromosomes. The Selection process considers the fitness value of the chromosomes. The process is divided into two main processes:

1. Calculating fitness value
2. Selection process

Process 1: Calculating fitness value

The calculation process is as follows:

1. Calculate the fitness value for each chromosome. Objective function is calculated using (15). Since it is a minimization formulation and the problem is solved with GA, then it has to be changed to a maximization formulation. Another way is we can use the effectiveness of routes (5) to calculate the fitness value. In this project, we choose the formulation of effectiveness of routes instead of convert the minimization formulation.
2. Calculate total fitness of the population.
3. Calculate selection probability (p_n) for each chromosome.
4. Calculate cumulative probability (q_n) for each chromosome.

Process 2: Selection

The selection process is done with the following steps:

1. Generate a random number (r_n) in the range of [0 - 1]. The random numbers is associated with the chromosomes.
2. If $r \leq q_1$, then select the first chromosome, otherwise select the n-th chromosome ($1 \leq n \leq pop\ size$) such that $q_{n-1} < r \leq q_n$.

Crossover

Crossover process exchanges some parts of a chromosome with the same parts of other chromosome. The part to be exchanged are from *gene* $\overset{k}{x}$ to *gene* $\overset{k}{y}$. Crossover process is divided into two main processes. The processes are:

1. Selecting chromosome.
2. Exchanging certain parts of the chromosome.

Process 1: selecting chromosome

Selecting chromosome is done with the following steps:

1. Generate random number (r_n) in the range of [0-1].
2. If r_n is equal or less than crossover rate (p_c), and then select the chromosome.

Process 2: exchanging parts of the chromosome

Exchanging parts of the chromosome is done as follows:

1. Generate a random number (r_k) in the range of [0-1] as much as k. The number is associated with sub-chromosome.
2. If r_k is equal or less than crossover rate (p_c), and then select the sub-chromosome.
3. Exchange genes from a gene after gene $\overset{k}{x}$ to a gene before gene $\overset{k}{y}$of the selected chromosome with the same part of the next selected chromosome. The illustration of the exchanged parts is

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

shown above in Fig. 3. This step is repeated for all selected sub-chromosome in the same chromosome.
4. If the constraint is not satisfied, then redo steps 1, 2, and 3. If still not satisfied, redo steps 1- 3 till *k* times.
5. Steps 1, 2, 3, and 4are repeated for all selected chromosomes.

Mutation

Mutation process exchanges a part of a sub-chromosome with the same part of other sub-chromosome in the same chromosome. Unlike the crossover, mutation deal with the sub-chromosome instead of the chromosome itself. The part to be exchanged are from gene $\frac{k}{x}$ to gene $\frac{k}{s}$. Mutation process is divided into two main processes. The processes are:

1. Selecting sub-chromosome
2. Exchanging certain part of selected sub-chromosome with the same part of other selected sub-chromosome.

Note that these two processes are repeated sequentially for all of the chromosomes.

Process 1: selecting sub-chromosome

Process of selecting sub-chromosome is done as follows:

1. Generate random numbers (r_k) in the range of [0-1]. The generated numbers are associated with sub-chromosomes.
2. If r_k is equal or less than mutation rate (p_c), and then select the sub-chromosome. The selected sub-chromosomes become inputs for process 2.

Process 2: exchanging part of the sub-chromosomes

This process consists of the following steps:

1. Select two sub-chromosome from the result of process 1
2. Exchange genes from gene $\frac{k}{y}$to gene $\frac{k}{s}$ of the selected sub-chromosome with the same genes of the next selected sub-chromosome.
3. Check the constraints. If the constraint is not satisfied, then undo the exchanging.

Breeder GA's

Breeder GA's is a selection process after the genetic operations. This process gives a bigger opportunity to some best chromosomes to survive. The best chromosomes from parent generation replace some randomly chosen chromosomes in offspring generation.

The process of selection is divided into the following steps:

1. Generate random number (r_n) in the range of [0 - 1]. The numbers are associated with chromosomes.
2. If the random number is less than or equal to the breeder rate (p_r), then select the chromosome.
3. Sort parent chromosomes using the fitness value in a descending order.
4. Replace the selected chromosomes with parent chromosome that possess highest fitness value. Note that the parent chromosome can only be selected once. The next parent chromosomes to replace are the second highest, the third highest, and so on.
5. Repeat steps 1, 2, 3 and 4 for all chromosomes.

Hybrid GA

A hybrid GA is proposed to solve the problem. The method is combination of fixed radius near neighbor method and general GA process. First step is generating *n* chromosome as initial population using fixed-radius near neighbor. The generated chromosome has the same form with

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

the chromosome of general GA. The second step is improving the fitness value of the initial solution by involving them in general GA processes, i.e. roulette wheel selection, crossover, mutation, and breeder GA's.

To generate initial population, these six steps are taken:

1. Generating Solution.
   There are three procedures in generating solution. Each procedure is for different aim.
   Procedure 1: aims to generate the first port. The processes are:
   i. Generating random number (r_s) in the range of [1 – p]
   ii. If all port has been served or port r_s is never served, save port r_s as the first port.

   Procedure 2: aims to generate the other ports (except for port $\frac{k}{x}$ andport $\frac{k}{y}$). The processes are:
   i. Generating random number (r_s) in the range of [1 – p]
   ii. If the distance between port r_s and the previous port can be travelled by ship k within 30 hours, and then port r_s as port s. Some experiments are done to define this radius. If it is too long or too short, the route is formed slowly, or not at all.
   Procedure 3: aims to generate ports for fuel ports (port $\frac{k}{x}$ andport $\frac{k}{y}$). The processes are:
   i. Select the nearest port from the previous port.
   ii. Save the port.
2. Calculating time travel and maximum distance constraints.
   Calculating Time Travel:
   Travel time is calculated using (13).
   Calculating Maximum mileage:
   Mileage from the first port (port $\frac{k}{1}$) to the first fuel port (port $\frac{k}{x}$) is calculated using (16).
   $$\sum_{i=1,j=i+1}^{x} d_{ij} + \sum_{i=1,j=i+1}^{x} d_{ji} \leq d_{max}^{k} \quad (16)$$
   Mileage from the first fuel port (port $\frac{k}{x}$) to the second fuel port (port $\frac{k}{y}$) is calculated using (17) and (18).
   $$\sum_{i=x,j=i+1}^{y} d_{ij} \leq d_{max}^{k} \quad (17)$$
   $$\sum_{i=x,j=i+1}^{y} d_{ji} \leq d_{max}^{k} \quad (18)$$
   Mileage from the second fuel port (port $\frac{k}{y}$) to the last port (port $\frac{k}{s}$) is calculated using (19).
   $$\sum_{i=y,j=i+1}^{s} d_{ij} + \sum_{i=y,j=i+1}^{s} d_{ji} \leq d_{max}^{k} \quad (19)$$
3. If the constraints are not satisfied, redo steps 1 and 2.
4. Redo steps 1, 2 and 3 till all of routes are generated.
5. If not all of port is served, do repairing procedures. The repairing procedures are developed to satisfy the constraint of the routes that all of the ports must be served. The processes are:
   a. If all of the ports are served, terminate this process.
   b. If port i of route k is served only once, continue to the next port.
   c. If port i of route k is served more than once, then randomly pick up a port that never served.
   d. Calculate the time and distance constraints. If the time and distance constraints are not satisfied, then undo the changing and pick another port that has not been served yet.
   e. If the time and distance constraints are satisfied or all the ports has been picked, then do steps 1, 2, 3, and 4 for the next ports, until all ports in all routes are checked.
6. If the constraints are still not satisfied, redo 1, 2, 3, 4 and 5.

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

RESULT AND DISCUSSION

All computational experiments are carried out using an Intel(R) Core(TM) i7-10700. The algorithms are implemented in C++ programming language using Microsoft Visual Studio 2022. The algorithms are tested 5 times on each problem. The best-found values over 5 runs are kept and evaluated. This experiment is to evaluate the performance of the algorithms.

General GA

It is evident that the performance of the genetic algorithm (GA) evolves over successive generations as can be seen in Table 1. Initially, in the 1st generation, the algorithm yielded a value of 8,162,132.60. As the algorithm iterated through generations, there was a noticeable improvement in performance, as indicated by a decreasing trend in the values obtained.

Table 1 Fuel Consumption Obtained by General GA Over Generations

Generation	Value Obtained
The best value in the 1st generation	8162132.60
The best value in 100th generation	8122223.22
The best value in 200th generation	7963241.68
The best value in 300th generation	7918784.94
The best value in 400th generation	7914335.24
The best value in 500th generation	8104581.52
The best value over 500 generations (generation 409)	7892659.83

By the 100th generation, the algorithm achieved a value of 8,122,223.22, showcasing a substantial reduction compared to the initial generation. This trend continued, with further improvements observed in subsequent generations. By the 200th generation, the algorithm attained a value of 7,963,241.68, followed by a continued decrease in values in the 300th, 400th, and 500th generations, with respective values of 7,918,784.94, 7,914,335.24, and 8,104,581.52.

However, it's essential to note that the algorithm's performance may not consistently improve with each generation, as demonstrated by the value obtained in the 500th generation being slightly higher than that of the 400th generation. Such fluctuations are common in evolutionary algorithms due to factors such as population diversity and exploration-exploitation trade-offs. The maximum and minimum values obtained by general GA over generations are shown in Fig 3.

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

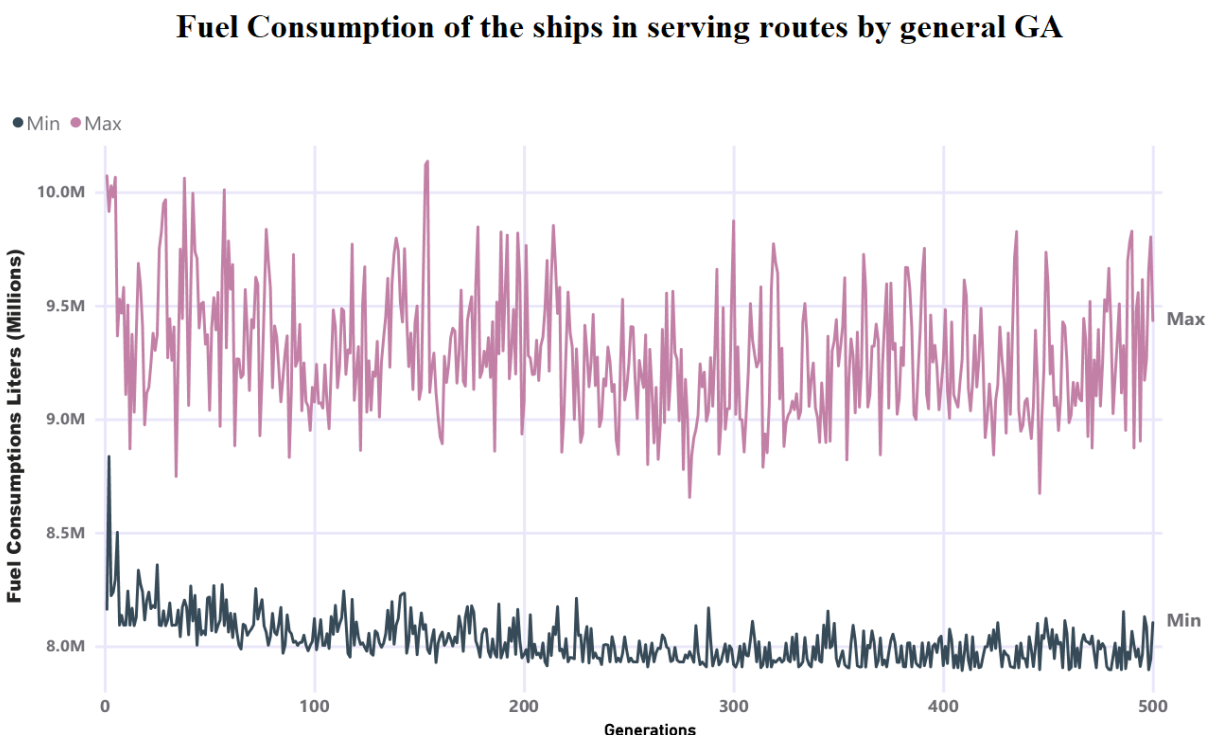

Fig. 3 Fuel consumption values obtained by general GA over generations

Nevertheless, the overall trend indicates that the GA effectively explores the solution space, gradually refining solutions and converging towards optimal or near-optimal solutions. This iterative refinement process is a hallmark of genetic algorithms, where solutions evolve over generations through selection, crossover, and mutation operations.

Furthermore, the best value obtained over 500 generations, recorded in the 409th generation with a value of 7,892,659.83, underscores the algorithm's ability to reach a highly competitive solution within a reasonable computational time. This result demonstrates the effectiveness of the GA in optimizing the given problem, albeit with fluctuations in performance across generations.

Hybrid GA

The performance of the hybrid genetic algorithm (GA) can be elucidated through an analysis of the values obtained across successive generations, as provided in the Table 2. Initially, in the 1st generation, the algorithm yielded a value of 8,455,515.53. From this starting point, the algorithm iterated through generations, demonstrating a notable improvement in performance over time.

Table 1 Fuel Consumption Obtained by Hybrid GA Over Generations

Algorithms	Hybrid GA
The best value in the 1st generation	8455515.53
The best value in 100th generation	7875255.95
The best value in 200th generation	7817752.56
The best value in 300th generation	7741495.97
The best value in 400th generation	7627008.76
The best value in 500th generation	7627008.76
The best value over 500 generations (generation 486)	7563095.12

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

By the 100th generation, the algorithm achieved a substantial reduction in the value obtained, reaching 7,875,255.946. This trend of improvement continued as the algorithm progressed through subsequent generations. By the 200th generation, the value obtained further decreased to 7,817,752.552, followed by continued improvements in the 300th and 400th generations, with respective values of 7,741,495.967 and 7,627,008.763.

Notably, the best value obtained in the 500th generation remained consistent with that of the 400th generation, both recording 7,627,008.763. This observation suggests that the algorithm may have converged to a local optimum or reached a plateau in performance, as evidenced by the stabilization of values across successive generations.

However, it's important to highlight the significant improvement in performance achieved by the hybrid GA compared to its initial generation. The considerable reduction in the value obtained over 500 generations, culminating in a best value of 7,563,095.12 recorded in the 486th generation, underscores the effectiveness of the hybrid GA in optimizing the given problem.

The hybridization of genetic algorithms with other optimization techniques or heuristics likely contributes to the algorithm's ability to explore the solution space more effectively, leading to improved convergence and solution quality. Additionally, the stability observed in the latter generations suggests that the hybrid GA may exhibit robustness in its performance, maintaining competitive solutions even after extensive iterations.

In summary, the performance analysis of the hybrid genetic algorithm reveals its capability to progressively refine solutions and converge towards optimal or near-optimal solutions over successive generations. Despite fluctuations in performance, the hybrid GA demonstrates significant improvements in solution quality, making it a promising approach for optimizing the given problem. The maximum and minimum values obtained by hybrid GA over generations are shown in Fig 4.

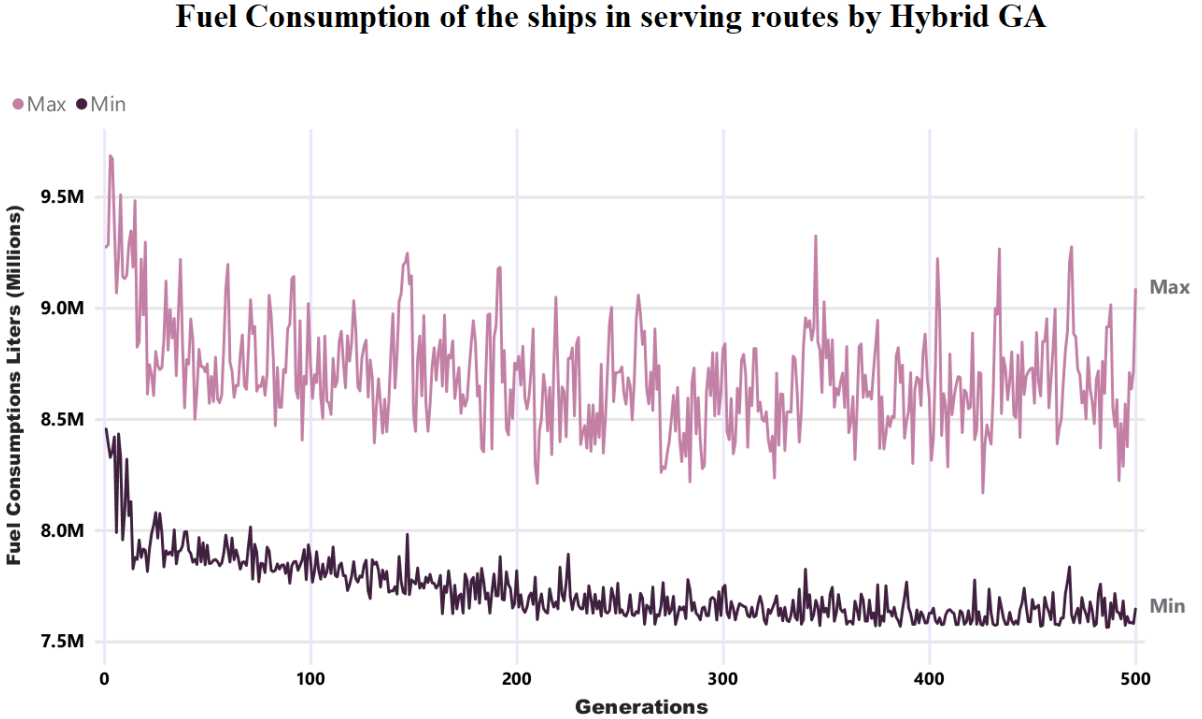

Fig. 4 Fuel consumption values obtained by hybrid GA over generations

DIMENSI, Volume 11 Nomor 3 : 710-724
NOVEMBER 2022
ISSN: 2085-9996
https://www.journal.unrika.ac.id/index.php/jurnaldms

Comparison of general GA and hybrid GA

To compare the performance of the general genetic algorithm (GA) and the hybrid genetic algorithm, we can analyze the values obtained by both algorithms across corresponding generations.

In the case of the general GA, the best value obtained over 500 generations was 7,892,659.83, recorded in the 409th generation. On the other hand, the hybrid GA achieved a best value of 7,563,095.12 in the 486th generation.

Comparing these best values, we observe that the hybrid GA outperformed the general GA in terms of solution quality. The best value obtained by the hybrid GA is lower, indicating a better objective function value, compared to the best value obtained by the general GA.

Additionally, we can analyze the trends in performance across generations for both algorithms. While both algorithms demonstrated improvements in solution quality over time, the hybrid GA exhibited a more consistent and pronounced decrease in values across generations. This suggests that the hybrid GA may converge to better solutions more efficiently compared to the general GA.

Furthermore, it's worth noting that the general GA experienced fluctuations in performance, as evidenced by variations in the best value obtained across successive generations. In contrast, the performance of the hybrid GA appeared to stabilize in the latter generations, indicating a more robust convergence to competitive solutions.

Overall, the comparison highlights the superior performance of the hybrid genetic algorithm over the general genetic algorithm in terms of solution quality and efficiency in converging to optimal or near-optimal solutions for the given problem.

To get a better view of the performance of general GA and hybrid GA, the best values obtained by general GA and hybrid GA over generations are drawn in Fig. 5.

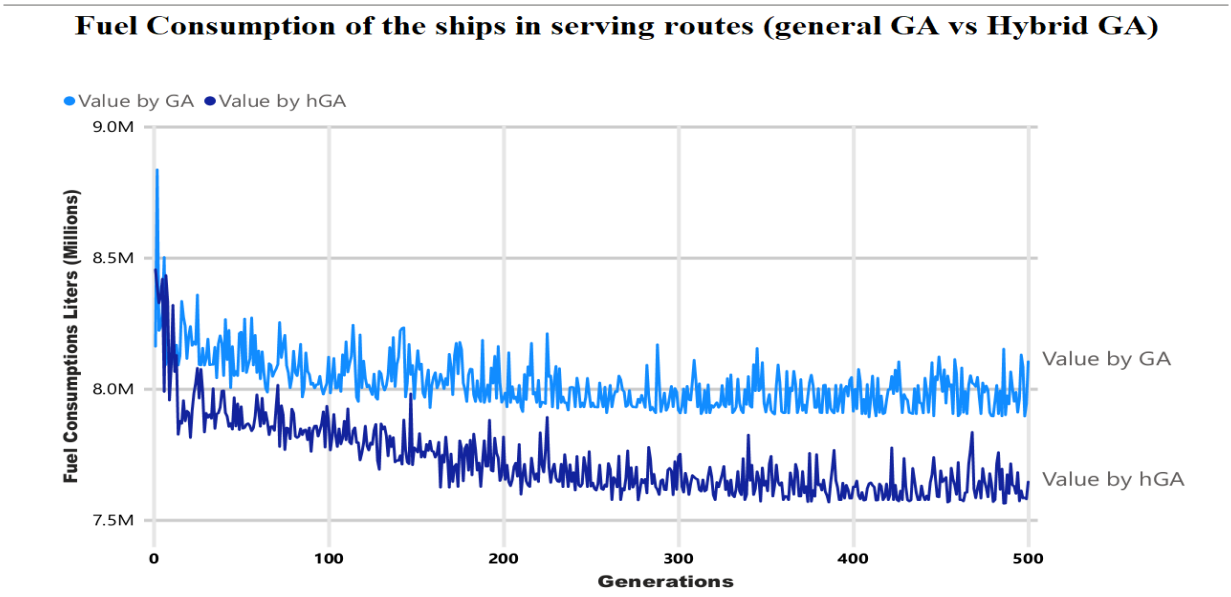

Fig. 5 The best values obtained by general GA vs hybrid GA over generations

CONCLUSION

The findings from the comparison between the general genetic algorithm (GA) and the hybrid genetic algorithm provide valuable insights into their respective performances in optimizing the given problem. Through an analysis of the values obtained across generations, it is evident that the hybrid GA outperforms the general GA in terms of solution quality. The best value obtained by the hybrid GA is significantly lower than that achieved by the general GA, indicating a superior objective function value and, by extension, a better solution.

Furthermore, the trends observed in the performance of both algorithms shed light on their convergence behaviors. While both algorithms demonstrated improvements in solution quality over successive generations, the hybrid GA exhibited a more consistent and pronounced decrease in values, indicating a more efficient convergence to competitive solutions. In contrast, the general GA experienced fluctuations in performance, suggesting less stability in its convergence process.

Overall, these findings underscore the effectiveness of hybridizing genetic algorithms with other optimization techniques or heuristics in enhancing solution quality and convergence efficiency. The superior performance of the hybrid GA highlights its potential as a robust and efficient optimization approach for the given problem.

In conclusion, the results of this study provide compelling evidence supporting the adoption of hybrid genetic algorithms for optimizing the problem at hand. Future research endeavors could focus on further refining the hybrid GA and exploring its applicability to other optimization problems, ultimately contributing to advancements in the field of optimization and algorithm design.

REFERENCES

Agany Manyiel, J. M., Kwang Hooi, Y., & Zakaria, M. N. B. (2021). Multi-Population Genetic Algorithm for Rich Vehicle Routing Problems. *Proceedings-International Conference on Computer and Information Sciences: Sustaining Tomorrow with Digital Innovation, ICCOINS 2021*, 213–219.

Bahalke, U., Hamta, N., Shojaeifard, A. R., Alimoradi, M., & Rabiee, S. (2022). A new heuristic algorithm for multi vehicle routing problem with and/or-Type precedence constraints and hard time windows. *Operational Research in Engineering Sciences: Theory and Applications*, *5*(2), 28–60.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, *99*, 300–313.

Keçeci, B., Altparmak, F., & Kara, I. (2021). A mathematical formulation and heuristic approach for the heterogeneous fixed fleet vehicle routing problem with simultaneous pickup and delivery. *Journal of Industrial and Management Optimization*, *17*(3), 1069–1100.

Kumar, S. N., & Panneerselvam, R. (2012). *A survey on the vehicle routing problem and its variants*.

Li, D., Cao, Q., Zuo, M., & Xu, F. (2020). Optimization of green fresh food logistics with heterogeneous fleet vehicle route problem by improved genetic algorithm. *Sustainability*, *12*(5), 1946.

Masmoudi, M. A., Hosny, M., Demir, E., & Cheikhrouhou, N. (2018). A study on the heterogeneous fleet of alternative fuel vehicles: Reducing CO2 emissions by means of biodiesel fuel. *Transportation Research Part D: Transport and Environment*, *63*, 137–155.

Miao, L., Ruan, Q., Woghiren, K., & Ruo, Q. (2012). A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. *RAIRO-Operations Research*, *46*(1), 63–82.

Panessai, I. Y., Baba, M. S., & Iksan, N. (2019). Solving Rich Vehicle Routing Problem Using Three Steps Heuristic. *International Journal of Artificial Intelligence*, *1*(1), 1–19. https://doi.org/10.36079/lamintang.ijai-0101.9

Perboli, G., Tadei, R., & Vigo, D. (2011). The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, *45*(3), 364–380.

Pertambangan Minyak Nasional. (2008). *Panduan Pelayanan BBM Bunker*. Direktorat Pemasaran dan Niaga.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*(8), 2403–2435.

Shi, Y., Lv, L., Hu, F., & Han, Q. (2020). A heuristic solution method for multi-depot vehicle routing-based waste collection problems. *Applied Sciences*, *10*(7), 2403.

Tao, Y., Lin, C., Wei, L., & others. (2022). Metaheuristics for a Large-Scale Vehicle Routing Problem of Same-Day Delivery in E-Commerce Logistics System. *Journal of Advanced Transportation*, *2022*.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, *60*(3), 611–624.

Yldrm, U. M., & Çatay, B. (2015). An ant colony-based matheuristic approach for solving a class of vehicle routing problems. *Computational Logistics: 6th International Conference, ICCL 2015, Delft, The Netherlands, September 23-25, 2015, Proceedings 6*, 105–119.

Yuan, X., Zhu, J., Li, Y., Huang, H., & Wu, M. (2021). An enhanced genetic algorithm for unmanned aerial vehicle logistics scheduling. *IET Communications*, *15*(10), 1402–1411.

Zhou, Z., Ha, M., Hu, H., & Ma, H. (2021). Half open multi-depot heterogeneous vehicle routing problem for hazardous materials transportation. *Sustainability*, *13*(3), 1262.