LogForum

> Scientific Journal of Logistics <

2023, 19 (4), 577-593

http://doi.org/10.17270/J.LOG.2023.904

http://www.logforum.net p-ISSN 1895-2038 e-ISSN 1734-459X

ORIGINAL PAPER

USING A GENETIC ALGORITHM TO SOLVE A VEHICLE ROUTING PROBLEM INVOLVING SIMULTANEOUS DELIVERIES AND PICKUPS WITH SPLIT LOADS AND TIME WINDOWS (A CASE STUDY FOR A SHIPPING COMPANY)

Shea Amanda Ferdianti, I Gede Agus Widyadana

Department of Industrial Engineering, Faculty of Industrial Technology Petra Christian University, East Java, **Indonesia**

ABSTRACT. Background: This research addresses a Vehicle Routing Problem with Simultaneous Delivery and Pickup, Split Loads, and Time Windows (VRPSDPSLTW). In this research, the VRPSDPSLTW problem is adapted for Company X, a shipping company based in Surabaya. The main goal is to enhance the optimal utilization of vessel capacity in the field of shipping transportation and logistics. Little previous research has been done on VRPSDPSLTW at a shipping company.

Methods: The optimization approach employed was the Genetic Algorithm (GA), which serves as a metaheuristic to effectively optimize vessel capacity utilization. The algorithm uses One Point Crossover and Swap Mutation operators and analyzes various mutation parameters to determine the best configuration. The GA was coded in R, and experiments were conducted to obtain the best parameter for the GA.

Results: The research yielded several outcomes, including route plans, loaded and unloaded Twenty-Foot Equivalent Units (TEUs), travel times, and trip utility from the point of loading (POL) to the point of delivery (POD). In total, there were 85 port visits, surpassing the initial count of 35 ports. Some ports were visited multiple times, with the exception of Surabaya, which served as the home base for a fleet of 15 vessels. The average trip duration was approximately 35 days. Through experimentation, it was determined that employing 1,000 generations along with a mutation probability of 0.2 produces improved solutions. The Genetic Algorithm solution enhanced the average vessel capacity utilization, increasing it to 80.93%. This represents a significant 21.23% increase compared to the global average of 59.7% observed for similar vessel usage scenarios.

Conclusions: Furthermore, through the introduction of novel route opportunities, the contributions of each vessel were effectively enhanced. This achievement resulted in an optimal average vessel capacity utilization that met the demand. The findings strongly advocate for the employment of the Genetic Algorithm, highlighting its potential to substantially improve vessel capacity utilization. Consequently, this approach has played a pivotal role in elevating the efficiency of transportation and logistics operations for Company X.

Keywords: vehicle routing problem, simultaneous deliveries and pickups, split loads, time windows, optimization, genetic algorithm

INTRODUCTION

The Vehicle Routing Problem (VRP) was first analysed by Dantzig and Ramser [1959] to find a solution for fuel delivery. It is now crucial for efficient cargo and travel services, covering various vehicles and customer demands, including sea and air transport. Optimizing delivery routes minimizes costs and travel time and maximizes vessel capacity use in shipping. This involves sequencing visits using multiple vehicles from a central depot [D.M. Utama et al. 2020]. Depot and vehicle capacities, along with customer requests, influence route design [F. Arnold and K. Sörensen 2019]. VRP has evolved into variations like VRPPDTW, which has been explored by researchers like Sitek et al. [2021] and Dewi and Utama [2021].

This research examines Shipping Company X, which has operated in Surabaya (Indonesia)

Copyright: Wyższa Szkoła Logistyki, Poznań, Polska

BY-NC

Citation: *Ferdianti S. A., Widyadana I. G. A., 2023. Vehicle routing problem simultaneous deliveries and pickups with split loads and time windows with genetic algorithm (case study in shipping company). LogForum 19 (4), 577-593*, http://doi.org/10.17270/J.LOG.2023.904

Received: 01.09.2023, Accepted: 02.10.2023, on-line: 10.12.2023

since 1984, aiming to enhance its shipping services and economic impact. While offering various services, including port-to-port and international shipping, the company seeks to optimize cargo routes for greater vessel capacity utilization. Current manual processes hinder route efficiency, as evidenced by some vessels operating below 60% capacity. With a fleet of over 40 vessels of varying capacities, simultaneous pickup and delivery, travel time considerations, and load distribution complexities are all significant challenges in route optimization for Company X.

The Genetic Algorithm (GA), by John Holland [1975], optimizes via genetic selection and natural processes, using crossover and mutation. GAs streamline function modeling, reduce errors, and are effective in engineering [J. Protopopova and S. Kulik 2020]. Applied to routing problems [S. Karakatič 2021], GAs address VRP variants, as can be seen in enhanced solutions by M. A. Mohammed et al. [2017], W. Ho et al. [2008], and P. R. de Oliveira da Costa [2018], for delivery efficiency and cost reduction. GA handles capacitated vehicle routing [H. Nazif and L. S. Lee 2012, R. Saxena et al. 2020] and fleet size [S. Liu et al. 2009], and improves VRPTW solutions with decomposition [C.-B. Cheng and K.-P. Wang 2009], which is relevant to VRPSDPSLTW.

This research introduces a customized Genetic Algorithm (GA) to address VRPSDPSLTW challenges for Shipping Company X. It aims to optimize vessel capacity utilization while adhering to split-load, vessel capacity, and travel time constraints. Utilizing historical data (January 2018 to February 2023), with anonymized port and vessel names for confidentiality purposes, it focuses on classifying VRP, particularly VRPSDPSLTW. It incorporates route-specific details such as demand, distances, travel times, and vessel attributes, primarily analyzing Surabaya-based vessels of Company X. The study assumes uniform speed, exclusive vessel use, and a one-week testing period. R is employed for GA, Power BI for visualization, and Minitab for statistical analysis. Notably, the research assumes equal port accessibility and excludes size-based limitations at specific ports.

MATERIALS AND METHODS

The NP-Hard Vehicle Routing Problem (VRP) has drawn significant research interest from researchers seeking to improve efficiency. Traditionally, it assumes a single depot, single-visit customers, and capacity limits. However, real-world scenarios require adjustments. Dror and Trudeau [1989] and Dror et al. [1994] introduced the Split Delivery VRP (SDVRP), dividing customer demands among vehicles to reduce distance and vehicle count.

Overcoming traditional VRP constraints, the Vehicle Routing Problem with Simultaneous Delivery and Pickup, Split Loads, and Time Windows (VRPSDPSLTW) model emerges. It involves vehicles from a depot serving customers while considering simultaneous delivery and pickup within time windows. This applies even when demand surpasses vehicle capacity, enabling multiple visits or multiple-vehicle service. In research by Wang et al. [2013], the VRPSDPLTW method was utilized, with ordered elements denoted as $J = 1,2, 3...$, n. $N0$ (where $N0$ includes the depot marked as 0 and 1,2, 3..., n represent customers). Routes involved sequential visits by individual vehicles, connecting successive customers. All routes shared the same depot for departure and return, leading to consistent origins and destinations.

Notations:

Q	capacity of each vehicle
V	set of vehicles, where $k \in V$
V_k	1 if vehicle k is selected to serve a customer, 0 otherwise
J	set of customers $\{1, 2, 3, \dots, n\}$ $\forall i, j \in J$, and $i \neq j$
d_{ij}	travel cost (travel distance) between customer i and customer j
t_{ij}	travel time between customer i and customer j
v_{ij}	travel speed between customer k and customer j
D_j	delivery demand at customer j

R_j	pickup demand at customer j
x_{ijk}	1 if vehicle k travels directly from customer i to customer j, 0 otherwise
y_{ijk}	amount of goods delivered by vehicle k using the route from customer i to customer j
z_{ijk}	amount of goods taken from customer j by vehicle k using the route from customer i to customer j
y'_{ijk}	remaining amount of goods to be delivered form customer i to customer j for vehicle k
$[ai, bi]$	time window for each customer $\forall i \in N0$
r_{ik}	time when vehicle k starts serving customer i.

Objective functions:

$$\min F_1(x) = \sum_{k \in V} |V_k| \sum_{j \in N_0} x_{0_{jk}} \tag{1}$$

$$\min F_2(x) = \sum_{k \in V} \sum_{i \in N_0} \sum_{j \in N_0} d_{ij}\, x_{ijk} \tag{2}$$

Constraints:

$$\sum_{j \in N_0} x_{0_{jk}} = 1 \quad \forall k \in V \tag{3}$$

$$\sum_{i \in N_0} x_{iuk} - \sum_{j \in N_0} x_{ujk} = 0 \quad \forall u \in J, \forall k \in V \tag{4}$$

$$\sum_{i \in N_0} x_{ijk} \geq 1 \quad \forall j \in J, \forall k \in V \tag{5}$$

$$\sum_{k \in V} \sum_{i=0, i \neq j}^{n} y_{ij} = D_j \qquad \forall j \in J \tag{6}$$

$$\sum_{k \in V} \sum_{i=0, i \neq j}^{n} Z_{ij} = R_j \qquad \forall j \in J \tag{7}$$

$$\sum_{k \in V} \sum_{i \in N} x_{ijk} \geq 1 \quad \forall j \in J, \forall k \in V \tag{8}$$

$$y'_{ijk} + \sum_{i=0, i \neq j}^{n} Z_{ijk} \leq Q \qquad \forall j \in J, \forall k \in V \tag{9}$$

$$t_{ij} = \frac{d_{ij}}{v_{ij}} \quad \forall i, j \in J \tag{10}$$

$$r_{ik} + t_{ij} - (b_i + t_{ij} - a_j)(1 - x_{ijk}) \leq r_{jk} \qquad \forall j \in J, \forall i \in N_{0,} \ \forall k \in V \tag{11}$$

$$a_i \leq r_{ik} \leq b_i \qquad \forall i \in N_{0,} \ \forall k \in V \tag{12}$$

$$r_{ik} + t_{i0} - (b_i + t_{i0} - a_0)(1 - x_{i0k}) \leq b_0 \qquad \forall i \in J, \forall k \in V \tag{13}$$

$$r_{0k} = a_0 \quad \forall k \in V \tag{14}$$

$$x_{iik} = 0 \qquad \forall i \in N_{0,} \ \forall k \in V \tag{15}$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in N_{0,} \ \forall k \in V \tag{16}$$

Objective (1) is intended to minimize vehicles on delivery routes, and objective (2) is intended to minimize travel costs. Constraint (3) ensures each of the k vehicles goes from the depot to customer j in set J. Constraint (4) mandates vehicles to serve customers in sequence before returning to the depot. Constraint (5) permits a single vehicle to serve a customer multiple times. Constraints (6) and (7) define total delivery and pickup demands at customer j. Constraint (8) allows multiple vehicles to serve a customer in varying quantities. Constraint (9) ensures vehicle k's remaining delivery and pickup fit its capacity. Constraint (10) calculates travel time using distance and speed. Constraint (11) regulates the arrival time of vehicle k at customer j after the time $rik + tij$ if vehicle k chooses the route from customer i to customer j or earlier than the difference $aj - bi$ if vehicle k does not choose that route. Constraint (12) enforces service within customer-specified time frames. Constraint (13) requires vehicles to reach the depot before closing time. Constraint (14) sets vehicle departure after the depot opening time a_0.

This research set out to use Genetic Algorithms to effectively optimize simultaneous pickup and delivery in the Vehicle Routing Problem (VRP) and to improve vessel capacity utilization compared to traditional routing methods. The study was intended to develop solutions that consider split-loads, vessel capacity constraints, and travel time constraints, as outlined in the Vehicle Routing Problem Simultaneous Deliveries and Pickups with Split Loads and Time Windows (VRPSDPSLTW). Therefore, the primary objectives of this research were to enhance vessel capacity utilization efficiency within the VRP while maintaining compliance with the constraints specified by VRPSDPSLTW.

The outlined scheme (Fig 1) offers a structured approach to address Company X's issues. It commences with a thorough issue analysis, followed by the establishment of research goals and constraints. A literature review was carried out to gather valuable insights, and initial data was collected. The scheme's core is the development of a Genetic Algorithm-based model for VRPSDPSLTW. Verification and validation tests include a feedback loop for model refinement. The Genetic Algorithm solution was then compared with current conditions, leading to a comprehensive analysis. The scheme culminates in conclusions and recommendations, providing a structured path for addressing Company X's challenges.

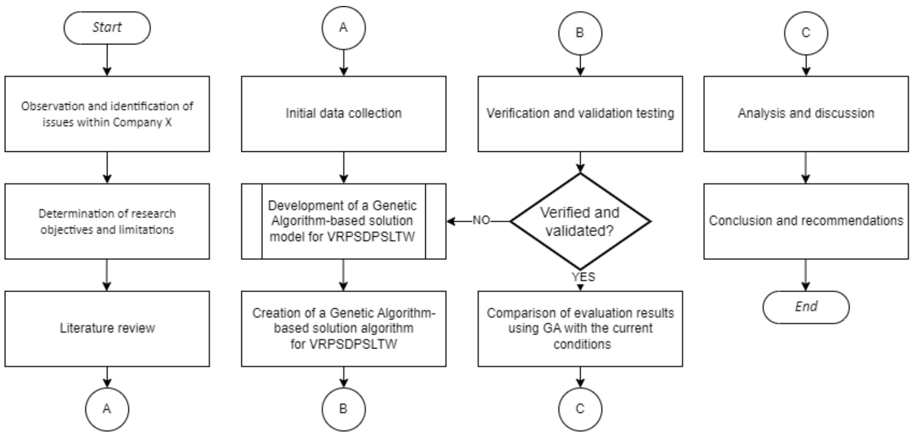

Fig. 1. Flowchart steps of conducted research
Source: Own work

This research employed a variety of data types to implement the Genetic Algorithm effectively using the process shown in the flowchart in Fig. 2. These encompass customer booking details (origin, destination, vessel information, container quantities), port information categorized by region (WEST/EAST), standardized port names, Company X's vessels grouped by homebase, essential vessel specifications (ID, name, dimensions, capacity), detailed cargo delivery records, accurate port-to-port distances and coordinates, and travel durations from port of loading (POL) to port of discharge (POD).

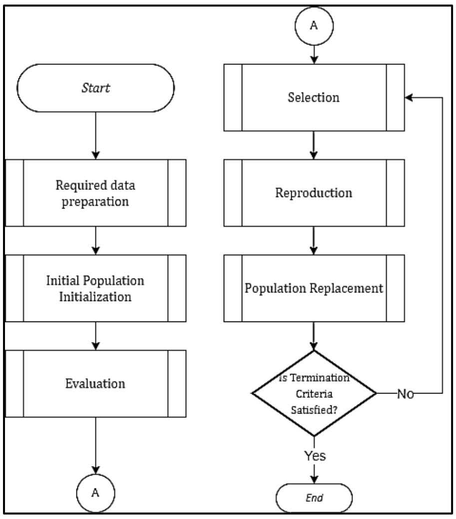

Fig. 2. Flowchart model for solving VRPSDPSLTW based on genetic algorithm
Source: Own work.

The Genetic Algorithm (GA) consists of the following stages: data preparation, Initial Population Initialization, Evaluation, Selection, Reproduction (Crossover and Mutation), and Population Replacement. Iterations continue until the target generation count is reached. Evaluation is crucial, assessing individual quality and performance. It picks individuals with potential traits for better solutions in the next generation. The research employs a fitness value that combines %utility, penalties for demand failure, and overtime. Evaluation follows defined stages and formulas.

Shipping Load (TEUs) = Previous Remaining Journey (TEUs) + Loading Quantity (TEUs) (20)

Remaining (TEUs) = Shipping Load (TEUs) – Unloading Quantity (TEUs) (20)

%Utility = Shipping Load (TEUs) / Payload (TEUs) * 100 (20)

Travel Time (Days) = Shipping Time (Days) + Port Time (Days) (21)

Total Failed Demand (TEUs) = Total Demand (TEUs) – Total Demand Fulfilled (TEUs) (22)

$$\textbf{Average \%Utility per chromosome} = \frac{\Sigma\ (\text{\%Utility from POL to POD in 1 chromosome})}{\text{Number of POL–POD journeys in 1 chromosome}} \quad (23)$$

Overtime (Days) = (Travel Time of trip 1 (Days) – 31) + (Travel Time of trip 2 (Days) – 31) + ⋯ + (Travel Time of trip n (Days) – 31) (24)

Penalty Time = {0 if Overtime = 0 day (25)
100, if Overtime < 5 days
1000, if Overtime ≥ 5 days}

Penalty Demand = Total Failed Demand (26)

Fitness Value = %Utility – Penalty Demand – Penalty Time (27)

Company X does not enforce a specific time limit for completing a trip, but it does provide standard estimates for sailing hours and port time at each port. For this research, an ideal total travel time per trip was defined as not exceeding 31 days. Any days that surpassed this limit were deemed as overtime, resulting in a penalty. The implementation of penalty time through a Genetic Algorithm (GA) is intended to attain the shortest travel time per trip (penalties are applied to steer the solution towards the 31-day mark), thereby optimizing utility while minimizing unfulfilled demand.

Integrating these elements into the fitness value enables GA to assess route performance. The selection favors higher %Utility and lower Penalty Demand and Time, progressively refining solutions for optimal routes and enhanced vessel capacity utilization. The generated solutions were verified, validated, and adjusted for Company X's needs. Result analysis is intended to produce high-performance solutions aligned with goals. Comparing historical data and constraints is a way of evaluating the effectiveness of improving cargo delivery vessel capacity use.

RESULTS

Data Collection and Analysis

Scenario testing utilized Company X's demand data over a one-week period, which was used as an input for the GA. Table 1 shows the first 5 rows of the booking data that were used as forecast demand and further processed. One TEU is equal to one 20-foot container. In total, 1518 rows of data on forecast demand were collected.

Table 1. Forecast demand for scenario testing

Row	POL	POD	TOTALTEUS
1	IDSUB	IDSSS	57
2	IDGGG	IDSUB	9
3	IDSSS	IDSUB	5
4	IDSSS	IDSUB	1
5	IDSSS	IDSUB	10

Source: Own work.

The TOTAL TEU values for matching POL and POD pairs were subsequently added up. This summed the TEUs for each corresponding POL-POD combination, leading to a data transformation as depicted in Table 2.

Table 2. Aggregated forecast demand based on POL and POD pairs

POL	POD	TEUs	POL	POD	TEUs	POL	POD	TEUs	POL	POD	TEUs
IDZZZ	IDSUB	7	IDSUB	IDFFF	199	IDNNN	IDSUB	328	IDFFF	IDSUB	352
IDYYY	IDSUB	323	IDSUB	IDGGG	367	IDNNN	IDXXX	1	IDEEE	IDSUB	268
IDXXX	IDSUB	702	IDSUB	IDHHH	70	IDMMM	IDSUB	117	IDDDD	IDSUB	156
IDVVV	IDABC	201	IDSUB	IDJJJ	40	IDLLL	IDSUB	383	IDCCC	IDSUB	33
IDVVV	IDALM	161	IDSUB	IDKKK	375	IDKKK	IDABC	7	IDBBB	IDSUB	892
IDVVV	IDSUB	116	IDSUB	IDLLL	372	IDKKK	IDAEF	18	IDAVW	IDSUB	22
IDVVV	IDXXX	10	IDSUB	IDMMM	207	IDKKK	IDALM	43	IDANO	IDSUB	5
IDSUB	IDAAA	89	IDSUB	IDNNN	123	IDKKK	IDCCC	8	IDAMN	IDSUB	63
IDSUB	IDABC	86	IDSUB	IDPPP	26	IDKKK	IDDDD	53	IDAKL	IDSUB	595
IDSUB	IDAEF	217	IDSUB	IDQQQ	193	IDKKK	IDGGG	35	IDAJK	IDQQQ	3
IDSUB	IDAEG	14	IDSUB	IDSSS	693	IDKKK	IDMMM	48	IDAJK	IDSUB	545
IDSUB	IDAGH	236	IDSUB	IDUUU	54	IDKKK	IDNNN	21	IDAHI	IDSUB	236
IDSUB	IDAHI	238	IDSUB	IDVVV	217	IDKKK	IDQQQ	30	IDAGH	IDSUB	2
IDSUB	IDAJK	117	IDSUB	IDWWW	2	IDKKK	IDSSS	63	IDAEF	IDSUB	125
IDSUB	IDALM	90	IDSUB	IDXXX	158	IDKKK	IDSUB	140	IDAEF	IDYYY	261
IDSUB	IDAMN	94	IDSUB	IDYYY	302	IDKKK	IDUUU	6	IDAEF	IDZZZ	40
IDSUB	IDAUV	9	IDSUB	IDZZZ	35	IDKKK	IDVVV	42	IDABC	IDSUB	18
IDSUB	IDBBB	639	IDSSS	IDSUB	774	IDKKK	IDYYY	26	IDABC	IDXXX	9
IDSUB	IDCCC	203	IDQQQ	IDSUB	217	IDJJJ	IDSUB	94	IDAAA	IDBBB	1
IDSUB	IDDDD	150	IDPPP	IDSUB	36	IDGGG	IDKKK	1	IDAAA	IDSUB	562
IDSUB	IDEEE	114	IDOOO	IDSUB	90	IDGGG	IDSUB	336			

Source: Own work.

To determine port visit counts, the maximum TEUs achieved from past booking and vessel journey data (2018–2022) were identified. The highlighted maximum TEUs (Table 2) were calculated by finding the most transported TEUs from POL to POD in a single vessel journey. Next, unique identifiers were assigned to ports and their visit counts (VISIT) were determined. VISIT represents required visits per port. VISITNR was calculated by dividing TOTALTEUS (demand) by MAXTEUSPERVISIT (highlighted), giving estimated visits. The VISIT column in Table 3 contains rounded VISITNR values. For example, the highest VISIT of 3 for port IDYYY (POD) means 3 visits. Table 4 shows resulting visits for forecast ports. This led to 85 port visits, exceeding the initial 35, as some ports were visited multiple times (except Surabaya, dividing or serving as home base).

Table 3. Example of calculating the number of visits for IDYYY

POL	POD	TOTALTEUS	MAXTEUSPERVISIT	VISITNR	VISIT
IDYYY	IDSUB	323	668	0.484	1
IDAEF	IDYYY	261	523	0.499	1
IDKKK	IDYYY	26	47	0.553	1
IDSUB	IDYYY	302	111	2.721	3

Source: Own work

Table 4. The required number of visits (count) to each port (35 ports in demand data, except for idsub as the home base) for each chromosome

Port	**Count**	**Port**	**Count**	**Port**	**Count**	**Port**	**Count**	**Port**	**Count**	**Port**	**Count**
IDAAA	2	IDXXX	2	IDFFF	3	IDKKK	5	IDPPP	1	IDAJK	2
IDBBB	7	IDYYY	3	IDGGG	4	IDLLL	4	IDAEG	1	IDAKL	3
IDCCC	2	IDAUV	1	IDAVW	1	IDMMM	3	IDAGH	3	IDALM	3
IDDDD	3	IDZZZ	1	IDHHH	2	IDNNN	2	IDQQQ	2	IDAMN	2
IDVVV	3	IDANO	1	IDABC	2	IDOOO	1	IDAHI	2	IDUUU	2
IDWWW	1	IDEEE	1	IDJJJ	2	IDAEF	3	IDSSS	5		
Total Count = 85											

Source : Own work

Genetic Algorithm

Here, the decision was taken to establish an initial population with 16 chromosomes, each representing port collections. Genes in a chromosome followed the visit counts in Table 4. Notably, IDSUB is different; it acts as a both start and an end point, dividing chromosomes into trips. The initial Population generated 16 random gene combinations. This ensured diverse starting points. Each chromosome included 35 ports, totaling 85 visits. The initial count of genes in a chromosome was 87 (85 visits + 2 IDSUBs at the beginning and end of the chromosome). However, gene count grows with each generation.

Chromosomes were split into trips via trimming, randomly picking 3 to 7 genes (excluding IDSUB) per trip. The "Subtle" column labels trips, with IDSUBs marking partitions. This breaks 1 chromosome into 15 trips (115 genes), adding 28 new IDSUB genes. Data were reshaped for journey analysis, streamlining the structure for fitness calculation from POL to POD. Table 6 demonstrates this transformation, simplifying the original format.

Table 5. Original data set format

Port	Chromosome	Subgroup
IDSUB	1	A
IDAGH	1	A
IDAUV	1	A
IDBBB	1	A
IDSUB	1	A

Source: Own work

Table 6. Data set format after transformation

POL	POD	Chromosome	Subgroup
IDSUB	IDAGH	1	A
IDAGH	IDAUV	1	A
IDAUV	IDBBB	1	A
IDBBB	IDSUB	1	A

Source: Own work

Table 7. Demand from IDSUB to IDAEF and number of visits to IDAEF

POL	POD	TOTALTEUS	POD VISIT
IDSUB	IDAEF	217	3

Source: Own work

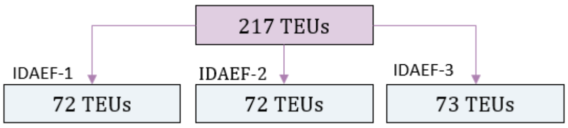

Fig. 3. Concept of split loads for demand from IDSUB to IDAED
Source: Own work

Load-unload quantities were determined based on total TEUs in demand data and visits from relevant ports. For instance, if there was a demand of 217 TEUs from POL IDSUB to POD IDAEF in a given week, with IDAEF visited three times within a chromosome, successful transport required IDSUB (POL) to precede IDAEF (POD). Proportional cargo allocation was crucial, and was achieved by dividing total TEUs by required visits. During three visits from IDSUB to IDAEF, a total of 217 TEUs were transported (Table 7), with allocations of 72 TEUs, 72 TEUs, and 73 TEUs on each visit (Fig. 3). Unsuccessful visits could occur when the POL IDSUB was not visited before the POD IDAEF for all three visits, resulting in untransported demand (145 TEUs if the 2nd and 3rd visits fail, or 73 TEUs if only the 3rd visit fails).

Loading involved direct loading of subsequent ports' demands at each POL. At POD, a specific portion was unloaded. This continued until the trip cargo was 0. The shipping load calculation considered TEUs from POL to POD.

Next, vessel placement maximized each trip's cargo. Each chromosome was assigned a vessel sequentially, matching the cargo to the closest vessel capacity. If this was between ideal and historical capacity, the utility calculations used the maximum cargo. Vessel assignment proceeded until each trip had a vessel. Table 8 demonstrates this process for a trip in one chromosome, outlining load, unload, shipping load, and vessel placement.

Table 8. Example calculation results: load, unload, shipping cargo, and vessel placement in a trip for a single chromosome

POL	POD	Chromosome	Subgroup	LOAD	UNLOAD	Shipping Load	Max Shipping Load	VESSEL ID	VESSEL NAME	PAY LOAD
IDSUB	IDAGH	1	A	188	80	188	188	BBB	BBBBBB	208
IDAGH	IDAUV	1	A	2	9	110	188	BBB	BBBBBB	208
IDAUV	IDBBB	1	A	0	99	101	188	BBB	BBBBBB	208
IDBBB	IDSUB	1	A	136	138	138	188	BBB	BBBBBB	208

Source: Own work

In Table 9, the evaluation results of the fitness values for all chromosomes in the initial population are presented, along with a detailed breakdown of the components contributing to the fitness value.

Table 9. Fitness value calculation results for all chromosomes in the initial population

Chromosome	%Utility	FAILED	PENALTYDEMAND	OVERTIME	PENALTYTIME	fitness value
1	80.23	880	880	166	1000	-1799.77
2	77.2	943	943	130	1000	-1865.8
3	77.22	589	589	147	1000	-1511.78
4	73.91	960	960	145	1000	-1886.09
5	76.67	862	862	131	1000	-1785.33
6	75.76	1024	1024	127	1000	-1948.24
7	75.64	811	811	116	1000	-1735.36
8	79.88	909	909	90	1000	-1829.12
9	78.6	927	927	108	1000	-1848.4
10	76.07	928	928	143	1000	-1851.93
11	80.01	867	867	141	1000	-1786.99
12	76.52	799	799	105	1000	-1722.48
13	73.09	759	759	115	1000	-1685.91
14	77.09	898	898	129	1000	-1820.91
15	75.74	729	729	111	1000	-1653.26
16	74.62	777	777	109	1000	-1702.38

Source : Own work

Subsequently, selection probability was calculated by dividing each individual's fitness value by the total fitness value of the population, establishing probabilities for the mating pool. Employing the roulette wheel method, individuals were chosen for the pool using a random number in the "rand" column. The first chromosome with a cumulative probability exceeding the random number entered the pool. After pool selection, individuals were paired as parent pairs for crossover or genetic recombination if criteria were fulfilled, as shown in Table 10.

Table 10. Roulette wheel results and parent pairs for crossover

rand	Chromosome	Cumulative probability	Pair	rand	Chromosome	Cumulative probability	Pair
0.722	12	0.738	1	0.966	16	1	5
0.08	2	0.1277	1	0.378	7	0.4059	5
0.041	1	0.0657	2	0.316	6	0.3477	6
0.814	14	0.8675	2	0.489	9	0.5381	6
0.597	10	0.6041	3	0.468	8	0.4716	7
0.497	9	0.5381	3	0.239	4	0.2439	7
0.663	11	0.6693	4	0.969	16	1	8
0.131	3	0.1915	4	0.408	8	0.4716	8

Source : Own work

After obtaining chromosome pairs, the subsequent step involved selecting pairs for crossover using a crossover probability (Pc) set at 0.9. Crossover occurs if the randomly generated number for each pair is below 0.9. Employing one-point crossover, a random cutting point is assigned to each pair. Chromosomes serve as "parents," yielding two offspring or "children" with equal chromosomes. Initial mutation probability (pm) ranges from 0.01 to 0.2 for real-world optimization. Following crossover, each chromosome's randomly generated number between 0 and 1 determines mutation. If the number is below pm, the chromosome mutates, exchanging two random gene points (swapping mutation) and forming a new offspring chromosome.

New offspring resulting from crossover and mutation are merged with the initial population, then evaluated for fitness. Retaining the two best chromosomes, merged chromosome averages are calculated. If the target generation is not reached, chromosomes ranked from 3rd to 100th become the next generation's initial population, ensuring genetic diversity and optimal outcomes.

Optimization Results of Scenario Testing

In this research, the optimization of GA solutions for the VRPSDPSLTW problem consisted of three key variables: population size, mutation probability, and crossover probability. The crossover probability was set at 0.9, and the initial population size was 16 individuals. The mutation probability was tested at 0.01 and 0.2, each repeated five times with the same initial population size and 500 generations per trial. Analysis of the best fitness value graphs in Fig. 4 and Fig. 5 reveals convergence at the 500th generation for both mutation probabilities. Consequently, it can be inferred that under these conditions, mutation probabilities of 0.01 and 0.2 yield similar qualities, facilitating a meaningful comparison.

Table 11. Analysis of mutation probability parameters

Replicaton	First Best Fitness Value		%Utility		Failed Demand (TEUs)		OverTime (Days)	
	pm = 0.01	pm = 0.2	pm = 0.01	pm = 0.2	pm = 0.01	pm = 0.2	pm = 0.01	pm = 0.2
1	-978.6	-920.85	76.4	81.15	55	2	141	101
2	-980.8	-929.72	72.2	79.28	53	9	103	125
3	-931.05	-921.18	79.95	79.82	11	1	115	127
4	-938.98	**-919.11**	79.02	80.89	18	**0**	153	116
5	-939.78	-928.74	80.22	80.26	20	9	124	79
Mean	-953.84	-923.92	77.56	80.28	31.4	4.2	127.2	109.6
Worst	-980.8	-929.72	72.2	79.28	55	9	153	127
Best	-931.05	**-919.11**	80.22	81.15	11	0	103	79

Source: Own work

Based on the findings in Table 11, a mutation probability of 0.2 demonstrates the highest potential for achieving optimal fitness values. In this research, higher fitness values indicated improved performance and closer alignment with objectives. Remarkably, the peak fitness value of -919.11 was achieved with a 0.2 mutation probability, surpassing the 0.01 probability results. The subsequent sections delve deeper into these optimal fitness value variables and compare them with the current conditions. The consistency of testing with a mutation probability of 0.2 is evident, as the best solution (-919.11) closely matches the average of all tests (-923.92). Fig. 4 and Fig. 5 provide supporting visuals, showcasing the best fitness values for mutation probabilities of 0.01 and 0.2, respectively.

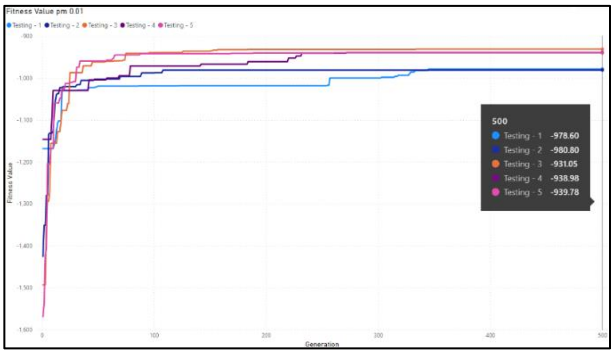

Fig. 4. The best fitness values for each generation with 500 generations and a mutation probability of 0.01 for each test
Source: Own work

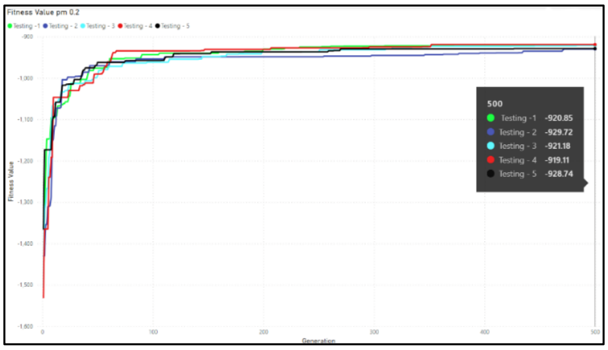

Fig. 5. The best fitness values for each generation with 500 generations and a mutation probability of 0.2 for each test
Source: Own work

The optimal fitness value achieved after 500 generations of testing was -919.11, representing a substantial improvement of 21.13%. This signifies that the average % capacity utilization of the tested vessels reached 80.89%, exceeding the historical average % capacity utilization of all vessels in Company X's history, which was 59.76% based on the data from previous operations. A mutation probability of 0.2 exhibits a higher likelihood of producing better results compared to 0.01. Although convergence solutions were achieved using 500 generations, the solution still shows a sloping, increasing trend; therefore the number of generations was increased to 1000. The number of generations was also increased to further reduce the number of overtime days. Fig. 6 presents the best fitness values for each generation with 1000 generations for each test, while Table 12 provides detailed results for each test with 1000 generations.

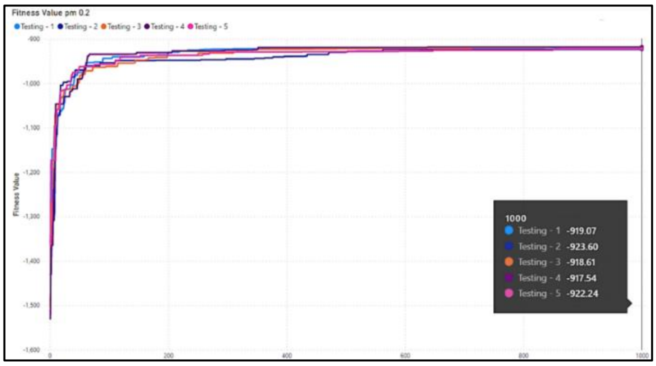

Fig. 6. Best fitness values across 1000 generations with a mutation probability of 0.2 for each test
Source: Own work

Table 12. The results of five replications with a mutation probability of 0.2 and 1000 generations

<table>
<tr><th rowspan="2">Replication</th><th colspan="2">First Best Fitness Value</th><th>%Utility</th><th>Failed Demand</th><th>OverTime</th></tr>
<tr><td colspan="2">pm = 0.2</td><td>pm = 0.2</td><td>pm = 0.2</td><td>pm = 0.2</td></tr>
<tr><td>1</td><td colspan="2">-919.07</td><td>80.93</td><td>0</td><td>94</td></tr>
<tr><td>2</td><td colspan="2">-923.6</td><td>78.4</td><td>2</td><td>138</td></tr>
<tr><td>3</td><td colspan="2">-918.61</td><td>82.39</td><td>1</td><td>123</td></tr>
<tr><td>4</td><td colspan="2">-917.54</td><td>83.46</td><td>1</td><td>115</td></tr>
<tr><td>5</td><td colspan="2">-922.23</td><td>79.76</td><td>2</td><td>76</td></tr>
<tr><td>Mean (1000 generations)</td><td colspan="2">-920.21</td><td>80.99</td><td>1.2</td><td>109.2</td></tr>
<tr><td colspan="6">Current Best Solution:</td></tr>
<tr><td colspan="2">First Best Fitness Value</td><td colspan="2">%Utility</td><td>Failed Demand</td><td>OverTime</td></tr>
<tr><td colspan="2">-919.11</td><td colspan="2">80.89</td><td>0</td><td>116</td></tr>
<tr><td colspan="6">Mean of mutation probability 0.2 with 500 Generations (all tests)</td></tr>
<tr><td colspan="2">First Best Fitness Value</td><td colspan="2">%Utility</td><td>Failed Demand</td><td>OverTime</td></tr>
<tr><td colspan="2">-923.92</td><td colspan="2">80.28</td><td>4.2</td><td>109.6</td></tr>
</table>

Source: Own work

Test 4, with 1000 generations, outperformed the previous best solution with higher %Utility (+2.57%), lower Overtime (-1 day), and improved fitness value (+1.57). However, it experienced unmet demand, impacting %Utility. As a result, Test 1 was chosen as the best solution, showing increased %Utility (+0.04) and reduced Overtime (-22 days) compared to the previous best solution, with no failed demands, providing more consistent results than Test 4. Although Test 500 met individual requirements, Test 1000 showed potential for better overall solutions with higher %Utility, lower demand failure rate, and shorter delivery times.

The increase in fitness value per generation from the optimal solution is visualized in Fig. 7. The visualization includes the second-best fitness value and the average fitness value of the entire population for each generation.

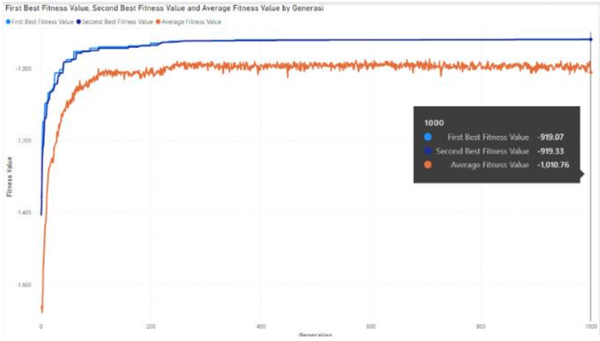

Fig. 7. Trend of fitness value per generation for the optimal solution
Source: Own work

Table 13 provides a detailed breakdown of the output from the GA process that led to the optimal fitness value from a total of 15 trips (the table displays one sample trip). Fig 8. represents the practical results obtained from this single sample trip from a total of 15 trips. It presents the planned routes to fulfill the designated demand, including the load and unload quantities, utility, and time required for each journey between POL and POD.

Furthermore, the output also provides information regarding the average % capacity utilization of the vessel for the entire voyage (80.93%), no failed demand (0 TEUs), and the total sum of overtime for the entire journey (94 days). The following section will present a comprehensive analysis of these variables.

Table 13. Output of GA process for optimal fitness value (1 out of 15 trips)

POL	POD	Chromosome	Subgroup	LOAD	UNLOAD	Shipping Load	VESSEL ID	VESSEL NAME	PAY LOAD	% UTILITY TRIP	TOTAL TIME DAYS
IDSUB	IDAMN	66	A	193	46	193	TTT	TTTTTTT	453	42.6	6.14
IDAMN	IDFFF	66	A	31	67	178	TTT	TTTTTTT	453	39.29	5.7
IDFFF	IDAKL	66	A	118	0	229	TTT	TTTTTTT	453	50.55	5.51
IDAKL	IDAGH	66	A	199	80	428	TTT	TTTTTTT	453	94.48	4.48
ADAGH	IDSUB	66	A	2	350	350	TTT	TTTTTTT	453	77.26	2.07

Source: Own work

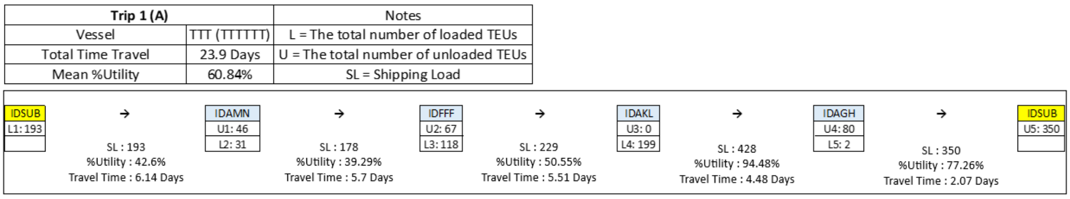

Fig. 8. Practical illustration of GA solution (1 out of 15 trips)
Source: Own work

The case study results displayed successful TEU demand, but various scenario tests uncovered potential forecast demand failures. In such cases, unmet demand and its POL and POD origins can be identified. Solutions include adapting forecast targets or allocating unmet demand to future trips.

Table 14 shows the required time for each trip to return to the home base in the proposed solution. This total time includes travel time from all POL to POD on the trip, added to port time (idle time of the vessel), and measured in days. Currently, the company does not have an ideal figure to determine how long one trip should take to fulfill the demand. Therefore, in the implemented GA program, the authors apply high penalties to solutions with significant total overtime (days > 31) across all trips. This is to minimize the total time of each journey, getting it as close as possible to the ideal figure. The results show that to meet the demand in the real-case scenario test, 15 trips were required with an average delivery time of 35.57 days.

The case study results revealed no failed TEU demand in the utilized scenario, yet various tests demonstrated potential forecast demand inaccuracies. In such instances, unfulfilled demand and its POL and POD origins can be displayed. Addressing this entails options like refining forecast targets or allocating unfulfilled demand to future trips, requiring careful consideration of utility maximization, demand fulfillment, and travel time trade-offs during the GA optimization process. Optimization focus adjustments can be realized by reevaluating parameters and penalty functions.

Table 14. The time taken by each trip from homebase to return to homebase

	Trip	VESSELID	VESSELNAME	**TOTALTIME (Days)**	**%UTILITY TRIP**
1	A	TTT	TTTTTT	23.90	60.8389
2	B	GGG	GGGGGG	25.01	71.8056
3	C	ABO	ABOABO	30.38	81.7243
4	D	WWW	WWWWWW	40.53	93.6330
5	E	ABY	ABYABY	54.22	68.3165
6	F	PPP	PPPPPP	46.16	79.2683
7	G	ABU	ABUABU	36.18	74.1385
8	H	OOO	OOOOOO	36.72	87.8021
9	I	BBB	BBBBBB	31.02	91.4530
10	J	ABS	ABSABS	48.54	79.9195
11	K	ABR	ABRABR	32.32	80.1060
12	L	QQQ	QQQQQQ	37.86	91.1635
13	M	RRR	RRRRRR	36.61	77.6126
14	N	ABI	ABIABI	24.51	86.2894
15	O	YYY	YYYYYY	29.51	89.8960
Mean				35.57	80.93

Source: Own work.

The GA program penalized solutions exceeding 31 days of overtime across trips to align with benchmarks, due to the absence of a standard trip duration. Real-case testing demanded 15 trips for demand fulfillment and successfully achieved an average trip time close to 31 days, specifically averaging 35.57 days, representing optimal results.

In the same table, the average vessel capacity utilization percentage for all trips is presented. This data offers a clear overview of trip-level utilization. Notably, each trip's average utilization surpasses 60%, with an overall average of 80.93%. These findings indicate a fairly optimal ship capacity utilization level from the GA algorithm. In the detailed analysis, the GA solution's utilization is compared with historical data to assess improvements in ship capacity utilization.

Table 15 shows the utility evaluation results for the selected vessel in GA scenario testing. "Result Utility" indicates post-GA optimization, while "History Utility" portrays historical usage (2019–2022) without GA, reflecting actual fulfillment rates. Notably, a 21.23% increase is observed. Fig. 9 further illustrates the positive impact of GA, with improved vessel performance in capacity utilization. All vessels experienced enhanced utility after GA optimization, showcasing its positive contribution to effective vessel capacity utilization for cargo delivery.

A t-test was conducted, with the null hypothesis (H0) that there was no difference between the population means of Result %Utility and History %Utility, and the alternative hypothesis (H1) that there was a higher mean for Result %Utility. The t-test results (P-Value (0.000) < alpha (0.05)) led to the rejection of H0 and the acceptance of H1. This indicates a substantial increase in the mean of Result %Utility in comparison to History %Utility.

Table 15. Results of utilization evaluation for the involved vessels in solving scenario tests with GA.

VESSELID	VESSELNAME	% Result Utility	% History Utility
ABS	ABSABS	79.92%	68.12%
TTT	TTTTTT	60.84%	55.54%
ABR	ABRABR	80.11%	54.64%
ABY	ABYABY	68.32%	57.36%
YYY	YYYYYY	89.90%	61.49%
BBB	BBBBBB	91.45%	66.23%
GGG	GGGGGG	71.81%	54.65%
RRR	RRRRRR	77.61%	52.20%
ABI	ABIABI	86.29%	76.32%
ABO	ABOABO	81.72%	53.46%
PPP	PPPPPP	79.27%	59.19%
OOO	OOOOOO	87.20%	63.22%
QQQ	QQQQQQ	91.16%	60.36%
WWW	WWWWWW	93.63%	46.76%
ABU	ABUABU	74.14%	65.67%
Mean		80.93%	59.70%

Source: Own work

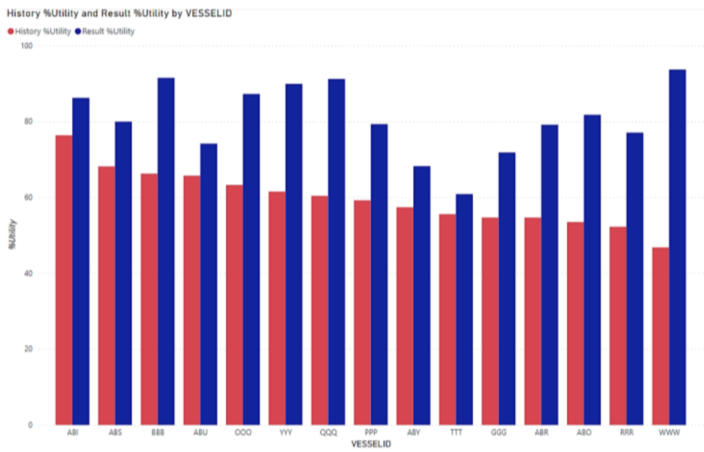

Fig. 9. Clustered Column Chart: Historical % Utility vs. % Utility Results Using GA

DISCUSSION

Comparing our study with research by Wang et al. [2013], both achieved substantial resource utilization enhancements, albeit in different contexts. Wang et al. [2013] improved land-based vehicle loading efficiency by 12.8% using a Hybrid Heuristic Algorithm inspired by local search. In contrast, our study focused on maritime operations, raising vessel capacity utilization by a remarkable 21.23%, from 59.7% to 80.93%, using a Genetic Algorithm.

This comparison highlights the versatility of VRPSDPSLTW optimization across logistics domains. Whether on land or at sea, optimization methodologies promise improved resource utilization. While Wang et al. [2013] employed a Hybrid Heuristic Algorithm, we used a Genetic Algorithm, addressing the same core problem of simultaneous deliveries and pickups with split loads and time windows (VRPSDPSLTW). These findings contribute to the discourse on optimizing logistics across diverse contexts.

CONCLUSION

The Genetic Algorithm (GA) effectively tackled the complex VRP with simultaneous pickup and delivery, split-loads, vessel capacity, and time window constraints (VRPSDPSLTW) optimization problem. Employing GA with 1000 generations, 0.9 crossover probability, and 0.2 mutation probability, vessel capacity utilization significantly rose to 80.93%, a remarkable 21.23% enhancement from the prior 59.7%. GA innovatively devised routes, yielding superior capacity utilization while considering empty container loads. Notably, the GA solution adeptly managed port visits, split-loads, time, and vessel capacity for POL to POD trips. The average trip duration was approximately 35 days, aligning closely with the 31-day target. Achieving a 100% sales target validated GA's efficacy.

However, it is essential to acknowledge the study's limitations. The research relied on anonymized historical data within a specific timeframe and uniform assumptions. Software choices simplify the model but might not account for the nuances of real-world operations. Additionally, the assumption of equal port accessibility and the exclusion of size-based limitations at specific ports could affect the generalizability of the findings. These limitations should be considered when interpreting the results and offer opportunities for further research to refine the model and address these constraints. Future research endeavors could explore multi-objective GA approaches for conflicting goals and incorporate real-world factors such as stochastic voyage time and time-dependent demand to enhance the applicability of the optimization solution.

RECOMMENDATIONS

To enhance the GA's performance in future research, the following suggestions can be considered:

- Modify the crossover operator: Test different crossover operators to explore the possibility of finding more efficient solutions for the VRPSDPSLTW context.
- Consider weather factors: Analyze the impact of weather conditions on vessel voyages for the researched routes and evaluate their effects on travel time.
- Expand research to involve all Company X's vessels from various homebases, including Jakarta port as the homebase for vessel journeys, and consider demand for all POL to POD services from the west and east regions.

By taking these recommendations into account, future research is expected to provide solutions that are closer to actual conditions, more efficient, and optimal in resolving the VRPSDPSLTW problem.

ACKNOWLEDGMENTS

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Arnold, F., Sörensen, K., 2019. What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. Computers & Operations Research, 106, 280-288. https://doi.org/10.1016/j.cor.2018.02.007.

Cheng, C.-B., Wang, K.-P., 2009. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. Expert Systems with Applications, 36, 7758-7763. https://doi.org/10.1016/j.eswa.2008.09.001.

Dantzig, G. B., Ramser, J. H., 1959. The truck dispatching problem. Management Science, 6(1), 80-91.

Dewi, S. K., Utama, D. M., 2021. A new hybrid whale optimization algorithm for the green vehicle routing problem. Systems Science & Control Engineering, 9, 61-72. https://doi.org/10.1080/21642583.2020.1863276.

Dror, M., Laporte, G., Trudeau, P., 1994. Vehicle routing with split deliveries. Discrete Applied Mathematics, 50(3), 239–254.

Dror, M., Trudeau, P., 1989. Savings by split delivery routing. Transportation Science, 23(2), 141–145.

Escobar-Falcón, L., Álvarez-Martínez, D., Wilmer-Escobar, J., Granada-Echeverri, M., 2021. A specialized genetic algorithm for the fuel consumption heterogeneous fleet vehicle routing problem with bidimensional packing constraints. International Journal of Industrial Engineering Computations, 12, 191-204. https://doi.org/10.5267/j.ijiec.2020.11.003.

Ho, W., Ho, G. T. S., Ji, P., Lau, H. C. W., 2008. A hybrid genetic algorithm for the multi-depot vehicle routing problem. Engineering Applications of Artificial Intelligence, 21, 548-557. https://doi.org/10.1016/j.engappai.2007.06.001.

Karakatič, S., 2021. Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. Expert Systems with Applications, 164, 114039. https://doi.org/10.1016/j.eswa.2020.114039.

Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. Transportation Research Part E: Logistics and Transportation Review, 45, 434-445. https://doi.org/10.1016/j.tre.2008.10.003.

Mohammed, M. A., Abd Ghani, M. K., Hamed, R. I., Mostafa, S. A., Ahmad, M. S., & Ibrahim, D. A., 2017. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. Journal of Computational Science, 21, 255-262. https://doi.org/10.1016/j.jocs.2017.04.003.

Nazif, H., Lee, L. S., 2012. Optimized crossover genetic algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling, 36, 2110-2117. https://doi.org/10.1016/j.apm.2011.08.010.

Oliveira da Costa, P. R. de, Mauceri, S., Carroll, P., Pallonetto, F., 2018. A genetic algorithm for a green vehicle routing problem. Electronic Notes in Discrete Mathematics, 64, 65-74. https://doi.org/10.1016/j.endm.2018.01.008.

Protopopova, J., Kulik, S., 2020. Educational intelligent system using genetic algorithm. Procedia Computer Science, 169, 168-172. https://doi.org/10.1016/j.procs.2020.02.130.

Saxena, R., Jain, M., Malhotra, K., Vasa, K. D., 2020. An optimized OpenMP-based genetic algorithm solution to the vehicle routing problem. In The Smart Computing Paradigms: New Progresses and Challenges (Vol. 767, pp. 237-245). https://doi.org/10.1007/978-981-13-9680-9_20.

Sitek, P., Wikarek, J., Rutczyńska-Wdowiak, K., Bocewicz, G., Banaszak, Z., 2021. Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. Neurocomputing, 423, 670-678. https://doi.org/10.1016/j.neucom.2020.02.126.

Toth, P., Vigo, D., 2002. The vehicle routing problem. SIAM. http://dx.doi.org/10.1137/1.9780898718515

Utama, D. M., Dewi, S. K., Wahid, A., Santoso, I., 2020. The vehicle routing problem for perishable goods: A systematic review. Cogent Engineering, 7, 1816148. https://doi.org/10.1080/23311916.2020.1816148.

Visutarrom, T., Chiang, T., 2019. An evolutionary algorithm with heuristic longest cycle crossover for solving the capacitated vehicle routing problem. In IEEE Congress on Evolutionary Computation (CEC), 2019 (pp. 673-680). https://doi.org/10.1109/CEC.2019.8789946.

Wang, Y., Ma, X., Lao, Y., Wang, Y., Mao, H., 2013. Vehicle routing problem simultaneous deliveries and pickups with split loads and time windows. Transportation Research Record Journal of the Transportation Research Board, 2378, 120-128. https://doi.org/10.3141/2378-13

Shea Amanda Ferdianti
Department of Industrial Engineering,
Faculty of Industrial Technology Petra Christian University, East Java, **Indonesia**
e-mail: c13190042@john.petra.ac.id

I Gede Agus Widyadana ORCID ID: https://orcid.org/0000-0001-9129-7468
Department of Industrial Engineering,
Faculty of Industrial Technology Petra Christian University, East Java, **Indonesia**
e-mail: gede@petra.ac.id