Volume 25 • Number 3 • July 2025

Cilt 25 • Sayı 3 • Temmuz 2025

Contents

EGE AKADEMİK BAKIŞ / *EGE ACADEMIC REVIEW*

Cilt 25 • Sayı 3 • Temmuz 2025
SS. 595/608
Doi: 10.21121/eab.20250309
Başvuru Tarihi: 17.12.2024 • Kabul Tarihi: 16.05.2025

Article Type: *Research Article*

A Smart - Selection - Based Genetic Algorithm for Delivery and Pickup Problem with Order Time Windows

Ural Gökay ÇİÇEKLİ[1], Aydın KOÇAK[2], Ege CİHANGİR[3]

ABSTRACT

In a highly competitive environment, businesses strive to optimize their distribution networks to reduce logistics costs. This study focuses on solving vehicle routing problems involving simultaneous delivery and pickup with time windows, addressing both backhaul and divisible delivery and pickup scenarios. A novel hybrid genetic algorithm incorporating smart selection and harem-based crossover methods is proposed to minimize transportation costs while adhering to capacity and time constraints. The smart selection method expedites the solution process by pre-selecting feasible vehicle-route combinations, significantly reducing the computational complexity. Computational experiments on real-world data from the automotive supply industry demonstrate that the proposed algorithm outperforms traditional approaches, achieving substantial cost reductions and high-quality solutions within shorter computation times.

Keywords: Vehicle Routing Problem, Delivery and Pickup Problem, Genetic Algorithm, Smart Selection, Hybrid Optimization.

JEL Classification Codes: C44, C61

Referencing Style: APA 7

INTRODUCTION

One of the fundamental ways to reduce costs in enterprises is by minimizing logistics expenses, which constitute a significant portion of product costs. Distribution costs, accounting for approximately 50% of logistics expenses, are crucial for operational efficiency. The Vehicle Routing Problem (VRP), first introduced by Dantzig and Ramser (1959), plays a significant role in optimizing distribution networks. It focuses on creating a set of minimum-cost routes for a fleet of vehicles to meet customer demands, often subject to constraints like capacity, time windows, and precedence rules (Baker and Ayechew, 2003).

This paper discusses a delivery and pickup problem which is an extension to the VRP. The vehicles should not only deliver goods to the suppliers but also pick up some goods at the supplier locations. In this type of problem, there are two types of suppliers: linehaul and backhaul (Wang and Chen, 2013). A linehaul supplier requests a specific quantity of products to be delivered from the depot, while a backhaul supplier requires a certain quantity of products to be collected and transported back to the depot (Liu et al., 2013). Our problem considers every supplier as either "as both linehaul and backhaul" or "as either linehaul or backhaul". Hence, we must deliver to every supplier and pick up from every supplier for delivery to the depot.

Our paper also contains vehicle routing problems with backhauls (VRPB), which necessitates that all deliveries must be made on each route before any pickups can be made (Goetschalckx and Jacobs-Blecha, 1989). This is also named the delivery-first, pickup-second strategy. Allowing pickups before completing all deliveries could lead to vehicle overloading during the trip, which may render the tour infeasible (Wang and Chen, 2013). The above-mentioned strategy eliminates such infeasible vehicle tours. In this strategy, the vehicle can visit the supplier twice, first for delivery and then for pickup service. Therefore, this is called the divisible delivery and pickup problem.

Zhong and Cole (2005) introduced a local search heuristic specifically designed to address the VRPB.

[1] Assoc. Prof. Ural Gökay Çiçekli, Ege University, Faculty of Economics and Administrative Sciences, Bornova, 35100, Izmir, Türkiye, gokay.cicekli@ege.edu.tr

[2] Assoc. Prof. Aydın Koçak, Ege University, Faculty of Economics and Administrative Sciences, Bornova, 35100, Izmir, Türkiye, aydin.kocak@ege.edu.tr

[3] Ege Cihangir (Independent Researcher), Mansuroğlu District, 1593/1 Street, No: 14, Apartment: 17, Bayraklı, 35530, Izmir, Türkiye, egecihangirr@gmail.com

Customer precedence and time windows increased the problem complexity. They developed a new technique of section planning to solve the problem. Gajpal and Abad (2009) presented a multi-ant colony system algorithm for solving VRPB.

On the other hand, sometimes our problem includes simultaneous pickups and deliveries. The Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) is a problem where pickup and delivery services are simultaneously operated. The term simultaneous refers to the distribution of delivery and pickup of goods once the destination is reached. Vehicles visit the supplier only once and depart from the supplier after completing the delivery and pickup operation simultaneously. The problem has garnered significant attention from researchers due to its efficiency in managing both services effectively.

VRPSPD was first introduced by Min (1989) for the library distribution system. Dethloff (2001) developed a heuristic algorithm based model for the VRPSPD in reverse logistics processes. Dell'Amico, Righini, and Salani (2006) proposed a dynamic programming model based on the branch and price algorithm for a solution to the VRPSPD in reverse logistics processes. Ashouri and Yousefikhoshbakht (2017) presented metaheuristic models based on the ant colony optimization (ACO) algorithm to solve the VRPSPD.

In this study, we address a complex VRP scenario involving simultaneous delivery and pickup with order time windows, capacity constraints, and real-world operational requirements. Unlike traditional approaches, we propose a novel hybrid genetic algorithm (GA) that incorporates smart selection and harem-based crossover methods. The smart selection method expedites the solution process by pre-selecting feasible vehicle-route combinations, significantly reducing the computational complexity. The harem-based crossover mechanism enhances genetic diversity and accelerates convergence to high-quality solutions (Çiçekli, 2012).

Key contributions of this study include:

1. Development of a hybrid GA tailored for simultaneous delivery and pickup operations with time windows.
2. Application of the algorithm to real-world data from the automotive supply industry, demonstrating its practical effectiveness in reducing transportation costs.
3. Integration of the harem-based crossover method into the GA framework, inspired by biological processes, to improve solution quality and computational efficiency.

By addressing these theoretical and practical challenges, this paper provides a robust framework for optimizing logistics operations under competitive and real-world constraints.

PROBLEM STATEMENT

The Vehicle Routing Problem with Time Windows (VRPTW) is a well-established NP-Hard problem, as identified by Lenstra and Kan (1981). An NP-Hard problem is characterized by the exponential growth of computational complexity as the problem size increases, making classical solutions impractical for large-scale instances. Anily (1996) further highlights that solving such problems requires heuristic or metaheuristic approaches, as exact methods become computationally infeasible. These heuristic methods aim to find near-optimal solutions within polynomial running times, balancing solution quality and computational efficiency.

This study addresses the VRP of a company operating in the automotive supply industry. Due to competitive concerns, the names of the company and its suppliers are kept confidential. Different processes are carried out with the suppliers, depending on the product structures of the company. As a result, either empty or filled boxes can be delivered to the suppliers, while only filled boxes are picked up from them.

The problem involves three different suppliers, referred to as S1, S2, and S3. Route costs vary based on vehicle type and route length, reflecting the operational complexity of the problem.

The company uses the following two types of vehicles that have different maximum permissible weights: semi-trailer trucks and trucks. The maximum load limit is 25000kg for a semi-trailer truck and 12000kg for a truck.

Since the company outsources vehicles, the study had no vehicle limitation. Therefore, all route types [(S1), (S2), (S3), (S1, S2), (S1, S3), (S2, S3), (S1, S2, S3)] and two vehicle types (semi-trailer trucks and trucks) were available for use each day within the specified time periods. As a result, 15 different vehicle-route combinations were created for each day. If a vehicle visits S1, S2, and S3 in sequence, it returns via the reverse route, S3, S2, and S1. Deliveries are made during the outbound journey, while pickups occur on the return journey. No pickups

Table 1: Example of Delivery and Pickup Orders with Time Windows

Order No	Supplier	Box (pcs)	Weight (kg)	Earliest (day)	Latest (day)	Order No	Supplier	Box (pcs)	Weight (kg)	Earliest (day)	Latest (day)
DO1	S3	36	160	6	8	PO1	S3	15	250	6	8

can take place before the outbound route is completed. Therefore, vehicle capacity is calculated separately for deliveries and pickups. Additionally, empty or filled boxes delivered to or picked up from any supplier had to adhere to specific time intervals. It was known how many empty or filled boxes had to be delivered to which supplier on which dates and how many filled boxes had to be picked up. For convenience, each order was encoded as "DO" for departures and "PO" for returns. Table 1 shows an example of departure and return orders.

In the VRP, costs vary depending on route length and vehicle type. The objective of the VRP was to minimize transportation costs. This study aimed to minimize the cost of selected routes and vehicles within the specified period.

A method called smart selection was developed to make the model run faster. Using smart selection, clusters of vehicles with suitable routes and schedules were generated for each order. If smart selection had not been used, there would have been 6,5332E+127 different potential solutions to the problem. Since orders can be assigned to appropriate vehicles, the vehicles are only checked for weight capacity. There is no volume restriction on vehicles, since the company's orders involve low volume high weight goods. Given these conditions, there are 1,2349E+72 different potential solutions to the problem. The problem addressed in this study extends the complexity of VRPTW by incorporating simultaneous delivery and pickup operations with time windows, vehicle capacity constraints, and multiple route scenarios. To handle these challenges effectively, this study proposes a hybrid GA that integrates smart selection and harem-based crossover mechanisms. The smart selection method pre-screens infeasible routes, significantly reducing the solution space and ensuring computational feasibility. Meanwhile, the harem-based crossover mechanism enhances genetic diversity within the population, preventing convergence to suboptimal solutions.

MATHEMATICAL MODEL

This section presents the mathematical model developed to solve the VRPSPD, time windows, and capacity constraints. The model minimizes the total transportation cost while ensuring that operational and logistical constraints are satisfied.

Sets

S: $Set\ of\ suppliers, S = \{s_1, s_2, s_3\}$.

V: Set of vehicles, $V = \{v_1, \dots, v_v\}$.

D: Set of delivery orders, $D = \{d_1, \dots, d_i\}$.

P: Set of pickup orders, $P = \{p_1, \dots, p_i\}$.

Parameters

C_v: Capacity of vehicle $v \in V$.

T_v: Scheduling time of vehicle $v \in V$.

R_m^v: Cost of delivery route m for vehicle $v \in V$.

R_j^v: Cost of pickup route j for vehicle $v \in V$.

W_{d_i}: Weight of delivery order $d_i \in D$.

W_{p_i}: Weight of pickup order $p_i \in P$.

e_{p_i}, l_{p_i}: Earliest and latest pickup times for $p_i \in P$.

e_{d_i}, l_{d_i}: Earliest and latest delivery times for $d_i \in D$.

Decision Variables

$$x_v = \begin{cases} 1, & \text{if vehicle } v \in V \text{ travels,} \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{p_i}^v = \begin{cases} 1, & \text{if vehicle } v \in V \text{ transports pickup order } p_i \in P, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{d_i}^v = \begin{cases} 1, & \text{if vehicle } v \in V \text{ transports delivery order } d_i \in D, \\ 0, & \text{otherwise.} \end{cases}$$

Objective Function

The objective is to minimize the total cost of delivery and pickup operations:

$$Minimize\ Z = \sum_{v \in V} \sum_{m} R_m^v \cdot x_v + \sum_{v \in V} \sum_{j} R_j^v \cdot x_v$$

The first term minimizes the cost of delivery routes, while the second term minimizes the cost of pickup routes.

Constraints

Vehicle Capacity Constraints:

$$\sum_{p_i \in P} W_{p_i} y_{p_i}^{v} \leq C_v \qquad \forall v \in V$$

The total weight of all pickup orders assigned to vehicle v must not exceed its capacity C_v.

$$\sum_{d_i \in D} W_{d_i} y_{d_i}^{v} \leq C_v \qquad \forall v \in V$$

The total weight of all delivery orders assigned to vehicle must not exceed its capacity C_v.

The capacity constraints determine whether the problem is simultaneous pickup and delivery. Also, the manager decides whether to transport loads between orders according to the capacity constraints.

Order Assignment Constraints:

Each delivery order must be assigned to exactly one vehicle:

$$\sum_{v \in V} y_{d_i}^{v} = 1 \qquad \forall d_i \in D$$

Each pickup order must be assigned to exactly one vehicle:

$$\sum_{v \in V} y_{p_i}^{v} = 1 \qquad \forall p_i \in P$$

Time Window Constraints:

Also, if vehicle v handles pickup or delivery order, pickup or delivery time T_v must fall within the time window specified for:

$$e_{p_i} y_{p_i}^{v} \leq T_v y_{p_i}^{v} \leq l_{p_i} y_{p_i}^{v} \qquad \forall p_i \in P, \forall v \in V$$

$$e_{d_i} y_{d_i}^{v} \leq T_v y_{d_i}^{v} \leq l_{d_i} y_{d_i}^{v} \qquad \forall d_i \in D, \forall v \in V$$

Binary Decision Variables:

$$x_v, y_{p_i}^{v}, y_{d_i}^{v} \in \{0,1\} \qquad \forall v \in V, \forall p_i \in P \; \forall d_i \in D$$

This mathematical model incorporates simultaneous delivery and pickup operations while adhering to time window and capacity constraints. The smart selection method simplifies the solution process by predefining feasible vehicle-route combinations, allowing for faster computation and high-quality solutions. A total of 360 different vehicle-route-time combinations were generated, considering time, route and capacity. With smart selection, suitable vehicle-route-time alternatives were determined in advance for each order. For example, for the 1st picking order, the number of alternatives decreased from 360 to 27.

METAHEURISTICS and SOLUTION PROCEDURE

Metaheuristic methods are a means of solving difficult and complex problems in various fields. Metaheuristic methods are used to efficiently find optimum or near optimum solutions by combining different concepts to explore the search space (Osman and Laporte, 1996). Many metaheuristic algorithms such as simulated annealing, GA, PSO, ACO, artificial neural networks, and tabu search are used to solve VRP (Çiçekli, 2012). Michalewicz et al. (1991) define a GA as a class of algorithms related to the genetic processes occurring in nature with randomly generated population and towards developing better solutions. GA makes it possible to discover solutions to a much wider range of problems than other algorithms (Hsiao et al., 2010). GA enables exploration of a problem over a much wider range than with other algorithms.

The proposed hybrid GA incorporates smart selection and harem-based crossover mechanisms to optimize VRPSPD with time windows. This section outlines the key steps of the algorithm and highlights its unique components.

Solution procedure

The GA developed in this study follows the steps below to optimize the VRPSPD:

1. Load Parameters:

Load all problem parameters, such as vehicle capacities, order weights, time windows, and other constraints.

2. Smart Selection:

Apply the smart selection method to pre-filter infeasible routes. This reduces the solution space and ensures that the initial population contains only feasible solutions.

3. Generate Initial Population:

Create the initial population of chromosomes (potential solutions) based on the feasible routes identified by the smart selection method.

4. Evaluate Fitness:

Evaluate the fitness of every chromosome in the population to determine their suitability as potential solutions. The fitness function considers factors such as total cost, time window satisfaction, and capacity constraints.

5. Select Best Chromosomes:

Identify the best-performing chromosomes (solutions) based on their fitness values.

6. Harem-Based Crossover (Step 1):

Select the best chromosome as the "leader" (first parent). Choose subsequent chromosomes in the population as "followers" (second parents) to form the harem. This ensures diversity and avoids premature convergence.

7. Generate Offspring (Step 2):

Combine the leader and followers to generate offspring using the crossover operator. The harem-based crossover ensures genetic diversity and high-quality solutions.

8. Apply Mutation:

Introduce mutations into the offspring population with a predefined mutation rate. Mutation helps explore new areas in the solution space and prevents stagnation.

9. Evaluate New Generation:

Calculate the fitness values of the new generation of chromosomes.

10. Replace Population:

Replace the current population with the new generation if the fitness of the new generation is better.

11. Check Termination Criteria:

The algorithm terminates when one of the following conditions is met:

- The maximum number of iterations is reached.
- The improvement between successive generations falls below a predefined threshold.

12. Output Best Solution:

Output the best chromosome (solution) as the optimal route for the problem.

Chromosome Encoding

One of the problems in the GA is the identification of encoding. Çiçekli (2012) notes that there are four categories of coding: binary, real number, permutation, and data structure.

In real-number encoding, chromosomes are made up of the real value of parameters (Herrera et al., 1998). Goldberg (1991) and Eshelman and Schaffer (1993) used real-coded GA for function optimization that represents the real number vector of chromosomes. These real-coded GA performed better than traditional bit-based (binary coding) GA. Wright (1991) listed the strengths of real-coded GA as increased efficiency, increased precision, and greater freedom in using different mutations and crossover techniques.

This study employed real-number encoding to achieve more efficient outcomes. For the representation of chromosomes as per the GA, genes were allocated to all routes and vehicle types within each day in the program. Considering many factors such as the variety of routes and schedules, and the maximum weight limit of vehicles, real-number encoding was used to achieve a more efficient outcome. Table 2 displays the codes of semi-trailer trucks and trucks for 24 days in line with the route schemes and their costs.

As shown in Table 2, two different types of vehicles were encoded based on days for all potential routes. "ST1_1" represents a semi-trailer truck with the route "S1", while

Table 2: Vehicle Types, Costs, and Routes Encoded by Dates

Day	ST1_1	ST2_1	ST3_1	ST12_1	ST13_1	ST23_1	ST123_1	T1_1	T2_1	T3_1	T12_1	T13_1	T23_1	T123_1	T123_2
5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Cost	675	675	850	795	1525	1525	1645	550	550	700	650	1250	1250	1350	1350

Table 3: Real-Coded Chromosome

24	7	11	156	225	75	128	140	139	33	57	148	203	21	88	154	3	193	136	10	62	192	31	164	63

"T123_1" represents a truck with the routes S1, S2, and S3. The vehicle encoded 20 represents a semi-trailer truck at the cost of 1525 with the routes S1 and S3 running on day 6. What is important here is the encoding shaped by the dates.

By making use of the coding of semi-trailer trucks and trucks, real-coded values make up the sample chromosome sequence as shown in Table 3.

As seen in the real-coded chromosome example in Table III, code 24 in the first gene means that the first order in the order list will be carried by the truck with the route S2 on day 6. Similarly, code 7 in the second gene indicates that the second order in the order list will be carried by the semi-trailer truck with the route S1-S2-S3 on day 5. Following the real-number encoding, the initial population is generated.

Initial Population

An initial population (generation) of random individuals which shows solutions of the related problem is created (Baker and Ayechew, 2003). One of the parameters of GA is population size, showing how many individuals are in the population (Çiçekli and Kaymaz, 2015). The size of the population is important for an efficient GA. If the number of individuals is small, only a certain part of the solution space can be reached, and there is not much choice for crossover (Grefenstette, 1992).

Individuals in the population are obtained by randomly generated real-coded chromosome sequences. However, these randomly generated chromosome sequences often do not provide the constraints of route and time intervals. In order to prevent this and with the aim of reaching the best solution more quickly, the smart selection method is used to create the population. Therefore, in the mathematical model, route and time interval constraints are not used.

Evaluation: Fitness Function

The process to be performed after the initial population is created is to calculate the fitness values of the generated chromosomes. VRP aims to reveal the most cost-effective chromosome sequence in such a way as to provide constraints. GA operators are included in the process, and after the new chromosomes are obtained this fitness value is calculated again.

In the study, the fitness function tried to reach the minimum cost. The calculation of the fitness function took place by collecting the current costs of the selected vehicles, regardless of departure and return. Even if a vehicle was used in departure and not used on return, both the departure and the return costs were generated. Therefore, the model aimed to use vehicles in both directions. Also, if the capacities of the vehicles were not suitable for the necessary transport (if the total of the departure or return weights was more than the weight of the vehicle), they were exposed to the penalty cost, and they were left out of the solution.

Selection Harem

After the initial population and identified fitness function selection are made in the GA, replication of the chromosomes in the current population and the possibility of placement in the new generation are proportional to their fitness (Whitley, 1994). Parent selection is the undertaking of allotting regenerative opportunity to every individual in the population. Elitism is one of the selection methods which arrange the chromosomes in increasing order according to their fitness value (Yadav and Sohal, 2017). We used elitism method for our algorithm. For solving the problem using the elitism method, the best 21 chromosomes in the population are selected. The 1st place chromosome takes the name "leader chromosome". The subsequent 20 chromosomes constitute the "harem". This method prepares the ground for the crossover operator.

Crossover

The crossover operator is designed to merge key attributes from two high-quality "parent" solutions, aiming to produce superior "offspring" solutions. Its primary objective is to create valid solutions that inherit the strengths of both parents. The specific coding of the problem directly influences the structure and implementation of the crossover operator (Poon and Carter, 1995).

It has been seen that crossover processes are employed for parent selection using the harem structure, which is used in many aspects of natural life. The harem represents the chromosome pool where the best chromosomes are collected as a result of the processes. The harem contains

Table 4: Crossover Example

Parent 1	29	86	103	92	116	94	123	107	149	206	205	201	247	251	251
Parent 2	48	86	87	107	98	114	119	109	130	169	209	232	200	238	239
Bit	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
Offspring	48	86	87	107	116	114	123	107	130	206	209	232	247	238	251

the best chromosomes in the population, behind the best chromosome called the leader chromosome. It has been seen that the harem method brings diversity by making loops efficient in GA (Çiçekli, 2012).

In our test studies, it was decided to determine the harem size as 20. Therefore, 20 chromosomes after the leader chromosome were included in the harem. The critical point here is that the leader remains constant in every crossover and that the chromosomes in the harem in turn enter the crossover with the leader. At this stage, the crossover operation is performed by the dominance code. Dominance code is randomly generated as bits. If the value is 1, the leader is the dominant, if the value is 0, the second parent will be dominant. Dominance code determines which parent will affect the genes of the new generation. As a result of each crossover, a new generation occurs.

Mutation

The crossover operator used to investigate gene potential may sometimes not produce the desired solution. In such a case, a mutation operator with the ability to produce new chromosomes from the existing chromosomes is required. Mutation aims to reveal new genetic material from an existing individual (Engelbrecht, 2007).

The new generation is added to the population after the fitness values are calculated. However, the crossover operator alone may not be enough to achieve the best solution. In such an eventuality, implementation of the mutation operator is necessary. The most accurate mutation process can take place with the most appropriate mutation rate (Kocamaz and Çiçekli, 2010).

After crossover and mutation operators, sometimes duplicate chromosomes can occur. Lim and Chew (1997) and Brown et al. (1994) performed operations ensuring that no duplicate chromosomes were permitted within the population. As duplicate chromosomes contain the same information, all duplicated chromosomes were removed from the population in this study.

The best chromosome is obtained after the crossover operator is mutated. The application of the mutation operator depends on the mutation rate, randomly assigned to each gene of the best-selected chromosome. If the gene is not covered by the specified mutation rate, the gene remains the same on the new chromosome. If the gene is covered by the determined mutation rate, the gene is added to the chromosome by taking advantage of the previously mentioned smart selection structure. This is done for all genes in the chromosome. Thus, the new chromosome generated is expected to give better results.

COMPUTATIONAL RESULT

In this paper, the GA was developed in Visual Basic for Applications (VBA) for Excel. The advantage of Excel as a development environment is that it provides capabilities that permit analysis and control of the output and results. By creating the "Userform", the algorithm is processed according to the values determined by the decision maker. The GA we developed was run on 40 equivalent computers in the computer lab.

Impact of population size

It is an important issue to determine the size of the initial population. Determining the optimal population size will create a more favorable environment for the solution. Figure 1 shows the tests to find the optimal population size. We tested population size from 100 to 1500 in multiples of 100. We ran each population size 50 times. The tests identified the optimal initial population size as 700.

Smart selection helped eliminate those genes that made it impossible to speed up the solution to the GA, while creating the population. As a result, it was observed that improvements in the solution to the problem were achieved within a short time.

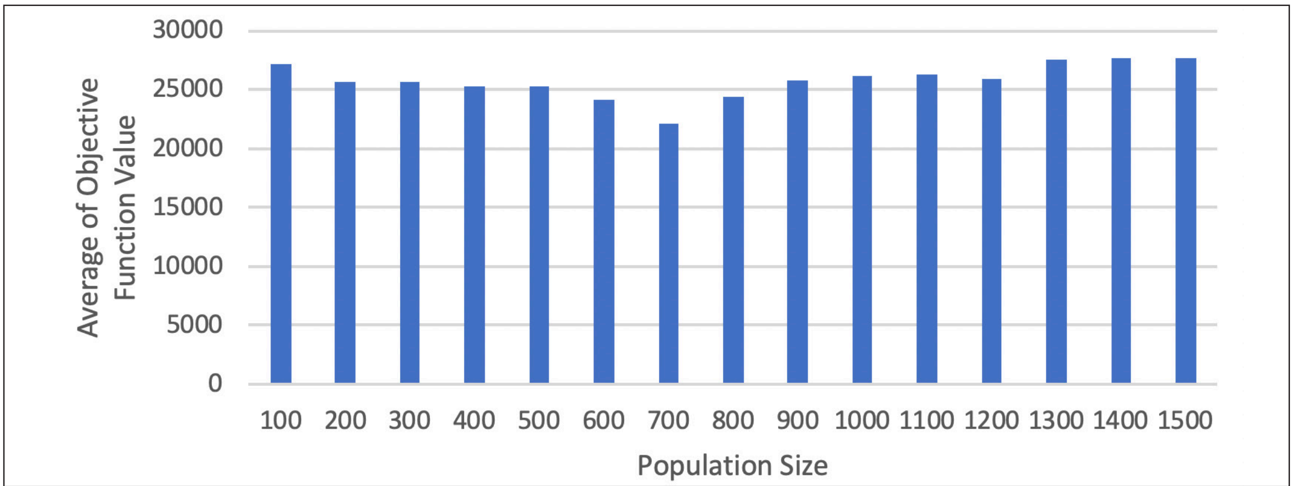

Figure 1: Impact of Population Size on the Objective Function Value

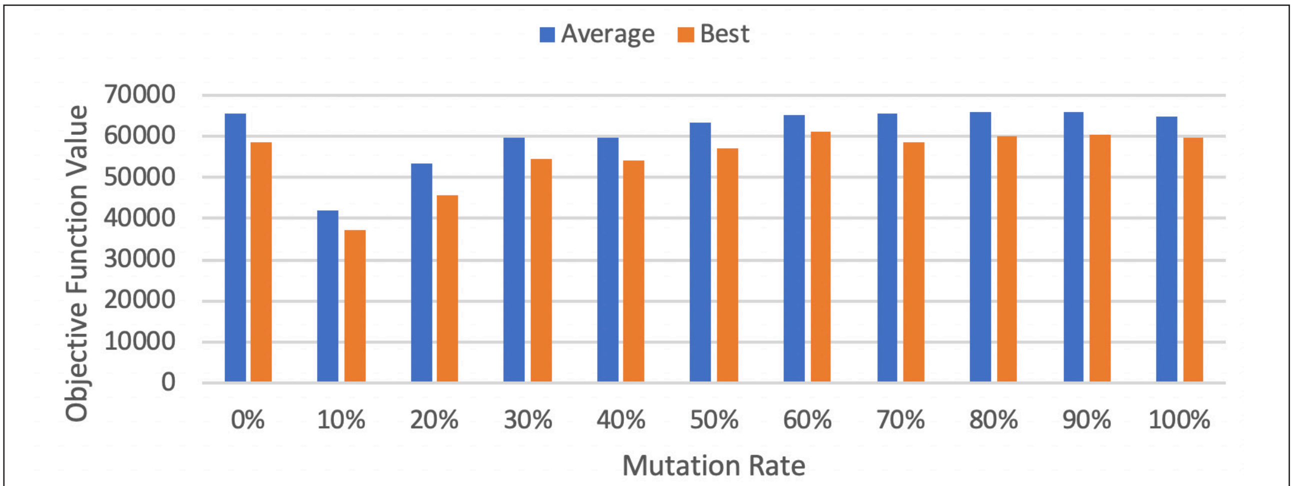

Figure 2: Impact of Mutation Rate on the Objective Function Value (0-100)

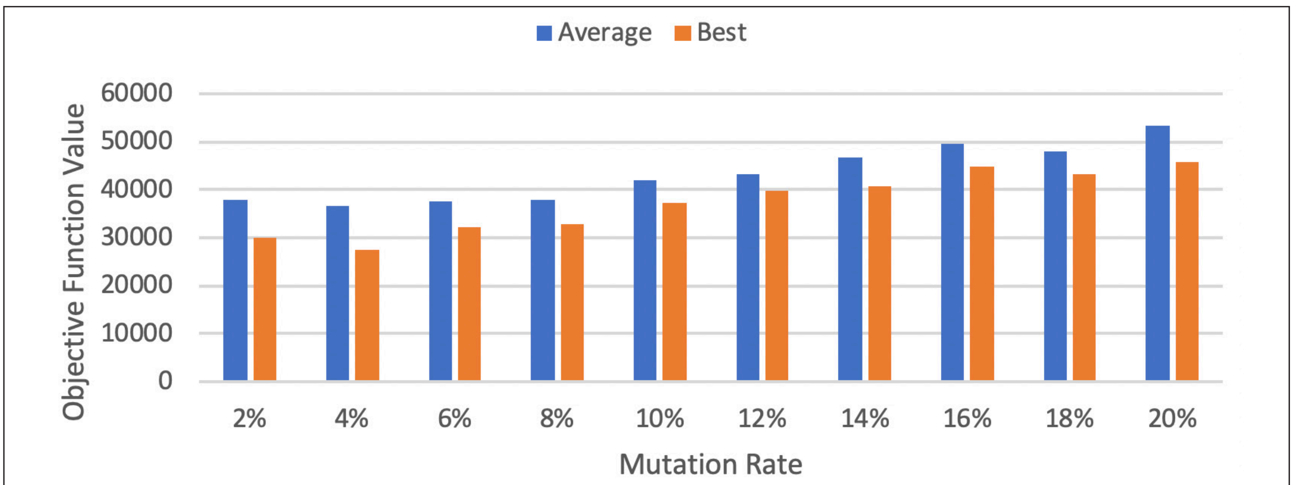

Figure 3: Impact of Mutation Rate on the Objective Function Value (2-20)

Impact of mutation rate

We have carried out various tests to find the most appropriate mutation rate for the problem structure. We tried to find the segment range where the appropriate mutation rate would take place. From 0% up to 100%, mutation rates were tested to increase by 10%. In the tests, each mutation rate was run 50 times. Fifty trials were performed with 1000 iterations for each mutation rate. As shown in Figure 2 the lowest average and the best value were found in 10% mutation rate. Therefore, we performed more detailed analyses for 2-20% segment (Figure 3).

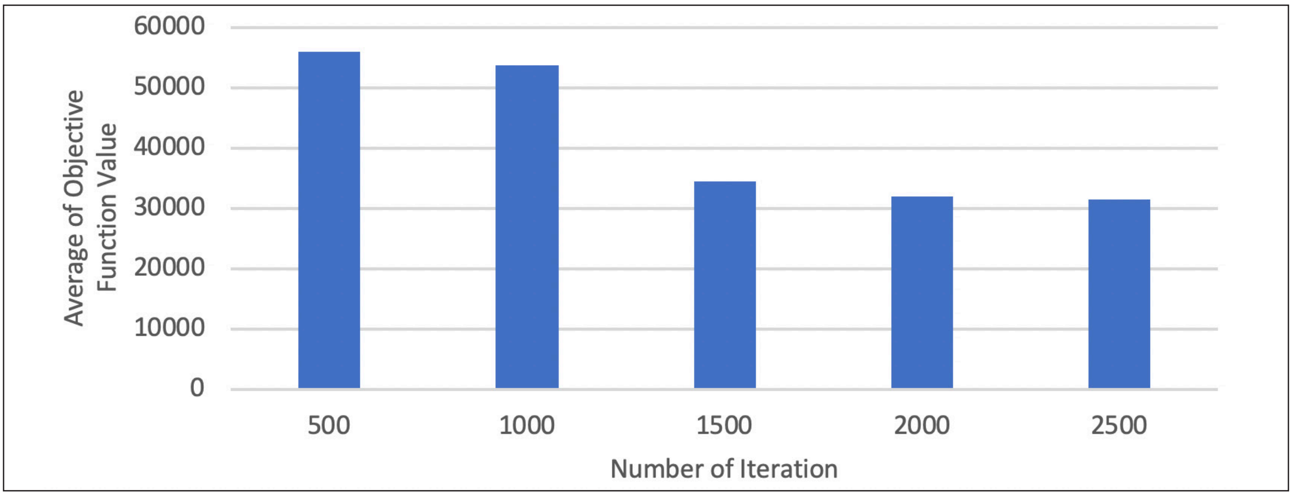

Figure 4: Impact of the Number of Iterations on the Objective Function Value

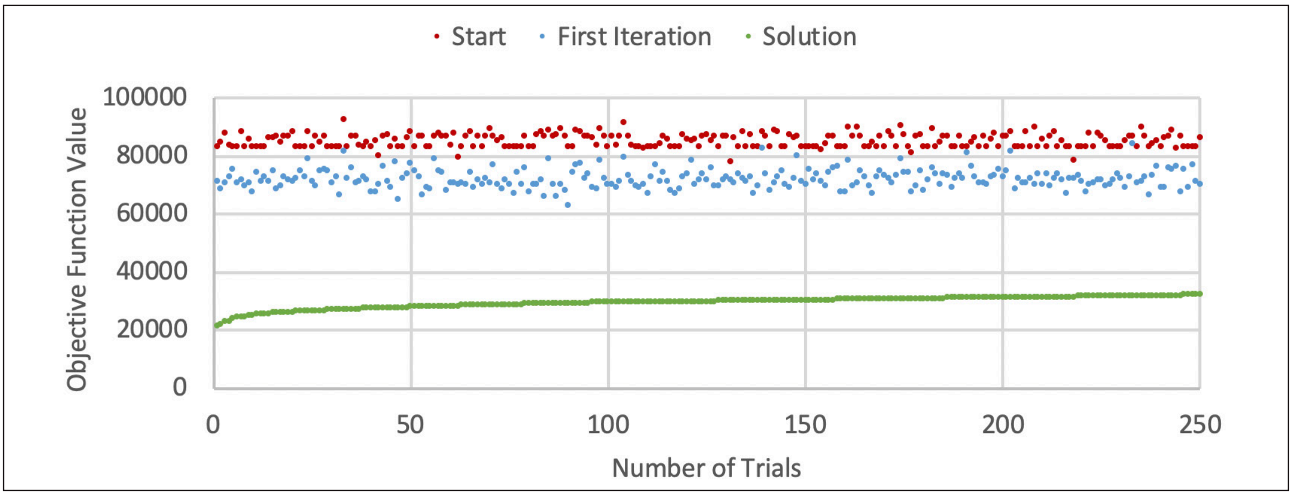

Figure 5: Objective Function Value of Initial, First Iteration, and Solution for Each Iteration

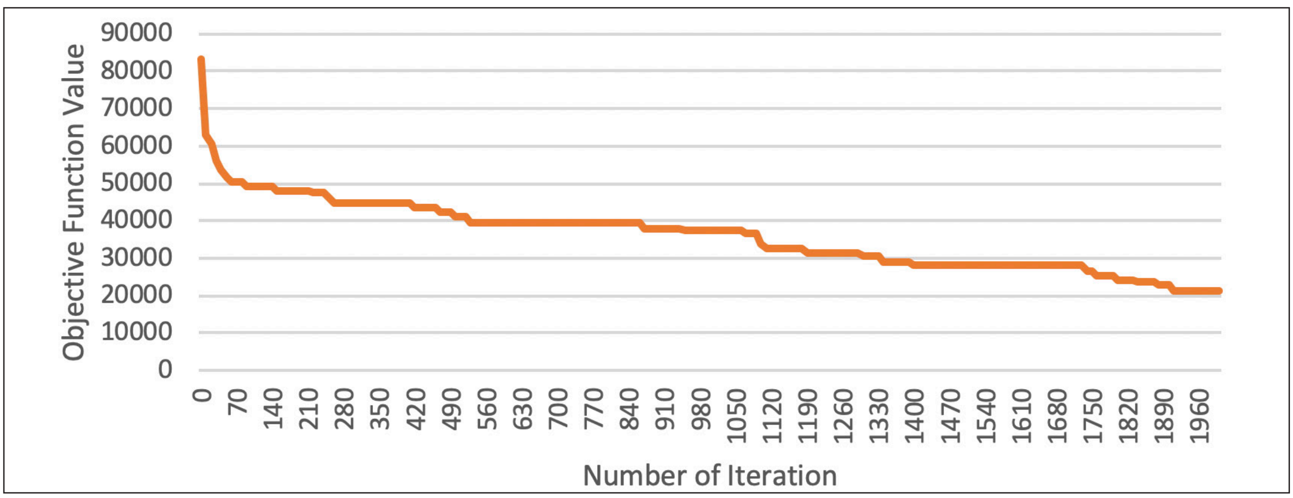

Figure 6: Evolution of the Best Solution

Figure 3 shows the detailed analysis of mutation rates between 2-20%. In the tests, each mutation rate was run 50 times. As a result of our tests, we determined the best mutation rate for this problem at 4%.

Impact of the number of iterations

We tested the developed model for 500, 1000, 1500, 2000, and 2500 iterations. Figure 4 shows the impact of the number of iterations on the objective function value. As a result of our tests, we determined that the best number of iterations for this problem was 2000.

Table 5: The Chromosome of the Best Solution

D	25	86	103	86	86	114	103	103	130	189	190	219	218	219	234	218	234	237	324	310	324	286	341	341	341
P	25	86	103	86	86	114	103	114	130	190	189	190	218	219	234	237	234	237	324	310	324	286	341	341	341

Best Solution

The developed model was run over 250 trials as 2000 iterations. Figure 5 presents the values of each repetition. These values include the initial value, the first iteration value, and the solution value. Solutions are sorted in ascending order. The average value of 250 trials is 29453, and the standard deviation is 1925. The computation time for 2000 iterations was 268 seconds.

Figure 6 shows the iteration values of the best result. At the end of the tests, our algorithm achieved the best result 21150 TL at 4% mutation as the best solution.

Table 5 gives the chromosome of the best solution obtained from 250 trials.

Table 6 presents the orders delivered and picked up by the vehicles daily. Table 6 shows the capacity and route of the vehicle, as well as which suppliers are visited during departure and return. For example, a truck with route S2 is preferred on day 12.

The decision maker can interpret the solution shown in Table 6 in different ways. The decision maker will decide whether the delivery and pickup for the supplier orders will be carried out simultaneously. On Day 27, V341 truck will deliver to S1 and S2 suppliers and pick up from them. Due to the load weight-fuel consumption relationship, the decision maker may not want to transport the PO23 order from the supplier S1 to the supplier S2. The decision maker may decide to collect the PO23 order either when the truck comes to supplier S1 for the first time, or when the truck comes back to supplier S1.

RESULTS and DISCUSSION

The proposed hybrid GA was evaluated using real-world data from an automotive supply chain, focusing on simultaneous delivery and pickup operations with time window constraints. The results demonstrate that the algorithm outperforms existing methods in terms of both cost efficiency and computational performance.

The algorithm achieved a 20% reduction in transportation costs compared to the company's current practices, highlighting its effectiveness in optimizing vehicle routes.

Compared to traditional GA, the hybrid approach reduced computation time, enabling faster decision-making for logistics planning.

The harem-based crossover mechanism contributed to generating high-quality solutions by maintaining genetic diversity and avoiding premature convergence.

The algorithm's scalability was tested with varying problem sizes, ranging from 10 to 500 orders. Results indicate that the model maintains its efficiency across different scales, demonstrating robustness for larger logistics networks.

The application of the proposed algorithm in a real-world automotive supply chain demonstrated its capability to handle complex logistical challenges. By adhering to time window constraints and optimizing vehicle capacities, the algorithm ensured timely delivery and pickup while minimizing operational costs. This makes it a valuable tool for decision-makers aiming to enhance efficiency in competitive industries.

CONCLUSIONS

This study presented a novel hybrid GA to address the VRPSPD, incorporating time window and capacity constraints. The proposed method integrates a smart selection structure and a harem-based crossover mechanism to optimize vehicle routes, reduce transportation costs, and enhance computational efficiency. Computational results demonstrated that the algorithm not only meets real-world logistical requirements but also outperforms traditional methods in terms of cost savings and solution quality.

In this problem, each supplier's order is subject to specific earliest and latest deadlines, creating a time window constraint for both delivery and pickup operations. The proposed model ensures that all orders are fulfilled within their respective date ranges, adhering to operational requirements. This feature makes the model particularly suitable for real-world applications where precise scheduling is critical.

The decision-maker can utilize the model as a flexible decision support system. Depending on the

Table 6: Summary of Results for Real-World Test Cases

Day	Vehicle	Delivery			Pickup		
6	V25 (12000)	DO1 (S3)			PO1 (S3)		
	(3)	5760			3750		
10	V86 (12000)	DO5 (S1)	DO2 (S2)	DO4 (S2)	PO5 (S1)	PO2 (S2)	PO4 (S2)
	(1-2)	200	5950	350	1800	1400	350
11	V103 (12000)	DO3 (S3)	DO7 (S3)	DO8 (S2)	PO3 (S3)	PO7 (S3)	
	(2-3)	4160	2720	3150	6750	750	
12	V114 (12000)	DO6 (S2)			PO6 (S2)	PO8 (S2)	
	(2)	6650			5950	3850	
13	V130 (12000)	DO9 (S3)			PO9 (S3)		
	(3)	9920			8250		
17	V189 (12000)	DO11 (S3)			PO11 (S3)		
	(2)	4000			7750		
	V190 (12000)	DO10 (S2)			PO10 (S2)	PO12 (S2)	
	(3)	7700			1400	5950	
19	V218 (12000)	DO13 (S1)	DO16 (S1)		PO13 (S1)		
	(1)	150	200		1500		
	V219 (12000)	DO12 (S2)	DO14 (S2)		PO14 (S2)		
	(2)	4200	4900		7000		
20	V234 (12000)	DO15 (S2)	DO17 (S2)		PO15 (S2)	PO17 (S2)	
	(2)	350	350		1050	2100	
	V237 (12000)	DO18 (S3)			PO16 (S1)	PO18 (S3)	
	(1-3)	8480			5400	6250	
24	V286 (25000)	D022 (S1)			PO22 (S1)		
	(1)	150			24000		
25	V310 (12000)	DO20 (S3)			PO20 (S3)		
	(3)	4640			8000		
26	V324 (12000)	DO19 (S2)	DO21 (S2)		PO19 (S2)	PO21 (S2)	
	(2)	7700	4200		3500	2100	
27	V341 (12000)	DO23 (S1)	DO24 (S2)	DO25 (S2)	PO23 (S1)	PO24 (S2)	PO25 (S2)
	(1-2)	100	5250	700	1200	2100	350

operational context, they can decide whether delivery and pickup operations for a supplier should be carried out simultaneously or separately. This flexibility allows the problem's structure to dynamically adjust between the VRPSDP or the Divisible Delivery and Pickup Problem (VRPDDP). Such adaptability makes the model applicable to various logistical scenarios, enhancing its practical value.

Since this is a new problem formulation and no established benchmarks exist, the results of the proposed metaheuristic model were compared with the firm's current practices. The computational experiments demonstrated that the algorithm achieves superior performance, yielding better cost savings and operational efficiency than existing methods used by the firm.

Key findings include:

- The algorithm successfully handled complex constraints, including time windows, capacity limits, and simultaneous delivery and pickup.
- The use of the smart selection method significantly narrowed the solution space, enabling faster and more efficient optimization.
- The harem-based crossover method provided a unique genetic diversity mechanism, leading to high-quality solutions.
- The model's flexibility allows it to be tailored to varying operational conditions, providing decision-makers with actionable insights.

Implications for Future Work:

While the proposed algorithm demonstrated strong performance, there are several avenues for future research:

- Dynamic Adjustments: Integrating real-time data, such as traffic updates and dynamic customer demands, into the algorithm could enable adaptive and responsive logistics solutions.
- Fuel Efficiency and Sustainability: Incorporating environmental factors, such as fuel consumption and carbon emissions, into the optimization process could align the model with green logistics objectives.
- Parameter Optimization: Exploring advanced techniques for tuning GA parameters, such as mutation rates and population sizes, could further enhance solution quality.
- Scalability: Investigating the algorithm's scalability for larger problem instances, including more suppliers, vehicles, and orders, could broaden its practical applicability.
- Multi-Objective Optimization: Extending the model to address multiple objectives, such as minimizing costs and maximizing customer satisfaction, could increase its versatility in complex logistics environments.

The findings from this study contribute to the growing body of literature on advanced optimization techniques for VRP. The proposed algorithm provides a robust framework for businesses aiming to optimize their distribution networks in competitive and resource-constrained environments. Future work should investigate how innovative technologies like AI and IoT can be utilized to optimize the algorithm's performance and broaden its use in diverse logistical contexts.

REFERENCES

Anily, S. (1996). The vehicle-routing problem with delivery and back-haul options. *Naval Research Logistics (NRL), 43*(3), 415-434.

Ashouri, M., & Yousefikhoshbakht, M. (2017). A Combination of Meta-heuristic and Heuristic Algorithms for the VRP, OVRP and VRP with Simultaneous Pickup and Delivery. BRAIN. *Broad Research in Artificial Intelligence and Neuroscience, 8*(2), 81-95.

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research, 30*(5), 787-800.

Brown, R. D., Jones, G., Willett, P., & Glen, R. C. (1994). Matching two-dimensional chemical graphs using genetic algorithms. *Journal of Chemical Information and Computer Sciences, 34*(1), 63-70.

Çiçekli U.G. (2012). *Çok Kriterli Dağıtım Rotası Probleminin Melez Bir Model İle Optimizasyonu: Genetik Algoritma Temelli Melez Model Uygulaması*, Lambert Academic Publishing, Mauritius.

Çiçekli UG, Kaymaz Y (2016) A genetic algorithm for the allocation of dangerous goods containers in a storage yard for freight villages and dry ports. *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 18:264-282

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

Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation science, 40*(2), 235-247.

Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum, 23*(1), 79-96.

Engelbrecht, A. P. (2007). *Computational intelligence: an introduction*. John Wiley & Sons.

Eshelman, L. J., & Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. *In Foundations of Genetic Algorithms* (Vol. 2, pp. 187-202). Elsevier.

Gajpal, Y., & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research, 36*(12), 3215-3223.

Goetschalckx, M., & Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls. *European Journal of Operational Research, 42*(1), 39-51.

Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems, 5*(2), 139-167.

Grefenstette, J. J. (1992, September). Genetic algorithms for changing environments. In *Ppsn* (Vol. 2, pp. 137-144).

Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial intelligence review, 12*(4), 265-319.

Hsiao, S. W., Chiu, F. Y., & Lu, S. H. (2010). Product-form design model based on genetic algorithms. International Journal of Industrial Ergonomics, 40(3), 237-246.

Lenstra, J. K., & Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*(2), 221-227.

Lim, A., & Chew, H. K. (1997). Joining of compacted cells using genetic algorithm. *Electronics Letters, 33*(23), 1944-1945.

Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research, 230*(3), 475-486.

Michalewicz, Z., Vignaux, G. A., & Hobbs, M. (1991). A nonstandard genetic algorithm for the nonlinear transportation problem. *ORSA Journal on computing, 3*(4), 307-316.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General, 23*(5), 377-386.

Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations research*, *63*, 511-623.

Poon, P. W., & Carter, J. N. (1995). Genetic algorithm crossover operators for ordering applications. *Computers & Operations Research, 22*(1), 135-147.

Wang, H. F., & Chen, Y. Y. (2013). A coevolutionary algorithm for the flexible delivery and pickup problem with time windows. *International Journal of Production Economics, 141*(1), 4-13.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing, 4*(2), 65-85.

Wright, A. H. (1991). Genetic algorithms for real parameter optimization. *In Foundations of Genetic Algorithms* (Vol. 1, pp. 205-218). Elsevier.

Yadav, S. L., & Sohal, A. (2017). Comparative study of different selection techniques in genetic algorithm. *International Journal of Engineering, Science and Mathematics*, *6*(3), 174-180.

Zhong, Y., & Cole, M. H. (2005). A vehicle routing problem with backhauls and time windows: a guided local search solution. *Transportation Research Part E: Logistics and Transportation Review, 41*(2), 131-144.