APPLICATION OF THE GRASSHOPPER OPTIMIZATION ALGORITHM FOR ROUTE OPTIMIZATION IN FOOD PICKUP-DELIVERY

Ananda Rizky Ardha Saputra[1)]
Helen Burhan[2)]
Yudi Satria[3)]
[1)] Public Administration Study Program, Faculty of Law, Social Science and Politic Science, Universitas Terbuka, Tangerang Selatan, Indonesia
[2,3)] Mathematics Study Program, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok, Indonesia
e-mail: ananda.rizky01@sci.ui.ac.id

ABSTRACT

Food pickup and delivery is one of the services provided by online transportation platforms, where couriers collect food orders from restaurants and deliver them to customers. As the demand for this service continues to increase, a larger number of couriers is required to meet customer requests. To address this challenge, a double-order scheme is introduced, allowing a single courier to simultaneously handle two orders destined for nearby customers. In this study, the double-order scheme is employed to optimize food delivery routes with the objectives of minimizing operational costs while maintaining food quality within the specified delivery time windows. This optimization problem is formulated as the Pickup and Delivery Problem with Time Windows (PDPTW). The Grasshopper Optimization Algorithm (GOA), a metaheuristic optimization method inspired by the foraging behavior of grasshoppers, is used to solve the problem. GOA simulates how grasshoppers search for food and communicate food locations to one another through a chemical signal known as the 4-VA pheromone. The proposed method was evaluated using simulation data consisting of 50 customer orders, 300 iterations, and a population of 10 grasshoppers. The results demonstrate that the proposed approach can reduce operational costs by up to 33.71% and decrease the number of active couriers by 50% compared with the conventional single-order delivery scheme.

Keywords: double-order scheme, food pickup-delivery service, grasshopper optimization, route optimization.

INTRODUCTION

In the digital era, advances in information and communication technologies have significantly transformed various aspects of human life. One notable change is the evolution of food delivery services, now facilitated by online transportation platforms. These services provide urban communities with the convenience of enjoying their favorite meals without leaving their homes or offices, offering a practical solution for busy people. Kusumawardhani (2022), citing research by Google et al. (2021), found that individuals with fixed incomes frequently utilize these services to enhance productivity, explore the latest culinary trends, and engage in social activities.

Food delivery services were already popular before the COVID-19 epidemic hit. People become more dependent on these platforms as a result of health regimen enforcement and limitations on outdoor activities. Demand was still strong even after the outbreak had passed. According to research by Google et al. (2024), the gross merchandise value of online transportation and food delivery services reached USD 9 billion in 2024, a 13% increase from the previous year.

With the convenience of digital payment methods, these services have become integral to contemporary life.

Optimizing operations is crucial to keep up with the increased demand while keeping customers happy. With the new multi-order service model, a driver may now make many deliveries to different locations in a single journey. Burhan et al. (2021) examined this approach and suggested merging two ride-hailing services to handle two customers' requests simultaneously. This method cuts down on operating expenses and delivery times, but it also makes drivers more responsible for keeping food fresh and meeting delivery deadlines.

A variation of the Vehicle Routing Problem (VRP), the Pickup and Delivery Problem with Time Windows (PDPTW) models the problems of the multi-order delivery system. Toth & Vigo (2002) state that VRP is an NP-Hard problem, meaning that the solution complexity grows exponentially with the size of the issue. The use of heuristic and metaheuristic procedures is hence justified as accurate methods are often unworkable.

The Grasshopper Optimization Algorithm (GOA), Ant Colony Optimization (ACO), Firefly Algorithm (FA), and Particle Swarm Optimization (PSO) are among the metaheuristic approaches used to tackle VRP. As an offshoot of the Traveling Salesman Problem (TSP), VRP research started with Dantzig & Ramser (1959). The VRP with Simultaneous Pickup and Delivery was solved by Montané & Galvão (2006) using Tabu Search, and other variations of VRPPD were found by Parragh et al. (2008). To solve the VRPPD with Time Windows and incorporate customer satisfaction, Teng et al. (2021) used a Genetic Algorithm. A further topic covered by Burhan et al. (2021) was the use of platform-sharing programs by logistics companies.

PDPTW is only one of several VRP issues for which GOA has shown promise in tackling using metaheuristics. Through social interaction, attraction to the optimal solution, and adaptive parameters, GOA continually updates the candidate solutions, which are modelled as grasshopper postures. These locations represent the order of routes that vehicles may take within the VRP framework. Minimizing travel time or distance while fulfilling restrictions like vehicle capacity and time limits is the goal of the updating process. In light of this, GOA provides a practical means of meeting the difficulties of multi-order delivery services.

Previous studies by Boro & Kumar Behera (2021), Soto-Mendoza et al. (2020), and Naomi et al. (2022) have used GOA to VRP. The goal of this research is to find out how GOA can help with the time window issue of food pickup and delivery. The goal of this study is to increase the usefulness of GOA in addressing issues linked to VRP and to make food delivery services more operationally efficient.

METHOD

This study models the food pickup delivery problems as (PDPTW). The problem is based on a simulation dataset of fifty orders, each consisting of a merchant–customer pair, served within a specific time period. The data were obtained from Google Maps and are shown on the map in Figure 1. Since the data points represent real-world locations, the coordinates used are not Cartesian but geographic, involving latitude and longitude.

Given fifty customer orders, the problem consists of 101 nodes, including one depot, fifty merchant (pickup) nodes, and fifty customer (delivery) nodes. Two service schemes are evaluated in this study. The single-order scheme assigns one pickup–delivery pair to each courier in a single trip; after completing the delivery, the courier proceeds to the next assigned order. The double-order scheme allows one courier to serve two pickup–delivery pairs within a single route while satisfying vehicle capacity, pickup–delivery precedence, and time-window constraints.

Both routing schemes are optimized using the Grasshopper Optimization Algorithm (GOA). The objective is to minimize the total operational cost. After optimization, the operational costs and the number of operating vehicles obtained from both schemes are compared to determine the more efficient routing strategy.

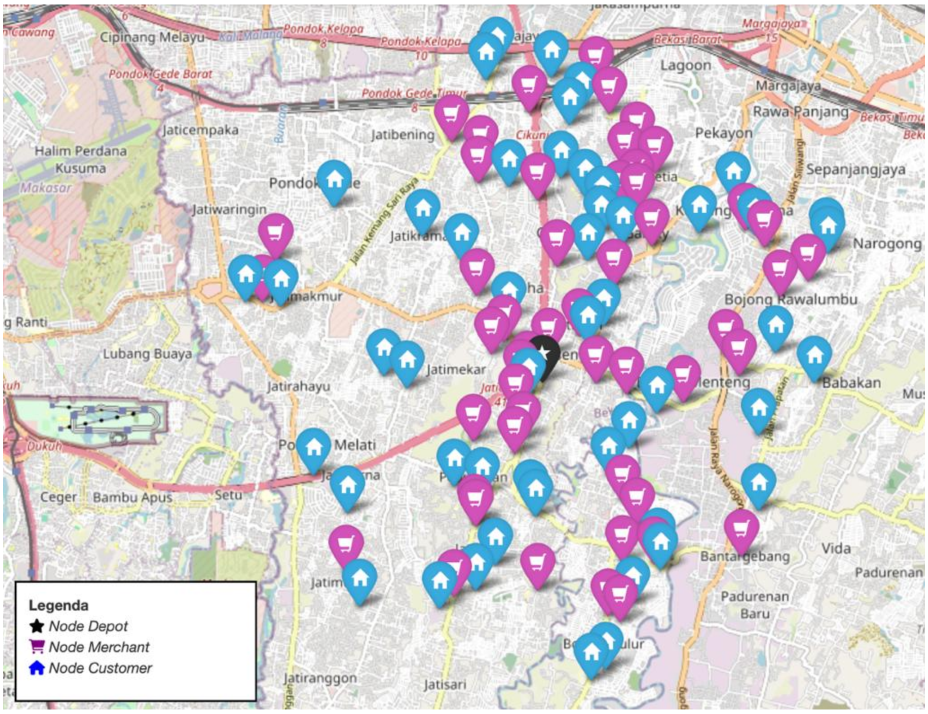

Figure 1. Distribution of Research Data Points on the Map (Lovani, 2024)

In measuring distance between two points, this study applies the Haversine distance formula as shown in Equation (1). By taking the curvature of the Earth into consideration, the great-circle distance between any two places on its surface may be determined using the Haversine formula. Originally introduced by James Andrew in 1805, this formula is considered reasonably accurate because it does not account for elevation or depth variations across the Earth's surface (Budiyanti, 2022). According to Nichat (2013), the Haversine distance is defined by the following equation:

$$d_h = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\Phi_2 - \Phi_1}{2}\right) + \cos(\Phi_1)\cos(\Phi_2)\sin^2\left(\frac{\Psi_2 - \Psi_1}{2}\right)}\right) \quad (1)$$

as:

R : Earth's radius (6,371 km)
Φ_n : Latitude of point n, with $n = \{1,2\}$ (in radians)
Ψ_n : Longitude of point n, with $n = \{1,2\}$ (in radians)

Pickup Delivery Problem with Time Windows (PDPTW)

The Vehicle Routing Problem (VRP) is a combinatorial optimization challenge that involves planning the most efficient routes for a group of vehicles to go from a central depot to many consumers located in different parts of the world. According to González-Feliu (2008), the goal of solving the VRP is to meet all client expectations while respecting operational limits and minimizing transportation costs. The primary parts of VRP, according to Toth & Vigo (2002), are the depot (where routes begin and end), the fleet of vehicles (which have capacities, deadlines, and operational costs), the customers (who have specific needs and deadlines), and the routes that vehicles take (which are parts of a larger transport network). The VRP was first developed by

Dantzig & Ramser (1959) and is an extension of the famous Traveling Salesman Problem (TSP) that allows multiple vehicles to serve different customers.

In the most fundamental form of vehicle response pricing (VRP), known as Capacitated VRP (CVRP), there is complete uniformity in vehicle capacity and predictable consumer demand. In CVRP, the goal is to find routes that use the fewest feasible cars while minimizing overall trip time or distance (Meliawati, et al., 2025). Another important variety is Time-constrained VRP (T-VRP), which focuses on temporal limitations, and Distance-constrained VRP (D-VRP), which focuses on geographical constraints. Complex versions integrate many characteristics. For example, the Pickup and Delivery Problem with Time Windows (PDPTW) results from combining service time windows (VRPTW) with pickup and delivery (VRPPD).

Customer requests are paired with pickup and delivery times in PDPTW, and services must be fulfilled within specific time frames. Every client request in this model has a pickup and a delivery node. The model accounts for vehicle capacity limits and states that pickups must always occur before deliveries. Toth & Vigo (2002) state that the goal is to reduce overall operating cost while meeting capacity and schedule limitations, ensuring that each vehicle route begins and ends at the depot and visits each node only once.

Each client has a unique window of opportunity to get the service under the VRPTW model, denoted as $[a_i, b_i]$. Departure schedules, travel times, service periods at each stop, and total load must all be considered while planning vehicle routes. The optimization of routes becomes considerably more complex due to these extra limitations. A number of research projects in Indonesia have investigated potential uses for PDPTW. For instance, according to research by Garside & Cahyanti (2018), 3 kilogram gas cylinders sold in retail stores might have transportation costs cut by as much as 13.72 percent when compared to more traditional approaches. In their PDPTW concept, Kohar & Jakhar (2025) considered the possibility of making multiple deliveries and pickups within a single journey. In PDPTW, pickup sites must be visited before delivery locations, which is a significant difficulty.

All things considered, PDPTW is a highly applicable model for contemporary logistics systems that need to strike a balance between efficiency and customer satisfaction. A rigorous yet practical approach is provided to increase service efficacy while lowering delivery costs by concurrently addressing windows, vehicle capacity limits, and routing (González-Feliu, 2008).

Teng et al. (2021) state that a transportation network modelled by a graph $G = (N, A)$ may be used to represent the Pickup and Delivery Problem with Time Windows (PDPTW). In formal terms, a set of non-empty vertices (V) and a set of edges (E) constitute a graph $G = (V, E)$, as stated by Rosen (2012). One or two vertices, called endpoints, are connected to every edge. A connection or link between the vertices involved is represented by an edge, which is said to connect its ends. Nodes for collection and delivery, routes, vehicles, and time restrictions are all essential parts of the PDPTW architecture. If we want to capture the complexities of routing challenges in actual food delivery situations, we need this model. The model's parameters are defined by their respective functions and restrictions with respect to operating expenses, service windows, vehicle capacity, and route optimization.

Here is a table that lists the main notations used in the PDPTW model:

Table 1. PDPTW Parameter Notations

Parameter	Definition
U	Set of all merchant nodes or pickup nodes.
V	Set of all customer nodes or delivery nodes
N	Set of all nodes, including depot, merchant, and customer nodes
$A = \{(i,j) \mid i \in N, j \in N\}$	Set of all directed edges or arcs in the network.
R	Set of all orders or requests
Ω	Set of all available vehicles or fleet
$Q = 2$	Maximum number of orders that a single vehicle can serve
L	Maximum travel distance of a vehicle
D	Number of orders that have not yet been delivered
v	Speed of the delivery vehicle
$[e_{i^-}, l_{i^-}]$	Time window for delivering order to customer i^-, where L_{i^-} is the latest delivery time, and $L_{i^-} > l_{i^-}$
$[e_{i^+}, l_{i^+}]$	Time window for picking up order from merchant i^+, where L_{i^+} is the latest pickup time, and $L_{i^+} > l_{i^+}$
t_i	Service duration required at node i
η_1	Coefficient of start-up cost for using a vehicle
η_2	Coefficient of riding cost
η_3	Coefficient of riding or travel cost
d_{ij}	Distance between node i and node j
T_{ij}	Travel time between node i and node j
τ_{A_i}	Arrival time at node i
τ_{L_i}	Departure time from node i

Grasshopper Optimization Algorithm (GOA)

The Grasshopper Optimization Algorithm (GOA), introduced by Saremi et al. (2017), is a population-based metaheuristic optimization method inspired by the swarming behavior of grasshopper colonies. Meraihi et al. (2021) emphasized that the core inspiration behind GOA originates from the collective foraging behavior exhibited during the gregarious phase of grasshoppers, during which individuals are influenced by population density, pheromone signals, environmental stimuli, and social interaction. Yang et al. (2023) highlighted the role of a specific pheromone, 4-vinylanisole (4VA), which is crucial in attracting other grasshoppers to aggregate and forage cooperatively. This behavior reflects a coordinated mechanism for navigating toward optimal food sources, which is abstracted in GOA as a search for the global optimum in a solution space.

To solve the PDPTW, GOA finds the best vehicle routes by emulating the swarm intelligence of grasshoppers. There are essentially two stages to how the algorithm works: exploration and exploitation. In order to find a variety of possible solutions, grasshoppers randomly traverse larger areas during exploration. As they enter the exploitation phase, their actions hone down on the most promising answers they've encountered. By mathematically modeling the swarming behavior of grasshoppers, GOA explores the solution space to identify vehicle routes that minimize the total operational cost while satisfying the constraints of the Pickup and Delivery Problem with Time Windows (PDPTW). In this study, each grasshopper represents a candidate routing solution encoded as a pickup–delivery sequence. During each iteration, the candidate solution is decoded into vehicle routes and evaluated using the objective function. Feasibility is verified by checking vehicle capacity, pickup–delivery precedence, and customer time-window constraints. Candidate solutions that violate any of these constraints are repaired by restoring the required pickup–delivery order and adjusting the route until all constraints are satisfied before the objective value is computed. Consequently, GOA searches among feasible routing solutions and iteratively converges toward lower-cost routes.

Using formulations suggested by Saremi et al. (2017) and Topaz et al. (2008), which describe how each grasshopper's (solution candidate's) position is updated based on interactions with others and attraction toward the best global solution, Naomi et al. (2022) mathematically modeled this swarming behavior. After a grasshopper i moves, its location is described by Equation (2), which takes into account attractive forces, social interaction zones, and adaptive coefficients to provide balanced search dynamics.

$$X_i = c\left(\sum_{j=1}^{B} c\frac{ub_d - lb_d}{2}a(r)\hat{r}\right) + \widehat{T_d} \tag{2}$$

Upper and lower boundaries of the grasshopper's location are denoted by ub_d and lb_d, respectively, while the decreasing coefficient of the interaction zone is represented by the variable c. So far, the optimal solution has been denoted by the phrase ($\widehat{T_d}$). To find how far apart two grasshoppers are, one uses the formula$r = \left|x_j - x_i\right| = \sqrt{\sum_{k=1}^{B}\left(x_{ik} - x_{jk}\right)^2}$, B as total number grasshoppers, also $\hat{r}$ unit vector between them.

The interaction strength is defined by the function $a(r) = f \cdot \exp\left(-\frac{r}{L}\right) - \exp(-r)$, f is intensity interaction also L is interaction range. This function determines the social behavior between grasshoppers based on the value of $a(r)$. If $a(r)$ is positive, the grasshoppers are attracted to one another; if negative, they repel each other. When $a(r) = 0$, the grasshoppers are in a neutral or comfort zone, neither moving closer nor farther apart.

The coefficient c is computed using Equation (3), where p as current iteration index also p^* as maximal number iterations.

$$c = c_{max} - \frac{p}{p^*}(c_{max} - c_{min}) \tag{3}$$

In solving the PDPTW using the GOA, each grasshopper is analogized as a potential solution representing a delivery route. Every grasshopper has coordinates in a multi-dimensional space. At context of (VRP), this space contains information about sequence nodes visited a vehicle. For example, if a grasshopper has coordinates (7, 10, 9, 15), this indicates that the route follows node 1, then node 3, followed by node 2, and finally node 4. Each dimension in the grasshopper's position corresponds to a specific node in the route. Thus, a grasshopper's position illustrates the order in which a vehicle should perform pickup and delivery tasks.

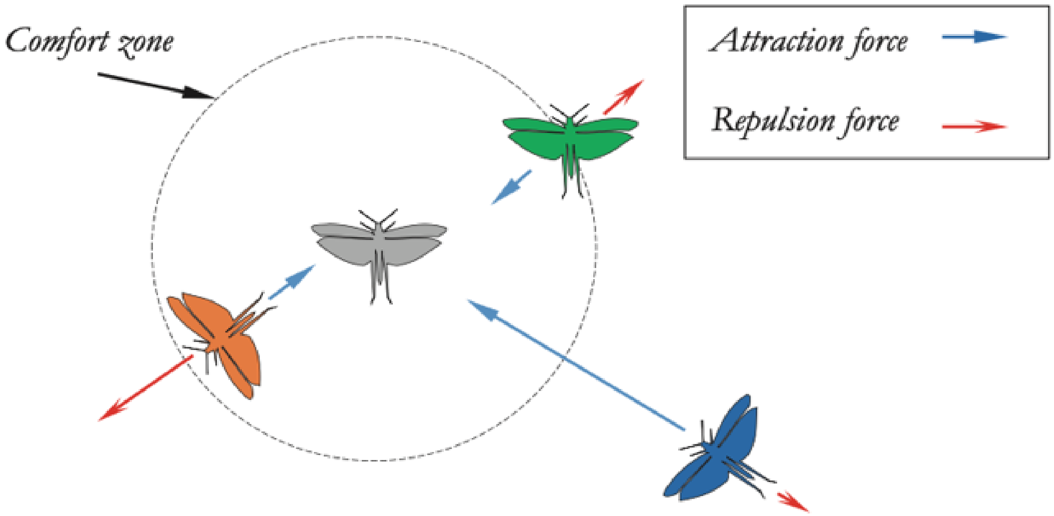

Figure 2. Illustration of Grasshopper Interaction Zones (Saremi et al., 2017)

RESULTS AND DISCUSSION

The mathematical model for the PDPTW, aimed at minimizing the travel cost of couriers from the depot to a set of merchant and customer nodes and back to the depot, was formulated by Teng et al. (2021) as follows:

a. Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle k travels from node i to node j} \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{rk} = \begin{cases} 1 & \text{if vehicle k serves request r} \\ 0 & \text{otherwise.} \end{cases}$$

b. Cost variables

i. Start-up cost

This includes courier wages, vehicle maintenance, and depreciation costs. It is defined by Equation (4).

$$F_1 = \eta_1 \sum_{k \in \Omega} \sum_{j \in N} x_{0jk} \tag{4}$$

ii. Riding cost

This represents fuel and refueling costs, expressed in Equation (5).

$$F_2 = \eta_2 \sum_{i \in N} \sum_{j \in N} \sum_{k \in \Omega} x_{ijk} d_{ij} \tag{5}$$

iii. Penalty cost

This reflects customer dissatisfaction related to service time, calculated as in Equation (6).

$$F_3 = \eta_3 \left(\sum_{i \in N} \max(e_i - \tau_{A_i}, 0) + \sum_{i \in N} \max(\tau_{L_i} - l_i, 0) \right) \tag{6}$$

c. Mathematical model

i. Objective function

Goal is for minimizing total delivery cost as defined at Equation (7), where $\rho_1 + \rho_2 + \rho_3 = 1$ represent the weight (importance) of each cost component.

$$\min(\rho_1 F_1 + \rho_2 F_2 + \rho_3 F_3) \tag{7}$$

ii. Constrains

$$\sum_{k \in \Omega} z_{rk} = 1, \forall r \in R \tag{8}$$

$$\sum_{j \in N} \sum_{k \in \Omega} x_{ijk} = 1, \forall i \in N - \{0\} \tag{9}$$

$$\sum_{j \in N} x_{0jk} = 1, \forall k \in \Omega \tag{10}$$

$$\sum_{i \in N} x_{i0k} = 1, \forall k \in \Omega \tag{11}$$

$$\sum_{i \in N} x_{ihk} - \sum_{i \in N} x_{hjk} = 0, \forall k \in \Omega, \forall h \in N - \{0\} \tag{12}$$

$$\sum_{i \in N} \sum_{k \in \Omega} x_{0jk} \leq \Omega \tag{13}$$

$$\sum_{r \in R} z_{rk} \leq Q = 2, \forall k \in \Omega \tag{14}$$

$$\sum_{i \in N^+} \sum_{j \in N} x_{ijk} - \sum_{i \in N} \sum_{j \in N^-} x_{ijk} \leq D = 1, \forall k \in \Omega \tag{15}$$

$$\tau_{A_i} + t_i = \tau_{L_i}, \forall i \in N - \{0\}, \forall k \in \Omega \tag{16}$$

$$\tau_{L_i} + T_{ij} = \tau_{A_j}, \text{ketika } x_{ijk} = 1, \forall i \in N, \forall j \in N - \{0\}, \forall k \in \Omega \tag{17}$$

$$\tau_{A_i} \leq \tau_{A_j}, \forall i \in U, \forall j \in V \tag{18}$$

$$\tau_{L_0} = 0 \tag{19}$$

It is implied by constraint (8) that a single courier is assigned to each delivery order. The fleet is only allowed to visit each client node once, as stated in constraint (9).

According to constraint (10) every vehicle is required to leave the depot. At the conclusion of their excursion, all vehicles are required by constraint (11) to return to the depot. Because of constraint (12), couriers cannot make intermediate stops at client nodes. That there aren't more cars leaving the depot than there are available ones is what constraint (13) is trying to prevent. There is a limit to how many orders a courier may carry, as stated in constraint 14. A courier may have at most 15 orders that have not been delivered once they have picked them up, as shown in constraint (15).

The earliest time a courier may complete a pickup or delivery service upon reaching a client node is represented by constraint (16). Time constraints and priority requirements between the pickup (merchant) and delivery (customer) nodes are enforced by constraints (17) and (18). Finally, the departure time from the depot is set to zero by constraint (19).

In order to solve the PDPTW using the Grasshopper Optimization Algorithm (GOA), the total number of orders to be serviced and the distance data between nodes are entered first. In this study, each dimension of a grasshopper represents one decision variable corresponding to the visiting sequence of customer requests. The continuous position generated by GOA is subsequently transformed into a valid permutation representing the order in which pickup and delivery nodes are visited. This permutation is then decoded into a feasible vehicle route that satisfies the PDPTW constraints. Following this, the algorithm parameters were initialized using the parameter settings reported by Naomi et al. (2022), which were adopted because they had been validated for a similar pickup-and-delivery optimization problem. These parameters include the dimensionality, the number of grasshoppers (B), the maximum number of iterations (p^*), the adaptive coefficients c_{max} and c_{min}, and the upper and lower search bounds (ub_d and lb_d, respectively), as shown in Table 2. Furthermore, the time window parameters, including the service duration at each node, which is specified $t_i = \begin{cases} 5 \text{ minutes}, & i \in U \\ 3 \text{ minutes}, & i \in V \end{cases}$ and the delivery time windows, defined as

$$[e_i, l_i] = \begin{cases} [10.00, 11.20], & i \in U \\ [10.01, 11.30], & i \in V \\ [10.00, 11.40], & i = 0 \end{cases}$$

Table 2. Parameters Used in the Optimization Process (Naomi et al., 2022)

Parameter	Value	Parameter	Value
D	50	Q	2
B	10	v	30km/hr
p^*	300	ρ_1	0.24
c_{max}	0.00001	ρ_2	0.12
c_{min}	0.3	ρ_3	0.64
$[ub_d, lb_d]$	$[0,1]$	η_1	IDR10,000
f	0.5	η_2	IDR2.5/km
L	1.5	η_3	IDR1,000

Afterward, the initial positions of the grasshoppers are randomly generated within the range $[lb_d, ub_d]$ for each individual in the population, repeating this process for all B grasshoppers. These initial positions serve as the starting solutions. Each grasshopper's position is then mapped into a delivery route, where dimension n corresponds to merchant m_n. The route is determined by sorting the values of each dimension in ascending order.

Once the initial route is constructed, it is segmented according to the maximum number of orders each courier can handle. Since each courier may serve at most two orders, the generated sequence is split into sub routes of two merchants from left to right. Then, the corresponding customer nodes are inserted to complete each pickup-delivery pair.

After adding the customer nodes to the route configuration, the following step is to evaluate the goal function using this information. To begin, we'll be focusing on the grasshopper whose objective function value is the lowest. The interaction intensity among grasshoppers is controlled by updating the attraction coefficient c to match the current iteration. The influence force is determined by calculating the pairwise distance r between grasshoppers and using the interaction function a(r), which is defined by the parameters f and L.

Each grasshopper uses this contact force to adjust its position according to a specified formula. The new location is rolled back to its old one if it goes over the allowed range $[lb_d, ub_d]$. After that, if any grasshopper manages to achieve more objective value than the previous goal, a new target is selected. The present objective will not alter if no progress is made.

Once the maximum number of iterations p^* has been reached or the solution has converged, the optimization process stops. In the PDPTW case, GOA finds the best option by iteratively minimizing the overall trip cost.

The study data was processed in the Google Collaboratory environment using the Python programming language, version 3.10.12. Experiments were conducted to obtain the best-known routing solution. Each run produced different routes and total operational costs, although the number of operating drivers remained consistent. The fifty simulated orders were generated from real merchant and customer locations obtained from Google Maps. The dataset includes geographically dispersed pickup and delivery points with heterogeneous travel distances and predefined delivery time windows. Each order consists of one merchant–customer pair, while the double-order scheme permits combining two compatible orders into a single vehicle route, subject to vehicle capacity and temporal feasibility. **Figure 3** presents a comparative summary of the solution results for the fifty incoming delivery requests between the single-order and double-order service schemes.

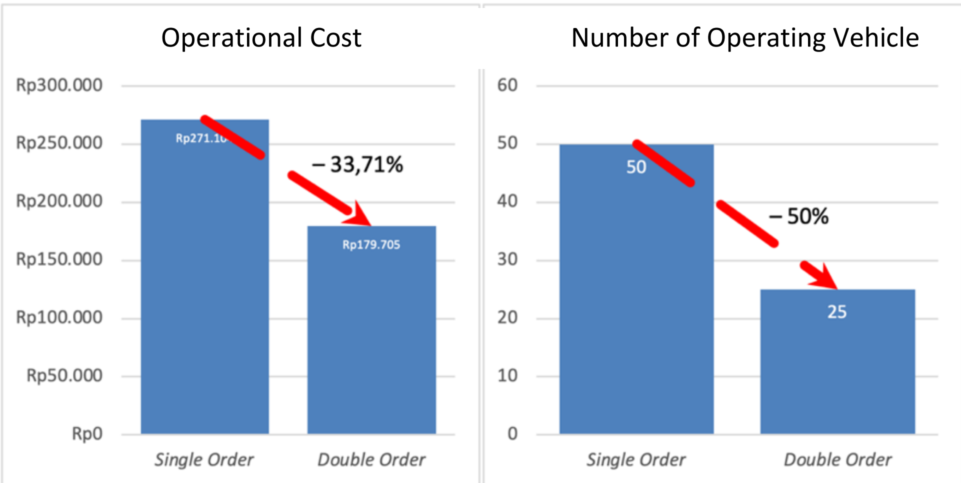

Figure 3. Comparison of Results between Single-Order and Double-Order Service Schemes

The double-order service scheme successfully reduced the number of operating vehicles by 25 units and lowered the total operational cost by IDR91,399 compared to the single-order scheme. Based on these outcomes, could be revealed implementation of GOA in solving the PDPTW for this dataset is effective in reducing fleet usage by 50% and operational costs by 33.71%.

CONCLUSION

Based on the problem formulation, research objectives, and the results obtained, it can be concluded that the mathematical modeling process of the food delivery route optimization problem has been successfully constructed and explained through a simple illustrative case. Furthermore, the application of the GOA to optimize food delivery routes using a double-order scheme was found to outperform the one-order method, both in the illustrative scenario and in the research dataset. When tested on a dataset of 50 delivery orders, the double-order scheme significantly improved operational efficiency, reducing total delivery costs by 33.71% and cutting the number of active couriers by up to 50%. These findings indicate that GOA is a promising method for enhancing the efficiency of food delivery services, particularly under multi-order conditions.(2025)

REFERENCE

Montané, F. A. T., & Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, *33*(3), 595–619. https://doi.org/10.1016/j.cor.2004.07.009

Boro, S., & Kumar Behera, S. (2021). An approach for vehicle routing problem using grasshopper optimization algorithm and simulated annealing. *International Journal of Advanced Research*, *9*(03), 59–64. https://doi.org/10.21474/IJAR01/12554

Budiyanti, E. (2022). Implementasi metode haversine formula pada sistem informasi geografis rumah kost daerah Tangerang Selatan. *LIAISON JOURNAL OF BEST*, *1*(1).

Burhan, H., Soehodho, S., & Nahry, N. (2021). Model development of ride splitting service with resource sharing scheme on ride sourcing (online taxi) services in Jakarta. *IIUM Engineering Journal*, *22*(1), 175–190. https://doi.org/10.31436/iiumej.v22i1.1520

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

Garside, A. K., & Cahyanti, D. N. (2018). Penyelesaian Vehicle Routing Problem with Simultaneous Pick Up and Delivery dengan Algoritma Tabu Search. *Jurnal Ilmiah Teknik Industri*, *17*(2), 125. https://doi.org/10.23917/jiti.v17i2.6703

González-Feliu, J. (2008). *Models and methods for the city logistics: The two-echelon capacitated vehicle routing problem* [Doctoral Thesis]. Politecnico di Torino.

Google, Temasek, & Bain & Company. (2021). *E-Conomy SEA 2021—Roaring 20s: the SEA Digital Decade*.

Google, Temasek, & Bain & Company. (2024). *e-Conomy SEA 2024 Profits On The Rise, Harnessing SEA's Advantage*.

Kabul, P. M., Fitriani, R., Arifin, J., & Damayanti, A. (2025). Penentuan Rute Terpendek Pendistribusian Galon Menggunakan Metode VRP (Vehicle Routing Problem) PT. ABC. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *9*(3), 282–290. https://doi.org/10.30998/string.v9i3.25605

Kohar, A., & Jakhar, S. K. (2025). A capacitated multi pickup online food delivery problem with time windows: a branch-and-cut algorithm. *Annals of Operations Research*, *354*(2), 735–756. https://doi.org/10.1007/s10479-021-04145-6

Kusumawardhani, S. (2022). *Survei persepsi & perilaku konsumsi online food delivery (OFD) di Indonesia*.

Lovani, S. M. (2024). *Optimasi rute pesan-antar makanan yang mempertimbangkan kepuasan pelanggan menggunakan metode ant colony optimization (ACO)* [Skripsi]. Universitas Indonesia.

Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper Optimization Algorithm: Theory, Variants, and Applications. *IEEE Access*, *9*, 50001–50024.

https://doi.org/10.1109/ACCESS.2021.3067597

Naomi, A., Pratiwi, A. B., & Suprajitno, H. (2022). Grasshopper Optimizaton Algorithm (GOA) untuk Menyelesaikan Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). *Tensor: Pure and Applied Mathematics Journal*, *3*(2), 73–84. https://doi.org/10.30598/tensorvol3iss2pp73-84

Nichat, M. K. (2013). Landmark based shortest path detection by using A* Algorithm and Haversine Formula. *International Journal of Innovative Research in Computer and Communication Engineering*, *1*(2).

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, *58*(1), 21–51. https://doi.org/10.1007/s11301-008-0033-7

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Soto-Mendoza, V., García-Calvillo, I., Ruiz-y-Ruiz, E., & Pérez-Terrazas, J. (2020). A Hybrid Grasshopper Optimization Algorithm Applied to the Open Vehicle Routing Problem. *Algorithms*, *13*(4), 96. https://doi.org/10.3390/a13040096

Teng, R., Hong-bo, X., Kang-ning, J., Tian-yu, L., Ling, W., & Li-ning, X. (2021). Optimisation of takeaway delivery routes considering the mutual satisfactions of merchants and customers. *Computers & Industrial Engineering*, *162*, 107728. https://doi.org/10.1016/j.cie.2021.107728

Topaz, C. M., Bernoff, A. J., Logan, S., & Toolson, W. (2008). A model for rolling swarms of locusts. *The European Physical Journal Special Topics*, *157*(1), 93–109. https://doi.org/10.1140/epjst/e2008-00633-y

Toth, P., & Vigo, D. (2002). *The Vehicle Routing Problem* (P. Toth & D. Vigo, Eds.). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718515

Yang, J., Yu, Q., Yu, J., Kang, L., & Guo, X. (2023). 4-Vinylanisole promotes conspecific interaction and acquisition of gregarious behavior in the migratory locust. *Proceedings of the National Academy of Sciences*, *120*(37). https://doi.org/10.1073/pnas.2306659120