



Journal Homepage: [-www.journalijar.com](http://www.journalijar.com)

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/24393
DOI URL: <http://dx.doi.org/10.21474/IJAR01/24393>



RESEARCH ARTICLE

A TWO-PHASE DECOMPOSITION HEURISTIC VERSUS A HYBRID GENETIC ALGORITHM FOR SOLVING A MULTI-PRODUCT, MULTI-COMPARTMENT VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS PICKUP AND DELIVERY AND TIME WINDOWS FOR THE TRANSPORT OF PERISHABLE PRODUCTS

Gueu Ounse Eric-Noel¹, Kaye Bi Kouai Bertin², Coulibaly Kpinna Tiekoura³ and Oumtanaga Souleymane¹

1. LARIT, Institut National Polytechnique Félix Houphouët-Boigny (INP-HB), BP 1093 Yamoussoukro, Côte d'Ivoire.
2. LAMI, Université Félix Houphouët-Boigny (UFHB), 01 BP V34 Abidjan 01, Côte d'Ivoire.
3. LASTIC, École Supérieure Africaine des TIC (ESATIC), 18 BP 1501 Abidjan 18, Côte d'Ivoire.

Manuscript Info

Manuscript History

Received: 12 August 2026

Final Accepted: 14 September 2026

Published: October 2026

Key words:-

vehicle routing problems; perishable products; multi-compartment vehicles; simultaneous pickup and delivery; time windows; genetic algorithm; Solomon insertion heuristic; two-phase decomposition heuristic.

Abstract

We compare two approximate methods for solving the MPMC-VRPSPDTW, a vehicle routing problem for the transport of perishable products. The problem combines the following features: multiple products, multiple compartments, simultaneous pickup and delivery, time windows and a product degradation that depends on the road category. A single objective function minimizes the transport cost and the opportunity cost due to the loss of economic value of the products. The first solution method is a two-phase decomposition heuristic (TPDH) that works as follows: nearest-neighbour clustering of customers, then inter-route local search. The second is a hybrid genetic algorithm (GA) whose initial population is built from a feasible solution provided by an insertion heuristic inspired by Solomon's II. On 45 configurations derived from Solomon's instances C101, R104 and RC107, the hybrid GA obtains a lower cost in 34 configurations and reduces the mean cost by 5.1% and the mean number of vehicles by 25.8%. Its mean run time is 1.5 times that of the TPDH (497 s vs 339 s). On the 23 configurations solved to optimality by B&C CPLEX, its mean optimality gap is 4.4%, against 12.6% for the TPDH. Paired Wilcoxon tests confirm these differences ($p \leq 1.1 \times 10^{-4}$).

"© 2026 by the Author(s). Published by IJAR under CC BY 4.0. Unrestricted use allowed with credit to the author."

Introduction:-

The distribution of fresh products must balance transport cost, the quality of the delivered products and fleet size. Road condition is a major cause of mechanical damage to fruits and vegetables during transport, particularly in countries where the road network is heterogeneous (Chonhenchob et al., 2009; Fernando et al., 2018; Idah et al., 2007; Jarimopas et al., 2005). A shorter route may therefore cost more, in lost value, than a longer route on better roads. (Buelvas Padilla et al., 2018) integrated this effect of road roughness into a routing problem with a single product and single-compartment vehicles. We extended this framework to the MPMC-VRPSPDTW, a multi-

Corresponding Author:-Gueu Ounse Eric-Noel

Address:-LARIT, Institut National Polytechnique Félix Houphouët-Boigny (INP-HB), BP 1093 Yamoussoukro, Côte d'Ivoire.

product, multi-compartment vehicle routing problem with simultaneous pickup and delivery and time windows (Gueu et al., 2025). In this problem, each customer receives several perishable products and returns empty packaging. The problem is formulated as a mixed-integer linear program (MILP) that minimizes the sum of the transport cost and a weighted opportunity cost. This problem generalizes the VRP with time windows and is therefore NP-hard. In our experiments, the Branch-and-Cut (B&C) of CPLEX reaches the 7,200 s limit in 22 of the 45 configurations. Approximate methods are therefore needed. Two families of methods were available to us: a constructive heuristic followed by local search, which is simple and fast, and a population-based metaheuristic, which is more expensive but more robust. The choice between the two depends on the trade-off between route quality and computation time, which we measure in this paper.

Our contributions are as follows:-

1. a two-phase decomposition heuristic (TPDH) adapted to the MPMC-VRPSPDTW, which clusters customers by feasible nearest neighbour and then improves the routes by inter-route exchanges;
2. a hybrid genetic algorithm (GA) whose initial population is built from a feasible solution provided by an insertion heuristic inspired by the I1 heuristic of (Solomon, 1987), and whose operators all preserve feasibility;
3. a paired comparison of the two methods on 45 configurations derived from Solomon's instances, in terms of objective cost, number of vehicles and computation time, with the B&C solutions as a reference and non-parametric statistical tests.

Section 2 positions our work with respect to the state of the art. Section 3 recalls the problem and its model. Section 4 describes the two methods. Section 5 presents the experimental protocol and Section 6 presents the results. Section 7 discusses the limitations of these results. Section 8 concludes the paper.

Related work:-

VRP variants:-

The multi-compartment VRP (MCVRP) aims to serve customers who request several incompatible products, transported in separate compartments (Derigs et al., 2011; Ostermeier et al., 2021). The VRP with simultaneous pickup and delivery (VRPSPD), introduced by (Min, 1989), was solved by tabu search (Alfredo Tang Montané and Galvão, 2006) and extended to time windows by (Wang and Chen, 2012). Its non-monotonic load along the route makes capacity constraints harder to satisfy (Koç et al., 2020). For perishable products, (Osvald and Stirn, 2008) model degradation as a function of travel time and (Amorim and Almada-Lobo, 2014) address a bi-objective case. (Buelvas Padilla et al., 2018) relate the mechanical degradation of a product to road roughness.

Two-phase decomposition heuristics:-

"Cluster-first, route-second" heuristics first assign customers to vehicles, then build or improve the routes (Fisher and Jaikumar, 1981). They are fast and easy to adapt and implement. Their quality, however, depends strongly on the assignment phase: a greedy clustering may open too many routes, which the subsequent local search only partly corrects. Classical constructive heuristics, such as the savings algorithm (Clarke and Wright, 1964) or the I1 insertion heuristic of (Solomon, 1987), are also used as starting points for more elaborate methods.

Genetic algorithms for the VRP:-

Genetic algorithms are among the most widely used metaheuristics in combinatorial optimization (Meniz and Tiryaki, 2025). For the VRP, (Prins, 2004) showed that a simple GA, starting from a population partly built by heuristics, competes with the best methods. (Wang and Chen, 2012) apply a GA to the VRPSPD with time windows. For highly constrained problems, a random initial population mostly contains infeasible solutions. Starting from feasible solutions built by an insertion heuristic avoids spending the first generations on recovering feasibility.

Positioning:-

Table 1 positions our work. (Gueu et al., 2025) introduced the MILP model of the MPMC-VRPSPDTW. The present paper proposes and compares two approximate solution methods for this model.

Table 1. Positioning with respect to the closest studies (SPD: simultaneous pickup and delivery; TW: time windows).

Study	Multi-product	Multi-compart.	SPD	TW	Perishability	Roughness	Method
(Derigs et al., 2011)	✓	✓	–	–	–	–	Heuristics
(Alfredo Tang Montané and Galvão, 2006)	–	–	✓	–	–	–	Tabu search
(Wang and Chen, 2012)	–	–	✓	✓	–	–	Co-evolutionary GA
(Osvald and Stirn, 2008)	–	–	–	✓	✓	–	Tabu search
(Amorim and Almada-Lobo, 2014)	✓	–	–	✓	✓	–	MILP (ϵ -constraint)
(Buelvas Padilla et al., 2018)	–	–	–	✓	✓	✓	NSGA-II
(Gueu et al., 2025)	✓	✓	✓	✓	✓	✓	MILP
This work	✓	✓	✓	✓	✓	✓	TPDH and hybrid GA, compared with B&C

The MPMC-VRPSPDTW:-**Description and assumptions:-**

In the MPMC-VRPSPDTW, a homogeneous fleet of multi-compartment vehicles leaves a depot to deliver several types of perishable products and to collect empty packaging from a set of customers (Figure 1). Each arc belongs to a road category that sets the damage rate suffered by each product.

The assumptions of the problem are as follows:-

- each vehicle leaves the depot and returns to it at most once;
- each customer is visited exactly once, by a single vehicle, within its time window, and its delivery and pickup demands are fully satisfied;
- each compartment is dedicated to one product and its capacity is respected along the whole route;
- on each arc, each product suffers a damage rate per unit of time that depends on the road category; this damage accumulates along the route and cannot exceed 100%;
- the transport cost and the opportunity cost are minimized in a single objective function.

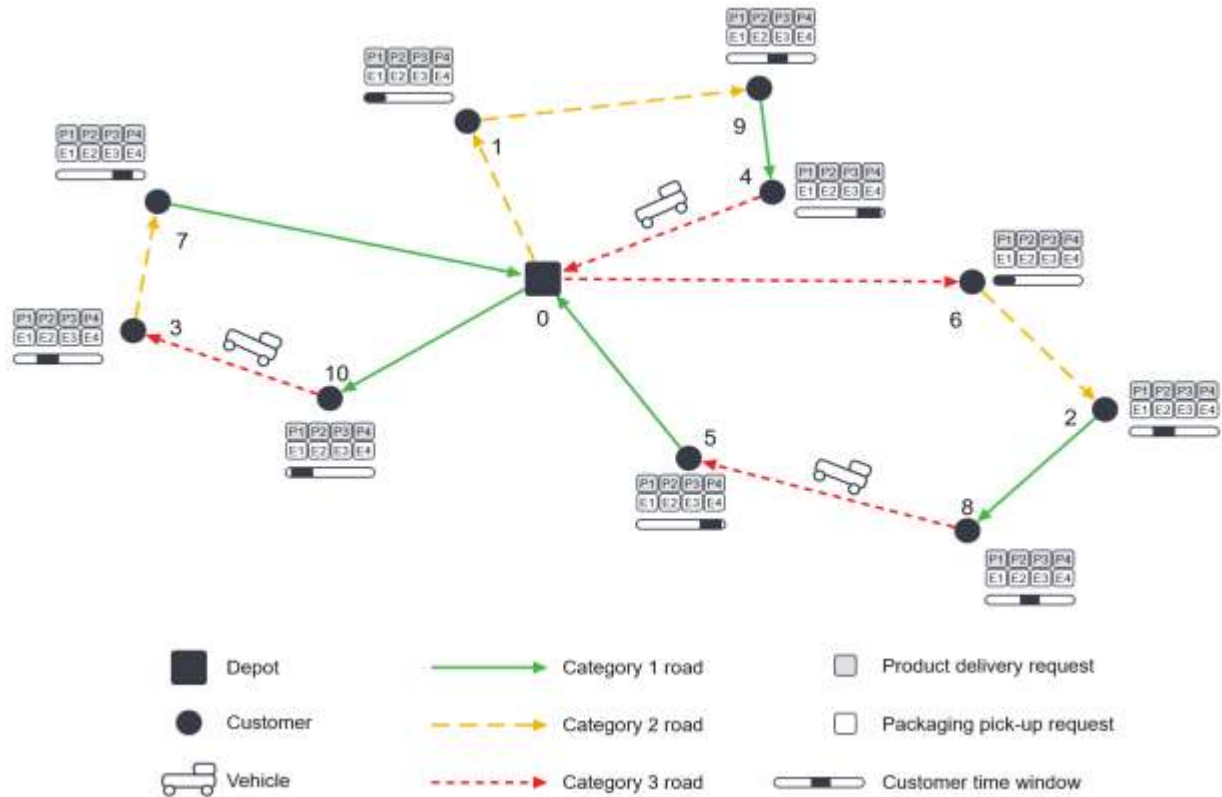


Figure 1. Diagram of the MPMC-VRPSPDTW (P1–P4: delivery demands; E1–E4: packaging to be collected; categories 1 to 3: road types).

Notation:-

The problem is defined on a directed graph $G = (V, A)$, where $V = \{0, \dots, n\}$ contains the depot 0 and the customers $N = \{1, \dots, n\}$. $K = \{1, \dots, |K|\}$ is the fleet and $P = \{1, \dots, m\}$ the set of products.

Table 2. Notation of the model.

Symbol	Meaning
c_{ij}, t_{ij}	Distance and travel time from i to j (including the service time at i)
T_{ijp}	Damage rate per unit of time of product p on arc (i, j) , as a fraction; it depends on the road category
q_{ip}, r_{ip}	Quantity of product p delivered to customer i ; quantity of packaging of p collected from i
Q_p	Capacity of the compartment reserved for product p
e_i, l_i	Bounds of the time window of node i
C_p	Unit price of product p
$F = f_1 + f_2 + f_3 + f_4$	Operating cost per unit of distance (fuel, maintenance, tires, depreciation)
A	Weight of the opportunity cost
M	Large number
x_{ij}^k	1 if vehicle k travels directly from i to j , 0 otherwise
λ_{ik}	1 if vehicle k visits node i , 0 otherwise
z_{ijkp}, y_{ijkp}	Flows of product p to be delivered and of collected packaging of p , carried on (i, j) by k
s_{ik}	Arrival time of vehicle k at node i

MILP model:-

The objective function (1) adds the transport cost and the opportunity cost (economic value lost by damaged products), weighted by the factor α .

$$\min Z = \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} F_{ij} x_{ij}^k + \alpha \sum_{j \in V} \sum_{p \in P} C_p q_{jp} H_{jp} \quad (1)$$

Constraints (2) and (3) require each vehicle to leave the depot and return to it at most once. Constraints (4) to (6) assign each customer to a single vehicle. Constraint (7) limits the number of vehicles used and constraints (8) ensure flow conservation.

$$\sum_{j \in N} x_{0j}^k \leq 1 \quad \forall k \in K \quad (2)$$

$$\sum_{i \in N} x_{i0}^k \leq 1 \quad \forall k \in K \quad (3)$$

$$\sum_{j \in V} \sum_{k \in K} x_{ij}^k = 1 \quad \forall i \in N \quad (4)$$

$$\sum_{j \in V} x_{ij}^k = \lambda_{ik} \quad \forall i \in V, k \in K \quad (5)$$

$$\sum_{k \in K} \lambda_{ik} = 1 \quad \forall i \in N \quad (6)$$

$$\sum_{k \in K} \lambda_{0k} \leq |K| \quad (7)$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0 \quad \forall j \in N, k \in K \quad (8)$$

Constraints (9) and (10) define the pickup and delivery flows. Constraints (11) to (13) bound the load of each compartment on each arc and over the route, and constraints (14) and (15) ensure that all demands are satisfied.

$$\sum_{i \in V} \sum_{k \in K} y_{ijkp} - \sum_{i \in V} \sum_{k \in K} y_{ijkp} = r_{jp} \quad \forall j \in N, p \in P \quad (9)$$

$$\sum_{i \in V} \sum_{k \in K} z_{ijkp} - \sum_{i \in V} \sum_{k \in K} z_{ijkp} = q_{jp} \quad \forall j \in N, p \in P \quad (10)$$

$$y_{ijkp} + z_{ijkp} \leq Q_p x_{ij}^k \quad \forall i, j \in V, k \in K, p \in P \quad (11)$$

$$\sum_{i \in N} \sum_{j \in V} r_{ip} x_{ij}^k \leq Q_p \quad \forall k \in K, p \in P \quad (12)$$

$$\sum_{i \in N} \sum_{j \in V} q_{ip} x_{ij}^k \leq Q_p \quad \forall k \in K, p \in P \quad (13)$$

$$\sum_{i \in N} \sum_{k \in K} y_{i0kp} = \sum_{i \in N} r_{ip} \quad \forall p \in P \quad (14)$$

$$\sum_{i \in N} \sum_{k \in K} z_{i0kp} = \sum_{i \in N} q_{ip} \quad \forall p \in P \quad (15)$$

Constraints (16) to (18) enforce the time windows, eliminate subtours and set the return to the depot.

$$e_i \leq s_{ik} \leq l_i \quad \forall i \in V, k \in K \quad (16)$$

$$s_{ik} + t_{ij} - s_{jk} \leq (1 - x_{ij}^k)M \quad \forall i, j \in V, k \in K, j \neq 0 \quad (17)$$

$$s_{ik} + t_{i0} \leq l_0 + (1 - x_{i0}^k)M \quad \forall i \in N, k \in K \quad (18)$$

Inspired by (Buelvas Padilla et al., 2018), constraints (19) accumulate the damage of each product along the route: the damage at customer j is at least that of the preceding node i , plus the damage suffered on arc (i, j) . Constraints (20) and (21) bound the cumulative damage between 0 and 100%, and constraints (22) and (23) define the domains of the decision variables.

$$H_{jp} \geq H_{ip} + T_{ijp} t_{ij} + (x_{ij}^k - 1)M \quad \forall i, j \in V, k \in K, p \in P, j \neq 0 \quad (19)$$

$$H_{0p} = 0, \quad H_{jp} \geq 0 \quad \forall j \in V, p \in P \quad (20)$$

$$H_{jp} \leq 1 \quad \forall j \in V, p \in P \quad (21)$$

$$x_{ij}^k, \lambda_{ik} \in \{0, 1\} \quad \forall i, j \in V, k \in K \quad (22)$$

$$s_{ik}, y_{ijkp}, z_{ijkp} \geq 0 \quad \forall i, j \in V, k \in K, p \in P \quad (23)$$

All constraints are linear: the flow variables y and z and the formulations of (17) to (19) avoid products of variables. The model is detailed in (Gueu et al., 2025).

Solution methods:-

Both solution methods evaluate a solution with the same fitness, equal to the objective function (1):

$$\text{Fitness}(s) = F \times \text{distance}(s) + \alpha \times \text{cost}_{\text{opp}}(s) \quad (24)$$

Both handle only feasible solutions. A customer is inserted or moved only if the resulting route respects the vehicle compartment capacities, the time windows and the cumulative damage threshold of 100% for each product. Figure 2 compares the processes of the two methods.

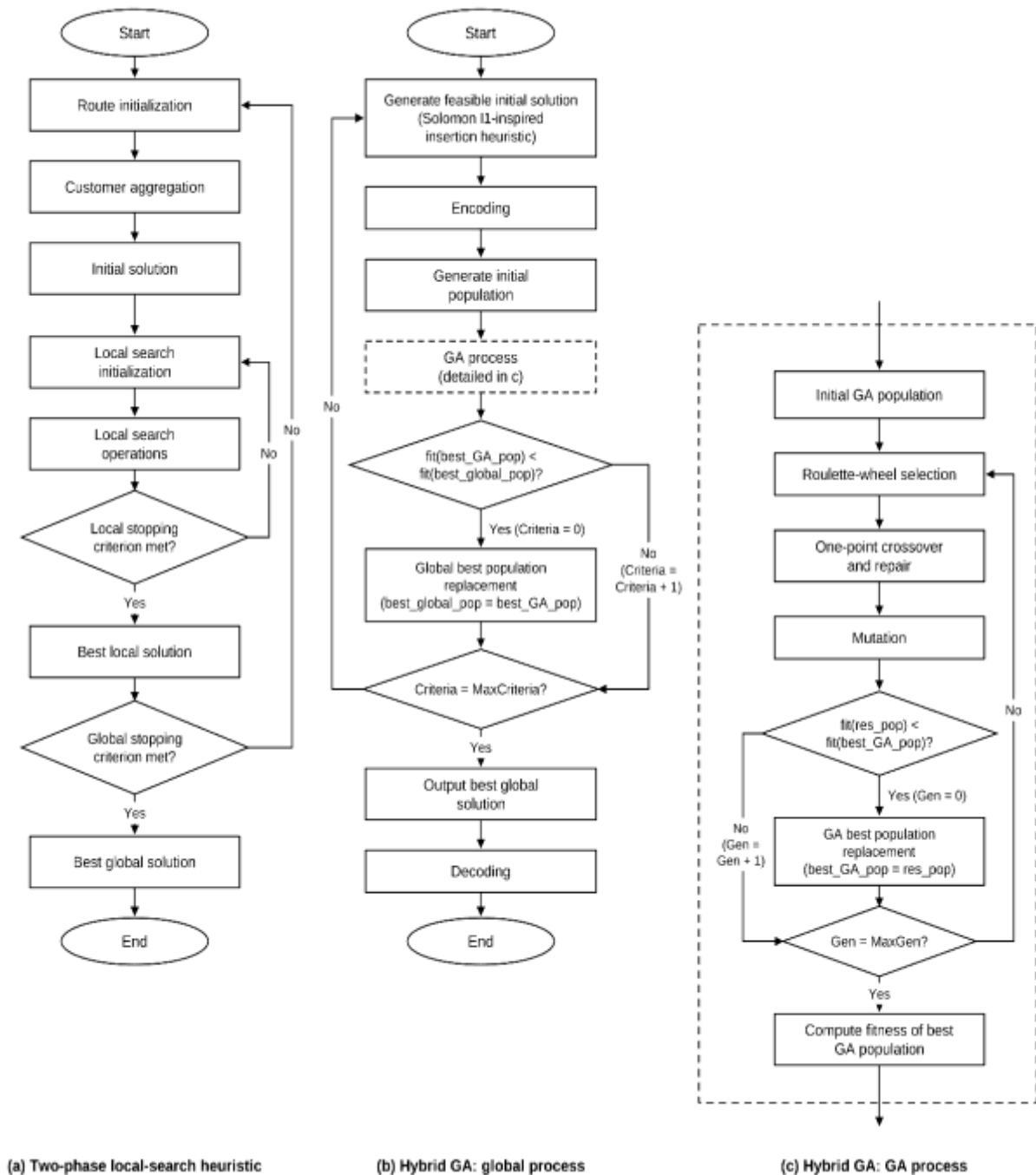


Figure 2. Compared processes of the TPDH and the hybrid GA.

Two-phase decomposition heuristic (TPDH):-

The TPDH follows the “cluster-first, route-second” principle. It repeats two phases within a global loop (Algorithm 1):-

Phase 1: customer clustering. Routes are built one at a time. The first customer of a route is drawn at random among the unserved customers. The following customers are chosen as the nearest neighbour of the last inserted customer, provided that the insertion is feasible. A route is closed when no more customers can enter it, and a new one is opened as long as customers and vehicles remain.

Phase 2: inter-route local search. Two customers from two different routes, drawn at random, are swapped. The swap is kept if the routes are feasible and if it reduces the fitness. Otherwise, the swap is undone. Phase 2 stops after MaxIter consecutive swaps without improvement.

Global loop. The solution obtained by the local search replaces the best known solution if it is better. The algorithm stops after MaxCriteria consecutive global iterations without improvement.

Algorithm 1. TPDH

Input : instance, MaxIter, MaxCriteria

Output: best solution s^*

```

1  criteria  $\leftarrow 0$  ;  $s^* \leftarrow \emptyset$ 
2  while criteria < MaxCriteria do
3       $s \leftarrow \text{CLUSTERING}()$  // phase 1: feasible nearest neighbor
4      iter  $\leftarrow 0$ 
5      while iter < MaxIter do // phase 2: local search
6           $s' \leftarrow \text{RANDOM\_SWAP}(s)$  // two customers from two routes
7          if  $s'$  feasible and  $f(s') < f(s)$  then  $s \leftarrow s'$  ; iter  $\leftarrow 0$ 
8          else iter  $\leftarrow \text{iter} + 1$ 
9      end while
10     if  $s^* = \emptyset$  or  $f(s) < f(s^*)$  then  $s^* \leftarrow s$  ; criteria  $\leftarrow 0$ 
11     else criteria  $\leftarrow \text{criteria} + 1$ 
12 end while
13 return  $s^*$ 
```

Hybrid genetic algorithm:-

The hybrid GA replaces the greedy construction of the TPDH by a least-cost insertion heuristic, then explores around this solution with a population (Algorithm 2):-

Initial solution: the insertion heuristic is inspired by the I1 heuristic of (Solomon, 1987). The first customer of each route is drawn at random. At each step, the insertion of each remaining customer is tested at every position of the current route, and the feasible insertion of least cost is kept. When no insertion is possible, a new route is opened.

Encoding and initial population: a chromosome is a sequence of integers: the customers are numbered from 1 to n , and the depot 0 is placed at the beginning and end of each route. The initial population contains the insertion solution, then solutions obtained by successive random swaps of two customers between two routes. Each new solution derives from the previous one and is added only if it is feasible, until the set population size (PopSize) is reached.

Selection: an inverted roulette wheel gives each individual a weight inversely proportional to its fitness. PopSize/2 parents are drawn according to these weights.

Crossover and repair: two distinct parents are crossed with probability P_c by a one-point crossover. Each child receives the first half of the genes of one parent and the second half of the other. The repair then replaces duplicated customers by the missing ones. If one of the two children is infeasible, both are rejected.

Mutation: with probability P_m , two customers from two distinct routes are removed and each is reinserted at its best feasible position in the route of the other customer. If one of the insertions fails, the original chromosome is kept.

Replacement and stopping: the new population replaces the best GA population if the sum of the fitness values of its individuals is lower. The GA stops after MaxGen generations without improvement. A global loop then restarts a new insertion solution and a new GA, and keeps the best population found. It stops after MaxCriteria iterations without improvement, and the best solution of this population is decoded.³

Algorithm 2. Hybrid GA

Input : instance, PopSize, Pc, Pm, Max Gen, Max Criteria

Output: best solution s^*

```

1  criteria  $\leftarrow 0$  ; B  $\leftarrow \emptyset$  // best global population
2  while criteria < Max Criteria do
3      s0  $\leftarrow$  INSERTION_I1() // feasible solution
4      P  $\leftarrow$  INITIAL_POPULATION (s0, PopSize) // successive feasible swaps
5      gen  $\leftarrow 0$ 
6      while gen < MaxGen do
7          parents  $\leftarrow$  INVERTED_ROULETTE(P, PopSize/2)
8          R  $\leftarrow$  CROSSOVER_REPAIR(parents, Pc) ; MUTATION(R, Pm)
9          if  $\Sigma f(R) < \Sigma f(P)$  then P  $\leftarrow$  R ; gen  $\leftarrow 0$ 
10         else gen  $\leftarrow$  gen + 1
11     end while
12     if B =  $\emptyset$  or  $\Sigma f(P) < \Sigma f(B)$  then B  $\leftarrow$  P ; criteria  $\leftarrow 0$ 
13     else criteria  $\leftarrow$  criteria + 1
14 end while
15 return the best individual of B, decoded

```

Differences between the two methods:-

Both methods share the inter-route swap, used as local search by the TPDH and as a source of diversity by the GA. They differ on two points. First, construction: the nearest-neighbour rule of the TPDH looks only at the next customer, whereas the least-cost insertion of the GA compares all positions of the route. Second, search: the TPDH improves a single solution, while the GA recombines a population of 100 feasible solutions.

Experimental protocol:-

Hardware and software:-

The three methods (B&C, TPDH and hybrid GA) are implemented in C++. The B&C is that of CPLEX 12.6, applied directly to the MILP with a single thread, a 7,200 s time limit and automatic cuts, heuristics and preprocessing. When the limit is reached, CPLEX returns the best feasible solution found. The experiments are run on a 64-bit computer with an 8th-generation Intel Core i7 processor at 2.3 GHz and 8 GB of RAM.

Instances:-

We use three instances of (Solomon, 1987) that differ in customer distribution and time-window tightness: C101 (clustered customers, very tight windows), R104 (scattered customers, tight windows) and RC107 (mixed structure, moderately tight windows). Each instance has up to 100 customers and 25 vehicles of capacity 200. Solomon's instances have only one demand per customer.

We complete them as follows to fit our model:-

- the demand q_i of each customer is split equally among four products ($q_{ip} = q_i / 4$);
- the collected packaging equals the delivered quantity of the same product ($r_{ip} = q_{ip}$);
- the vehicle of capacity 200 is divided into four compartments, one per product;
- one time unit of the instances is taken as one minute.

Damage rates and road scenarios:-

The damage rates T_{ijp} are derived from the measurements of (Chonhenchob et al., 2009) on four products transported in Thailand. For each measured trip, the percentage of loss per unit of load is divided by the mean travel time, itself equal to the distance divided by the mean speed. The three measured trips define three road categories (Table 3).

Table 3. Damage rates T_{ijp} (% per minute) by road category (Chonhenchob et al., 2009).

Road category (measured trip)	Lettuce	Cabbage	Chinese pear	Chinese plum
Rural (collection centres $\rightarrow$ packing house, Chiang Mai)	0.469	0.521	0.406	0.156
Intercity (Chiang Mai $\rightarrow$ distribution centre, Bangkok)	0.050	0.067	0.048	0.017
Urban (distribution centre $\rightarrow$ retailers, Bangkok)	0.667	1.000	1.400	0.333

As in (Buelvas Padilla et al., 2018), we define four network composition scenarios. Scenario 1 splits the roads into three thirds (intercity, rural, urban). Scenarios 2, 3 and 4 combine in equal shares, respectively, intercity and rural

roads, intercity and urban roads, and rural and urban roads. The category of each arc is drawn according to these proportions, symmetrically ($w_{ij} = w_{ji}$). For a given instance-scenario pair, the three methods use the same matrix.

Costs:-

Product prices are observed average prices (Table 4). Operating costs per kilometre are taken from Barnes and Langworthy (2004): fuel $f_1 = \$0.214$, maintenance $f_2 = \$0.131$, tires $f_3 = \$0.044$ and depreciation $f_4 = \$0.10$, i.e. $F = \$0.489/\text{km}$. These costs are taken as constant on all arcs: the road category only affects product degradation.

Table 4. Product prices (\$/kg), source: France Agri Mer.

Product	Price range	Average price used
Lettuce	1.57 – 2.20	1.90
Cabbage	0.60 – 1.20	0.90
Chinese pear	2.50 – 3.00	2.75
Chinese plum	2.50 – 3.50	3.00

Parameters and procedure:-

The parameters of the two algorithms are given in Table 5.

Table 5. Parameters of the two methods.

Method	Parameter	Value
TPDH	Consecutive swaps without improvement in local search (MaxIter)	10
TPDH	Consecutive global iterations without improvement (MaxCriteria)	15
Hybrid GA	Population size (PopSize)	100
Hybrid GA	Crossover probability (P_c)	0.8
Hybrid GA	Mutation probability (P_m)	0.05
Hybrid GA	Consecutive generations without improvement (MaxGen)	2
Hybrid GA	Consecutive global iterations without improvement (MaxCriteria)	2

For each instance-scenario pair, the number of customers starts at 10 and increases in steps of 5 up to the largest size the B&C could handle. This yields 45 configurations: 20 for C101 (10 to 35 customers), 13 for R104 (10 to 25) and 12 for RC107 (10 to 20). The weight α is set empirically for each configuration, between 0.4 and 4, and is identical for the three methods. Since α varies from one configuration to another, we compare the methods configuration by configuration, with paired tests, rather than through mean costs alone.

Results:-

Detailed results:-

Table 6 gives, for each configuration, the objective cost, the computation time and the number of vehicles of the three methods. The GA/TPDH gap equals (GA cost – TPDH cost) / TPDH cost; a negative value indicates a gain for the GA. The B&C time is not reported: it is below 7,200 s for the 23 optimal configurations and reaches this limit for the other 22.

Table 6. Results by configuration (Sc.: scenario; costs in \$; †: 7,200 s limit reached by the B&C, value not proven optimal).

Instance	Sc.	n	α	B&C cost	TPDH cost	GA cost	GA/TPDH gap (%)	TPDH CPU (s)	GA CPU (s)	B&C veh.	TPDH veh.	GA veh.
C101	1	10	0.5	81.64	88.16	81.64	-7.39	116.9	107.4	3	3	3
C101	1	15	0.5	188.43	201.49	189.54	-5.93	178.9	132.7	3	6	4
C101	1	20	0.5	268.00	305.90	271.89	-11.12	443.8	137.1	3	8	5
C101	1	25	0.5	308.27	367.15	329.95	-10.13	334.4	1,264.1	5	8	7

Instance	Sc.	n	α	B&C cost	TPDH cost	GA cost	GA/TPDH gap (%)	TPDH CPU (s)	GA CPU (s)	B&C veh.	TPDH veh.	GA veh.
C101	1	30	0.5	398.89	408.76	434.15	6.21	862.9	1,162.8	5	10	8
C101	1	35	0.4	409.02	450.43	441.00	-2.09	1,020.7	2,368.1	6	11	8
C101	2	10	0.5	67.90	76.18	67.90	-10.87	29.3	297.8	2	3	2
C101	2	15	0.5	138.39	163.90	142.91	-12.81	48.4	312.4	3	5	3
C101	2	20	0.5	245.64	260.15	248.27	-4.57	315.6	623.8	3	6	4
C101	2	25	0.5	272.99	325.32	292.62	-10.05	288.2	733.3	4	10	5
C101	2	30	0.4	307.75 †	370.97	349.20	-5.87	940.9	914.0	5	9	6
C101	3	10	0.5	70.61	89.87	70.61	-21.43	69.6	106.7	3	4	3
C101	3	15	0.5	148.08	175.77	151.21	-13.98	158.9	219.1	4	5	4
C101	3	20	0.5	229.49	270.55	237.53	-12.21	185.9	234.3	4	8	5
C101	3	25	0.4	320.80	327.75	327.07	-0.21	456.2	759.9	5	8	6
C101	3	30	0.4	347.78	405.38	402.72	-0.66	4,422.2	2,010.3	6	11	7
C101	3	35	4	1,183.62 †	1,093.42	1,205.37	10.24	1,208.2	821.3	6	12	7
C101	4	10	0.5	129.22	129.22	129.22	0.00	54.9	85.2	5	5	5
C101	4	15	0.4	252.48	252.48	252.48	0.00	308.7	436.1	8	8	8
C101	4	20	0.7	443.20 †	450.83	444.73	-1.35	390.3	429.1	10	10	11
R104	1	10	0.5	131.85	150.73	134.33	-10.88	30.2	100.6	2	3	2
R104	1	15	0.7	178.33 †	235.69	211.69	-10.18	69.4	419.0	3	4	4
R104	1	20	0.5	228.22 †	289.56	259.66	-10.33	115.2	195.8	5	6	4
R104	2	10	1.5	144.23	153.91	159.18	3.43	14.2	126.4	2	3	3
R104	2	15	2	193.05 †	253.45	235.52	-7.07	48.9	229.9	4	5	3
R104	2	20	3	272.17 †	355.77	321.31	-9.69	120.7	1,620.0	5	7	5
R104	3	10	2	180.51	189.62	180.60	-4.76	49.0	52.8	2	4	2

Instance	Sc.	n	α	B&C cost	TPDH cost	GA cost	GA/TPDH gap (%)	TPDH CPU (s)	GA CPU (s)	B&C veh.	TPDH veh.	GA veh.
R104	3	15	1.5	204.92 †	261.56	233.71	-10.65	182.6	157.1	3	6	3
R104	3	20	1	250.41 †	298.02	297.65	-0.12	325.1	528.8	4	5	4
R104	3	25	0.4	294.41 †	350.82	368.67	5.09	665.3	886.7	5	7	5
R104	4	10	2.5	257.51	284.04	265.14	-6.66	57.0	215.7	3	4	4
R104	4	15	0.7	229.32 †	263.94	267.50	1.35	71.5	253.3	6	5	4
R104	4	20	0.4	249.82 †	297.40	273.07	-8.18	219.2	373.5	6	6	5
RC107	1	10	0.5	110.12	143.47	121.32	-15.44	32.5	173.8	2	3	2
RC107	1	15	0.5	133.76 †	196.94	155.96	-20.81	73.5	193.3	2	4	2
RC107	1	20	4	940.18 †	943.48	958.00	1.54	114.2	917.0	4	6	5
RC107	2	10	0.5	98.26 †	98.26	103.88	5.71	196.0	54.9	2	2	2
RC107	2	15	3	176.38 †	239.37	247.87	3.55	90.4	210.1	2	3	3
RC107	2	20	0.5	191.20 †	291.70	230.51	-20.98	142.8	267.6	3	5	3
RC107	3	10	0.5	107.68	134.25	123.17	-8.25	47.9	44.5	2	3	2
RC107	3	15	2	167.79 †	211.62	226.34	6.96	57.5	128.7	2	3	3
RC107	3	20	4	466.67 †	739.60	548.74	-25.81	373.9	1,124.8	4	7	5
RC107	4	10	2	315.44 †	358.41	350.84	-2.11	41.5	57.0	3	3	2
RC107	4	15	3	534.11 †	689.79	650.18	-5.74	103.8	528.3	4	6	5
RC107	4	20	0.5	348.10 †	408.31	347.51	-14.89	156.2	330.2	4	7	5

Summary by instance:-

Table 7 presents the mean values of the indicators (objective cost, CPU time, number of vehicles used) by instance.

Table 7. Means by instance (costs in \$, times in s).

Instance	Config.	B&C cost	TPDH cost	GA cost	GA/TPDH gap	TPDH CPU	GA CPU	B&C veh.	TPDH veh.	GA veh.	GA < TPDH in cost
C101	20	290.61	310.68	303.50	-2.3%	591.7	657.8	4.65	7.50	5.55	16 / 20
R104	13	216.52	260.35	246.77	-5.2%	151.4	396.9	3.85	5.00	3.69	10 / 13
RC107	12	299.14	371.27	338.69	-8.8%	119.2	335.9	2.83	4.33	3.25	8 / 12
Overall	45	271.48	312.30	296.50	-5.1%	338.5	496.6	3.93	5.93	4.40	34 / 45

The analysis of the performance of the algorithms according to the three indicators is as follows:-

Objective cost: the hybrid GA obtains a lower cost than the TPDH in 34 of the 45 configurations, an equal cost in 2 and a higher cost in 9. Its mean cost is 5.1% lower; the median gap per configuration is -6.7%. On mean costs, the gain is smallest for C101 (-2.3%) and largest for RC107 (-8.8%). The means of the per-configuration gaps are closer (-5.7% for C101, -5.3% for R104, -8.0% for RC107): only RC107 stands out clearly.

Number of vehicles: the GA uses fewer vehicles than the TPDH in 35 configurations, as many in 9 and one more in a single case (C101, scenario 4, 20 customers). The mean fleet falls from 5.93 to 4.40 vehicles, i.e. -25.8%, with a decrease close to 25% on all three instances. The B&C remains the most economical, with 3.93 vehicles on average.

Computation time: the GA is slower than the TPDH in 36 configurations. Its mean time is 496.6 s against 338.5 s, a ratio of 1.47. The median ratio per configuration is 1.70. The gap is largest on R104 and RC107, where the TPDH converges in 120 to 150 s on average. Both methods remain much faster than the B&C, whose mean time is 3,797 s.

Optimality gap:-

On the 23 configurations where the B&C proves optimality, the mean optimality gap is 4.4% for the hybrid GA against 12.6% for the TPDH (Figure 3). The GA reaches the optimum in 5 configurations, the TPDH in 2. The TPDH is closer to the optimum in only 2 configurations: C101 scenario 1 with 30 customers (2.5% vs 8.8%) and R104 scenario 2 with 10 customers (6.7% vs 10.4%). On the 22 unsolved configurations, the best B&C solution is beaten twice: by the TPDH for C101 scenario 3 with 35 customers (1,093.42 vs 1,183.62) and by the GA for RC107 scenario 4 with 20 customers (347.51 vs 348.10).

The hybrid GA stays within 4.4% of the optimum on average, versus 12.6% for TPDH

Gap to the B&C optimal cost (%), 23 configurations solved to optimality

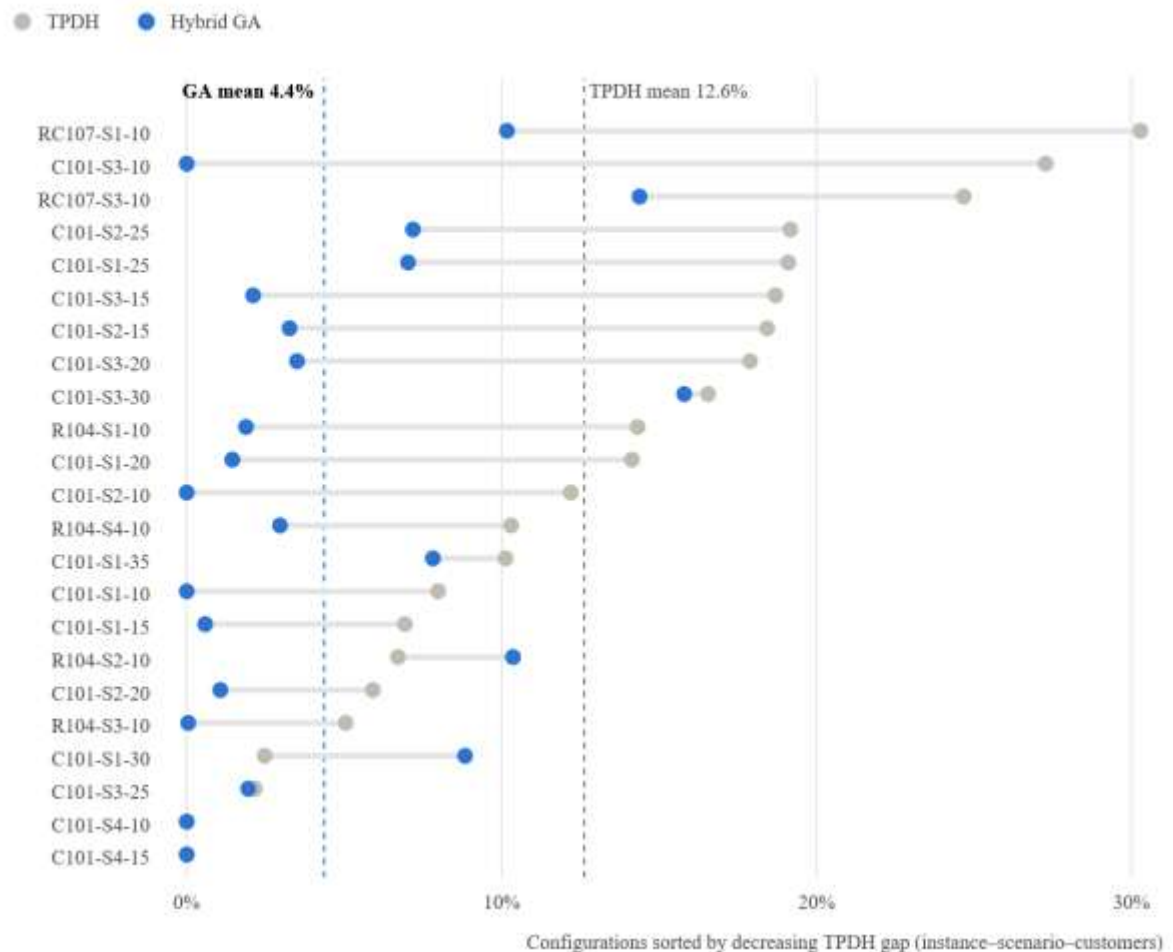


Figure 3. Optimality gap of the TPDH and the hybrid GA on the 23 configurations solved to optimality by the B&C. The GA is closer to the optimum in 19 configurations and tied in 2.

Statistical tests:-

Since the three methods are applied to the same configurations, we use paired non-parametric tests. The Friedman test rejects the equality of the three methods for cost ($\chi^2 = 62.4$; $p = 2.9 \times 10^{-14}$; mean ranks: 1.13 for the B&C, 2.14 for the GA, 2.72 for the TPDH) and for the number of vehicles ($\chi^2 = 56.3$; $p = 5.9 \times 10^{-13}$). Pairwise comparisons with the Wilcoxon signed-rank test and Holm's correction show that the GA and the TPDH differ significantly in cost (adjusted $p = 1.1 \times 10^{-4}$) and in number of vehicles (adjusted $p = 4.2 \times 10^{-7}$). The GA is also significantly slower than the TPDH ($p = 2.4 \times 10^{-5}$).

Limitations:-**Our results call for several caveats:-**

- **Different computational budgets:** both methods stop after a number of iterations without improvement, not after a common time or number of evaluations. Part of the quality gap may therefore come from the additional time given to the GA. A comparison with equal time could clarify this point.
- **Stochastic nature:** both methods are randomized. Repeated independent runs for each configuration would make it possible to measure the variability of their results.
- **Instance size:** the configurations stop at 35 customers, the largest size handled by the B&C. The behaviour of the two heuristics on 50 to 100 customers remains to be established.

Conclusion:-

We compared two approximate methods for solving the MPMC-VRPSPDTW: a two-phase decomposition heuristic (TPDH) and a hybrid genetic algorithm initialized by an insertion heuristic inspired by Solomon's I1. On 45 configurations derived from Solomon's instances C101, R104 and RC107, the hybrid GA obtains a lower cost in 34 configurations and reduces the mean cost by 5.1% and the mean fleet size by 25.8%. Its mean optimality gap is 4.4%, against 12.6% for the TPDH, on the 23 configurations solved by the CPLEX B&C. These gains come at the price of a mean run time 1.5 times longer (497 s vs 339 s), which remains far below that of the B&C (3,797 s). The TPDH remains useful when a solution must be obtained very quickly. The hybrid GA offers the best trade-off between route quality, fleet size and computation time on the tested configurations. Three extensions are planned. The first is a comparison with an equal computational budget, over several runs and on instances of 50 to 100 customers. The second is a multi-objective version of the problem, which avoids setting α a priori and treats the number of vehicles as an objective in its own right. The third is the parallelization of the GA, whose population evaluation lends itself well to computation on multi-core architectures.

References:-

1. Alfredo Tang Montané, F., Galvão, R.D., 2006. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research* 33, 595–619. <https://doi.org/10.1016/j.cor.2004.07.009>
2. Amorim, P., Almada-Lobo, B., 2014. The impact of food perishability issues in the vehicle routing problem. *Computers & Industrial Engineering* 67, 223–233. <https://doi.org/10.1016/j.cie.2013.11.006>
3. Buelvas Padilla, M.P., Nisperuza Canabal, P.A., López Pereira, J.M., Hernández Riaño, H.E., 2018. Vehicle routing problem for the minimization of perishable food damage considering road conditions. *Bundesvereinigung Logistik (BVL) e.V., DE*.
4. Chonhenchob, V., Sittipod, S., Rachtanapun, P., Singh, S.P., Singh, J., 2009. Effect of Truck Vibration during Transport on Damage to Fresh Produce Shipments in Thailand. *Journal of Applied Packaging Research*, Vol. 3, No. 1.
5. Clarke, G., Wright, J.W., 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* 12, 568–581. <https://doi.org/10.1287/opre.12.4.568>
6. Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U., 2011. Vehicle routing with compartments: applications, modelling and heuristics. *OR Spectrum* 33, 885–914. <https://doi.org/10.1007/s00291-010-0194-3>
7. Fernando, I., Fei, J., Stanley, R., Enshaie, H., 2018. Measurement and evaluation of the effect of vibration on fruits in transit—Review. *Packag Technol Sci* 31, 723–738. <https://doi.org/10.1002/pts.2409>
8. Fisher, M.L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11, 109–124. <https://doi.org/10.1002/net.3230110205>
9. Gueu, O.É.-N., Coulibaly, K.T., Kayé, B.K.B., Oumtanaga, S., 2025. An optimization model for the distribution of perishable products with consideration of road roughness conditions. *IJBAS* 14, 1–11. <https://doi.org/10.14419/1x6arf11>

10. Idah, P.A., Ajisegiri, E.S.A., Yisa, M.G., 2007. Fruits and Vegetables Handling and Transportation in Nigeria. Department Agricultural Engineering, Federal University of Technology Minna, Niger State, Nigeria 175–183.
11. Jarimopas, B., Singh, S.P., Saengnil, W., 2005. Measurement and analysis of truck transport vibration levels and damage to packaged tangerines during transit. Packag. Technol. Sci. 18, 179–188. <https://doi.org/10.1002/pts.687>
12. Koç, Ç., Laporte, G., Tükenmez, İ., 2020. A review of vehicle routing with simultaneous pickup and delivery. Computers & Operations Research 122, 104987. <https://doi.org/10.1016/j.cor.2020.104987>
13. Meniz, B., Tiryaki, F., 2025. Genetic Algorithm Optimization with Selection Operator Decider. Arab J Sci Eng 50, 6931–6941. <https://doi.org/10.1007/s13369-024-09068-5>
14. Min, H., 1989. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research Part A: General 23, 377–386. [https://doi.org/10.1016/0191-2607\(89\)90085-X](https://doi.org/10.1016/0191-2607(89)90085-X)
15. Ostermeier, M., Henke, T., Hübner, A., Wäscher, G., 2021. Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. European Journal of Operational Research 292, 799–817. <https://doi.org/10.1016/j.ejor.2020.11.009>
16. Osvald, A., Stirn, L.Z., 2008. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. Journal of Food Engineering 85, 285–295. <https://doi.org/10.1016/j.jfoodeng.2007.07.008>
17. Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002. [https://doi.org/10.1016/S0305-0548\(03\)00158-8](https://doi.org/10.1016/S0305-0548(03)00158-8)
18. Solomon, M.M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research 35, 254–265. <https://doi.org/10.1287/opre.35.2.254>
19. Wang, H.-F., Chen, Y.-Y., 2012. A genetic algorithm for the simultaneous delivery and pickup problems with time window. Computers & Industrial Engineering 62, 84–95. <https://doi.org/10.1016/j.cie.2011.08.018>