International Journal of Basic and Applied Sciences, 14 (6) (2025) 51-62

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/pjenr873
Research paper

Modeling Fleet Management for The Urban Distribution of Perishable Goods According to Customer Proximity

Doua Etienne [1] *, Diaby Moustapha [2], Kayé Bi Kouaï Bertin [3], Coulibaly Adama [1]

[1] LAMAP, Université Félix Houphouët-Boigny (UFHB); Côte d'Ivoire, 01 BP V34 Abidjan 01
[2] LASTIC, École Supérieure Africaine des TIC (ESATIC); Côte d'Ivoire, 18 BP 1501 Abidjan 18
[3] LAMI, Université Félix Houphouët-Boigny (UFHB); Côte d'Ivoire, 01 BP V34 Abidjan 01
**Corresponding author E-mail: etiennedoua1997@gmail.com*

Received: April 25, 2025, Accepted: August 27, 2025, Published: October 4, 2025

Abstract

Managing the distribution of perishable goods is a major challenge due to the need to maintain their quality throughout the transport process. Fleet management plays a crucial role in this regard. This paper proposes a fleet management model that integrates customer segmentation based on their proximity to the depot. Customers are classified into two categories: nearby customers and distant customers. This is made possible by adding a grouping constant. Ordinary vehicles serve nearby customers, while refrigerated vehicles serve distant customers to preserve product quality by regulating temperature. This distinction allows us to adapt delivery strategies, optimize routes, and guarantee the quality of the products delivered, while reducing logistics costs. This will therefore reduce product waste through strict control of delivery times and reduce the carbon footprint through the rational use of energy-intensive refrigerated vehicles. The grouping constant has a strong influence on the model's performance, which is why a sensitivity analysis was carried out to select the ideal grouping constant for solving the developed model. Experimental tests were carried out on instances from the literature. The results of the experiments show that for the CPLEX solver, 76.17% of instances were solved within the allotted time, compared to 23.83% of instances that were not solved within the allotted time, while the decomposition heuristics solved all the instances used. Comparisons between the CPLEX solver and the decomposition heuristic (TPDH) were made on instances for which we obtained solutions with both methods. The results show that the average cost difference between the two methods is 4.70%. They also show an average gain of 94% in computation time in favor of TPDH.

Keywords: *Customer proximity, Fleet Management, Perishable Goods, Urban distribution, VRP.*

1. Introduction

Managing the distribution of perishable products is a major challenge in the field of logistics, due to the need to preserve their quality throughout the transport process [3]. Most studies on perishability have focused on stock management, as evidenced by the work of Nahmias [21], Raafat [22], Goyal and Giri [14], and Karaesmen et al [17]. Perishable products, such as fresh food, medicines, or other temperature-sensitive items, require special attention to ensure that they arrive at their destination in optimal conditions. This challenge becomes even more complex when it comes to optimizing vehicle routes while considering geographical, temporal, and economic constraints. Current distribution networks often use refrigerated vehicles and ordinary vehicles to meet the needs of urban logistics [5]. In the latter, type 1 customers, who are to be delivered by refrigerated vehicles, are determined, and type 2 customers are assigned to type 1 customers according to their proximity, and are to be delivered by ordinary vehicles. In this study, we propose a distribution model that integrates vehicles of different types and a segmentation of customers according to their proximity to the depot. Here, customers are grouped into nearby customers and distant customers. Ordinary vehicles are used for nearby customers, and refrigerated vehicles are used for distant customers. The temperature in refrigerated vehicles is assumed to be sufficient to preserve the quality of the perishable products transported. We use two approaches to solve this problem. The first is the exact resolution with the CPLEX solver, and the second is a two-phase decomposition heuristic (TPDH), based on a genetic algorithm. The performance of the algorithms will be evaluated on the P instances developed by Augerat et al in 1995 [4]. This work will be organized as follows: literature review (section 2), description and mathematical formulation of the model (section 3), identification of clients (section 4.1), decomposition heuristics (section 4.2), experimentation (section 5), computational results (section 6), and discussion (section 7).

2. Literature Review

The Vehicle Routing Problem (VRP) originates from the Traveling Salesman Problem (TSP), which was first proposed by Dantzig and Ramser [7]. Since then, various versions of VRP have been developed, including Closed VRP [9], Open VRP [12], Static VRP [9], Dynamic VRP [1], VRP with Time Windows [15], etc. In this context, Fu et al. (2005) published a paper on the Open Vehicle Routing Problem

(OVRP), which they solved using Tabu Search [12]. Moreover, García-Nájera et al [13] published an article in 2015 on VRP with Backhauls. They divided customers into two groups: linehaul customers and backhaul customers, depending on whether the product needs to be delivered or picked up. The model is solved using an evolutionary algorithm based on similarity-based selection. Readers are referred to Laporte [19] and Eksioglu et al. [10] for more general VRP problems. Numerous studies propose various routes and strategies to optimize transportation models. Although the field of perishable goods transport and supply chain optimisation remains relatively unexplored, several studies have proposed variants of the VRP adapted to the constraints associated with perishability. As early as 1993, Wee [27] defined perishability as degradation, deterioration, evaporation, obsolescence, or loss of marginal value of the product. This definition has been expanded in more recent work that incorporates shelf life [5], sensitivity to time and temperature, and realistic parameters for optimising the cold chain while reducing costs and emissions (Tirkolaee et al [26]). Therefore, it is necessary to find an efficient vehicle routing policy and supply chain design for perishable goods. The work of Tarantilis and Kiranoudis (2001) [24] focuses on fresh milk distribution and formulates the problem as a heterogeneous fixed fleet VRP. A threshold acceptance-based algorithm was designed to meet the needs of a company that had to produce a schedule recurrently, several times a day. In 2002, Tarantilis and Kiranoudis [25] looked at an open vehicle routing problem with multiple depots, applied to the distribution of fresh meat from depots to customers located in an Athens neighbourhood. They proposed a stochastic search metaheuristic to solve this problem. Perishable products are products that have a short shelf life and therefore require special attention during their distribution to maintain their quality throughout the distribution process [3]. Refrigerated vehicles are often used to maintain product quality during distribution. Amorim and Almada-Lobo (2014) [2] developed a scenario-based multi-objective model, aiming to reduce distribution costs while optimizing product freshness. For small-sized instances, they used the constraint method, while for large-scale problems, they opted for an evolutionary algorithm. Hsiao et al. (2017) [16] focused on the distribution planning of perishable products, integrating cold chains to address quality-related issues. This was defined based on the estimated shelf life, different for each product, and deteriorates over time. To solve this problem, they applied a biogeography-based optimization (BBO) method. Except that the use of these vehicles for the distribution of goods leads to high costs for the company at the expense of ordinary vehicles. To reduce company expenses, it would be interesting to alternate general vehicles and refrigerated vehicles optimally to further minimize distribution costs. In this sense, Song and Ko (2016) [23] worked on the VRP, which includes two types of vehicles: refrigerated vehicles and ordinary vehicles. Their two-vehicle-type VRP model aims to maximize the quality of perishable products while minimizing distribution costs, considering the combination of several types of products and the specific characteristics of the vehicles. Another relevant example is that of Partha Sarathi Barma et al. (2022) [5], who proposed a multi-objective model for the CVRP, where several types of vehicles are used in an m-ring star-type distribution network. This model seeks to balance costs and the quality of delivered products, while respecting vehicle capacity constraints. In this article, the model we develop represents a distribution network for perishable products, where two types of vehicles are used: refrigerated vehicles and ordinary vehicles, just like [5]. Our approach focuses on a single type of perishable product, and the tour time of an ordinary vehicle is limited by the shelf life of the transported products, to ensure that the products do not deteriorate during transport. This constraint is essential to ensure that product quality is not compromised by overly long journeys, while maintaining reduced distribution costs. To solve the model, we apply both an exact method via the CPLEX solver [8] and a two-phase decomposition heuristic, based on a genetic algorithm [20]. The two methods are compared on solution quality and computation time criteria.

3. Description and Mathematical Formulation of The Model

3.1. Model description

The distribution model that we propose is a delivery network for perishable products, in which customers are classified as either nearby or distant, to optimise vehicle routes (see Fig.1). Nearby customers are located within the area defined by the threshold distance r, while distant customers are located outside this area or on its periphery. In our distribution strategy, nearby customers are served by ordinary vehicles, while distant customers are served by refrigerated vehicles to help maintain product quality through temperature regulation. The temperature in refrigerated vehicles is sufficient to preserve the quality of perishable products. However, for products transported by ordinary vehicles, the tour time is limited by the shelf life of the products, to prevent deterioration in quality during transportation. This means that ordinary vehicles can only serve customers whose delivery can be made within a time limit of ρ from the depot. Neither ordinary nor refrigerated vehicles are owned by the company, and both are subject to fixed start-up costs. Each vehicle leaves the depot, makes its rounds, and returns after serving nearby or distant customers, respectively. Due to the high cost of using refrigerated vehicles, the number of refrigerated vehicles is limited to m. Additionally, each customer can only be served by one vehicle.

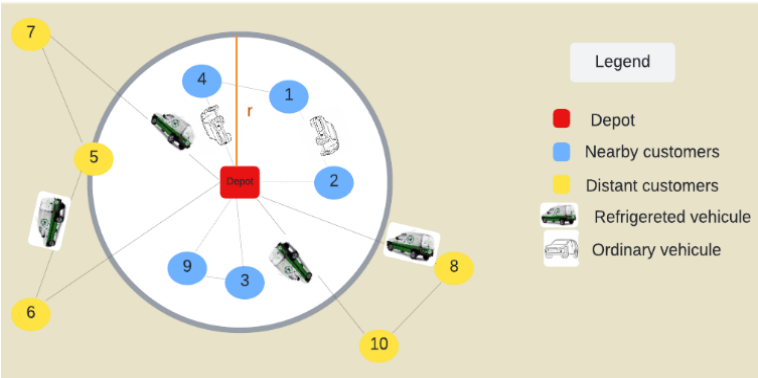

Fig. 1: Descriptive Diagram of the Distribution Network.

3.2. Mathematical problem formulation

The developed model uses a formulation based on the edges [8]and [5]. This edge modelling makes it possible to effectively represent the relationships between the different delivery points and the vehicles. Consider a non-directed graph G(V, A) where V is the set of vertices

and A is the set of edges. The set $V = \{0\} \cup C$, where 0 represents the depot and C represents the set of customers. Let r be the grouping constant. We define

$$B(0, r) = \{x \in C / \text{distance}(0, r) < r\},$$

The ball of radius r centered at O. Where "distance" is the usual distance defined on $\mathbb{R}$.

- Let be a customer $i \in C$. We say that i is a nearby customer of the depot if $i \in C$ and $\text{distane}(0, i) < r$. Let us denote C_p the set of nearby customers.
- Let be a customer $i \in C$. We say that i is a distant customer, if the client i It is not close to the depot. Let us denote C_e the set of all distant customers.
- Index
- i, j An An An Index representing customers.
- k index representing vehicles.
- (i, j) the edge between clients i and j

Parameters

- c_{ij}^1: the cost of transporting a vehicle travelling on an edge (i, j) with $i, j \in C_p$.
- c_{ij}^2: the cost of transporting a vehicle travelling on an edge (i, j) with $i, j \in C_e$.
- t_{ij}: the travel time of an ordinary vehicle travelling on an edge (i, j) with $i, j \in C_p$.
- d_i: request from customer i.
- K: the number of ordinary vehicles.
- Q_1: capacity of a regular vehicle.
- CG: fixed cost of starting up a regular vehicle.
- m: number of refrigerated vehicles.
- Q_2: capacity of a refrigerated vehicle.
- CF: fixed cost of starting up a refrigerated vehicle.
- ρ: the maximum shelf life of all product properties if transported by an ordinary vehicle.
- r: is the clustering constant.

Decision variables

- t_k: the total tour time of an ordinary vehicle k.
- binary variables:
- x_{ijk}^1 Is 1 if the ordinary vehicle k crosses the edge (i, j); 0 otherwise.
- S_{ik}^1 Is 1 if an ordinary vehicle k visits a customer i; 0 otherwise.
- x_{ijk}^2 Is 1 if the refrigerated vehicle k crosses the edge (i, j); 0 otherwise.
- S_{ik}^2 Is 1 if a refrigerated vehicle k visits a customer i; 0 otherwise.

The goal is to minimize total costs while meeting capacity constraints. The mathematical formulation is given by:

$$\min \sum_{i \in C_p} \sum_{j \in C_p} \sum_{k=0}^{K} c_{ij}^1 x_{ijk}^1 + \sum_{i \in C_e} \sum_{j \in C_e} \sum_{k=0}^{m} c_{ij}^2 x_{ijk}^2 + \sum_{k=0}^{K} CG\, S_{0k}^1 + \sum_{k=0}^{m} CF\, S_{0k}^2 \quad (1)$$

Constraints

$$\sum_{k=0}^{K} S_{ik}^1 = 1, \forall i \in \{0, \ldots, |C_p|\}, \quad (2)$$

$$S_{ik}^1 \leq S_{0k}^1, \forall i \in \{0, \ldots, |C_p|\}, \forall k \in \{0, \ldots, K\}, \quad (3)$$

$$S_{0k+1}^1 \leq S_{0k}^1, \forall k \in \{0, \ldots, K-1\}, \quad (4)$$

$$\sum_{i=0}^{|C_p|} x_{ijk}^1 = S_{jk}^1, \forall j \in \{0, \ldots, |C_p|\}, \forall k \in \{0, \ldots, K\} \quad (5)$$

$$\sum_{j=0}^{|C_p|} x_{ijk}^1 = S_{ik}^1, \forall i \in \{0, \ldots, |C_p|\}, \forall k \in \{0, \ldots, K\} \quad (6)$$

$$\sum_{i=0}^{C_p} d_i S_{ik}^1 \leq Q_1 S_{0k}^1, \forall k \in \{0, \ldots, K\} \quad (7)$$

$$T_k = \sum_{i=0}^{C_p} \sum_{j=0}^{C_p} t_{ij} x_{ijk}^1 \leq \rho, \forall k \in \{0, \ldots, K\} \quad (8)$$

$$u_{ik} - u_{jk} + (|C_p|) x_{ijk}^1 \leq |C_p| - 1, \forall i, j \in \{1, \ldots, |C_p|\}, \forall k \in \{1, \ldots, K\} \quad (9)$$

$$\sum_{k=0}^{m} S_{0k}^2 \leq m, m: \text{integer} \quad (10)$$

$$\sum_{k=0}^{m} S_{ik}^2 = 1, \forall i \in \{0, \ldots, |C_e|\} \quad (11)$$

$$S_{ik}^2 \leq S_{0k}^2, \forall i \in \{0, \ldots, |C_e|\}, \forall k \in \{0, \ldots, m\} \quad (12)$$

$$\sum_{i=0}^{|C_e|} x_{ijk}^2 = S_{jk}^2, \forall j \in \{0, \ldots, |C_e|\}, \forall k \in \{0, \ldots, m\} \quad (13)$$

$$\sum_{j=0}^{|C_e|} x_{ijk}^2 = S_{ik}^2, \forall i \in \{0, \dots, |C_e|\}, \forall k \in \{0, \dots, m\} \quad (14)$$

$$\sum_{i=0}^{|C_e|} d_i S_{ik}^2 \leq Q_2 S_{0k}^2, \forall k \in \{0, \dots, m\} \quad (15)$$

$$u_{ik} - u_{jk} + (|C_e|)x_{ijk}^2 \leq |C_e| - 1, \forall i, j \in \{1, \dots, |C_e|, \forall k \in \{1, \dots, m \quad (16)$$

$$T_k \geq 0, \forall k \in \{1, \dots, K \quad (17)$$

$$x_{ijk}^1 \in \{0,1\}, \forall i, j \in \{0, \dots, |C_p|\}, \forall k \in \{0, \dots, K\} \quad (18)$$

$$S_{ik}^1 \in \{0,1\}, \forall i \in \{0, \dots, |C_p|\}, \forall k \in \{0, \dots, K \quad (19)$$

$$x_{ijk}^2 \in \{0,1\}, \forall i, j \in \{0, \dots, |C_e|\}, \forall k \in \{1, \dots, m\} \quad (20)$$

$$S_{ik}^2 \in \{0,1\}, \forall i \in \{0, \dots, |C_e|\}, \forall k \in \{1, \dots, m\} \quad (21)$$

Constraints 2 and 11 ensure that a client is served by only one vehicle. The constraints 3 and 12 mean that a vehicle k can only visit the client. i If it leaves the repository. Constraint 4 maintains order in the output of vehicles. Thanks to constraints 5, 6, 13, and 14, we are reassured that each vehicle that visits a customer will leave this customer after it is served. The constraints 7 and 15 maintain the capacity condition. Constraint 8 ensures that the total tour time of an ordinary vehicle must not exceed a certain threshold. ρ To preserve the quality of the products. The constraints 9 and 16 are sub-tower elimination constraints Miller-Tucker-Zemlin (MTZ) [11]. Indeed, a subtour is a cycle that does not pass through the depot. This means that a vehicle could travel in a loop between several customers without returning to the depot. However, this is prohibited because all routes must start and end at the depot. To ensure this, specific constraints are added to the mathematical model. The constraints 17 - 21 define the decision variables.
Now that the model has been clearly defined, we will present the resolution methodologies adopted to evaluate its performance, both in terms of the quality of the solutions obtained and the computing time required.

4. Methodology

4.1. Customer segmentation

Consider a set C of 20 Customers. That is to say $C = \{1,2,\dots,20\}$.The d_{0i} Distances from the Customers to the depot are assumed to be as follows:

$d_{01} = 2.5, d_{02} = 9.8, d_{03} = 5.1, d_{04} = 12.3, d_{05} = 7.6,$

$d_{06} = 3.2, d_{07} = 10.1, d_{08} = 6.8, d_{09} = 13.5, d_{010} = 4.9,$

$d_{011} = 8.2, d_{012} = 11.7, d_{013} = 1.9, d_{014} = 9.3, d_{015} = 5.7,$

$d_{016} = 14.1, d_{017} = 7.1, d_{018} = 3.8, d_{019} = 10.9, d_{020} = 6.3.$

Let's take $r = 7.64$. Nearby customers C_p are those for which $d_{0i} < r$, and distant Customers C_e are those for which $d_{0i} \geq r$. As a result, the Nearby customers are:

$C_p = \{1, 3, 5, 6, 8, 10, 13, 15, 17, 18, 20\}$

And distant customers are:

$C_e = \{2, 4, 7, 9, 11, 12, 14, 16, 19\}$

Algorithm 1 is the determination of nearby customers and distant customers algorithm.
The graphical representation is given in Fig.2

Algorithm 1: Determining nearby customers and distant customers			
Require sef lei Co d_{0i}: Distances to depot.			
Ensure C_p: Nearby Customers, C_e: Distant Customers.			
1	. $C_p \leftarrow \emptyset$		
2	. $C_e \leftarrow \emptyset$		
3	for all $i \in C$ do		
4		If $d_{0i} < r$ then	
5			. $C_p \leftarrow C_p \cup \{i\}$
6		else	
7			$C_e \leftarrow C_e \cup \{i\}$..
8		end if	
9	end for		
10	return C_p, C_e		

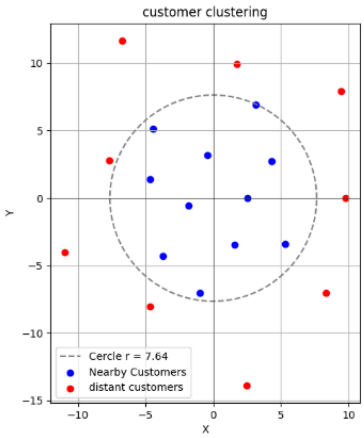

Fig. 2: Customer Grouping.

In the following subsection, we assume that customers are already grouped into nearby customers and distant customers, according to the principle defined above.

4.2. Two-phase decomposition heuristic for model resolution (TPDH)

The decomposition heuristic used in this work to solve the CVRP by distinguishing between nearby and distant customers was inspired by [20] and [18].

1) Phase 1

In the first phase, we develop Algorithm 2, which consists of decomposing the capacity-constrained vehicle routing problem (VRP) into several TSP problems, grouping customers on routes while respecting the capacity limits of the vehicles. Indeed, this algorithm takes as input a list of customers (Customers), their respective requests (d), the maximum capacity of the vehicles (Q), a list of routes (TT), and a matrix of distances between customers (From). The algorithm iterates if there are still customers to visit. For each route, it initializes an empty route, sets the current capacity to zero, and adds the depot as a starting point. Then, it randomly selects a first customer by shuffling the customer clues and choosing the first from the shuffled list. If the addition of this customer does not exceed the vehicle's capacity, it adds them to the route, updates the current capacity and the last customer visited, and removes them from the list of remaining customers. Then, it adds the next customers by choosing each time the customer closest to the last customer visited [6], as long as the addition does not exceed the capacity of the vehicle. If the addition of a customer exceeds capacity or there are no more customers to add, the tour is over. The full route is added to the route list (TT), and the algorithm continues until all customers have been assigned to a route. This algorithm works well for the generation of routes for refrigerated vehicles and ordinary vehicles. But for ordinary vehicles, in addition to the capacity constraint, we consider constraint 8.

```
Algorithm 2: Aggregation
Procedure AGREGATION (Clients, d, Q, TT, D_p)
While Non-Empty Clients do
    Tour leftTour leftleftleftTour lefteftleftTour leftteftleftTour← []
    q ← 0
    Add depot (0) to Tour
    Select a random initial customer.
    If charge ≤ Q,  then
        Add the initial client to the Tour.
        Update Load q
        Remove the initial client from Clients.
        While Non-Empty Clients
            Find from Clients, the closest to the last client added.
            If the customer found then
                If adding a nearby customer is possible, then
                    Add nearby client to Tour
                    Update Load q
                    Remove the initial client from Clients.
                else
                    break
                end if
            else
                break
            end if
        end while
        If the Tour contains at least one customer, then
            Add Tour to TT
        end if
    end if
end while
end procedure
```

2) Phase 2

In the subsequent phase, a genetic Algorithm 3 is developed to enhance the solutions to each TSP that have been obtained in phase 1. The genetic algorithm is a computational optimization method that draws inspiration from the process of natural evolution. The process

commences with an initial population of potential solutions, which are evaluated according to a fitness function that measures their quality. The optimal solutions are selected as parents for the subsequent generation, where they are combined by crossbreeding to create new individuals. In this instance, a selection by wheel and a one-point crossover were utilised. Subsequent to a period of remediation, the solutions are evaluated and the children are categorised. Subsequently, a random mutation is applied to introduce diversity. The assessment, sorting, and replacement of mutated children is a process that involves the replacement of a proportion of the existing population. This iterative process is characterised by a recurrent reassessment and categorisation of the population with each generation. The termination of the algorithm is initiated by a stopping criterion, which is based on either convergence or the number of generations. The optimal solution identified in the previous generation is then reversed.
The genetic algorithm is described as follows:

```
Algorithm 3: Genetic Algorithm
1   Procedure GENETICALGORITHM (vec1, vec2)
2       population ← generatePopulation(vec1, pop_size)
3       EVALUATE (population, vec2)
4       SORTDESCENDING (population)
5       . backkopu. lationackopulationack(). back()
6       Whilia do
7           Left ← select(population)
...8        CROSSOVER(population, population, p_c)
9           REPAIR(population0)
10          MUTATE(population0, p_m)
11          REPLACE(population0)
12          currentobjvapulation. back(). back()
13          If currentobjval then
14                  + 1
15          else
16                  objval ← cuentobjval
17                  Crit ← 0
18          end if
19          population0. clear()
20      end while
21      Inationback()
22      population. clear()
23  end procedure
```

The table below presents the parameters of the genetic algorithm used to solve the optimization problem.

Table 1: Parameters of Genetic Algorithm

Population size	100
Number of generations	150
Max Criteria	20
Crossover Rate	0.8
Mutation rate	0.1

Building on this methodology, we now present the experimental results.

5. Experimentation

The experiments were conducted on a series of well-known test instances from the literature. Indeed, one of the sets of instances developed by Augerat [4] in the literature on the CVRP, namely the P set, is used for these tests. All tests were conducted in C++ on a PC with an Intel Core. i $5-6300U(2.40\ GHz)$ Processor and 8.00 GB of RAM. We used IBM CPLEX's Concert Technology with the C++ language to solve the problem exactly. The resolution by the decomposition heuristic was done only with the C++ programming language. We report computation time and limits to show the performance of our algorithms. There are no instances in the literature that are adapted to the proposed model. Therefore, we modify instances of the P set that were originally designed for CVRP problems. Each instance contains a set of customers, their contact information, their requests, and the vehicles that will serve them. All instances assume that all vehicles are of the same type. However, just like Barma et al (2022) [5], we did not consider vehicle information, as our study considers two different types of vehicles. For example, the P_n60_k10 test instance has 60 customers with their contact information and specifies 10 vehicles to serve them as part of the problem of vehicle routes with capacity. In our study, we only used the contact details of the 60 customers and their requests. In addition, the 10 vehicles of the body are considered ordinary vehicles and we consider 10 refrigerated vehicles of the same capacity. The 60 customers are divided into close customers and distant customers according to their proximity to the depot. For testing, we set ρ at 150-time units. Each execution of our algorithm for resolving an instance was given a time limit of 45000 seconds. The cost of starting up a refrigerated vehicle is $CF = 100$ And the cost of starting up an ordinary vehicle is $CG = 50$.

6. Computational Results

We resolved the P instances with CPLEX. Click the link below to download these instances: http://www.vrp-rep.org/datasets/item/2014-0009.html.

6.1. Sensitivity analysis: choice of r

During our study, we calculated the average distance to the depot r_0 For a significant proportion of instances, P, given by the following formula:

$r_0 = \frac{1}{|C|}\sum_{i\in C} d_{0i}$

Next, we defined an interval around the value. r_0 .That is to say, the interval was defined as: $[r_0 - 50\%; r_0 + 50\%]$. A selection of 9 values of r was made within the range specified earlier. This approach was undertaken for each instance to determine the values of r that achieve an optimal balance between costs and resolution time. These values have been chosen as follows: the first 4 values are between $r_0 - 50\%$ and r_0, the 5^{th} value is r_0 And the last 4 values are between r_0 and $r_0 + 50\%$. The results of this experiment are reported in Table 2. This table shows the number of nearby customers (Cp), the number of distant customers (Ce), the transportation cost (Cost), the number of refrigerated vehicles used, the execution time (Time), and the Gap (%) for the 9 values of r considered for each instance. Examination of the results reveals that the parameter r Has a complex influence on the model's performance, with marked effects depending on the size of the instances. For small instances $n \leq 20$, such as P_n19_k2, using $r = r_0 = 24$ Yields an optimal solution in just 0.46 seconds with a single vehicle. For medium-sized instances $(20 < n \leq 40)$, such as P_n22_k8, using $r = r_0 = 25$ achieves the best compromise with 5 vehicles, a cost of 1336, and a reasonable computation time of 22.165 Seconds.
Very large instances $(n > 40)$ are more sensitive, requiring lower values of r $(20 - 22)$ To maintain their stability. However, it has been observed that when the parameter r is large (greater than 30), critical behaviours appear. These manifest themselves in a significant increase in computation time when the instance is. P_n16_k8 reaches 27335s with $r = 33$, and in the appearance of significant gaps, notably 22.37% for P_n23_k8 with $r = 33$. In addition, convergence failures were recorded for certain instances, such as P_n40_k5 With a parameter r equal to 33. This is because r values that are too large encourage extreme solutions, leading to unsatisfactory results, such as deficits or excessive computation times to achieve feasibility. When r is too low (less than 20), costs are generally higher. For example, for instance P_n16_k8, the cost is 1200 with a parameter $r = 13$. This increase in costs is directly related to an increase in the number of refrigerated vehicles required, due to the increased dispersion of distant customers. Convergence failures are also observed for certain instances, such as P_n23_k8 with a parameter r equal to 15 and P_n40_k5 With a parameter r equal to 13. These observations highlight the existence of an optimal range for r, generally located around $\pm 25\%$ of the reference value r_0. To justify this, a heatmap of the cost-time trade-off as a function of the value of r was created. It reveals interesting and contrasting trends depending on the instances. Overall, the darker areas (scores close to 0) correspond to the best balance between cost and execution time, while the lighter areas (scores close to 1) indicate unbalanced solutions, often characterised by very high cost or time. The scores are calculated as follows:

$$\text{Balanced score} = \lambda.\frac{\text{Cost}}{\text{Max cost}} + (1-\lambda).\frac{\text{Time}}{\text{Max time}}$$

- $\lambda \in [0,1]$ Is the weight of the cost and $1 - \lambda$ Is the weight of the time.
- The values are normalised so that they can be compared with each other.
- To be fair, we take: $\lambda = 0.5$

It has been observed that, in most cases, the most satisfactory compromises are found in the indicated area, thus validating the relevance of this choice. In other words, solutions that offer an optimal balance between cost and computation time, with relatively small delays and optimal use of resources (limited number of vehicles), are concentrated in this area. As illustrated in Fig.4, a graphical representation of the cost-time trade-offs for five instances is presented. The blue stars indicate the best trade-offs. However, some areas contain clear values even around the optimal r, suggesting that for some instances, there is significant variability in performance. The presence of NaN illustrates combinations that have not been evaluated or are not viable. Based on this analysis, a consolidation process was undertaken to identify an overall compromise range. This range should contain at least one good compromise point for each instance, while remaining narrow enough to ensure a certain degree of consistency. This process led to the selection of the range. $[20, 27]$, which is the smallest range in which each of the instances tested has at least one satisfactory solution in terms of cost and time. An illustration is provided with four instances in Fig.3. This global zone thus appears to be a strategic balance, making it possible to limit excessive use of resources (refrigerated vehicles), avoid excessive calculation times, and maintain an acceptable solution quality. It therefore constitutes a robust and consistent basis for decision-making for multi-instance implementation and favourably guides the choice of r. Thus, the analysis shows that, for all the instances tested, r_0 lies in the interval $[20, 27]$. Furthermore, it appears that, for the majority of these instances, $r = r_0$ Provides an optimal balance between cost and time. Consequently, it can be concluded that r_0 It is an ideal value for the parameter r.
In the remainder of this study, we will assume that, for each test sample., $r = r_0$

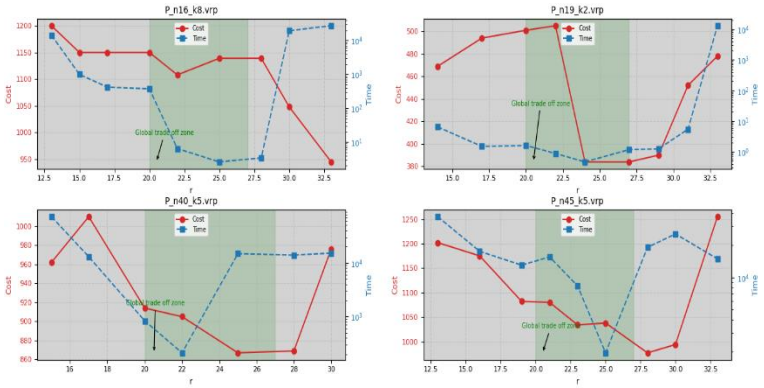

Fig. 3: Cost and Time Depending on r: Unified Compromise Zone for All Instances.

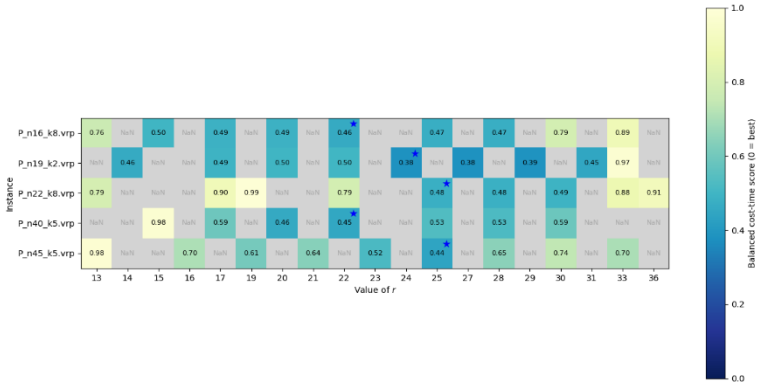

Fig. 1: Representation of Cost-Time Trade-Offs as a Function of r for 5 Instances.

Table 2: Sensitivity Analysis of r

Instances	r	C_p	C_e	Cost	number VF	Time	GAP (%)
	13	2	14	1200	7	14397.61	0.00%
	15	3	13	1150	6	998.01	0.00%
	17	3	13	1150	6	419.73	0.00%
	20	3	13	1150	6	374.08	0.00%
Instance P_n16_k8.vrp	r_0=22	6	12	1108	5	6.41	0.00%
	25	8	8	1139	4	2.59	0.00%
	28	8	8	1139	4	3.41	0.00%
	30	10	6	1048	3	19477.21	0.00%
	33	15	1	945	1	27335.02	13.00%
	14	2	17	469	2	6.38	0.00%
	17	4	15	494	2	1.48	0.00%
	20	5	14	501	2	1.58	0.00%
	22	6	13	505	2	0.87	0.00%
Instance P_n19_k2.vrp	r_0=24	9	10	384	1	0.46	0.00%
	27	9	10	384	1	1.16	0.00%
	29	10	9	390	1	1.23	0.00%
	31	12	7	452	1	5.40	0.00%
	33	16	3	478	1	13015.23	0.00%
							0.00%
	14	2	18	482	2	14.70	0.00%
	17	4	16	505	2	2.71	0.00%
	19	4	16	505	2	2.56	0.00%
	22	6	14	512	2	1.22	0.00%
Instance P_n20_k2.vrp	r_0=24	9	11	384	1	0.28	0.00%
	27	9	11	384	1	0.30	0.00%
	29	10	10	390	1	0.46	0.00%
	31	13	7	465	1	63.97	0.00%
	33	17	3	481	1	13094.17	1.51%
	14	3	18	486	2	21.04	0.00%
	17	5	16	505	2	2.00	0.00%
	19	5	16	505	2	2.01	0.00%
	21	5	16	505	2	1.98	0.00%
Instance P_n21_k2.vrp	r_0=24	10	11	386	1	0.25	0.00%
	25	10	11	386	1	0.34	0.00%
	28	10	11	386	1	0.39	0.00%
	30	12	9	401	1	1.32	0.00%
	32	15	6	480	1	580.33	0.00%
							0.00%
	14	3	19	490	2	11.87	0.00%
	17	5	17	509	2	6.03	0.00%
	19	5	17	509	2	5.99	0.00%
	21	5	17	509	2	6.03	0.00%
Instance P_n22_k2.vrp	r_0=23	9	13	535	2	0.83	0.00%
	25	10	12	538	2	0.51	0.00%
	28	11	11	390	1	0.57	0.00%
	30	13	9	473	1	1122.76	0.00%
	32	16	6	481	1	1202.38	0.00%
							0.00%
	13	4	18	1392	7	13005.16	0.00%
	17	5	17	1369	6	18128.09	0.00%
	19	6	16	1354	6	22177.83	0.00%
	22	7	15	1386	6	13003.32	0.00%
Instance P_n22_k8.vrp	r_0=25	10	12	1336	5	22.165	0.00%
	28	10	12	1336	5	184.21	0.00%
	30	12	10	1340	5	449.94	0.00%
	33	16	6	1306	3	18046.09	12.6%
	36	17	5	1270	3	20105.00	13.29%

	13	4	19	xxx	xxx	xxx	xxx
	15	5	18	xxx	xxx	xxx	xxx
	17	6	17	1352	7	13002.15	0.00%
	20	6	17	1352	7	13415.51	0.00%
Instance P_n23_k8.vrp	r_0=22	8	15	1309	6	61.267	0.00%
	25	11	12	1260	5	157.96	0.00%
	28	12	11	1177	4	16238.35	0.00%
	30	14	9	1251	4	26729.02	13.63%
	33	20	3	983	1	25532.46	22.37%
	13	6	34	xxx	xxx	xxx	xxx
	15	8	32	962	4	72277.22	0.00%
	17	10	30	1010	4	13003.75	0.00%
	20	14	26	914	3	834.49	0.00%
Instance P_n40_k5.vrp	r_0=22	16	24	905	3	211.604	0.00%
	25	21	19	867	2	14814.89	3.48%
	28	24	16	869	2	13930.42	5.37%
	30	28	12	976	2	15180.02	13.02%
	33	35	5	xxx	xxx	xxx	xxx
	13	6	39	1202	5	37229.6650	0.00%
	16	9	36	1175	4	17557.9410	0.00%
	19	14	31	1082	4	13013.0510	0.00%
	21	14	31	1080	4	15566.4040	0.00%
Instance P_n45_k5.vrp	r_0=23	19	26	1034	3	8300.398	2.42%
	25	21	24	1038	3	1942.4410	4.74%
	28	25	20	977	2	19156.1470	18.56%
	30	29	16	994	2	25610.9270	17.00%
	33	38	7	1255	1	14922.4530	41.77%

7. Results with CPLEX

In the case of the 24 instances of Issue P, a total of 19 were resolved within the specified timeframe, thereby achieving a success rate of 76.17%. Of these solutions, 9 correspond to proven optimals, while the remaining 10 are suboptimal solutions. It was not possible to resolve the remaining 5 instances. Table 3 displays the results for instances ranging from 16 to 76 customers. The table presents the following information for each instance: the optimal transportation cost identified (Cost); the T_k Transit times of the ordinary vehicles responsible for the deliveries of nearby customers; the compute time (Time CPU) in seconds; and the relative deviation (Gap%) obtained by CPLEX. It is evident from the data that the solver is capable of identifying optimal solutions for specific instances within the stipulated timeframe. Moreover, it has been demonstrated that, in each instance, the route time of each standard vehicle remains less than ρ, thereby ensuring the quality of the product upon its arrival at the customer's location. The results reveal significant trends in CPLEX performance on the P instances of the vehicle routing problem. For small instances ($n \leq 23$), CPLEX demonstrates remarkable efficiency, finding optimal solutions in very reasonable computation times ranging from 0.250s to 61.267s with zero gaps, while strictly respecting the time constraint $T_k < \rho$ For each vehicle. However, complexity increases exponentially with instance size. Indeed, for n ranging from 40 to 76 clients, the computation times explode from 211s to 40736s And the gaps increase to 23.54% for P_n55_k15, revealing the limits of the exact approach. We also observe that instances with a high number of vehicles. ($k \geq 8$) Pose particular difficulties, as evidenced by the gap of 20.77% for P_n65_k10 Requiring more than 11 hours of computation. Interestingly, some configurations like P_n76_k4 Have moderate gaps of 4.43% Despite their size, suggesting that the difficulty depends as much on the structure as on the sheer dimension of the problem. All solutions consistently adhere to transit time constraints, validating operational feasibility, but the inability to resolve 5 instances and the high gaps for $n \geq 50$ Underscore the need for alternative approaches for large problems, while confirming the suitability of CPLEX for small instances where optimality is crucial. It is precisely to respond to these limitations that we have developed the TPDH. In the following section, we compare the results obtained with CPLEX and those obtained with TPDH.

Table 3: Exact Solution with CPLEX

INSTANCES	Cost	Tour time T_k	Time CPU	GAP
Instance P_n16_k8.vrp	1108	T_0 =42<ρ T_1 =42<ρ T_2 =24<ρ	6.4100	0.00%
Instance P_n19_k2.vrp	384	T_0 =92<ρ	0.460	0.00%
Instance P_n20_k2.vrp	384	T_0 =92<ρ	0.284	0.00%
Instance P_n21_k2.vrp	386	T_0 =75<ρ	0.250	0.00%
Instance P_n22_k2.vrp	535	T_0 =88<ρ	0.827	0.00%
Instance P_n22_k8.vrp	1336	T_0 =53<ρ T_1 =65<ρ T_2 =98<ρ	22.165	0.00%
Instance P_n23_k8.vrp	1309	T_0 =35<ρ T_1 =30<ρ T_2 =53<ρ	61.267	0.00%
Instance P_n40_k5.vrp	905	T_0 =70<ρ T_1 =79<ρ	211.604	0.00%
Instance P_n45_k5.vrp	1034	T_0 =87<ρ T_1 =98<ρ	8300.398	2.42%

		T_2 =16<ρ		
Instance P_n50_k7.vrp	1147	T_0 =58<ρ T_1 =66<ρ T_2 =43<ρ	12262.8250	0.00%
Instance P_n50_k8.vrp	1482	T_0 =43<ρ T_1 =58<ρ T_2 =49<ρ T_3 =43<ρ	29398.66	10.04%
Instance P_n50_k10.vrp	xxx	xxx	xxx	xxx
Instance P_n51_k10.vrp	1675	T_0 =40<ρ T_1 =68<ρ T_2 =62<ρ T_3 =57<ρ T_4 =61<ρ	26313.516	14.17%
Instance P_n55_k7.vrp	1276	T_0 =34<ρ T_1 =66<ρ T_2 =76<ρ	27340.231	7.19%
Instance P_n55_k8.vrp	1541	T_0 =69<ρ T_1 =52<ρ T_2 =57<ρ	34241.975	7.00%
Instance P_n55_k10.vrp	xxx	xxx	xxx	xxx
Instance P_n55_k15.vrp	2455	T_0 =45<ρ T_1 =66<ρ T_2 =51<ρ T_3 =63<ρ T_4 =61<ρ T_5 =47<ρ	30654.186	23.54%
Instance P_n60_k10.vrp	1874	T_0 =58<ρ T_1 =64<ρ T_2 =72<ρ T_3 =66<ρ	28646.203	17.60%
Instance P_n60_k15.vrp	xxx	xxx	xxx	xxx
Instance P_n65_k10.vrp	1957	T_0 =76<ρ T_1 =53<ρ T_2 =38<ρ T_3 =66<ρ T_4 =45<ρ	40736.22	20.77%
Instance P_n70_k10.vrp	xxx	xxx	xxx	xxx
Instance P_n76_k4.vrp	1091	T_0 =133<ρ T_1 =87<ρ	26526.206	4.43%
Instance P_n76_k4.vrp	1201	T_0 =128<ρ T_1 =98<ρ T_2 =54<ρ	15054.396	17.80%
Instance P_n101_k4.vrp	xxx	xxx	xxx	xxx

8. Discussion

Table 4 gives the results of solving the model with the CPLEX solver and those of the decomposition heuristic. The columns provide information about the instances being tested, the costs of the solutions found by Cplex(Cost(Cplex)), the costs of the solutions found by the TPDH (cost(TPDH)), the percentage difference (Gap(cost)) between these costs, the computation times required by each method(time(Cplex), time (TPDH)), and the percentage difference between these times((Gap(time))). The comparison uses instances that produced results with the CPLEX solver. The cost gap between the CPLEX and the TPDH is given by the relation:

$$\text{Gap(cost)} = 100 \times \frac{\text{cost(TPDH)} - \text{Cost(Cplex)}}{\text{Cos(Cplex)}}$$

And the time gap between the CPLEX solver and the TPDH is given by the relationship:

$$\text{Gap (time)} = 100 \times \frac{\text{time(TPHD)} - \text{time(Cplex)}}{\text{time(Cplex)}}$$

Indeed, the table highlights significant trends in the comparative performance of the CPLEX exact solver and the TPD heuristic. For small instances ($n \leq 23$),demonstrates superiority in terms of computation time, with times ranging from 0.25s to 211.61s compared to 157.93s to 608.98s for TPDH. CPLEX also demonstrates superiority in terms of solution quality, with an average cost difference of 7%, confirming its suitability for small problems where optimality is crucial. However, its performance deteriorates drastically for larger instances (i.e., with execution times reaching 34,241.98s for P_n55_k7.vrp due to the combinatorial explosion characteristic of NP-hard problems. This is clearly visible in Graph 2 of Fig. 5. Conversely, in most cases, the resolution costs obtained by TPDH are quite close to those obtained using CPLEX (see Graph 1 in Fig. 5. The average cost of solving with CPLEX is 1,214.74, while the average cost of the decomposition

heuristics is 1,271.84. This represents an average cost gap of 4.70%, indicating that the heuristic approach produces solutions that are similar to those found by the CPLEX solver. TPDH also demonstrates remarkable robustness for medium to large instances, achieving resolution times that are, on average, 93.37% faster and exhibiting cost spreads ranging from −12.07% to +14.51%, which demonstrates an excellent trade-off between cost and time. TPDH's occasional exceptional performance in certain instances, such as the negative cost variance for P_n50_k5.vrp, is due to CPLEX being limited by certain configurations, resulting in inefficient cuts or cumulative rounding errors. TPDH's underperformance in certain instances, such as the positive gap of 43.45% for P_n76_k4.vrp, suggests avenues for improvement through integrating metaheuristics or machine learning techniques to guide the search. From a practical point of view, these results suggest an adaptive hybrid approach consisting of using CPLEX for small critical issues and TPDH for larger instances, while also considering alternative exact methods, such as branch-and-price, for intermediate cases. These considerations align with current supply chain challenges, where balancing solution accuracy, computation time, and energy sustainability is crucial, particularly in logistics contexts requiring fast, scalable decisions. TPDH's drastic reduction in computing time opens up interesting prospects for real-time applications and the optimisation of large fleets in sustainable urban contexts.

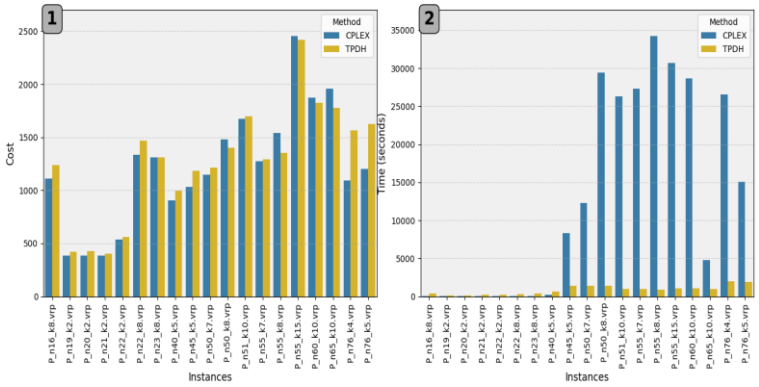

Fig. 5: Representation of CPLEX and TPDH Performance.

Table 4: Comparison of results between CPLEX and TPDH

INSTANCES	Cost (Cplex)	Cost (TPDH)	Gap (cost)	Time (Cplex)	Time (TPDH)	Gap (time)
Instance P_n16_k8.vrp	1108	1236	11.55%	6.41	339.71	5199.69%
Instance P_n16_k8.vrp	384	418	8.85%	0.46	157.931	34232.83%
Instance P_n19_k2.vrp	384	426	10.94%	0.284	163.336	57412.68%
Instance P_n21_k2.vrp	386	403	4.40%	0.25	178.844	71437.60%
Instance P_n22_k2.vrp	535	561	4.86%	0.827	183.52	22091.05%
Instance P_n22_k8.vrp	1336	1469	9.96%	22.165	262.22	1083.04%
Instance P_n23_k8.vrp	1309	1310	0.08%	61.267	351.399	473.55%
Instance P_n40_k5.vrp	905	997	10.17%	211.604	608.977	187.79%
Instance P_n45_k5.vrp	1034	1184	14.51%	8300.398	1362.2	-83.59%
Instance P_n50_k7.vrp	1147	1216	6.02%	12262.825	1416.48	-88.45%
Instance P_n50_k8.vrp	1482	1403	-5.33%	29398.66	1362.02	-95.37%
Instance P_n51_k10.vrp	1675	1695	1.19%	26313.516	1007.58	-96.17%
Instance P_n55_k7.vrp	1276	1292	1.25%	27340.231	931.681	-96.59%
Instance P_n55_k8.vrp	1541	1355	-12.07%	34241.975	872.034	-97.45%
Instance P_n55_k15.vrp	2455	2416	-1.59%	30654.186	1056.65	-96.55%
Instance P_n60_k10.vrp	1874	1823	-2.72%	28646.203	1051.92	-96.33%
Instance P_n65_k10.vrp	1957	1774	-9.35%	4736.22	948.126	-79.98%
Instance P_n76_k4.vrp	1091	1565	43.45%	26526.206	2015.34	-92.40%
Instance P_n76_k5.vrp	1201	1622	35.05%	15054.396	1883.91	-87.49%
Average	1214.74	1271.84	4.70%	12830.425	850.204	-93.37%

9. Conclusion

This innovative model for the distribution of perishable products provides an operational response to complex logistics challenges, reconciling product quality, guaranteed by the maximum transit time. ρ, and economic efficiency through the optimization of ordinary and refrigerated vehicles. The results demonstrate the complementarity of the CPLEX approaches for small instances with less than 40 customers and the TPDH heuristic for large instances with more than 40 customers, with gains of up to 93.37% In computation time. Beyond algorithmic performance, this model offers major strategic implications, such as the reduction of food waste through strict control of delivery times, the reduction of the carbon footprint through the rational use of energy-intensive refrigerated vehicles. Future research prospects should explore several promising avenues. First, extending the model to the management of multiple products with variable lifespans would increase its operational scope. Second, integrating contextual factors such as demand variability, traffic conditions, or time slots via dynamic or stochastic approaches seems to be a promising avenue. Finally, real-time route optimisation using artificial intelligence tools is a third particularly interesting area of study. It would also be relevant to develop inter-company pooling mechanisms based on cooperative game theory in order to reduce fixed costs, or to validate the model operationally on refrigerated electric fleets, thereby contributing to the link between logistics performance and ecological transition. All of these advances would position the model as an instrument of excellence for agri-food supply chains that are both resilient and sustainable.

References

[1] Abbasi-Tavallali, Pezhman, Mohammad Reza Feylizadeh, and Atefeh Amindoust. "A system dynamics model for routing and scheduling of cross-dock and transportation in reverse logistics network of perishable goods", *Journal of Intelligent \& Fuzzy Systems*, 40.6 (2021): 10417-10433. https://doi.org/10.3233/JIFS-200610.

[2] Amorim P, Almada-Lobo B, "The impact of food perishability issues in the vehicle routing problem", *Comput Ind Eng*, 67 (2014):223–233. https://doi.org/10.1016/j.cie.2013.11.006

[3] Amorim P, Meyr H, Almeder C, Almada-Lobo B, "Managing perishability in production-distribution planning: a discussion and review", *Flexible Services and Manufacturing Journal*, 25 (2013):389-413. https://doi.org/10.1007/s10696-011-9122-3.

[4] Augerat, P. "VRP problem instances." http://www. branchandcut. Org/VRP/data/ (1995).

[5] BARMA, Partha Sarathi, DUTTA, Joydeep, MUKHERJEE, Anupam, et al, "A multi-objective ring star vehicle routing problem for perishable items", *Journal of Ambient Intelligence and Humanized Computing*, 2022, p: 1-26.

[6] Clarke, Geoff, and John W. Wright. "Scheduling of vehicles from a central depot to a number of delivery points." *Operations research*, 12.4 (1964): 568-581. https://doi.org/10.1287/opre.12.4.568.

[7] Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manag Sci*, 6(1):80–91. https://doi.org/10.1287/mnsc.6.1.80

[8] Dellaert, Nico, et al. "A multi-commodity two-Echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach", *Computers \& Operations Research*, 127 (2021): 105-154. https://doi.org/10.1016/j.cor.2020.105154

[9] Desport, Pierre."Tactical planning for closed-loop supply chains: modelling, resolution, evaluation" *Modelling and simulation*. University of Angers, 2017. French. ⟨NNT: 2017ANGE0012⟩. ⟨tel-01679886⟩.

[10] Eksioglu, B., Vural, A. V., \& Reisman, A. (2009), "The vehicle routing problem: A taxonomic review", *Computers & Industrial Engineering*, 57(4), 1472–1483. https://doi.org/10.1016/j.cie.2009.05.009.

[11] FOUILHOUX, Pierre, "Combinatorial Optimisation: Linear Programming and Algorithms". Pierre and Marie Curie University, 2015.

[12] FU, Zhuo, EGLESE, Richard, et LI, Leon YO, "A new tabu search heuristic for the open vehicle routing problem", *Journal of the operational Research Society*, 2005, vol. 56, no 3, p. 267-274. https://doi.org/10.1057/palgrave.jors.2601817

[13] GARCÍA-NÁJERA, Abel, BULLINARIA, John A., et GUTIÉRREZ-ANDRADE, Miguel A, "An evolutionary approach for multi-objective vehicle routing problems with backhauls", *Computers & Industrial Engineering*, 2015, vol. 81, p. 90-108. https://doi.org/10.1016/j.cie.2014.12.029.

[14] Goyal S, Giri B (2001), "Recent trends in modeling deteriorating inventory", *Eur J Oper Res*, 134(1):1–16. https://doi.org/10.1016/S0377-2217(00)00248-4.

[15] Haddadene, Syrine Roufaida Ait, Nacima Labadie, and Caroline Prodhon, "The VRP with time windows, synchronization and precedence constraints: Application in home health care sector." MOSIM 2014, 10th Francophone Conference on Modelling, Optimisation and Simulation. 2014

[16] Hsiao Y, Chen M, Chin C (2017), "Distribution planning for perishable foods in cold chains with quality concerns: formulation and solution procedure", *Trends Food Sci Technol* 61:80–9. https://doi.org/10.1016/j.tifs.2016.11.016

[17] Karaesmen, Itir Z., Alan Scheller–Wolf, and Borga Deniz. "Managing perishable and aging inventories: review and future research directions", *Planning production and inventories in the extended enterprise: A state of the art handbook*, Volume 1 (2010): 393-436. https://doi.org/10.1007/978-1-4419-6485-4_15

[18] KAYÉ, Bi Kouaï Bertin, DIABY, Moustapha, N'TAKPÉ, Tchimou, et al, "Managing an External Depot in a Production Routing Problem", *IJACSA*, 2020, vol. 11, p. 321-330. https://doi.org/10.14569/IJACSA.2020.0110242.

[19] Laporte, G. (2007), "What you should know about the vehicle routing problem", *Naval Research Logistics (NRL)*, 54, 811–819. https://doi.org/10.1002/nav.20261.

[20] LEI, Lei, LIU, Shuguang, RUSZCZYNSKI, Andrzej, et al, "On the integrated production, inventory, and distribution routing problem", *IIE Transactions*, 2006, vol. 38, no 11, p. 955-970. https://doi.org/10.1080/07408170600862688.

[21] Nahmias S (1982), "Perishable inventory theory: a review", *Oper Res* 30(4):680–708. https://doi.org/10.1287/opre.30.4.680.

[22] Raafat F (1991), "Survey of literature on continuously deteriorating inventory models", *J Oper Res Soc*42:27–37. https://doi.org/10.1038/sj/jors/0420103.

[23] SONG, Byung Duk et KO, Young Dae, "A vehicle routing problem of both refrigerated-and general-type vehicles for perishable food products delivery", *Journal of food engineering*, 2016, vol. 169, p. 61-71. https://doi.org/10.1016/j.jfoodeng.2015.08.027

[24] Tarantilis, C., & Kiranoudis, C. (2001), "A meta-heuristic algorithm for the efficient distribution of perishable foods", *Journal of Food Engineering*, 50 ,1–9. https://doi.org/10.1016/S0260-8774(00)00187-4

[25] Tarantilis, C., & Kiranoudis, C. (2002), "Distribution of fresh meat", *Journal of Food Engineering*, 51, 85–91. https://doi.org/10.1016/S0260-8774(01)00040-1

[26] Tirkolaee, Erfan Babaee, et al. "A robust green traffic-based routing problem for perishable products distribution",*Computational intelligence* 36.1 (2020): 80-101. https://doi.org/10.1111/coin.12240.

[27] Wee H (1993), "Increasing efficiency in the supply chain for short shelf-life goods using RFID tagging", *Comput Ind Eng*, 3(24):449–458.