Nigerian Journal of Operations Research (NIJOR)

https://nijor.org.ng/
pISSN: 2141-7938
eISSN: 2141-7938

Adaptation of Artificial Bee Colony Algorithm for Solving Inventory Routing Problem

Stephen Munu[1,*], Asaju Bolaji La-aro[1], Moshood Abiola Hambali[1]

[1] Department of Computer Science,Faculty of Computing and Information System, Federal University Wukari ,Nigeria

ARTICLE INFO

Article history:
Received 9 March 2026
Revised 14 April 2026
Accepted 27 April 2026
Available online 1 May 2026

Keywords:

Artificial Bee Colony; Inventory Routing Problem; Modified Artificial Bee Colony; RAODEF-EURO Challenge 2016; Optimization Model.

ABSTRACT

This study applies the Artificial Bee Colony (ABC) algorithm to the Inventory Routing Problem (IRP), using real-world data from the RAODEF-EURO Challenge 2016 on liquid-gas distribution. The research develops a feasible optimization model for IRP scenarios involving 1–13 drivers and 1–15 trailers delivering products from a central depot to 200–324 customers over planning horizons ranging from one week to one month. The model incorporates standard IRP constraints and aims to minimize the logistic ratio, defined as delivery cost relative to transported volume over time. Addressing IRP is essential in operations research, as it requires balancing varying customer demands with long-term distribution efficiency. Although previous heuristic and hyper-heuristic approaches achieved partial success, many struggled with scalability in large, constraint-intensive instances. To overcome these limitations, this study evaluates both the standard Artificial Bee Colony algorithm and a Modified Artificial Bee Colony (MABC) variant. Their performances are assessed using fifteen benchmark instances (Instance B). Results show that both algorithms generated feasible solutions across all datasets. However, the MABC consistently achieved lower logistic ratios and improved cost efficiency in most instances, outperforming the standard ABC in solution quality. The overall findings demonstrate that the MABC offers a more robust and scalable approach for solving complex IRP scenarios, making it a promising tool for practical logistics planning and decision-making.

1. Introduction

The global manufacturing and service environment is increasingly driven by the need to deliver high-quality products and services with maximum efficiency and reliability. Central to achieving this objective is the effective scheduling of human and logistical resources, which ensures continuity of operations through proper shift planning, staff allocation, and coordinated service delivery (Baraka and Yadavalli, 2022). At the same time, rapid digital transformation and advances in information technology have intensified the pursuit of integrated supply chain optimization, where data-driven decision making is used to minimize logistics costs while improving responsiveness and service quality. A core component of this transformation is inventory management, which focuses on

* *Corresponding author.*
E-mail address: munu@fuwukari.edu.ng

maintaining appropriate stock levels to guarantee timely product availability across the supply network.

The integration of inventory management with transportation and scheduling decisions gives rise to a highly complex optimization challenge known as the Inventory Routing Problem. The Inventory Routing Problem simultaneously determines quantity to deliver, when to deliver, and how to route to the delivery point of each customer, with the objective of minimizing total operational costs, including transportation, inventory holding, and shortage penalties, while satisfying time windows and capacity constraints. Inventory Routing Problem is defined as the concurrent optimization of inventory control, vehicle routing, and delivery scheduling (Cao *et al*., 2020). The Inventory Routing problem is particularly relevant in Vendor-Managed Inventory systems, where suppliers are responsible for managing customer stock levels (H *et al*., 2020). Due to its combinatorial nature, uncertainty, and multiple conflicting objectives, the Inventory Routing Problem remains one of the most challenging problems in supply chain optimization.

Research on the Inventory Routing Problem has produced major categories of solution approaches. Exact methods, such as Mixed-Integer Linear Programming and Branch-and-Cut, have been employed to obtain provably optimal solutions for well-structured problem instances (Absi *et al*., 2020; Diabat *et al*., 2021). However, these approaches often suffer from excessive computational time when applied to large-scale or real-time problems (Schenekemberg *et al*., 2020).

To address scalability, heuristic and metaheuristic techniques have been widely adopted. Local search-based methods, including Tabu Search and Variable Neighborhood Search, have been applied (Lee *et al*., 2022; Liu et al., 2021; Tiwari, and Sharma, 2023). Population-based metaheuristics such as Genetic Algorithms and Particle Swarm Optimization have been employed to the solution space more broadly by evolving multiple candidate solutions simultaneously (Chan *et al*., 2020)

More recently, hybrid and hyper-heuristic frameworks have gained attention. Matheuristic approaches combining mathematical programming with metaheuristics were developed (He et al., 2020; He *et a*l., 2024), while hyper-heuristic strategies that automate the selection of low-level heuristics were explored using a Hidden Markov Model in the analysis of the ROADEF/EURO 2016 Challenge, further highlighted the effectiveness of such advanced hybrid strategies (Kheiri, 2020; André, 2020).

Despite the strengths, each methodological class exhibits limitations at some stage. Exact methods are computationally expensive and become impractical for large, highly constrained, or dynamic instances. Local search heuristics, although fast, tend to become trapped in local optima due to their limited exploration capability. Population-based metaheuristics offer better global search behavior but have been insufficiently explored for large-scale, constraint-intensive Inventory Routing Problem benchmarks. Hybrid and hyper-heuristic approaches improve robustness but still struggle to consistently escape deep local optima, particularly for complex real-world instances such as the ROADEF/EURO 2016 Air Liquide dataset, as observed (Absi *et al*., 2020; He *et al*., 2024; Kheiri, 2020).

The critical gap identified in the literature is the lack of a sufficiently strong global exploration mechanism capable of effectively navigating the vast and rugged search space of large-scale, highly constrained Inventory Routing Problem instances. Existing approaches rely heavily on localized improvement strategies (Local optima), which limit their ability to discover high-quality solutions far from initial search regions.

To address this gap, the present study proposes the application of the Artificial Bee Colony algorithm, a population-based, nature-inspired metaheuristic known for its balanced exploration and exploitation capabilities. By mimicking the foraging behavior of honey bees, the Artificial Bee Colony algorithm enables diversified search through employed, onlooker, and scout bee phases, making it particularly suitable for escaping local optima and exploring complex solution landscapes. This study adapts the Artificial Bee Colony algorithm to the Inventory Routing Problem and evaluates its performance on the challenging ROADEF/EURO 2016 benchmark, with the aim of contributing a robust and scalable computational framework for solving large-scale, real-world scheduling and routing problems.

2. Literature Review

This review provides a comprehensive synthesis of recent and influential contributions to the Inventory Routing Problem literature, with particular emphasis on methodological developments between 2024 and 2025, dominant research directions, and a critical unresolved gap that motivates future investigation.

Recent years have witnessed rapid progress in Inventory Routing Problem modeling and solution techniques, driven by the increasing need to represent real-world complexities such as sustainability constraints, uncertainty, perishability, and large-scale operational settings. A notable trend is the emergence of exact, matheuristic, and hybrid metaheuristic frameworks that combine mathematical rigor with computational efficiency.

Within the class of exact and matheuristic approaches, Mundim *et al*., (2025) proposed a bi-objective heterogeneous fleet Inventory Routing Problem that explicitly integrates environmental considerations through a validated vehicular emission model. Their augmented epsilon-constraint method combined with a Branch-and-Cut algorithm successfully generated Pareto-efficient solutions for 285 instances, clearly quantifying the trade-off between economic cost and carbon dioxide emissions. Although the study provides strong evidence of the feasibility of environmentally sustainable logistics optimization, it also reveals a substantial cost premium associated with emission reduction and highlights potential scalability limitations of exact Branch-and-Cut schemes for very large or highly dynamic systems.

Addressing uncertainty and perishability, He *et al.*,(2024) developed a robust multi-period Inventory Routing Problem for deteriorating products, solved using Robust Optimization and Benders Decomposition for small instances, and an Iterated Local Search–Benders Decomposition hybrid for larger ones. The results demonstrate that robust formulations can significantly enhance solution stability and cost performance under demand variability. However, the computational burden of exact robust counterparts and the sensitivity of results to the definition of uncertainty sets remain important limitations, necessitating heuristic or decomposition-based approximations for practical-scale problems.

Berbiche *et al*., extended Inventory Routing Problem modeling to a highly complex three-echelon humanitarian logistics context, incorporating non-stationary demand, hostility-induced attrition, and vehicle commodity compatibility (Berbiche *et al.,* 2024). The approach employed Mixed Integer Linear Programming formulation that captures unprecedented operational realism. The resulting NP-hard structure recorded efficient performance without the need to integrate other advanced techniques such as Lagrangian Relaxation or Branch-and-Price, which were suggested but not fully implemented in the study.

Similarly, Laganà *et al*., (2024) introduced a matheuristic framework combining Column Generation with Iterated Local Search for the Multi-Vehicle Inventory Routing Problem. The approach achieved superior solution quality and faster convergence on benchmark instances compared with existing heuristics. Nonetheless, the exponential growth of columns and the computational intensity of local search procedures pose challenges for real-time or very large-scale applications.

Parallel to these developments, metaheuristic and hyper-heuristic strategies have continued to evolve, particularly for multi-objective and stochastic variants of the Inventory Routing Problem. Prakash *et al*., (2024) formulated a sustainable assembly Inventory Routing Problem under demand uncertainty and solved it using a Hybrid Non-Dominated Sorting Genetic Algorithm II. The model implemented simultaneously accounts for economic cost, carbon emissions, and social incentives, and the proposed algorithm outperformed a Speed-Constrained Multi-Objective Particle Swarm Optimization approach in generating high-quality Pareto fronts. Despite its effectiveness, the inherent computational expense of multi-objective evolutionary optimization and the managerial burden of selecting among non-dominated solutions remain open concerns.

Zheng *et al.,* (2024) proposed a three-phase stochastic solution framework combining Progressive Hedging, route splitting, and the Lin–Kernighan Heuristic for a recyclable Inventory Routing Problem with incentive mechanisms. The method utilized achieved substantial cost reductions relative to conventional Genetic Algorithm and Iteration-Move-Search approaches. However, its multi-stage structure requires careful parameter tuning and becomes computationally demanding for very large problem instances.

Haseltalab and Karimi, (2025) formulated a Mixed Integer Nonlinear Programming based Deteriorating Inventory Routing Problem and developed hybrid Genetic Algorithms enhanced with Tabu Search and greedy mechanisms. The Genetic Algorithm Tabu Search hybrid consistently produced near-optimal solutions and outperformed simpler variants in terms of solution quality, albeit at the expense of higher computational time, reflecting the intrinsic difficulty of Mixed Integer Nonlinear Programming formulations.

Across these studies, several dominant research themes clearly emerge. First, sustainability has become a central objective, with explicit modeling of carbon dioxide emissions and environmental penalties now routinely integrated into Inventory Routing Problem formulations, enabling quantitative assessment of cost emission trade-offs rather than purely conceptual discussions of green logistics. Second, advanced treatment of uncertainty through Robust Optimization and Stochastic Programming is increasingly prevalent, supported by new risk and service-level metrics such as the Service Violation Index. Third, perishability and product degradation are receiving focused attention, particularly in food, medical, and fuel supply chains, where shelf-life and wastage costs critically influence routing and inventory decisions. Finally, there is growing emphasis on dynamic and real-time decision policies, reflecting the operational need for adaptive routing and replenishment in volatile and information-rich environments.

Despite these methodological advances, a significant research gap persists. A striking disconnect exists between the development of sophisticated algorithms and their validation on widely recognized standardized benchmarks. In particular, few recent studies published after 2020 evaluate their approaches on the ROADEF–EURO 2016 Challenge dataset for Air Liquide gas distribution, which represents one of the most realistic and rigorously defined Inventory Routing Problem benchmarks, incorporating specific operational rules and the well-known logistic ratio performance metric. The early Branch-Cut-and-Price and hyper-heuristic techniques used as reviewed in this study, demonstrated competitive performance on the dataset specifically utilized by Absi *et al*., (2020) and Kheiri, (2020) shown in Table 1. This lack of benchmarking limits cross-study comparability and obscures the true practical effectiveness of recent algorithmic innovations. Consequently, there is a clear need for future research to bridge this gap by applying and evaluating modern exact, matheuristic, and metaheuristic frameworks on standardized Inventory Routing Problem benchmarks, particularly the ROADEF–EURO 2016 instances, in order to establish reproducible performance baselines and advance the field toward unified, evidence-based methodological assessment.

Table 1: Summary of Literature Review

Author	Year	Title	Algorithm	Limitation
Absi et al.	2020	*A heuristic branch-cut-and-price algorithm for the ROADEF/EURO challenge on Inventory Routing*	Heuristic Branch-Cut-and-Price	Difficulty balancing inventory and capacity in dynamic settings.
He et al.	2020	*A matheuristic with fixed-sequence reoptimization for a real-life inventory routing problem*	Matheuristic combining MIP/LP and randomized greedy search	Lack of efficiency with increasing numbers of routes
Kheiri, A.	2020	*Heuristic sequence selection for inventory routing problem*	Hyper-heuristic using Hidden Markov Model for heuristic selection	Lack of efficiency with increasing numbers of routes
Laganà et al.	2024	*An iterated local search matheuristic approach for the multi-vehicle inventory routing problem*	Matheuristic combining Column Generation and Iterated Local Search	Exponential growth of variables and computational intensity limit real-time large-scale use.
Berbiche et al.	2024	*An Integrated Inventory-Production-Distribution Model for Crisis Relief Supply Chain Optimization*	Mixed Integer Linear Programming	Model is NP-hard and computationally intractable without advanced decomposition methods.
Prakash et al.	2024	*An enhanced multiobjective inventory routing model to meet sustainable goals under uncertainty*	Hybrid Non-Dominated Sorting Genetic Algorithm II	Computationally intensive and yields multiple Pareto solutions requiring managerial selection.
Zheng et al.	2024	*Joint optimization of recyclable inventory routing problem under uncertainties*	Progressive Hedging, Route Splitting, and Lin–Kernighan Heuristic	Requires careful tuning and is computationally heavy for very large instances.
Haseltalab and Karimi	2025	*Optimizing the deteriorating inventory-routing problem with heterogeneous vehicles*	Genetic Algorithm with Tabu Search and Genetic Algorithm with Greedy heuristic	Higher solution quality comes with increased computational cost; MINLP structure is difficult.
Mundim et al.	2025	*Bi-objective green inventory routing with heterogeneous fleet and fuels*	Augmented ε-constraint and Branch-and-Cut	Large cost increase required to achieve significant emission reduction, raising economic concerns.

3. Methodology

3.1 Data and Problem Instance

This research utilizes benchmark datasets from the 2016 ROADEF/EURO Challenge (Instance B), comprising 15 distinct problem instances. These instances model real-world liquid-gas distribution, featuring a mix of vendor-managed inventory (VMI) and call-in customers, with varying numbers of drivers (1–13), trailers (1–15), and customers (200–324). The planning horizon ranges from one week to one month.

Stage 1: Data Processing. The optimization process begins with Data Processing, where the necessary inputs are prepared. This involves reading the problem instance data from the IRP dataset files (Instance B). This stage ensures that all operational parameters, constraints, and instance-specific data are correctly loaded and structured for the proposed model shown in Fig 1.

Stage 2: Initialization. The Artificial Bee Colony (ABC) algorithm parameters are set, including the population size, maximum cycle number (MCN), and abandonment limit. The Food Source Memory (FSM) is initialized by randomly generating a population of feasible solutions ("food sources"), each representing a complete IRP solution that satisfies all hard constraints. This stage establishes the starting point for the iterative search process (Fig 1).

Stage 3: Improvement (Iterative Bee Phases**).** The core of the algorithm resides in the Improvement stage, which consists of three iterative phases executed in sequence (Fig 1):

i. Employed Bee Phase: Each employed bee is assigned to a specific food source. It performs a local search by applying Move, Swap, or Kempe chain neighbourhood operator operator Neighborhood operators to its assigned solution. If the newly generated solution has a better fitness value $f(x_j^{'}) < f(x_j)$, the food source is updated. This phase focuses on exploitation refining existing solutions.

ii. Onlooker Bee Phase: Onlooker bees probabilistically select promising food sources from the FSM based on their fitness values. They then apply the same Neighborhood operators (Move, Swap, and Kempe chain neighbourhood operator operator) to these selected solutions. If an improved solution is found, it replaces the original in the FSM. This phase directs search effort toward high-quality regions of the solution space.

iii. Scout Bee Phase: If a food source fails to improve over a predefined number of cycles (reaching the abandonment limit), it is considered stagnant. A scout bee then generates a new random food source to replace it, ensuring population diversity and enabling exploration of new regions of the solution space.

3.1. 2. Termination Condition

The iterative loop (Stage 3) in Figure 1 continues until a termination criterion is “met” specifically, when the loop counter exceeds the Maximum Cycle Number (MCN). At this point, the optimization process ends, and the best-found solution (food source with the lowest logistic ratio) is output as the final result.

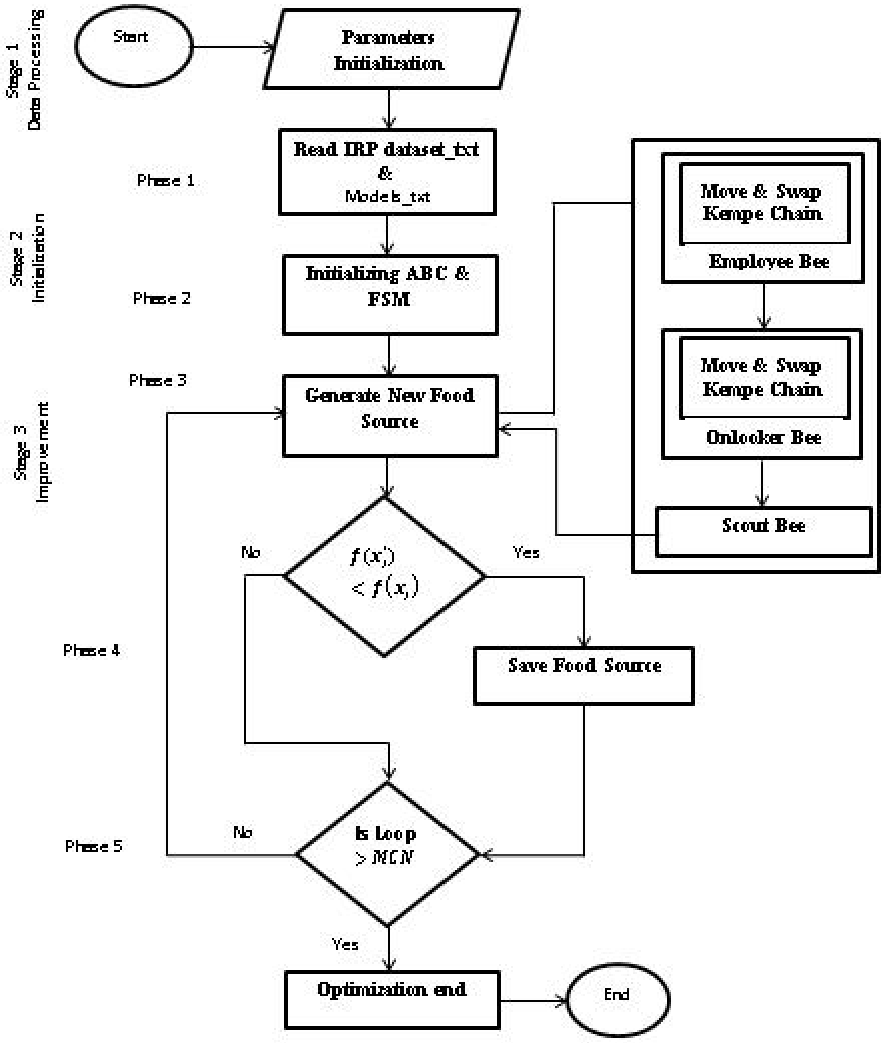

Fig 1: Model Diagram

3.1.3. Enhancement: The MABC with Kempe chain neighbourhood operator

While the standard ABC provides a robust balance of exploration and exploitation, its reliance on simple Swap and Move operators can sometimes limit its ability to execute more transformative changes to the solution structure, especially in highly constrained problems like the IRP. To address this, MABC algorithm is proposed.

The key enhancement is the integration of an advanced Kempe chain neighbourhood operator, which is activated alongside the standard Swap and Move operators during both the employed and onlooker bee phases. Unlike the binary Swap or unary Move operators, the Kempe chain neighbourhood operator performs a structured, multi-point interchange. It works by selecting two distinct delivery shifts or routes and identifying a connected chain (or set) of customers that can be feasibly reallocated between them. This operator facilitates a more dramatic yet controlled reorganization of the delivery schedule, effectively reshuffling customer groups across shifts.

The inclusion of the Kempe chain neighbourhood operator strengthens the algorithm in two critical ways:

i. Escaping Local Optima: By enabling larger, more complex moves, it allows the search to jump out of basins of attraction surrounding sub-optimal solutions that simpler operators cannot overcome.

ii. Accelerated Convergence: It promotes a more efficient exploration of the combinatorial solution space, often leading to the discovery of higher-quality solutions in less iteration by making significant, feasibility-preserving improvements to the solution structure

Thus, the MABC framework retains the adaptive population-based search of the original ABC while augmenting its local search power with a sophisticated, problem-specific Neighborhood operator (Kempe chain neighbourhood operator operator), making it particularly well-suited for the intricate constraints and large scale of the IRP

3.2 ROADEF-EURO 2016 Inventory Routing Problem Constraints

Here is the categorization of the constraints in the ROADEF/EURO 2016 Inventory Routing Problem into hard and soft constraints for this study:

3.2.1 Hard constraints

Hard constraints are constraints that must be satisfied for solving IRP. These are the mandatory physical and operational rules. They are typically related to the physical limitations of the problem, such as the capacity of trucks, the time available to complete the deliveries, and the safety of the drivers (Koç *et al*., 2024). The following constraints in the ROADEF/EURO 2016 IRP are hard constraints:

Constraints related to drivers: These constraints ensure that each driver does not work too many hours or drive too many kilometres more than the hours assigned for delivery.

Driver's constraints 1**:** Inter-shift duration is the duration separating two or more consecutive shifts for driver d. The duration or interval between two shifts assigned to a driver d and the duration separating the shift is called minimum inter-shift duration I_d^{min} . Therefore Eq. (1) show the relationship;

$$ts_s^{start} - ts_s^{end} \geq I_d^{min} \quad \forall s, s' \in S | s \prec s', Y_{s,d} = Y_{s',d=1}, d \in D \tag{1}$$

Driver's constraint 2: For the operations concerning shift s (with base final (s) operation inclusive) the maximum driving time is the total of time travelled on a shift in the previous operation together with the previous operation's travelled time to the current operation location. This implies that, the total driving time within T_O^{drive} any shift must not exceed the maximum Driving duration, D_d^{max} using Eq. (2,3).

$$\sum_{o \in O} T_O^{drive} \leq D_d^{max}, \ \forall s \in S \tag{2}$$

Therefore, the accumulated driving time up to operation o, in a shift s is presented as:

$$T_O^{drive} = T_{O'}^{drive} + T_{p(o'), p(o)} \tag{3}$$

Driver's Constraints 3: looking at the driver's time window, the driver's selected time windows must contain the start(s) and end(s) interval for every shift s in Eq. (4). That means any assigned shift must start and end within the driver's legal time window.

$W_p^{min} \leq t_s^{start}$ *and* $W_p^{max} \leq t_s^{end}$,
$\forall s \in S, d \in D$ (4)

Constraints related to trailers: These constraints ensure that trailers are not overloaded and that they are compatible with the shift assigned.

Trailer Constraint 1: There is no overlapping time for a single trailer with a different shift.

Looking at shifts s and s' assigned to a single trailer, this implies that is either shifts s ends before shifts s' begins or shifts s' ends before shifts s as expressed by Eq. (5)

$t_{s'}^{start} \geq t_s^{end}$ *or* $t_s^{start}\, t_{s'}^{end}\ \forall s, s' \in S | s \neq s'$,
$X_{s,r} = 1, X_{s',r} = 1, r \epsilon R$ (5)

Trailer constraints 2: The compatibility of the trailer attached to the driver in a shift. The trailer allocated for a shift to a driver should be a trailer driveable by the driver using Eq. (6).

$X_{s,r} . Y_{s,d} \implies Ad, r = 1, \forall s \in S, d \in D, r \in R$ (6)

Constraints related to shifts: These constraints ensure that deliveries are made during the allowed hours of operation.

Shift constraints 2: The travel, from the previous point and if necessary the layover duration determines the arrival at the current point Eq. (7).

$t_o^{arr} = t_{O'}^{dep} + T_{p(o'), p(o)} + L_{o'} . \delta_{Lay}, \forall o \in O$ (7)

The final operations before reaching the base have to be considered so long as shift only end and closed at the base. The actual period driver d spends at either the source or customer p waiting for attention before loading or delivery is considered in this constraint. That is the time in which the driver is waiting to be attended to. The departure time from an operation is the arrival time plus waiting and service duration which represented in Eq. (8) below.

$t_o^{dep} = t_o^{arr} + \omega_o + \sigma_o, \forall o \in O$ (8)

Shift constraints 3: Delivery to customers is only allowed during customers opening hours. The initial time window of each site in all operations consists of arrival operation and departure operation intervals. All operations must occur within the operational time windows of the respective location represented in Eq. (9).

$W_{p(o)}^{min} \leq t_o^{arr} \leq t_o^{dep} \leq W_{p(o)}^{max}\ \forall o \in O$ (9)

Shift constraints 4: The customer's capacity at the required location. The call-in customers are considered in this constraint. The quantity to be delivered must be more than the minimum required quantity for the customer used and also, less than the tank capacity of the customers for each operation in a shift shown in Eq. (10).

$0 \leq |Q_o^{end}| \leq C_r, \forall o \in O, r \in R | X_{s,r} = 1, V_{o,s} = 1$ (10)

The inventory level on the trailer must be updated correctly after each operation.

$Q_o^{end} \leq Q_o^{start} + Q_o, \forall o \in O$ (11)

Constraints related to Site: These constraints ensure that deliveries are made to the correct sites and that the inventory is not damaged. But, the customer on their side has to be ready since they made an order. Each site tank capacity is a must to consider, exception of Call-in customers. The inventory level at any customer location ($p \in C$) must remain within the tank capacity at all hourly timesteps h.

$0 \leq Z_{p,h} \leq C^p, \forall p \in C, h \in H$ (12)

The inventory balance equation ensures that tank levels change correctly based on consumption and executed deliveries. Therefore, the source and customer dynamic equation basic inventory is as follows:

$$Z_{p,h} = Z_{p,h-1} - F_{p,h} + \sum_{o\epsilon O} Q_o \text{ , } \forall \text{p} \in \text{C}, \text{h} \in \text{H} \quad (13)$$

Where,

$O_{p,h}$ is the set of delivery operations at customer p completed within time step h.

3.2.2 Soft constraints

Soft constraints are constraints that are not strictly enforced, if violated. The soft constraints reflect the performance and cost objectives which are typically optimized but, can be violated. They are typically related to the quality of the solution, such as the total distance travelled or the number of late deliveries (Shan *et al*., 2025). The following constraints in the ROADEF-EURO 2016 IRP are soft constraints:

Constraints related to Layover: These constraints are typically considered soft constraints because it adds flexibility to scheduling. While extending travel duration for customer delivery may be suboptimal, it may be manageable and tolerated to some extent. It is the fixed time added to a shift, extending the travel duration for customer delivery.

The layover is the added interval to a shift in an operation taken by a driver for delivery to a customer at a given location. Only one layover is allowed or required in a shift.

Layover Limit: A maximum of one layover is permitted per shift using Eq. (1). The associated cost is penalized in the objective function.

$$\sum_{o\epsilon O(\text{s})} L_o \leq 1, \forall \text{s} \in \text{S} \quad (14)$$

Constraints related to quality of service: These constraints penalize solutions that have late deliveries or that do not meet the customer's requirements.

Quality of Service: To avoid run-out; the tank level at the safety level or below the safety level for each of the Vendor Managed Inventory (VMI) customers triggers demand for supply. The customer's p level of the tank must always be above safety level else delivery to customer is the only option. Customer tank levels should ideally not fall to or below the safety stock level, Z_p^{safety} in Eq. (15). This is usually modelled by penalizing the deviation in the objective function.

$$Z_{p,h} \geq Z_p^{safety}, \forall \text{p} \in \text{C}, \text{h} \in \text{H} \quad (15)$$

3.3 Optimization Goal (Objective Function)

To achieve the objectives of this work the logistic ratio has to be minimized. This focused on the horizons of long-term distribution and supply cost to customers at various destinations, rather than to customer's location. To reduce the cost of distributions is the main issue of concern in this study. To achieve this goal is equal to meeting the customer's demand at the expected and favourable period. The summation of the cost incurred from shifts shared upon the total quantity of product delivered in the same shifts is referred to as the logistic ratio. The logistic ratio is been represented as:

$$Logistic\ Ratio = Min\left(\frac{\sum_{s\in S} Cost(s)}{\sum_{s\in S} Q_s^{delivered}}\right) \quad (16)$$

The cost of distribution (distribution cost) is clearly stated as the cost of a shift in the above expression, which consists of the followings:

i. Distance cost (DCOST): This covered the fuel consumed by the trailer, its maintenance, and the length of distance in a shift.
ii. Time cost (TCOST): some related driver's charges with salary and the period taken in a shift.
iii. Layover cost (LayCost): The extended interval of in a shift which serves as extra distance to cover or to make delivery. This only holds if a layover exists.

$$\text{Cost(s)} = \text{DCOST(s)} + \text{TCOST(s)} + \text{LayCost(s)} \quad (17)$$

Where,

$$\text{Distance Cost (DCOST)} = Cost^D . \sum_{o \in \text{O(s)}} T_{p(o'),p(o)} \quad (18)$$

$$Time\ Cost\ (TCOST) = Cost^T . (\text{t}_s^{\text{end}} - \text{t}_s^{\text{start}}) \quad (19)$$

$$Layover\ COST(LayCOST) = Cost^L . \sum_{o \in \text{O}(s)} L_o \quad (20)$$

Overall total quantity delivered:
Total Quantity Delivered ($Q_s^{delivered}$):

$$Q_s^{delivered} = \sum_{o \in \text{O}(s)} \max(0, Q_o) \quad (21)$$

3.4 Mathematical Model

3.4.1 Sets and Indices

i. Setup time: The required time to load a trailer at the base.
ii. Start/end: The start and the end of delivery at the base by the driver.
iii. Time windows: The designated time for drivers to be active for delivery and for customers to receive delivery.
iv. Minimum Inter shift Duration: The interval/period assigned for a driver to move between driving duration and resting time within time window.
v. Maximum driving duration: The required period for a driver to make delivery and return to the base.
vi. Layover duration: The only duration assigned for a driver to make delivery to a call-in customer.

3.4.2 Problem Definition

i. Problem Notation

The notations used in this work are expressed in the table 2 below

Table 2: Problem Notation

Symbol	Definition	Unit/set
D	Set of all drivers	
R	Set of all trailers	
L	Set of all locations (L=B∪C∪S)	
C	Set of customer locations	
S	Set of all shifts	
O	Set of all operations	
d,d'	Index for drivers, d∈D	
R	Index for trailers, r∈R	
p,q	Index for locations, p,q∈L	
s,s'	Index for shifts, s,s'∈S	
o,o'	Index for operations, o,o'∈O	
H	Index for inventory time step (e.g., hourly)	*H*

T	Total planning horizon duration	*Minutes*
$\tau_{p,q}$	Travel time from location p to q	*Minutes*
σ_o	Service/Loading time for operation o	*Minutes*
W_p^{min}, W_p^{max}	Time window for location p	*Minutes*
I_d^{min}	Minimum inter-shift duration for driver d	*Minutes*
D_d^{max}	Maximum driving time for driver d	*Minutes*
C_r	Capacity of trailer r	*Kg*
C_p	Tank capacity of customer p	*Kg*
Z_p^{safety}	Safety stock level for customer p	*Kg*
Ad,r	Binary parameter: 1 if driver d can tow trailer r, 0 otherwise	
$Cost^D$, $Cost^T$ $Cost^L$	Unit cost parameters for distance, time, and layover	*Unit*
$F_{p,h}$	Forecasted consumption rate for customer p in time step h	*kg/hour*
δ_{Lay}	Fixed duration of a layover	*Minutes*

ii. Decision Variable for IRP and its Formulation

To consider the categories of the constraints in IRP, hard and soft constraints aid in the implementation to be able obtain a feasible solution (Absi *et al.*, 2020). In fact, it is often the case that there are more hard constraints than soft constraints. In this case, the algorithm utilized shift variables expressed in table 3 to find a solution that satisfies all of the hard constraints, and then minimize the violation of the soft constraints (Kheiri, 2020).

Table 3: Shift/Route Variables

Symbol	**Definition**	**Type**
$Y_{s,d}$	Binary: 1 if shift s is assigned to driver d, 0 otherwise	*{0,1}*
$X_{s,r}$	Binary: 1 if shift s is assigned to trailer r, 0 otherwise	*{0,1}*
t_s^{start}, t_s^{end}	Start and end time of shift s	*R+*
$V_{o,s}$	Binary: 1 if operation o belongs to shift s, 0 otherwise	*{0,1}*
Operation Variables:		
t_o^{arr} , t_o^{dep},	Arrival and Departure time for operation o	*R+*
ω_o	Waiting time before service/loading for operation o	*R+*
Q_o	Quantity (delivery if Q_o>0, loading if Q_o<0, in operation o	*R (kg)*
L_o	Binary: 1 if a layover occurs afer operation o, 0 otherwise	*{0,1}*
$Q_o^{start}Q_o^{end}$	Trailer inventory level before and after operation o	*R+ (kg)*
Inventory Variables		
$L_{p,h}$	Inventory level at location p at the start of time step h	*R (kg)*

3.5 The Algorithmic Approach/Implementation

To achieve the objectives algorithm 1, 2, 3 and 4 were implemented one after the other according to ABC approach:

i. Initialization Phase

The necessary parameters for the initialization phase were established at the first three lines in Algorithm 1. This ranges from generating random IRP solutions, initializing the FSM for each solution and the limit or criteria for abandonment of an unimproved solution. Line 6 evaluate fitness solution, line 5 to 12 consists the iterative processes of generating the solution while line 8 established abandonment threshold for each IRP solution.

Algorithm 1: Initialization Phase

1. S_N →Number of initial food sources
2. f_i → Food source memory (FSM)
3. C_i →Abandonment criteria counter
4. Begin
5. For i = 1 to S_N DO
6. X_i → randomly generated feasible IRP solution including:
   - Assigned trailers and drivers,
   - Delivery quantities per customer per shift,
   - Routing structure,
   - Start and end times,
   - Layovers if applicable.
7. $TotalQty_i$ → sum of all quantities delivered in X_i
8. $Cost_i$ → sum over all shifts x ∈ X_i of:
   $Dcost * TrailerDist(x) + Tcost * WTime(x) + LayCost(x)$
9. $f_i = \frac{Cost_i}{TotalQty_i}$ → *logisticratio*
10. $C_i \leftarrow 0$
11. END FOR
12. END

ii. Employee Bee phase

During the Employee Bee phase, each bee is assigned to specific food source X_j. Where, j= 1, 2. . . S_N . The bee explores the Neighborhood of X_j probably applying one of the predefined local search operators such as swap, move and kempe chain neighbourhood operator in Algorithm 2. Line 1to 12 demonstrated the iterative processes of the employee bee with the application of the Neighborhood structures (kempe chain neighbourhood operator operator) to each IRP solution in Algorithm 2

This ensures lower ratio solutions have the higher chances of refinement. Once a solution X_j is selected an onlooker bee applies one of the operators (swap, move or kempe chain neighbourhood operator operator) to generate new solution under the condition that if the new solution is better its replaces the older solution in the FSM shown in Algorithm 2

Algorithm 2. Employee Bee Phase

```
Begin
  For j = 1 to S_N DO
    rnd ← U (0, 1)          ⟶generate a uniform random number
    IF rnd ≥ 0.9 THEN
      Apply Swap (X_j)      ⟶swap two customers or shifts
    ELSE IF rnd ≥ 0.15 THEN
      Apply Move (X_j)      ⟶reassign customer to new shift or route
    ELSE
      Apply Kempe (X_j)     ⟶reallocate customer groups across shifts
    END IF
  END FOR
END
```

iii. Onlooker Bee

This ensures lower ratio solutions have the higher chances of refinement. Once a solution X_j is selected an onlooker bee applies one of the operators (swap, move or kempe chain neighbourhood operator operator) to generate new solution under the condition that if the new solution is better its replaces the older solution in the FSM shown in Algorithm 3.

Algorithm 3. Onlooker Bee Phase

```
Begin
  Compute total fitness:
    Total Fitness ← ∑_{k=1}^{S_N} f(X_k)
  For j = 1 to S_N DO
    Compute selection probability:
      P_j ← f (X_j) / TotalFitness
    Select X_j using roulette wheel selection based on P_j
    rnd ← U(0, 1)           ⟶generate a uniform random number
    IF rnd ≥ 0.9 THEN
      Apply Swap (X_j)      ⟶swap two customers or shifts within X_j
    ELSE IF rnd ≥ 0.15 THEN
      Apply Move (X_j)      ⟶reassign a customer to a new shift or route
    ELSE
      Apply Kempe (X_j)     ⟶reallocate groups of customers between shifts
    END IF
    Evaluate newX_j:
      - Compute TotalQty_j  ← ∑ delivery quantities in updated X_j
      - Compute Cost_j      ← sum of trailer, waiting, and layover costs
      - F_new = Cost_j/TTotalQty_j   ⟶logistic ratio (fitness)
    IF F_new < f(X_j) THEN
      Update FSM: Replace X_j with the new improved solution
    END IF
  END FOR
End
```

iv. Scout phase

The scout bee algorithm 4 began looping from line 2 through each food source representing feasible delivery plan within FSM. Line 3 checks condition of threshold whether or not exceeded, if satisfied the current food source is to be replaced for been stagnant. Line 4 initiate replacement of stagnant solution and generate new solution through randomization, line sum the quantity delivered, line 6 sum the total cost accrued while line 7, 8, and 9 evaluate quality of fitness, reset trial counter of solutions and indicate the end of solution blocks. Line 10 close the main loop as line 11 concludes the scout phase.

Algorithm 4. Scout Bee Phase

1. Begin
2. For j = 1 to S_N DO
3. IF trial(X_j) ≥ limit THEN
4. X_j→ GenerateNewFeasibleSolution ()
   ⟶Randomly assign trailers, drivers, and customers
   ⟶Build shifts, routes, and time schedules
   ⟶Respect constraints on quantity, time, and layover
5. Compute $TotalQty_j$← ∑ quantities in X_j
6. Compute $Cost_j$→ sum of all cost components in X_j
   ⟶Includes Dcost, Tcost, and LayCost
7. f(X_j) = $Cost_j$ / $TotalQty_j$ ⟶logistic ratio
8. trial(X_j) ← 0 ⟶Reset trial counter
9. END IF
10. END FOR
11. End

3.6 Performance Evaluation Metrics

The algorithms' performance is assessed using two primary metrics:

i. Logistic Ratio (LR): The primary objective function, calculated as Total Distribution Cost divided by Total Quantity Delivered. Minimizing this ratio is the optimization goal.

$$Logistic\ Ratio = \ Min\left(\frac{\sum_{s\ \in S} Cost(s)}{\sum_{s\ \in S} Q_s^{delivered}}\right) \tag{22}$$

Where;

$$s\ \in S = There\ exist\ a\ shift\ in\ Set\ of\ all\ shifts$$
$$Cost(s) = The\ total\ cost\ of\ the\ entire\ shifts\ covered$$
$$Q_s^{delivered} = The\ total\ quantity\ delivered\ or\ the\ entire\ shifts\ covered$$

Ii. Feasibility Rate: The percentage of generated delivery shifts that satisfy all operational hard constraints.

$$\text{Feasibility Rate} = \frac{N_f}{N_t}(100) \tag{23}$$

Where;

N_f = Number of feasible shifts (shifts with delivery quantity greater than zero), N_t = Total number of generated shifts

$If\ N_t = 0$, then the feasibility rate is defined as 0% to avoid division by zero

4. Results and Discussion

4.1 Performance of ABC vs. MABC

The experimental results on Instance B are summarized in Table 4 and fig 2. Though, the result shows that some instances share identical structures which resulted to similar or identical outcomes. This occurred under MABC (V2.15 and V2.17, V2.13 and V2.19, V2.16 and V2.20), ABC (V2.12 and V2.22, V2.24 and V2.26). The Modified ABC (MABC) algorithm outperformed the standard ABC in 10 out of 15 instances, achieving significantly lower (better) logistic ratios. Notable improvements were observed in complex instances like V2.18, where MABC reduced the logistic ratio by 93.26%. The standard ABC performed better in 5 instances (V2.12, V2.16, V2.20, V2.22 and V2.23). This shows that the enhanced exploration of MABC is particularly beneficial for larger, more complicated scenarios but, may slightly underperform in simpler or more constrained settings.

Table 4: Logistic Ratio Results for ABC and MABC Algorithms

Instance B	ABC	MABC
V2.12	**0.0098549**	0.014468
V2.13	0.0306717	**0.0243827**
V2.14	0.0303930	**0.026597**
V2.15	0.0189651	**0.0120151**
V2.16	**0.0137475**	0.017778
V2.17	0.0189651	**0.0120151**
V2.18	0.1882200	**0.012678**
V2.19	0.0306717	**0.0243827**
V2.20	**0.0137475**	0.017778
V2.21	0.0136890	**0.012078**
V2.22	**0.0098549**	0.014468
V2.23	**0.0098560**	0.011094
V2.24	0.0069897	**0.0057018**
V2.25	0.0070260	**0.005409**
V2.26	0.0069897	**0.0057018**

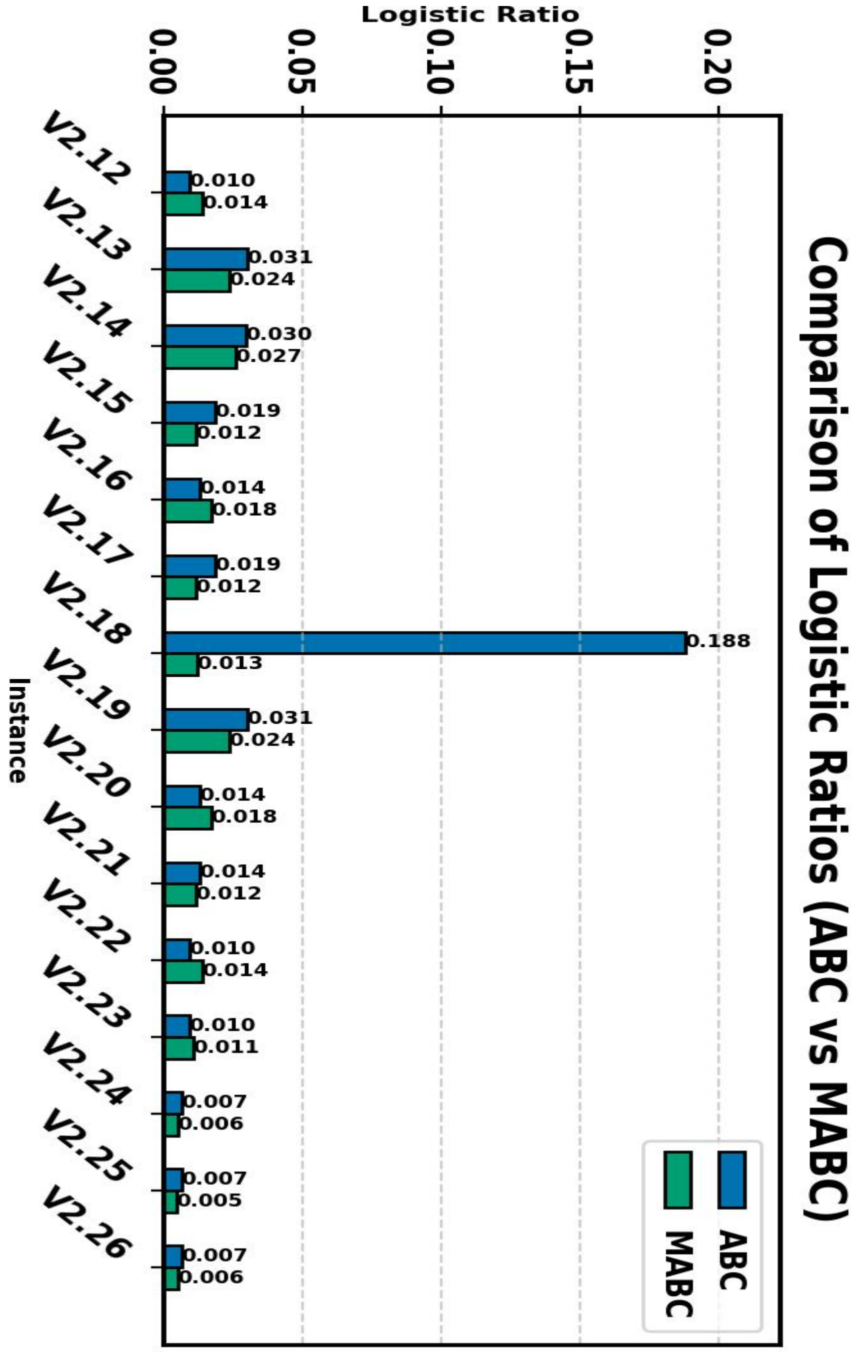

Fig 2: ABC vs. MABC Result Comparison

Table 5 compares the best-performing algorithm (MABC) against two leading methods from literature: Absi *et al.,* (2020) and Kheiri, (2020) with their logistic ratio values represented as LR_1 and LR_2. MABC demonstrated highly competitive performance, matching or surpassing the benchmark methods in the majority of instances, especially in medium and large-scale cases like V2.15, V2.17, and V2.18. The results table 5 confirmed that the MABC through the integration of Move, Swap and Kempe chain operator is highly effective in complex routing/ scheduling of delivery. The lower logistic ratio results indicate better feasibility and cost optimization. This shows that the algorithm enhanced exploration capability by escaping the block local optima that trap both

ABC and the benchmarked methods. In other words, MABC successfully restructured the rotes via Kempe chain operator leading to near optimal feasibility.

4.2 Convergence Behaviour

Convergence graphs for all instances showed that both ABC and MABC algorithms effectively reduced the logistic ratio over 1,000 iterations from fig 3 to 4. The MABC typically achieved a lower final cost, with a convergence pattern indicating a more thorough search of the solution space due to the Kempe chain neighbourhood operator.

The small instance has less structure or features compared to the large instances. The instance V2.12 (fig 3) is used to show example and represent the small instances (V2.12, V2.16, V2.20, V2.22, V2.23). In that case, the ABC model performances were better because there are not lengthy list of customers to attend to, less quantity delivery or minimal routes/shift. While, the large instance has more structure or features compared to the small instances. The instance V2.18 (figure 4) is used to show example and represent the large instances (V2.13, V2.14, V2.15, V2.17, V2.18, V2.19, V2.21, V2.24, V2.25, V2.26). In that case, the MABC model performances were better because it's suitable for instances with complex or many features.

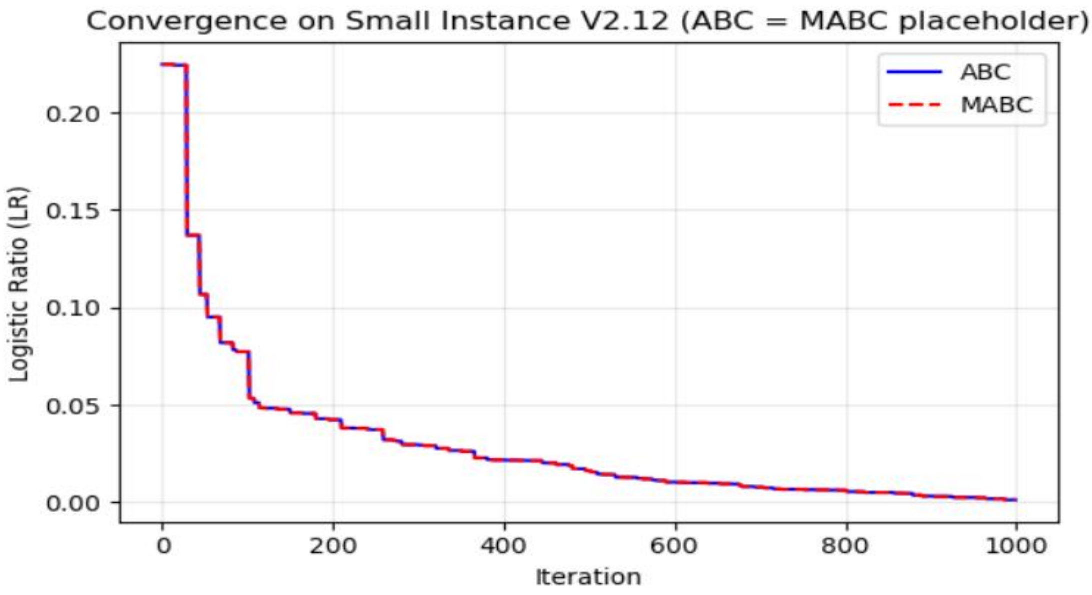

Fig 3: Small ABC vs, MABC

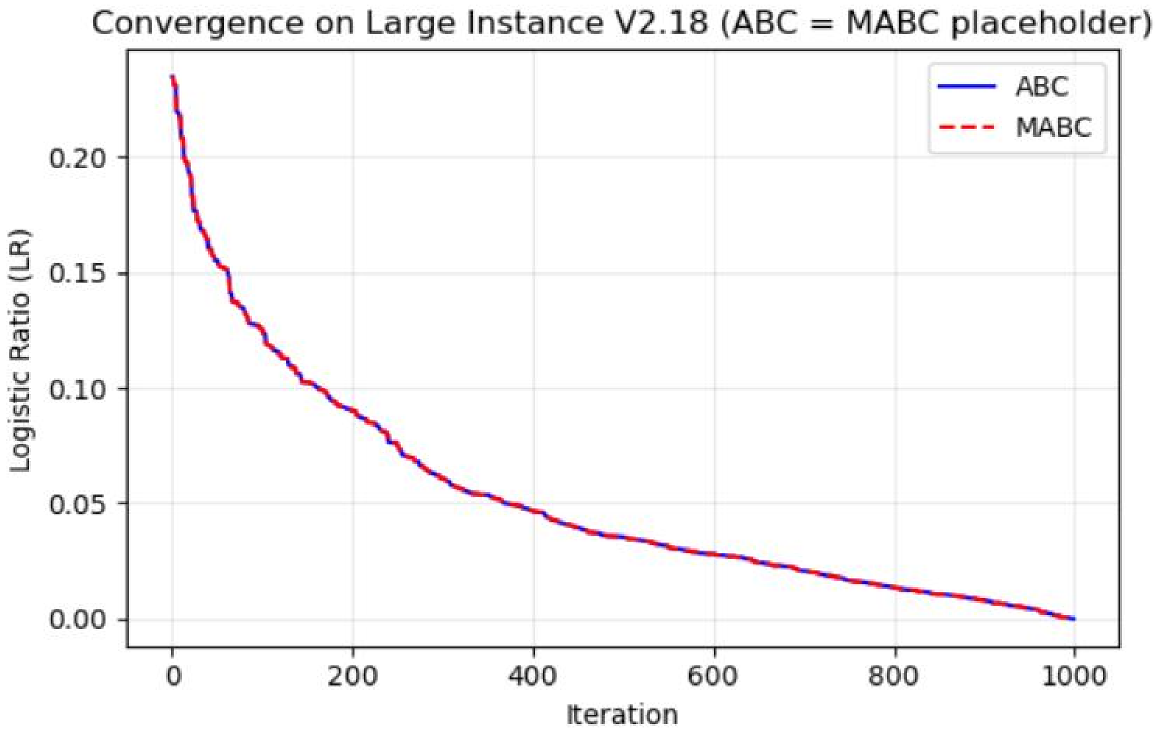

Fig 4: Large ABC vs. MABC

Therefore, MABC underperformed in smaller instances because integration of kempe Chain operator whose function constitutes both Move and Swap of customers delivery between two or more shifts/routes due to customers ordered product quantity and trailers capacity during shift excessively disrupt fine-tuned local optima maintain by ABC algorithm. Therefore, for smaller dataset/instance kempe chain should be disable to avoid inconsistency and efficient performance.

4.3 State of the Art Comparison

The performance of two well-known state-of-the-art methods: Absi *et al.,* (2020)*, as* LR_1, and Kheiri, (2020) as LR_2, the suggested Modified Artificial Bee Colony (ABC) algorithm is contrasted with the standard ABC in Table 5. The logistic ratio (LR), which continues to be the primary metric for assessing routing cost efficiency across benchmark instances, serves as the basis for the comparison.

Table 5: Comparative Results Summary

Instance B	ABC	MABC	LR_1	LR_2
V2.12	0.0098549	0.014468	**0.010024**	**0.010266**
V2.13	0.0306717	**0.0243827**	0.028875	0.030768
V2.14	0.0303930	**0.026597**	0.034971	0.037582
V2.15	0.0189651	**0.0120151**	0.024993	0.026608
V2.16	0.0137475	0.017778	**0.011783**	0.012420
V2.17	0.0189651	**0.0120151**	0.032130	0.031538
V2.18	0.1882200	**0.012678**	0.031882	0.033018
V2.19	0.0306717	**0.0243827**	0.034022	0.036018
V2.20	0.0137475	0.017778	**0.017486**	0.018656
V2.21	0.0136890	**0.012078**	0.016806	0.017210
V2.22	0.0098549	0.014468	**0.012667**	**0.012992**
V2.23	0.0098560	**0.011094**	0.012603	0.013311
V2.24	0.0069897	**0.0057018**	0.011219	0.013033
V2.25	0.0070260	**0.005409**	0.011451	0.012411
V2.26	0.0069897	**0.0057018**	0.011281	0.012866

Note: LR (ABC): The logistic ratio value from the Artificial Bee Colony algorithm. LR (MABC): The Ratio values from the Modified Artificial Bee Colony algorithm. LR_1 is the Logistic ratio by (Absi *et al.* 2020). LR_2 is the Logistic ratio values by Kheiri, (2020) in Table 5

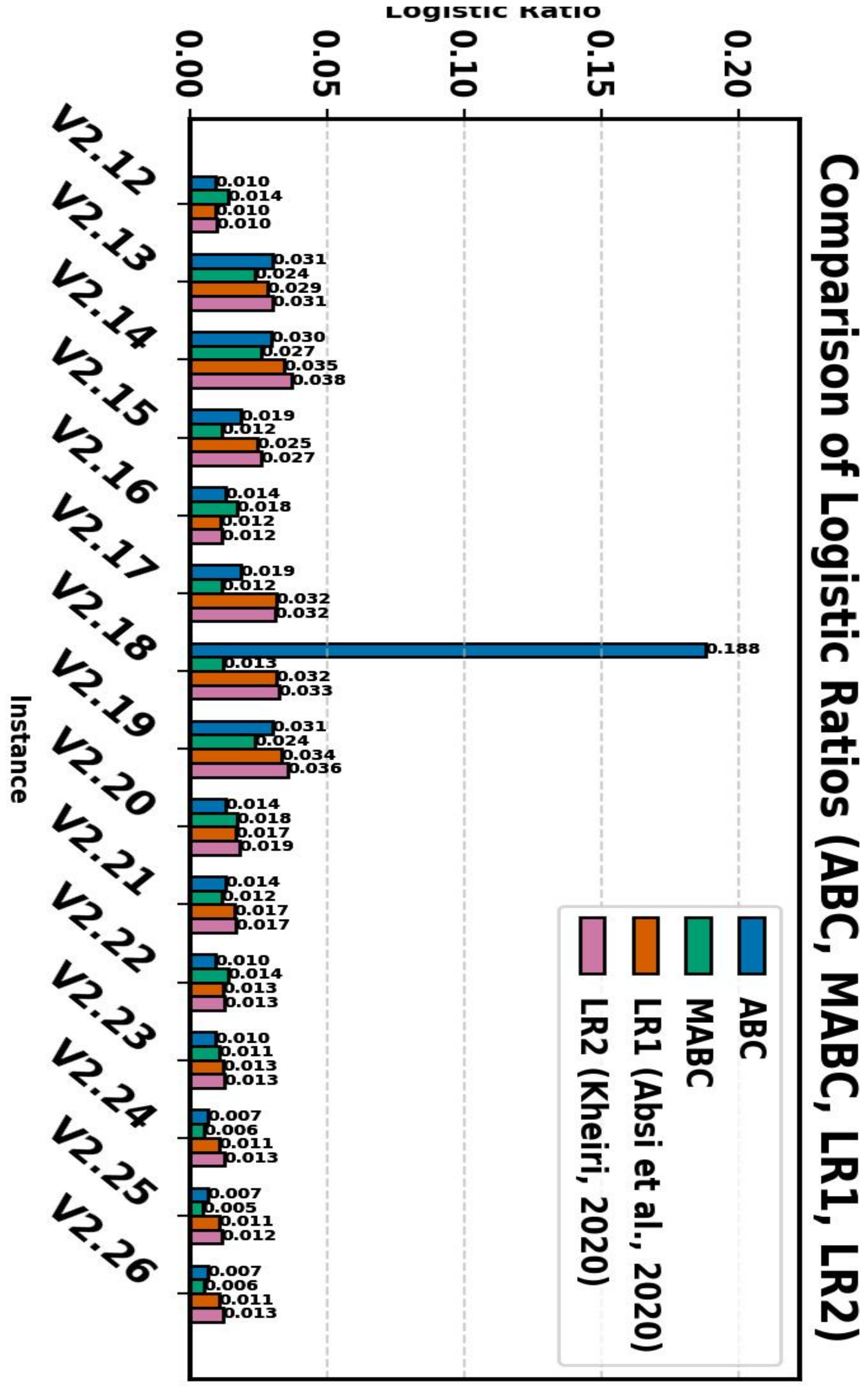

Fig 5: Comparison with State of art algorithm

Table 5, and fig 5 show that the Modified ABC (MABC) algorithm is highly competitive, often matching or outperforming leading methods, especially in medium and large IRP instances where routing becomes more complex. It achieves major improvements in cases like V2.15, V2.17, and V2.18, where its LR values are far lower than those of Absi and Kheiri, demonstrating strong optimization capability.

In some cases such as V2.14 and V2.19 MABC performs at the same level as the top algorithms or falls slightly behind. However, in smaller or more constrained instances like V2.12, V2.16, V2.20, and V2.22, MABC's performance drops, showing higher LR values than both benchmark approaches. A few instances also reveal that the standard ABC still outperforms MABC.

Therefore, MABC performs well in 11 out of 15 datasets of Instance B, especially in larger and more complex scenarios. This suggests that the added exploration mechanism strengthens the algorithm's ability to navigate wide solution spaces. The few cases of underperformance are minor and likely tied to the simpler structure of those instances B (datasets) and the natural randomness of heuristic methods.

Here is the clear expression of the Wilcoxon Signed-Rank test of the ABC, MABC, LR_1 and LR_2 results. MABC achievement is not merely numerical but, statistically significant as shown in Fig. 6, 7 and 8 below. MABC significantly outperformed LR_1 $(W = 17.0, p = 0.0062)$ and LR_2 $(W = 12.0, p = 0.0021)$ $\alpha = 0.05$ *significance level*. While the MABC yields better numerical results than ABC but, the difference is not statistically significant $(W = 35.0, p = 0.0776 > 0.05)$ indicates the need for further investigation to confirm the superiority.

	Comparison	Statistic (W)	p-value (one-tailed)	Significant (α=0.05)
0	MABC vs ABC	35.0	0.077609	False
1	MABC vs LR1	17.0	0.006226	True
2	MABC vs LR2	12.0	0.002136	True

Fig 6: Wilcoxon Comparison

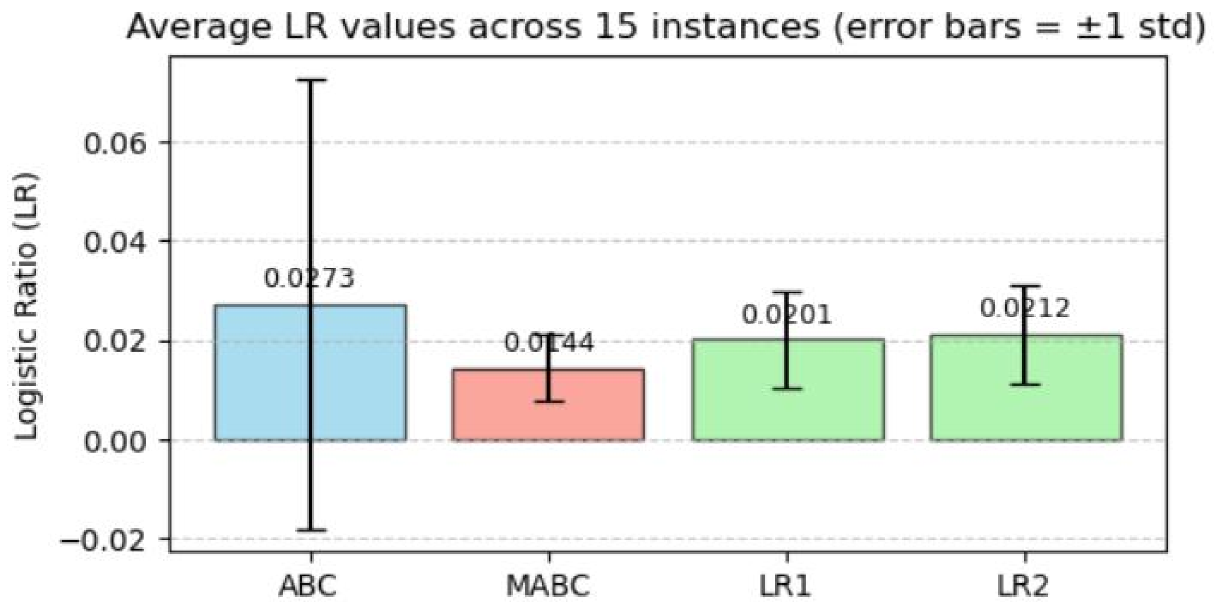

Fig 7: Visualization of Standard Deviation

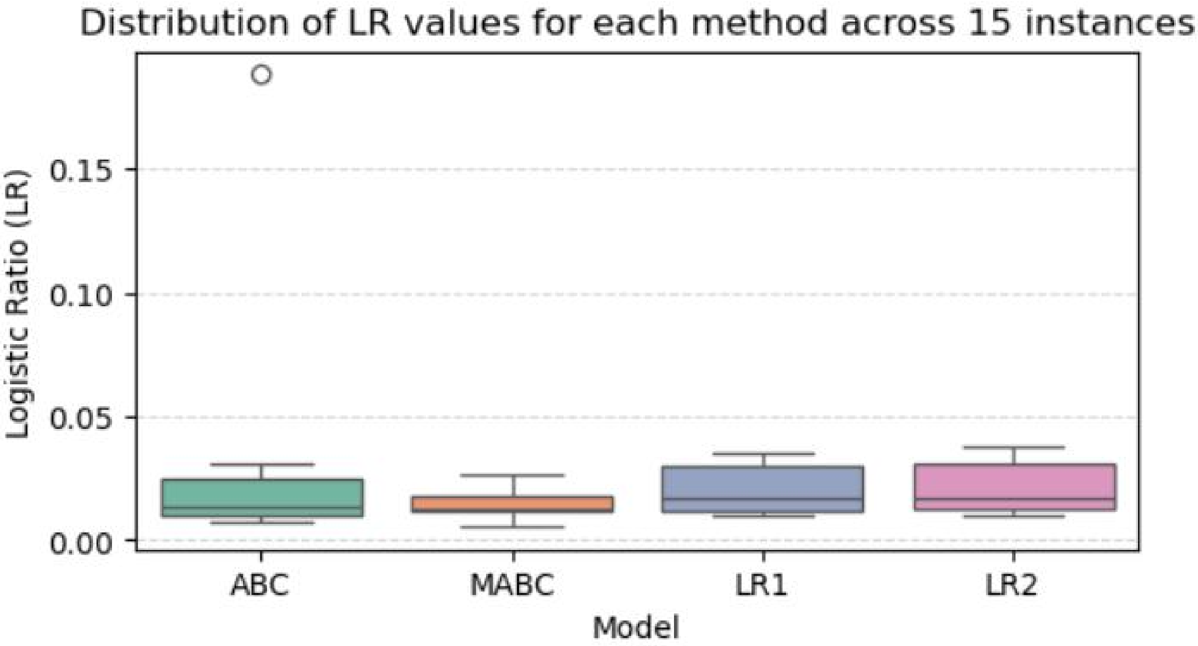

Fig 8: Distribution across instances

The standard significance of MABC improvement is confirmed by the application of Wilcoxon Signed-rank test which shows that MABC significantly outperformed LR_1 and LR_2 ($P<0.01$) while the advantage is not yet statistically significant ($p = 0.0776$) in fig 6-8.

5. Conclusion

This study successfully implemented and evaluated a Modified Artificial Bee Colony algorithm for solving the large scale Inventory Routing Problem. Using the ROADEF/EURO 2016 benchmark, the MABC algorithm demonstrated its efficacy by consistently producing feasible solutions and achieving lower logistic ratios than the standard ABC in most test instances. Its competitive performance against established state-of-the-art methods underscores its potential as a robust tool for complex logistics optimization. The integration of the Kempe chain neighbourhood operator search proved valuable in enhancing local exploitation and escaping local optima. Future work should focus on further hybridizing the approach, adaptive parameter tuning, and applying the MABC to other complex scheduling and distribution problems within operations research.

5.1 Recommendation

The population-based metaheuristics, particularly the ABC algorithm, shows potential strength for solving complex, high-dimensional optimization problems due to their flexibility, robustness, and global search ability. Existing studies demonstrate successful applications of ABC in Proportional Integral Derivative controller tuning, robotic path planning, and job shop scheduling, and neural network training, where it improves control accuracy, generates efficient paths, optimizes production schedules, and enhances learning by avoiding local minima.

5.2 Future Direction

The future research should focus on developing hybrid optimization models that combine ABC with local search or exact methods to improve convergence speed and solution quality. Further directions include real time; data driven Inventory Routing Problem models using Internet of Things data; robust approaches that handle uncertainty such as vehicle breakdowns and traffic delays; and ABC based models for vendor-managed inventory and perishable goods distribution, where shelf-life and spoilage constraints are critical.

Funding
This research received no external funding or grant from any individual and organisation.

Data Availability Statement
The data supporting reported results can be found in publicly archived datasets by ROADEF-UERO Challenge 2016 for Air Liquid Gas Company. The link: [Challenge ROADEF/EURO 2015 : Inventory Routing Problem]

Conflicts of Interest
The authors of this article declare that there are not known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Acknowledgement
This research was not funded by any grant

References
Absi, N., Cattaruzza, D., Feillet, D., Ogier, M., & Semet, F. (2020). A heuristic branch-cut-and-price algorithm for the ROADEF/EURO challenge on inventory routing. Transportation Science, 54, 313-329. https://doi.org/10.1287/trsc.2019.096

André, J., Bourreau, E., & Wolfler Calvo, R. (2020). Introduction to the special section: RAODEF-EURO Challenge 2016—inventory routing problem. Transportation Science, 54, 299-301. https://doi.org/10.1287/trsc.2019.0972

Baraka, J. C. M., & Yadavalli, S. (2022). Inventory management concepts and implementations: A systematic review. South African Journal of Industrial Engineering, 33(2), 15. https://doi.org/10.7166/33-2-2527

Berbiche, N., Chakir, M., Hlyal, M., & Alami, J. E. (2024). An integrated inventory-production-distribution model for crisis relief supply chain optimization: A systematic review and mixed integer programming formulation. Journal Européen des Systèmes Automatisés, 57, 899-920. https://doi.org/10.18280/jesa.570329

Cao, J., Gao, J., Li, B., & Wang, X. (2020). The inventory routing problem: A review. In Proceedings of the 20th COTA International Conference of Transportation Professionals (CICTP 2020) (p. 4488). https://doi.org/10.1061/9780784482933.385

Chan, F. T., Wang, Z. X., Goswami, A., Singhania, A., & Tiwari, M. K. (2020). Multi-objective particle swarm optimisation based integrated production inventory routing planning for efficient perishable food logistics operations. International Journal of Production Research, 58, 5155-5174. https://doi.org/10.1080/00207543.2019.1701209

Diabat, A., Archetti, C., & Najy, W. (2021). The fixed-partition policy inventory routing problem. Transportation Science, 55, 353-370. https://doi.org/10.1287/trsc.2020.1019

Haseltalab, S., & Karimi, H. (2025). Enhancing efficiency in delivering petroleum products: Optimizing the deteriorating inventory-routing problem with heterogeneous multi-compartment vehicles. International Journal of Management Science and Engineering Management, 20(2), 159-182. https://doi.org/10.1080/17509653.2024.2422308

He, Y., Artigues, C., Briand, C., Jozefowiez, N., & Ngueveu, S. U. (2020). A matheuristic with fixed-sequence reoptimization for a real-life inventory routing problem. Transportation Science, 54(2), 355-374. https://doi.org/10.1287/trsc.2019.0954

He, Z., Liu, Y., & Liu, K. (2024). Robust optimization approaches for the perishable product inventory routing problem with demand uncertainty. Journal of Industrial and Management Optimization, 20, 2740-2769. https://doi.org/10.3934/jimo.2024024

Kheiri, A. (2020). Heuristic sequence selection for inventory routing problem. Transportation Science, 54, 302-312. https://doi.org/10.1287/trsc.2019.0934

Koç, Ç., Bektaş, T., & Laporte, G. (2024). Decarbonizing road freight transportation: Recent advances and future trends. Journal of the Operational Research Society, 75, 43-63. https://doi.org/10.1080/01605682.2024.2446655

Laganà, D., Malaguti, E., Monaci, M., Musmanno, R., & Paronuzzi, P. (2024). An iterated local search matheuristic approach for the multi-vehicle inventory routing problem. Computers & Operations Research, 169, 1-11. https://doi.org/10.1016/j.cor.2024.106717

Lee, Y., Charitopoulos, V. M., Thyagarajan, K., Morris, I., Pinto, J. M., & Papageorgiou, L. G. (2022). Integrated production and inventory routing planning of oxygen supply chains. Chemical Engineering Research and Design, 186, 97-111. https://doi.org/10.1016/j.cherd.2022.07.027

Liu, P., Hendalianpour, A., Razmi, J., & Sangari, M. S. (2021). A solution algorithm for integrated production-inventory-routing of perishable goods with transshipment and uncertain demand. Complex & Intelligent Systems, 7, 1349-1365. https://doi.org/10.1007/s40747-020-00264-y

Mundim, A. A., Santos, M. O., & Morabito, R. (2025). Sustainable solutions analysis of a bi-objective green inventory routing problem with heterogeneous fleet and different types of fuels. RAIRO-Operations Research, 59, 549-578. https://doi.org/10.1051/ro/2024162

Prakash, S., Mukherjee, I., Soni, G., & Piplani, R. (2024). An enhanced multiobjective inventory routing model to meet sustainable goals for assembly supply network under uncertainty. Annals of Operations Research. https://doi.org/10.1007/s10479-024-05925-6

Schenekemberg, C. M., Scarpin, C. T., Pécora, J. E., Guimarães, T. A., & Coelho, L. C. (2020). The two-echelon inventory-routing problem with fleet management. Computers & Operations Research, 121, 104944. https://doi.org/10.1016/j.cor.2020.104944

Shan, Y., Li, Y., Liang, X., & Yue, L. (2025). Optimisation of multi-type vehicle routing considering soft capacity constraint and carbon emissions. International Journal of Systems Science: Operations & Logistics, 12, Article 2466105. https://doi.org/10.1080/23302674.2025.2466105

Tiwari, K. V., & Sharma, S. K. (2023). An optimization model for vehicle routing problem in last-mile delivery. Expert Systems with Applications, 222, 119789. https://doi.org/10.1016/j.eswa.2023.119789

Zheng, M., Li, Y., Du, N., Wang, Q., Huang, E., & Jiang, P. (2024). Joint optimization of recyclable inventory routing problem under uncertainties in an incentive-based recycling system. Computers & Industrial Engineering, 198, 110692. https://doi.org/10.1016/j.cie.2024.110692