Dynamic Integrated Production and Delivery Scheduling with Electric Vehicles: A Hybrid Ant Colony System Approach

Rachida Ben chabane*,[1], Simon Caillard[2], Hajar Nouinou[3], and Mourad Zghal[2]

***Abstract*— This paper addresses a dynamic Integrated Production and Delivery Problem (IPDP), where customer orders arrive in real time and production and transportation decisions must be continuously updated. To manage this dynamic environment, we develop D-HACS, a Dynamic-Hybrid Ant Colony System that adapts integrated schedules as new information becomes available. Computational experiments show that D-HACS maintains high solution quality even as both dynamism and problem size increase. These findings highlight the capability of D-HACS to ensure effective real-time and energy-aware coordination within integrated production–delivery systems.**

I. INTRODUCTION

Modern supply chains must jointly optimize production and distribution decisions under increasing environmental and operational constraints. The rapid deployment of low-emission zones (LEZ) across European cities, with more than 300 zones active in 2024 and projections exceeding 500 zones in the near future [1], [2], is accelerating the transition toward electric vehicles (EVs) for urban logistics. However, adopting EVs alone is not sufficient: inefficient coordination between production completion times and delivery operations leads to idle vehicles, missed time windows, and increased energy consumption. This highlights the need for tightly integrated production and delivery planning.

In many time-sensitive supply chains, production and distribution are strongly interdependent and must be scheduled jointly to ensure on-time deliveries while minimizing operational costs. This is particularly critical in make-to-order environments, where production is triggered by customer demand and delivery delays directly disrupt following operations. In such contexts, synchronization between production completion times and vehicle departures plays a central role in reducing waiting times, limiting inventory, and ensuring service reliability. Despite this interdependence, production and distribution decisions are still often optimized separately in practice, leading to suboptimal performance. This has motivated extensive research on the integrated production and delivery problem (IPDP), studied under various manufacturing environments such as job shop [3], flexible flow shop [4], and intelligent manufacturing systems [5].

The integration challenge becomes even more complex when transportation decisions involve electric vehicles. The electric vehicle routing problem with time windows (EVRPTW) has received increasing attention due to battery limitations, limited driving range, and charging requirements [6], [7]. These operational constraints directly impact route feasibility and vehicle scheduling, thereby strengthening the coupling between routing and production decisions. As a result, integrating EV routing within IPDP frameworks significantly increases the overall problem complexity.

Despite these advances, most existing works on integrated production and delivery remain limited to static settings, where all jobs and customer orders are assumed to be known in advance. This assumption is rarely satisfied in practice. Real-world supply chains operate in dynamic environments, where new orders arrive over time and unexpected disruptions such as machine breakdowns require continuous rescheduling [8]. Ignoring this dynamic dimension can lead to poor responsiveness and degraded system performance.

Recent studies have started to address dynamic and online variants of integrated production and distribution problems. For instance, [9] investigates an online setting with dynamic order arrivals and time-dependent costs, while [10] highlights the growing importance of such problems in make-to-order systems. In parallel, adaptive metaheuristics have been proposed to handle dynamic rescheduling in production environments [11]. However, to the best of our knowledge, no study simultaneously considers dynamic decision-making, integrated production and delivery, and routing constraints involving a mixed fleet of electric and combustion vehicles.

This paper addresses a dynamic integrated production and delivery problem with a mixed fleet composed of electric and combustion vehicles. Customer orders arrive over time and must be produced and delivered under time window and routing constraints. To solve this problem, we propose a hybrid ant colony optimization approach that jointly determines production sequencing and vehicle dispatching decisions in a dynamic environment.

The main contributions of this paper are as follows: (i) we formulate a dynamic IPDP with mixed fleet routing constraints and time-sensitive deliveries; (ii) we develop a position-based hybrid ant colony metaheuristic for joint production and distribution optimization under dynamic conditions; (iii) we evaluate the proposed approach on a set of newly generated benchmark instances.

The remainder of this paper is organized as follows. Section II presents the problem formulation. Section III describes the proposed solution approach. Section IV reports computational results. Section V concludes the paper.

II. PROBLEM FORMULATION

We consider an integrated production and distribution problem in which manufacturing and delivery decisions are

*Corresponding author: rbenchabane@cesi.fr
[1]CESI LINEACT Laboratory, Strasbourg, France
[2]CESI LINEACT Laboratory, Reims, France
[3]CESI LINEACT Laboratory, Nancy, France

jointly optimized over a finite planning horizon H. The problem combines a Job Shop Scheduling Problem (JSP) for the production stage and an Electric Vehicle Routing Problem with Time Windows (EVRPTW) for the distribution stage.

In the static setting, all customer orders are assumed to be known in advance, and the objective is to determine both a feasible production schedule and a set of delivery routes that satisfy operational constraints. However, in many real-world applications, new orders may arrive dynamically during execution. This leads to a dynamic extension of the problem, in which the current solution must be updated upon the arrival of new jobs, while preserving the feasibility and stability of the ongoing operations. The main assumptions underlying the problem are introduced below. Then, we describe the production and distribution subproblems in the static case, followed by the dynamic rescheduling framework.

A. Assumptions

The following assumptions govern both the production and distribution components, as well as their coordination:

- Production is organised as a job shop system. Each machine processes at most one operation at a time, and operations are non-preemptive. Operations already in progress at a dynamic event cannot be interrupted.
- The fleet is heterogeneous (electric and combustion vehicles), with fixed load capacities. A single depot serves as the origin and destination for all vehicles. Each vehicle may perform multiple successive tours, but an ongoing tour cannot be interrupted; the vehicle becomes available only after completing its current tour and returning to the depot.
- Each customer order must be delivered entirely by a single vehicle; split deliveries are not allowed.
- Energy and fuel consumption depend linearly on the transported load.
- Electric vehicles may recharge at designated stations during their routes. Recharge times depend on the battery level upon arrival.
- For electric vehicles, fuel consumption is set to zero; for combustion vehicles, battery-related parameters are ignored and fuel capacity is assumed to be sufficient throughout the planning horizon.
- Each job i becomes available only after its release date rd_i.
- Each job corresponds to a unique customer node in the routing graph. The depot, charging infrastructure, and fleet remain unchanged over time.

B. Production Subproblem

Let M be the set of machines and N_c the set of jobs (customer orders). Each job $i \in N_c$ is associated with a predefined sequence of machines $\text{Seq}_i = (m_{i,1}, m_{i,2}, \ldots, m_{i,|\text{Seq}_i|})$, where $m_{i,k} \in M$. A given machine may appear in the sequences of multiple jobs, possibly at different positions. These machine sequences correspond to operations. Accordingly, each operation is defined as a pair (i,m), where $i \in N_c$ and $m \in \text{Seq}_i$. The set of all operations is denoted by: $O = \{(i,m) : i \in N_c,\ m \in \text{Seq}_i\}$. Operations are non-preemptive, and each machine can process at most one operation at a time. For each job i, operations must respect the technological order induced by Seq_i. Each job $i \in N_c$ is also associated with a release date rd_i and a due date dd_i.

In a standard JSP formulation, a feasible schedule induces, for each machine $m \in M$, a processing sequence π_m specifying the order in which operations assigned to machine m are executed. In this work, we adopt an alternative representation based on a global sequence of operations, denoted by Π, which defines a total order over the set O. This sequence is compatible with the technological constraints of each job, and induces the machine-wise sequences π_m by projection onto the subset of operations assigned to each machine.

Given a global sequence Π, the start times of operations are determined by respecting both: (i) the machine cannot process simultaneously two operations, and (ii) the precedence constraints within each job. Let $S_{i,m}$ denote the start time of operation (i,m), and let $p_{i,m}$ be its processing time. The completion time of job i is defined as: $E_i = S_{i,\ lm_i} + p_{i,\ lm_i}$, where lm_i denotes the last machine in Seq_i. The value E_i represents the earliest time at which job i becomes available for delivery.

C. Distribution Subproblem

The distribution component is modelled as an EVRPTW. We consider a complete directed graph $G = (N,E)$, where $N = N_c \cup \{0\} \cup F$ comprises customer nodes N_c (job and client share the same representation: a job corresponds to a client), the depot 0, and a set of charging stations F. Each edge $(i,j) \in E$ is characterised by a travel distance D_{ij} and a travel time t_{ij}. In addition to the release date and due date, each customer $i \in N_c$ is associated with a demand d_i and an earliest delivery date edd_i. Each vehicle $k \in V$ has a load capacity Q_k, a battery capacity B_k, and a recharge rate σ_k. The energy and fuel consumption rates when empty are denoted h_k^0 and ρ_k^0, and when fully loaded h_k^* and ρ_k^*.

Let $R_k^\ell = \{v_1, v_2, \ldots, v_{|R_k^\ell|}\}$ denote the sequence of clients visited in the ℓ-th tour of vehicle k. The start time of a tour ℓ of vehicle k is defined as

$$t_{\ell k}^{\text{start}} = \begin{cases} \max\left(0, \max_{i \in R_k^1} E_i\right) & \text{if } \ell = 1, \\ \max\left(t_{k,v_{|R_k^{\ell-1}|}}^{\text{arr}} + s_{v_{|R_k^{\ell-1}|}} + t_{v_{|R_k^{\ell-1}|},0},\ \max_{i \in R_k^\ell} E_i\right) & \text{if } \ell > 1 \end{cases}$$

where $v_{|R_k^{\ell-1}|}$ is the last client of the previous tour.

Arrival times at customers within tour ℓ are computed sequentially as

$$\begin{aligned} t_{k,v_1}^{\text{arr}} &= t_{\ell k}^{\text{start}} + t_{0,v_1}, \\ t_{k,v_j}^{\text{arr}} &= t_{k,v_{j-1}}^{\text{arr}} + s_{v_{j-1}} + t_{v_{j-1},v_j}, \quad j > 1. \end{aligned}$$

This formulation ensures that:

- The tour cannot start before all jobs of its clients are completed.

- The tour start time also accounts for the return from the previous tour to the depot.
- Arrival times at customers are computed sequentially, including travel and service times.

D. Objective

The global objective of the integrated production–distribution problem is to *minimize the maximum delay*:

$$T_{\max} = \max_{i \in N_c} (t_{k,i}^{\text{arr}} - dd_i) \text{ with } k \in V.$$

And *the CO_2 emissions* generated during vehicle routing. For each combustion vehicle $k \in V_{\text{comb}}$, the emissions associated with traversing arc (i, j) depend on the travel time and on the load carried when departing from node i, denoted $\text{load}_k(i)$. Assuming a linear fuel consumption rate between ρ_0^k and ρ_k^*, the total emissions are computed as:

$$\text{CO}_2 = \sum_{k \in V_{\text{comb}}} \sum_{(i,j) \in \mathscr{R}_k} t_{ij} \cdot \left(\rho_0^k + \frac{\rho_k^* - \rho_0^k}{Q_k} \cdot \text{load}_k(i) \right).$$

Finally, the two criteria are aggregated into a weighted single-objective function:

$$\textit{Minimize} \quad w_1 T_{\max} + w_2 \text{CO}_2.$$

This formulation ensures:

- Coordination between production and delivery, through E_i,
- Feasibility with respect to vehicle capacities and battery constraints,
- A framework suitable for dynamic insertion of newly arrived jobs, which will be handled by the ant-based rescheduling algorithm.

E. Dynamic Rescheduling Framework

We consider a dynamic setting in which new jobs may arrive during the execution of the plan at decision times $t_{\text{event}} \in (0, H)$. At a given event time, the set of operations is partitioned into: (i) completed operations, which are removed from the schedule, (ii) ongoing operations, which cannot be interrupted and must retain their current processing, and (iii) operations that have not yet started. We denote by O_{old} the set of operations that remain to be processed (those that are not yet started), and by O_{new} the set of operations associated with newly arrived jobs. Let N_{old} and N_{new} denote the corresponding sets of customers/jobs.

The current solution is represented by a partial global sequence Π_{event}, obtained by removing completed and ongoing operations from the original sequence. Note that ongoing operations determine updated machine availability times. Thus, for each machine m, let t_m^{avail} denote the ending time of the ongoing operation. For each vehicle k, let t_k^{avail} denote the finishing time of the current tour plus the time required to return to the depot.

The rescheduling problem involves inserting O_{new} into Π_{event}, ensuring:

- Technological precedence constraints within each job,
- Machine availability respecting t_m^{avail},
- Vehicle availability respecting t_k^{avail},

Updated sequences naturally induce new machine-wise sequences π_m and updated delivery routes. Multiple new jobs are inserted jointly, and the procedure repeats at each subsequent event, ensuring continuity over the planning horizon.

III. PROPOSED METHOD

A. General Framework

To guide the construction of solutions, most ACO-based algorithms rely on pheromone trails that store information extracted from the best solutions generated so far, combined with problem-specific heuristic information. In comparison, hybrid schemes that combine distinct ACO variants exploit complementary search mechanisms: (i) the ACS [12] promotes diversification through a random proportional rule and local pheromone decay, while (ii) the MMAS [13] enforces intensification by confining pheromone values within explicit bounds $[\tau_{\min}, \tau_{\max}]$ and restricting reinforcement to elite solutions only. Recent studies on multi-objective ACO highlight that hybrid schemes are more robust in bi-objective problems, especially when the objectives interact during solution construction, as hybrid schemes better maintain search stability and adaptability under dynamic changes [14].

In this work, we propose a Dynamic Hybrid Ant Colony System (D-HACS) which synergistically combines ACS and MMAS within a unified rescheduling framework. At each dynamic event t_{event}, D-HACS operates over $\text{iter}_{\max}$ iterations. At each iteration: (1) a colony of $\text{ant}_{\max}$ ants concurrently constructs complete integrated solutions, as detailed in Sections III-A.2 and III-A.3, using both pheromone matrices $\tau^{\text{INS-JSP}}$, $\tau^{\text{INS-VRP}}$ and problem-specific heuristics; (2) solutions are evaluated, and the global-best solution Sol^* is updated accordingly; (3) pheromone matrices are updated through evaporation and selective reinforcement; (4) a stagnation detection mechanism resets both matrices whenever no improvement is recorded over it_{reset} consecutive iterations, restoring search diversity.

The D-HACS procedure relies on a sequence of coordinated modules, each responsible for updating a specific component of the integrated production–delivery plan at a dynamic event. As detailed in Section II-E, D-HACS operates on a partially completed plan in which all production operations and route prefixes already committed at t_{event} remain frozen. Consequently, the algorithm does not rebuild solutions from scratch but exploits the accumulated pheromone information to adjust only the modifiable subset of operations and routes. A description of the modules in Fig. 1 is provided below.

1) Initialisation: To guide the insertion of new jobs into the existing production sequence and vehicle routes, two pheromone matrices are initialized. The JSP insertion matrix is defined as:

$$\tau^{\text{INS-JSP}} \in \mathbb{R}^{(|N_{\text{old}}|+1) \times |N_{\text{new}}|},$$

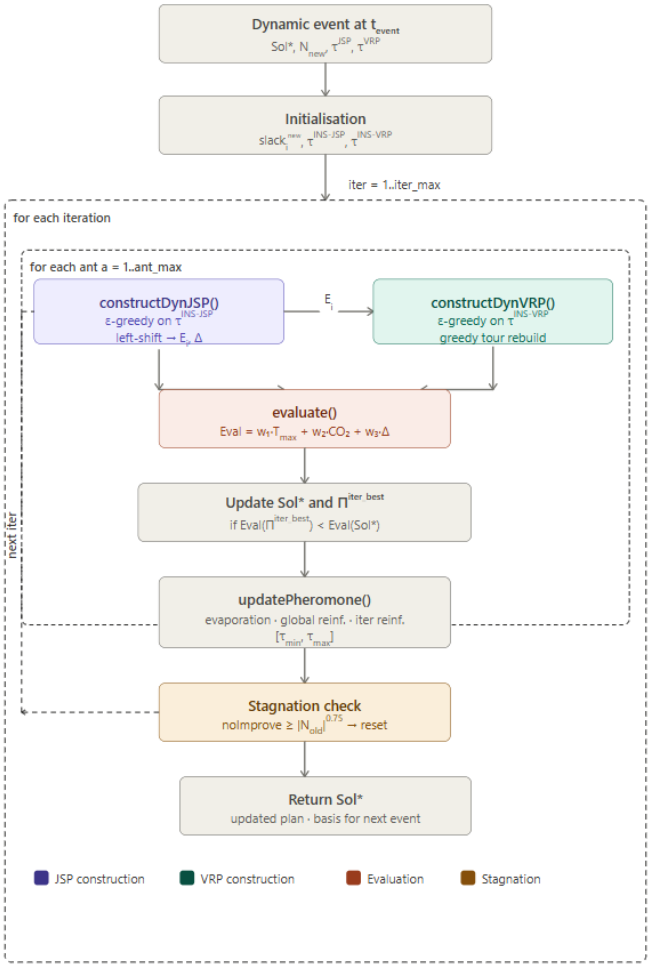

Fig. 1. Flowchart of the proposed algorithm.

where the $(|N_{\text{old}}|+1)$ enumerate all admissible insertion positions in the global sequence Π_{event} defined in Section II-E, namely insertion at the beginning or immediately after any job in N_{old}.

For the routing component, D-HACS follows the dynamic tour structure defined in Section II-C and updated under the rules of Section II-E. Let $\mathscr{R}$ denote the set of reoptimizable tours at time t_{event}, the set of tours associated with vehicles $k \in V$ whose next tour ℓ satisfies

$$t_{\ell k}^{\text{start}} \geq t_k^{\text{avail}}.$$

This restriction reflects the dynamic nature of the problem: at the time an event occurs, vehicles may already be executing an ongoing tour, which cannot be modified. Consequently, newly released customers cannot be inserted into currently active routes, and only tours scheduled to start after the vehicle returns to the depot (i.e., after t_k^{avail}) are considered for reoptimization.

The VRP insertion pheromone matrix is thus:

$$\tau^{\text{INS-VRP}} \in \mathbb{R}^{|\mathscr{R}|\times|N_{\text{new}}|},$$

where each element quantifies the desirability of inserting a newly released customer into a reoptimizable tour of vehicle k.

2) constructDynJSP(): For each ant, the function constructs a feasible partial schedule by inserting the operations of every new job $i_{\text{new}} \in O_{\text{new}}$ into the global sequence inherited from t_{event}. Candidate insertion positions are evaluated using both pheromone information $\tau^{\text{INS-JSP}}$ and a heuristic score:

$$S[j, i_{\text{new}}] = \left(\tau_{j,i_{\text{new}}}^{\text{INS-JSP}}\right)^{\alpha} \left(\eta(j, i_{\text{new}})\right)^{\beta},$$

and the position j^* is selected according to an ε-greedy rule.

Heuristic evaluation: The heuristic score combines the temporal impact of the insertion and the urgency of the new job:

$$\eta(j, i_{\text{new}}) = \mu_1\, \eta_{\text{time}}(j, i_{\text{new}}) + \mu_2\, \eta_{\text{urg}}(i_{\text{new}}), \qquad \mu_1 + \mu_2 = 1,$$

where μ_1 and μ_2 weight the relative influence of both terms.

For temporal impact, let $\mathscr{T}_{\text{impact}}(j, i_{\text{new}})$ denote the set of downstream jobs whose next required machine $m(u)$ satisfies $t_{\text{avail}}^{m(u)} < E_{i_{\text{new}}}$, meaning they experience a shift once i_{new} is inserted at j. The cumulative shift is approximated as:

$$C_{\text{time}}(j, i_{\text{new}}) = \sum_{u \in \mathscr{T}_{\text{impact}}(j, i_{\text{new}})} \left(E_u + p_{i_{\text{new}}, m(u)}\right),$$

$$\eta_{\text{time}}(j, i_{\text{new}}) = \frac{1}{1 + C_{\text{time}}(j, i_{\text{new}})}.$$

Urgency derives directly from slack, promoting jobs with limited temporal margin:

$$\eta_{\text{urg}}(i_{\text{new}}) = \frac{1}{1 + \text{slack}_{i_{\text{new}}}}.$$

Schedule reconstruction: After selecting j^*, the operations of i_{new} are inserted and a left-shift reconstruction is performed: each operation is scheduled as early as possible by scanning its machine sequence, always respecting (i) machine availability t_{avail}^m, (ii) the completion of the preceding operation of the job, and (iii) the non-preemptive machine constraint. Only non-started operations are modified; operations in progress remain fixed until t_{avail}^m.

Pheromone reinforcement: Once the full sequence has been reconstructed for the ant, pheromone values $\tau^{\text{INS-JSP}}$ are updated. For each job i_{new} inserted at position j^*, reinforcement is applied as:

$$\tau_{j^*,i_{\text{new}}}^{\text{INS-JSP}} \leftarrow (1-\rho)\, \tau_{j^*,i_{\text{new}}}^{\text{INS-JSP}} + \rho\, \Delta\tau(i_{\text{new}}),$$

where $\Delta\tau(i_{\text{new}})$ is inversely proportional to the resulting completion time or delay involving i_{new}. This local update integrates the constructive choice made in the JSP component into the global pheromone memory of D-HACS.

3) constructDynVRP(): Given the updated completion times E_i from `constructDynJSP()`, D-HACS inserts each new customer i_{new} into the modifiable suffix of the existing vehicle tours. For each candidate tour $R_k^\ell \in \mathscr{R}$, the algorithm computes:

$$S[R_k^\ell, i_{\text{new}}] = \left(\tau_{R_k^\ell, i_{\text{new}}}^{\text{INS-VRP}}\right)^{\alpha} \left(\eta(R_k^\ell, i_{\text{new}})\right)^{\beta},$$

and selects a tour through the ε-greedy rule:

$$R^* = \begin{cases} \arg\max_{R \in \mathscr{R}} S[R, i_{\text{new}}], & \text{with prob. } 1-\varepsilon, \\ \text{a random tour of } \mathscr{R}, & \text{with prob. } \varepsilon. \end{cases}$$

Heuristic evaluation: The insertion heuristic balances the impact on tour feasibility and the additional routing effort:

$$\eta(R_k^\ell, i_{\text{new}}) = \gamma_1 \, \eta_{\text{tard}} + \gamma_2 \, \eta_{\text{dist}}, \quad \gamma_1 + \gamma_2 = 1.$$

Tardiness component: A tour cannot depart before all its assigned customers have completed production, nor before the vehicle returns from its previous tour. Let $t_{\ell k}^{\text{arr,old}}$ denote the arrival time at the depot after executing the fixed prefix of tour ℓ of vehicle k in the pre-update solution. The start time of the tour before insertion is:

$$t_{\ell k}^{\text{start,old}} = \max\Big(t_{\ell k}^{\text{arr,old}}, \max_{i \in R_k^\ell} E_i\Big).$$

Inserting i_{new} modifies the latest production completion time, yielding:

$$\Delta t_{\text{start}}(R_k^\ell, i_{\text{new}}) = \max\big(E_{i_{\text{new}}}, \max_{i \in R_k^\ell} E_i\big) - \max_{i \in R_k^\ell} E_i.$$

The tardiness heuristic is therefore:

$$\eta_{\text{tard}}(R_k^\ell, i_{\text{new}}) = \frac{1}{1 + \Delta t_{\text{start}}(R_k^\ell, i_{\text{new}})}.$$

Distance component: To approximate routing impact, the insertion of i_{new} is evaluated at all feasible positions h of the modifiable suffix of R_k^ℓ. Let (v_{h-1}, v_h) be a pair of consecutive customers; the local additional distance is:

$$\Delta D_h = D_{v_{h-1}, i_{\text{new}}} + D_{i_{\text{new}}, v_h} - D_{v_{h-1}, v_h}.$$

The distance-based heuristic selects the best insertion position:

$$\eta_{\text{dist}}(R_k^\ell, i_{\text{new}}) = \frac{1}{1 + \min_h \Delta D_h}.$$

Tour reconstruction: Once R^* is selected, the algorithm inserts i_{new} at the position h^* minimizing ΔD_h. Arrival times are then recomputed, as defined in subsection II-E.

Only non-started legs of the tour are modified, ensuring consistency with the dynamic rules of Section II-E. After reconstruction, D-HACS proceeds to the next customer in N_{new} until all new clients have been assigned.

Pheromone reinforcement: After constructing a full VRP solution for the ant, the pheromone matrix $\tau^{\text{INS-VRP}}$ is updated. For the chosen tour R^* and position h^* where i_{new} was inserted, reinforcement is applied:

$$\tau_{R^*, i_{\text{new}}}^{\text{INS-VRP}} \leftarrow (1-\rho)\, \tau_{R^*, i_{\text{new}}}^{\text{INS-VRP}} + \rho \, \Delta\tau_{\text{VRP}}(i_{\text{new}}),$$

where $\Delta\tau_{\text{VRP}}(i_{\text{new}})$ is inversely proportional to the delay or distance increase induced by the chosen insertion:

$$\Delta\tau_{\text{VRP}}(i_{\text{new}}) = \frac{1}{1 + \Delta t_{\text{start}}(R^*, i_{\text{new}}) + \Delta D_{h^*}}.$$

This integrates routing decisions into the pheromone memory.

IV. COMPUTATIONAL RESULTS

To evaluate the performance of our D-HACS metaheuristic for the IPDP under dynamic operational conditions, we generated several families of test instances ranging from 5 to 100 customers, allowing for a comprehensive assessment of scalability and robustness under increasing levels of problem complexity, while introducing controlled uncertainty in customer availability. In all instances, only **20%** of customers are initially known at time $rd_i = 0$, while the remaining **80%** are released gradually over the planning horizon $H = [0, 100]$ to emulate real-world dynamic order arrivals. The release dates of delayed customers are drawn from two distinct stochastic schemes designed to capture different temporal demand structures. In the *Uniform (U)* scenario, all delayed customers receive strictly positive release dates sampled from a uniform distribution $U(0, 100)$, producing a steady and evenly distributed arrival pattern throughout the horizon. In contrast, the *Poisson-based (P)* scenario aims to reproduce bursty, highly variable demand: two random time points within H are selected, around which concentrated waves of customer arrivals are generated using a Poisson process with rate $\lambda = 10$, leading to clustered peaks of activity separated by quieter intervals.

TABLE I

COMPARISON OF $T_{\max}$ AND CO_2 EMISSIONS: D-HACS VS. CPLEX

Instances	**D-HACS – T_{max}**				**D-HACS – CO_2**			
	Avg	min	max	CPLEX	Avg	min	max	CPLEX
P_c5_m10_c_1	116.34	115.71	117.42	113	133.34	131.95	133.90	130
P_c5_m10_c_2	73.73	73.08	74.16	72	133.65	131.95	133.90	130
P_c5_m10_r_1	64.40	63.94	64.89	63	0.00	0.00	0.00	0
P_c5_m10_r_2	73.92	73.08	74.16	72	60.95	60.90	61.80	60
P_c5_m10_rc_1	79.40	79.17	80.34	78	99.02	98.45	99.91	97
P_c5_m10_rc_2	42.94	42.63	43.26	42	102.80	102.51	104.03	100
P_c5_m5_c_1	86.88	86.27	87.55	84	266.77	264.91	268.83	261
P_c5_m5_c_2	153.47	152.25	154.50	149	267.61	264.91	268.83	261
P_c5_m5_r_1	85.94	85.26	86.52	83	61.06	60.90	61.80	60
P_c5_m5_r_2	153.89	152.25	154.50	150	61.02	60.90	61.80	60
P_c5_m5_rc_1	85.74	85.26	86.52	84	129.56	127.89	129.78	126
P_c5_m5_rc_2	131.58	130.93	132.87	129	99.89	98.45	99.91	97
U_c5_m10_c_1	28.80	28.42	28.84	28	133.51	132.96	134.93	130
U_c5_m10_c_2	50.22	49.73	50.47	49	270.12	266.94	270.89	261
U_c5_m10_r_1	60.65	59.88	60.77	59	0.00	0.00	0.00	0
U_c5_m10_r_2	34.65	34.51	35.02	34	0.00	0.00	0.00	0
U_c5_m10_rc_1	46.30	45.67	46.35	45	99.24	98.45	99.91	97
U_c5_m10_rc_2	91.36	90.33	91.67	89	0.00	0.00	0.00	0
U_c5_m5_c_1	126.46	124.84	126.69	122	269.55	266.94	270.89	261
U_c5_m5_c_2	97.28	96.42	97.85	94	267.37	266.94	270.89	261
U_c5_m5_r_1	93.41	92.36	93.73	91	0.00	0.00	0.00	0
U_c5_m5_r_2	134.97	133.98	135.96	131	56.65	55.82	56.65	54
U_c5_m5_rc_1	66.57	65.97	66.95	65	98.71	98.45	99.91	97
U_c5_m5_rc_2	123.31	121.80	123.60	119	102.67	101.50	103.00	100
Average	86.70	86.27	87.55	85.21	112.06	111.65	113.30	110.13
P_c10_m10_c_1	252.19	248.85	255.96	237	364.70	357.00	367.20	340
P_c10_m10_c_2	233.01	231.00	237.60	220	489.00	475.65	489.24	453
P_c10_m10_r_1	216.36	213.15	219.24	203	395.64	389.55	400.68	371
P_c10_m10_r_2	204.76	202.65	208.44	193	268.69	263.55	271.08	251
P_c10_m10_rc_1	294.26	286.65	294.84	273	580.58	572.25	588.60	545
P_c10_m10_rc_2	240.61	236.25	243.00	225	657.77	643.65	662.04	613
P_c10_m5_c_1	430.68	422.10	434.16	402	285.35	279.30	287.28	266
P_c10_m5_c_2	317.68	311.85	320.76	297	389.19	381.15	392.04	363
P_c10_m5_r_1	392.55	389.55	400.68	371	191.61	186.90	192.24	178
P_c10_m5_r_2	288.42	286.65	294.84	273	301.41	299.25	307.80	285
P_c10_m5_rc_1	377.19	375.90	386.64	358	386.97	376.95	387.72	359
P_c10_m5_rc_2	301.59	300.30	308.88	286	557.12	544.95	560.52	519
U_c10_m10_c_1	293.77	287.70	295.92	274	360.68	353.85	363.96	337
U_c10_m10_c_2	233.48	233.10	239.76	222	412.33	402.15	413.64	383
U_c10_m10_r_1	266.13	264.60	272.16	252	298.52	295.05	303.48	281
U_c10_m10_r_2	224.65	220.50	226.80	210	210.72	210.00	216.00	200
U_c10_m10_rc_1	239.90	237.30	244.08	226	381.91	373.80	384.48	356
U_c10_m10_rc_2	168.67	164.85	169.56	157	504.72	500.85	515.16	477
U_c10_m5_c_1	423.40	423.15	435.24	403	262.21	259.35	266.76	247
U_c10_m5_c_2	304.34	299.25	307.80	285	386.08	381.15	392.04	363
U_c10_m5_r_1	351.90	349.65	359.64	333	344.50	340.20	349.92	324
U_c10_m5_r_2	299.05	295.05	303.48	281	264.14	263.55	271.08	251
U_c10_m5_rc_1	322.24	321.30	330.48	306	411.38	409.50	421.20	390
U_c10_m5_rc_2	304.81	302.40	311.04	288	549.35	544.95	560.52	519
Average	294.60	287.66	295.88	273.96	383.27	379.36	390.20	361.29

Table I shows that D-HACS achieves consistently strong performance across both the 5- and 10-customer dynamic instance families. For the 5-customer instances specifically, the method remains extremely close to the optimal CPLEX

solutions: across all 24 dynamic cases, the average deviation is approximately 2.8% for $T_{\max}$ and 2.7% for CO_2, with no instance exceeding a 4% difference. These consistently small gaps demonstrate that D-HACS can absorb high levels of dynamism without degrading solution quality at this problem scale. For the 10-customer instances, the deviations between D-HACS and CPLEX naturally increase as the problem size grows and 80% of customers are revealed during execution, but the method nevertheless maintains coherent and predictable performance patterns across all 24 cases for both $T_{\max}$ and CO_2, reflecting stable scalability under more demanding dynamic conditions. The results also confirm the prioritization embedded in the model, as the maximum delay $T_{\max}$ remains systematically closer to the CPLEX results than CO_2. This indicates that D-HACS effectively controls lateness and preserves temporal feasibility even under substantial dynamism.

TABLE II

COMPARISON OF C-HACS AND D-HACS PERFORMANCE ($T_{\max}$ | CO_2)

Family	C-HACS		D-HACS		Factors (vs. C-HACS)	
	med	SD	med	SD	med	SD
c5	90.2 | 111.6	1.0 | 0.4	92.0 | 113.8	1.2 | 0.5	× 1.02 | × 1.02	× 1.20 | × 1.25
c10	291.4 | 362.8	3.5 | 0.4	300.1 | 373.7	4.7 | 0.6	× 1.03 | × 1.03	× 1.35 | × 1.40
c20	479.1 | 633.3	5.3 | 7.9	503.1 | 658.6	8.2 | 12.6	× 1.05 | × 1.04	× 1.55 | × 1.60
c30	681.5 | 728.6	8.1 | 7.3	722.4 | 765.0	14.6 | 13.5	× 1.06 | × 1.05	× 1.80 | × 1.85
c50	1037.5 | 1079.6	6.7 | 8.3	1120.5 | 1155.2	14.7 | 19.1	× 1.08 | × 1.07	× 2.20 | × 2.30
c100	2069.6 | 2029.5	36.1 | 26.8	2276.6 | 2212.2	93.9 | 73.7	× 1.10 | × 1.09	× 2.60 | × 2.75
All	774.9 | 824.2	10.1 | 8.5	835.8 | 879.8	22.9 | 20.0	× 1.06 | × 1.05	× 1.95 | × 1.86

Table II provides a comparative analysis between C-HACS and D-HACS across instance families ranging from 5 to 100 customers under highly dynamic conditions. C-HACS represents a baseline obtained by testing our algorithm in a static context, assuming complete knowledge of all customer orders in advance. The results highlight the robustness of D-HACS in handling dynamic uncertainty, with an average performance degradation of only 7.8% for $T_{\max}$ and 6.7% for CO_2 emissions compared to the static case. For small instances (c5 and c10), the gap is minimal (between 2 and 3%), indicating that D-HACS efficiently manages real-time decision-making with near-optimal solutions. As instance size increases, the effect of dynamic arrivals becomes more pronounced; however, the degradation remains controlled, reaching around 8 to 10% for $T_{\max}$ and CO_2 in the largest instances, confirming the algorithm's scalability. Regarding solution stability, the dynamic setting primarily affects variability rather than central tendency. The standard deviation increases by factors ranging from approximately 1.2× for small instances to more than 2.5× for large-scale instances, reflecting the inherent stochasticity introduced by dynamic arrivals. Nevertheless, the close alignment between median values across both approaches indicates that D-HACS preserves a consistent solution structure. These findings validate that D-HACS achieves a favorable trade-off between solution quality and computational responsiveness in dynamic production-delivery environments.

V. CONCLUSIONS

This work introduced D-HACS, a Dynamic-Hybrid Ant Colony System developed to address the IPDP with a mixed fleet of electric and combustion vehicles, enabling the joint rescheduling of production and routing decisions under real-time order arrivals. Experiments on newly generated benchmarks show that D-HACS provides consistently high-quality solutions, remaining very close to CPLEX for small instances and maintaining stable performance as dynamism and problem size increase. By integrating electric vehicle constraints within a dynamic rescheduling framework, this study establishes a solid foundation for more responsive and sustainable production–delivery systems. Future work will aim to enhance scalability and robustness by exploiting the learning mechanisms already embedded in D-HACS. In particular, the pheromone–based adaptation process could be extended through a multi-agent architecture, enabling predictive reactions to dynamic events and improved coordination between production and routing decisions in highly uncertain environments.

ACKNOWLEDGMENT

This work was supported by the French Grand Est region and CESI Engineering School, France.

REFERENCES

[1] M. Delgado-Lindeman, R. Cordera, J. Moura, and A. Rodriguez, "Characteristics and effects of low emission zones in Europe: A systematic literature review", European Transport Research Review, vol. 17, 2025.

[2] Health Effects Institute, "State of Global Air 2024", Special Report, Boston, MA, 2024.

[3] E. Yağmur and S. E. Kesen, "Bi-objective coordinated production and transportation scheduling problem with sustainability: formulation and solution approaches", International Journal of Production Research, vol. 61, no. 3, pp. 774–795, 2023.

[4] J. Zheng, Y. Zhao, J. Li, W. She and Y. Li, "Integrated scheduling of material delivery and processing", Computers & Industrial Engineering, Elsevier, 2025.

[5] T. Liang, L. Zhou, and Z. Jiang, "Integrated scheduling of production and material delivery for the intelligent manufacturing systems", International Journal of Production Research, vol. 63, pp. 1–22, 2024.

[6] T. Luo, Y. Heng, L. Xing, T. Ren, Q. Li, H. Qin, Y. Hou and K. Wang, "A two-stage approach for electric vehicle routing problem with time windows and heterogeneous recharging stations", Tsinghua Science and Technology, vol. 29, no. 5, pp. 1300–1322, 2024.

[7] V. F. Yu and P. T. Anh, "The electric vehicle routing problem with time windows, partial recharges, and covering locations", International Transactions in Operational Research, 2025.

[8] S. K. Fuladi and C.-S. Kim, "Dynamic events in the flexible job-shop scheduling problem: Rescheduling with a hybrid metaheuristic algorithm", Algorithms, vol. 17, no. 4, p. 142, 2024.

[9] Y. Zhu, Z. Liu, F. Li, and J. Wang, "Online integrated production–transportation in a make-to-order environment", Journal of Scheduling, 2025.

[10] Z.-L. Chen, "Online integrated production and distribution scheduling: Review and extensions", INFORMS Journal on Computing, vol. 37, no. 2, pp. 360–380, 2024.

[11] S. Yang, J. Wang, W. Li, and Z. Xu, "Integrated optimization of dynamic scheduling and reconfiguration for distributed reconfigurable flowshops via iterated greedy algorithm", International Journal of Systems Science: Operations & Logistics, vol. 12, 2025.

[12] M. Dorigo and L. M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53–66, 1997.

[13] T. Stützle and H. H. Hoos, "MAX–MIN Ant System", Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000.

[14] M. A. Awadallah, S. N. Makhadmeh, M. A. Al-Betar, L. M. Dalbah, A. Al-Redhaei, S. Kouka and O. S. Enshassi, "Multi-objective ant colony optimization: Review", Archives of Computational Methods in Engineering, vol. 32, no. 2, pp. 995–1037, 2025.