Les Cahiers du GERAD ISSN: 0711–2440

A logic-based Benders decomposition for collaborative distribution with shared transfer hubs and coordinated routing

D. Ocampo-Giraldo, A. M. Anaya-Arenas, W. Klibi, J. Ortmann

G–2026–75

October 2026

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : D. Ocampo-Giraldo, A. M. Anaya-Arenas, W. Klibi, J. Ortmann (Octobre 2026). A logic-based Benders decomposition for collaborative distribution with shared transfer hubs and coordinated routing, Rapport technique, Les Cahiers du GERAD G– 2026–75, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2026-75) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: D. Ocampo-Giraldo, A. M. Anaya-Arenas, W. Klibi, J. Ortmann (October 2026). A logic-based Benders decomposition for collaborative distribution with shared transfer hubs and coordinated routing, Technical report, Les Cahiers du GERAD G–2026–75, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2026-75) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A logic-based Benders decomposition for collaborative distribution with shared transfer hubs and coordinated routing

Daniel Ocampo-Giraldo [a, b, c, d]

Ana María Anaya-Arenas [a, c, d]

Walid Klibi [d, e]

Janosch Ortmann [a, b, c, f]

[a] *Department of Analytics, Operations and IT (AOTI), School of Management (ESG), UQAM, Montréal, Canada*

[b] *Groupe d'études et de recherche en analyse des décision (GERAD), Montréal, Canada*

[c] *Research Centre for Smart2 Management of Complex Systems (CRI2GS), Montréal, Canada*

[d] *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montréal, Canada*

[e] *The Centre of Excellence in Supply Chain (CESIT), Kedge Business School, Bordeaux, France*

[f] *Centre de recherches mathématiques (CRM), Montréal, Canada*

ocampo_giraldo.daniel_mauricio@courrier.uqam.ca
anaya-arenas.ana_maria@uqam.ca
ortmann.janosch@uqam.ca

October 2026
Les Cahiers du GERAD
G–2026–75

Copyright © 2026 GERAD, Ocampo-Giraldo, Anaya-Arenas, Klibi, Ortmann

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : The Collaborative Routing Orchestration for Organic Food Distribution Problem (CRO-FDP) integrates shared hub activation, vehicle routing and commodity transfers in a collaborative distribution network for perishable goods. Transfers at shared hubs create temporal dependencies between vehicle routes, which must remain compatible with commodity release times and delivery deadlines. These dependencies tightly couple routing and scheduling decisions, making large instances difficult to solve to optimality.

To overcome this computational bottleneck, we develop a logic-based Benders decomposition tailored to collaborative distribution with shared transfer hubs and coordinated routing. A task-based master problem determines hub activation, vehicle sequences, and commodity transport decisions, while an auxiliary graph evaluates their temporal feasibility. The method is implemented in a branch-and-check scheme, using feasibility cuts to refine the master when temporal infeasibilities are identified. Spatial clustering and bin packing support initialization. Computational experiments on 100 instances compare the proposed approach with the Interval Relaxation (IR) method introduced in our previous work. Computational results show that the decomposition approach substantially outperforms the previous IR approach. For medium instances, the decomposition increases the proportion of optimal solutions from 10.6% to 68.1% and reduces the average optimality gap from 11.7% to 3.5%. For large instances, the decomposition reduces the average gap from 25.3% to 14.2%, demonstrating improved scalability and solution quality over IR.

Keywords: Collaborative routing; organic food distribution; vehicle synchronization; logic-based Benders decomposition; branch-and-check; perishable logistics

1 Introduction

The regional distribution of organic and local food has become a critical pillar of sustainable development, driven by increasing consumer demand for fresh, high-quality products with a lower environmental footprint (Baez et al., 2020). However, the agro-food sector faces structural logistical challenges that frequently compromise its economic viability. Small and medium-sized producers are typically fragmented and handle low volumes, leading to inefficient distribution operations characterized by vehicle load rates that often drop below 50% (Bosona and Gebresenbet, 2011). These inefficiencies are exacerbated by the extreme perishability of fresh produce, which requires high-frequency deliveries and strict temporal control to prevent quality degradation and food loss (Castro et al., 2019).

Horizontal logistics collaboration has emerged as an effective organizational paradigm to overcome these barriers. By sharing transportation capacity and intermediate transshipment hubs among independent producers operating at the same tier of the supply chain, collaborative networks can achieve significant economies of scale, improve consolidation, and reduce empty mileage and greenhouse gas emissions (Gharehyakheh et al., 2020). The Collaborative Routing Orchestration for Organic Food Distribution Problem (CRO-FDP) is grounded in the concrete operational challenges of the *Table de Concertation Bioalimentaire des Laurentides* (TCBL), a non-profit association coordinating independent growers, food processors, commercial retailers, and regional food banks in Quebec, Canada (Ocampo-Giraldo et al., 2026). Within this collaborative framework, the orchestrator simultaneously coordinates the selection of volunteer member facilities that serve as shared cross-docking transfer hubs and coordinates vehicle routes to satisfy pickup availability dates and tight delivery deadlines. A fundamental characteristic of this system is that hubs act purely as intermediate transshipment platforms with intermediate staging areas but no long-term or cold storage infrastructure, imposing strict temporal precedence between inbound and outbound vehicles.

From an operational research and computational standpoint, the CRO-FDP is an exceptionally difficult combinatorial problem. It tightly integrates multi-echelon vehicle routing, facility location, and continuous-time vehicle synchronization. In our previous work (Ocampo-Giraldo et al., 2026), this challenge was formulated through a monolithic time-expanded network (MIP-TE) and addressed via an Interval Relaxation (IR) matheuristic. While the IR method successfully identified feasible solutions for realistic instances, its bounding performance exposed severe computational bottlenecks inherent to time discretization: although small instances were largely solved to optimality, the average optimality gap climbed to 25.3% (and exceeded 42% on complex subsets) for large instances. These findings underscored that embedding high-resolution time grids directly into multi-echelon vehicle routing creates a combinatorial explosion that severely limits the dual bounding power and scalability of standard mathematical programming approaches.

To eliminate this computational hurdle, this research develops an exact decomposition framework rooted in the theory of Logic-Based Benders Decomposition (LBBD) (Hooker and Ottosson, 2003). Rather than relying on classical Benders decomposition—which requires continuous linear subproblems and fails to handle the combinatorial and precedence structure of cross-docking transfers—or resorting to standard echelon-based splits that preserve time-indexed routing within each tier, we completely decouple the spatial and routing decisions from the temporal scheduling dimension. The spatial decisions (hub activation, vehicle stop sequences, and commodity flow assignments) form a master problem termed the Vehicle Task Sequencing Problem (VTSP), formulated over a task-extended network where the continuous time horizon is entirely projected out. The temporal feasibility of candidate vehicle tours and inter-vehicle hub transfers is subsequently evaluated by an auxiliary subproblem formulated as a Temporal Constraint Graph (TCG) over a directed distance graph. Feasibility is assessed in polynomial time through shortest-path and negative-cycle detection via the Bellman-Ford algorithm, which functions as the inference dual of the LBBD scheme. When a negative cycle is encountered, it provides an exact combinatorial certificate of temporal infeasibility, which is converted into a logic-based feasibility cut that eliminates the conflicting combination of spatial moves and commodity assignments from the master problem. The framework is implemented within a single-tree Branch-and-Check

(BAC) architecture via solver callbacks at integer nodes, avoiding repeated restarts of the master problem (Thorsteinsson, 2001).

Relative to our previous CRO-FDP baseline (Ocampo-Giraldo et al., 2026) and existing LBBD applications in vehicle routing (Fachini and Armentano, 2020; Zhang et al., 2021; Fragkogios et al., 2024), the specific methodological contributions of this paper are as follows:

1. We introduce an exact decomposition for two-echelon collaborative routing with shared cross-docking that projects out the continuous time grid entirely from the master problem. The spatial master problem (VTSP) captures all cost-bearing decisions over a task-extended network, ensuring that the master objective function carries the entire operational cost and supplies a strict dual lower bound for the global problem.
2. We establish that temporal scheduling with one-way transfer precedence and discrete time windows reduces to cycle detection in a distance graph. By solving the LBBD inference dual via the Bellman-Ford algorithm, we extract minimal negative cycles and construct logic-based feasibility cuts that project temporal conflicts directly back into the master decision space without enumerating time steps.
3. To establish tight search bounds and prevent solver stalling, we design a preprocessing and initialization heuristic based on spatial clustering and greedy bin packing (detailed in Appendix B). This procedure aggregates compatible requests into Abstract Commodities to derive a high-quality initial primal bound and restrict the maximum task parameter ω and fleet size without discarding optimal solutions.
4. Computational experiments on 100 realistic benchmark instances demonstrate a structural performance improvement over both the monolithic MIP-TE model and the previous IR approach. For medium instances, the LBBD algorithm increases the proportion of proven optimal solutions from 10.6% to 68.1% while slashing the average optimality gap from 11.7% to 3.5%. For large instances, the average gap is reduced from 25.3% to 14.2% while achieving a 100% feasibility rate, confirming that spatial–temporal LBBD effectively overcomes the scalability barriers of collaborative multi-echelon distribution.

The remainder of this paper is organized as follows: Section 2 provides a review of the relevant literature on multi-echelon routing, synchronization, and Benders decomposition paradigms. Section 3 formalizes the problem definition of the CRO-FDP, detailing network assumptions, operational rules, and an illustrative comparison between direct and hub-based distribution. Section 4 presents the core LBBD framework, encompassing the VTSP master problem, the TCG subproblem, and the generation and proof of validity of negative-cycle feasibility cuts. Section 5 outlines the complete Branch-and-Check algorithm and its bounding properties. Section 6 reports the computational results and performance comparison against the IR and MIP-TE methods across all benchmark classes. Finally, Section 7 concludes the paper and discusses avenues for future research.

2 Literature review

The CRO-FDP lies at the intersection of multi-echelon routing, facility location, and strict temporal coordination. This review explores the methodological foundations necessary to address these complexities.

The Two-Echelon Location-Routing Problem (2E-LRP) represents a foundational challenge in modern logistics due to the inherent interdependence between strategic facility location and operational multi-tier routing decisions (Contardo et al., 2012). Because the problem is NP-hard, research has historically focused on balancing computational tractability with solution quality through both exact and heuristic methods. Early exact approaches, primarily based on Branch-and-Cut algorithms, were typically restricted to small-scale instances of approximately 20 to 50 customers (Crainic et al.,

2023; Rahmani, 2015). However, recent advancements in Branch-Cut-and-Price frameworks have significantly pushed these boundaries, enabling the optimal solution of instances involving up to 200 customers (Sadykov, 2019). Despite these gains, the vast scale of urban distribution networks often necessitates the use of metaheuristics. Adaptive Large Neighborhood Search has emerged as the state-of-the-art heuristic for the 2E-LRP, utilizing a "destroy and repair" logic to explore complex search spaces (Dereli et al., 2025; Rahmani, 2015). Other successful metaheuristic strategies include Tabu Search and Large Neighborhood Search, which frequently decompose the problem into sequential stages of facility location and multi-depot routing (Contardo et al., 2012; Lyu, 2020; Winkenbach et al., 2016).

A critical complexity within these multi-echelon systems is the requirement for synchronization at intermediate satellites, which function as dynamic transfer points rather than traditional warehouses (Mirhedayatian et al., 2021). In the context of organic food distribution, where product perishability is a primary concern, the need for precise temporal and spatial coordination is paramount (Soares et al., 2024). This synchronization typically manifests as temporal precedence, where second-echelon vehicles must wait for first-echelon deliveries to arrive before beginning their local distribution rounds (Mirhedayatian et al., 2021). In high-intensity scenarios with minimal satellite storage, researchers have developed models such as the Two-Echelon Multiple-Trip VRP with Satellite Synchronization (2E-MTVRP-SS), which use time-expanded formulations to ensure that vehicles from both tiers arrive simultaneously for cross-docking (Grangier et al., 2016). To manage the operational delays inherent in these synchronized handoffs, recent studies have proposed the use of designated waiting stations for city freighters and the organization of distribution into structured "delivery waves" to maximize hub throughput (Cleophas et al., 2019; Mirhedayatian et al., 2021).

To handle the resulting computational burden of integrated routing and timing decisions, decomposition methods are employed to decouple the overlapping layers of the 2E-LRP. Most frameworks treat the problem as a superposition of individual tier-specific routing problems linked by location decisions at the satellites (Contardo et al., 2012). Exact decomposition techniques, such as Benders decomposition, are particularly effective for separating strategic location choices from operational routing, using combinatorial cuts to eliminate infeasible configurations from the search space (Mohamed et al., 2023; Sadykov, 2019). This approach is especially valuable in stochastic or multi-period environments where uncertainty complicates the primary routing subproblems. Beyond mathematical decomposition, many researchers adopt hierarchical "cluster-first-route-second" strategies, which simplify the problem by clustering customers to identify optimal satellite locations before performing local route optimization (Dereli et al., 2025). Furthermore, within heuristic development, route-based decomposition—where subproblems are defined by the customers assigned to specific routes—has been shown to outperform more rigid path-based methods (Santini et al., 2023). These decomposition strategies remain essential for orchestrating the large-scale collaborative networks required for efficient urban logistics.

While these decomposition methods successfully separate strategic location from spatial routing—or decouple first-echelon from second-echelon routes—they still heavily rely on the assumption that temporal operations can occur independently. Existing decomposition schemes are not designed to handle the continuous-time dependencies required when independent actors share transfer hubs without storage capacity, which forces synchronization. Our proposed methodology builds upon the philosophy of exact decomposition but with a novel split: rather than decomposing by echelon, we decouple the spatial routing decisions (VTSP) entirely from the continuous-time scheduling (VTAP), utilizing logic-based valid cuts to bridge the two dimensions.

2.1 Benders decomposition in vehicle routing and location-routing

Benders decomposition approaches in vehicle routing (VRP) and location-routing problems (LRP) can be classified into two primary structural paradigms:

Classical benders decomposition is used when fixing binary master decisions leaves a continuous linear program (LP) with strong duality. Fragkogios et al. (2024) solves the Multi-Trip Time-Dependent Vehicle Routing Problem with Time Windows where the master problem Selects the customer sequence and the discrete time intervals for visits, treating travel times as partially time-dependent, and the subproblem solves a continuous linear problem with no duality gap that calculates the exact departure times and full time-dependency. Zhang et al. (2025) use a benders decomposition approach to solve Electric Vehicle Charging Station Location-Routing. The master problem Manages the integer decisions for constructing charging stations within a designated budget. The subproblem turns on a linear minimum cost network flow problem over a space-time-electricity network that evaluates transportation, fuel, charging, and waiting states. Classical Benders optimality and feasibility cuts derived from flow balance duals. These are injected dynamically via callbacks within a single Branch-and-Benders-cut (B&BC) tree. A benders has been also proposed to solve the Perishable Inventory Routing Problem. Alkaabneh et al. (2020) solves a integer problem defining vehicle routes across varying time periods, and a continuous dual linear problem evaluating holding costs, product shortages, and greenhouse gas emissions. For the Hub Location Problem Contreras et al. (2011) decompose the problem into an integer program focused on the strategic location and opening of hub facilities, and a continuous semi-assignment problem for routing multicommodity flows across the opened hubs, decoupled into independent subproblems for each commodity. A multicut Benders formulation is used to add independent cuts for subsets of commodities. To handle dual degeneracy, the algorithm solves a Pareto-optimal subproblem to generate strong, non-dominated optimality cuts.

Logic-Based benders decomposition applied when subproblems are combinatorial (NP-hard) or non-convex, meaning LP duality fails. Zhang et al. (2021) solves a Production and Inventory Routing problem decomposing the problem into a set partitioning model that determines plant production periods, inventory levels, and which customers are serviced in each period, and a Capacitated Vehicle Routing Problem for each period, solved exactly using Branch-and-Price. LP-based lifted optimality cuts derived from *ng*-path relaxations and valid inequalities added to the master formulation. Fachini and Armentano (2020) propose a logic benders decomposition for Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows. The master problem is a Generalized Assignment Problem (GAP) containing binary variables that allocate each customer to a specific vehicle type and manage capacity limits. The subproblem is an independent Traveling Salesman Problems with Time Windows (TSPTW) mapped to each vehicle. The cuts added are logic-based Benders optimality and feasibility cuts derived from the TSPTW. The algorithm is enhanced using a Branch-and-Check framework and clique valid inequalities added to the master problem to prevent incompatible customer subsets. On a allocation-routing problem, Mahéo et al. (2023) propose a benders decomposition approach for tactical municipal solid waste allocation and routing. The mastaer problem is a routing problem that dictates the collection routes, while the subproblem is a nonlinear resource allocation problem with integer variables used to assign waste bin combinations. Because classical linear duality does not apply to integer subproblems, the framework uses integer L-shaped cuts

3 Problem definition

The Collaborative Routing Orchestration for Organic Food Distribution Problem (CRO-FDP) models a two-echelon, multi-commodity distribution system where independent organic producers and regional food banks horizontally share logistical infrastructure under the centralized coordination of an orchestrator (Ocampo-Giraldo et al., 2026). The underlying network is represented as a directed graph $\mathcal{G} = (\overline{N}, \overline{A})$, where $\overline{N} = N \cup \{\Delta\}$ consists of physical locations N and a single central dummy depot Δ from which all routes start and end, and $\overline{A}$ is the set of direct arcs of the network. The physical node set N comprises producer and member pickup sites $\mathcal{O}$, client delivery locations $\mathcal{D}$ (retailers, institutional kitchens, and food banks), and a candidate set of intermediate transfer hubs H volunteered by network members. Because the orchestrator contracts a shared third-party fleet billed strictly from the initial pickup to the final drop-off, the depot Δ functions purely as a modeling anchor with zero travel

Table 1: Comparison of Benders decomposition frameworks

Reference & Problem Type	Master Problem	Subproblem(s)	Cut Types Generated	Acceleration & Refinements
Fragkogios et al. (2024)	Customer visit sequence & discrete velocity intervals (RMP)	Primal continuous LP of exact departure times per vehicle (PSP)	Classical dual feasibility & optimality cuts	Multi-cut generation, Suboptimal Master Solutions (SMS), temporal VIs
Fachini and Armentano (2020)	Generalized Assignment Problem (GAP) assigning customers to vehicles	Independent TSPTW per vehicle	Logic-based Benders optimality & feasibility cuts (LBBD)	Branch-and-Check (BAC via incumbent callbacks), clique VIs.
Zhang et al. (2025)	Station construction budget and opening decisions	Capacitated min-cost flow on space-time-electricity network	Classical LP dual optimality cuts (feasibility ensured by dummy arcs)	Single-tree Branch-and-Benders-Cut (B&BC).
Contreras et al. (2011)	Binary facility opening of hub locations	Continuous multicommodity flow / semi-assignment	Pareto-optimal (non-dominated) Benders optimality cuts	Multicut aggregation by origin node, reduced-cost variable elimination tests.
Zhang et al. (2021)	Plant production periods, stock levels, customer visit periods	CVRP per period solved via Branch-and-Price	Logic-based Benders cuts + lifted LP cuts from ng-path relaxations	Lower bounding route relaxations (LBR), inventory valid inequalities.
Mahéo et al. (2023)	Vehicle routing & collection day scheduling	Integer bin allocation problem per accumulation point (GAP)	Integer L-shaped cuts	Unified Branch-and-Benders-Cut (UB&BC), fleet symmetry breaking VIs.
Alkaabneh et al. (2020)	Routing decisions and customer visit schedules across time	Dual LP evaluating inventory shortages, holding, and emissions	Classical Benders optimality cuts	Perishable product life-cycle VIs, vehicle capacity VIs.
Çalık et al. (2021)	Station locations, vehicle arc traversal, and vehicle loads	Battery state of charge and recharge quantities per vehicle	Classical Benders feasibility cuts	Intermediate reduction process limiting station replica copies, symmetry breaking.
CRO-FDP Framework	Vehicle Task Sequencing Problem (VTSP) on task-extended network	Negative cycle on a graph	Logic-based feasibility cuts	Spatial clustering and bin-packing initialization.

time and cost to and from all non-depot nodes ($\tau_{\Delta n} = \tau_{n\Delta} = 0$, $\forall n \in N$). Between any two physical locations $m, n \in N$, the travel time $\tau_{mn} > 0$ is an integer multiple of the discretization parameter δ across the planning horizon $T = \{0, \delta, 2\delta, \ldots, \eta\delta\}$ and natively includes service and handling times upon arrival.

The distribution network operates under a flexible multi-tier collaborative structure. Commodity fulfillment is not restricted to rigid echelon boundaries: depending on temporal compatibility and overall routing costs, a commodity may be delivered directly from its origin to its destination by a single vehicle without visiting any hub, or it may be consolidated through one or more intermediate transfer hubs $h \in H$ if multiple transshipments yield a less costly routing plan. Likewise, vehicles are shared and flexible; they are not dedicated to a single echelon, and may perform both collection and distribution tasks within their tours. The CRO-FDP is fundamentally distinguished from standard Pickup and Delivery Problems with Transfers (PDPT) and traditional Vehicle Routing Problems with Cross-Docking (VRPCD) by the location decision. Unlike these problems, on the CRO-FDP places where croos-docking operations can occour are not defined beforehand. Moreover, transfer hubs in the CRO-FDP operate as cross-docking facilities equipped with an intermediate physical staging area where batches can wait unattended between being dropped off and collected. Hub synchronization is therefore

governed strictly by a one-way temporal precedence constraint—an outbound vehicle cannot depart earlier than the inbound vehicle arrives ($d_w^{l'} \geq e_w^l$)—while long-term or overnight storage is prohibited, requiring all shipments to clear the hub during the operational shift. Activating a candidate hub $h \in H$ incurs a fixed operational setup cost C_h representing the labor mobilized by the host member to handle physical transfers. Furthermore, vehicles are permitted to perform multi-visit routes, visiting the same physical location non-consecutively throughout the day to accommodate shipments released at different times.

Transportation demand is defined by a set of non-splittable commodities K, each characterized by an origin $o_k^P \in \mathcal{O}$, a destination $d_k^P \in \mathcal{D}$, an availability or release time $o_k^T \in T$ at the origin, and a strict delivery due time $d_k^T \in T$ driven by product freshness. In contrast to standard capacitated routing formulations—and departing from the preliminary compact model of the CRO-FNDP—vehicle carrying capacity is not a binding operational restriction in this setting. Because organic food shipments in this regional network are typically small, fragmented batches, and because perishable shelf-life imposes tight pickup and delivery time windows, temporal deadlines and vehicle working time limits represent the dominant constraints governing operational feasibility. The network utilizes a homogeneous fleet L where each vehicle is constrained solely by a maximum tour duration W (elapsed time between leaving Δ and returning), a fixed vehicle mobilization cost C_l, and a distance-proportional arc traversal cost c_a. Vehicles may dwell at any node, during which commodities remain aboard; unattended waiting without a vehicle is permitted exclusively within activated hubs. The objective of the CRO-FDP is to minimize the sum of hub activation costs, fixed vehicle dispatch costs, and variable routing costs while satisfying commodity flow continuity, route duration limits, and inter-vehicle temporal synchronization.

4 Decomposition framework

To address the severe computational burden arising from the simultaneous optimization of routing routes and time synchronization in the CRO-FDP, we design an exact decomposition framework rooted in the theory of Logic-Based Benders Decomposition (LBBD) (Hooker and Ottosson, 2003). In traditional Benders decomposition, the subproblem is restricted to a continuous linear program whose classical LP dual generates linear benders cuts. However, in coordinated distribution networks with shared transshipment hubs, temporal feasibility is governed by combinatorial sequencing and precedence relations that cannot be captured effectively by conventional dual polyhedra without incurring severe degeneracy and large time-expanded formulations (Ocampo-Giraldo et al., 2026).

Under the proposed LBBD architecture, the CRO-FDP is partitioned into two structurally distinct decision layers:

1. **Master Problem—Vehicle Task Sequencing Problem (VTSP):** Solved over a task-extended network, the master problem handles all combinatorial spatial decisions. It determines hub activation status (y_h), the sequential ordering of physical visits made by each vehicle ($x_{\hat{a}}^l$), and the multi-commodity flow paths assigning requests to vehicle segments (f_a^{kl}). Because the objective function of the CRO-FDP depends strictly on hub activation, vehicle dispatch, and distance-based arc traversals, the master problem carries the entire objective function while temporarily relaxing the time constraints. Consequently, a lower bound of the VTSP provides a valid dual lower bound for the CRO-FDP.
2. **Subproblem—Temporal Feasibility and Scheduling:** For any fixed integer master candidate $(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{f}})$, all spatial paths and visit orders are frozen. The subproblem evaluates whether there exists a mutually consistent temporal schedule on the time grid $T = \{0, \delta, \ldots, \eta\delta\}$ respecting commodity availability dates, delivery deadlines, route duration limits, and one-way cross-docking precedence at hubs. Rather than solving an expensive mixed-integer or time-expanded mathematical program, this evaluation is mapped to the detection of negative cycles in a directed distance graph (a time constraint graph).

When the subproblem is temporally feasible, an earliest schedule on the grid T is retrieved directly from shortest-path distances; because the operational cost is purely spatial, this schedule inherits the candidate's master objective value, supplying a valid global upper bound UB for the CRO-FDP.

The theoretical exactness of this LBBD framework rests on the following equivalence:

Proposition 1. (Equivalence of Zero-Repair Temporal Feasibility). *Let $(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{f}})$ be an integer-feasible solution to the master VTSP problem defining vehicle stop sequences Ω^ℓ and unambiguous occurrence-specific commodity transport and transfer assignments $(w_{\text{on}}(k,\ell), w_{\text{off}}(k,\ell))$. There exists an operational schedule assigning arrival times $e_w^\ell \in T$ and departure times $d_w^\ell \in T$ on the discrete grid $T = \{0, \delta, \ldots, \eta\delta\}$ satisfying all commodity release dates, delivery deadlines, route durations, and hub transfer precedences if and only if the associated directed distance graph $G = (V, E)$ contains no directed cycle of strictly negative total weight.*

Proof. ($\Longrightarrow$) Assume there exists an operational schedule on T assigning discrete event times $t_v \in T$ to all event vertices $v \in V$, anchored at the temporal reference $t_0 = 0$. By construction of the simple temporal network, every directed arc $(u, v) \in E$ with weight c_{uv} encodes a potential difference constraint of the canonical form:

$$t_v - t_u \leq c_{uv}. \tag{1}$$

Suppose, for the sake of contradiction, that G contains a directed cycle $\mathcal{C} = (v_1, v_2, \ldots, v_m, v_1)$ with strictly negative total weight:

$$\sum_{i=1}^{m-1} c_{v_i, v_{i+1}} + c_{v_m, v_1} = -\epsilon < 0. \tag{2}$$

Summing the corresponding potential inequalities (1) along cycle $\mathcal{C}$ yields:

$$(t_{v_2} - t_{v_1}) + (t_{v_3} - t_{v_2}) + \cdots + (t_{v_m} - t_{v_{m-1}}) + (t_{v_1} - t_{v_m}) \leq -\epsilon. \tag{3}$$

The telescoping sum on the left-hand side identically collapses to zero, resulting in $0 \leq -\epsilon < 0$, which is an immediate algebraic contradiction. Thus, no directed negative cycle can exist in G.

($\Longleftarrow$) Conversely, suppose that the directed distance graph G contains no negative cycles. Then shortest-path distances from the temporal reference node t_0 are well-defined, real-valued, and finite for all reachable event vertices $v \in V$. By the classical triangle inequality, for every directed arc $(u, v) \in E$ with weight c_{uv}, the shortest-path distances from t_0 satisfy:

$$D(t_0, v) \leq D(t_0, u) + c_{uv} \iff D(t_0, v) - D(t_0, u) \leq c_{uv}. \tag{4}$$

Setting each event time directly to its shortest-path potential anchored at $t_0 = 0$:

$$t_v = D(t_0, v), \quad \forall v \in V, \tag{5}$$

strictly satisfies every difference constraint $t_v - t_u \leq c_{uv}$, establishing that $\mathbf{t}$ is a feasible potential schedule.

Furthermore, by problem definition, all network parameters (travel durations τ_{mn}, commodity release times o_k^T, delivery deadlines d_k^T, maximum vehicle route duration W, and horizon span $\eta\delta$) are non-negative integer multiples of the time discretization parameter δ. Since every shortest-path distance $D(t_0, v)$ is formed by the finite summation of these arc weights, each resulting event time t_v is strictly an integer multiple of δ residing on the discrete time grid $T = \{0, \delta, 2\delta, \ldots, \eta\delta\}$.

Finally, because $D(t_0, v)$ represents the maximum lower bound imposed on event v by the cumulative release times and travel durations along the tightest precedence chain originating from t_0, the assignment $t_v = D(t_0, v)$ constitutes the unique earliest feasible operational schedule. Hence, the candidate plan is globally feasible for the original CRO-FDP without requiring any spatial repairs, route delays beyond minimum dwell, or vehicle insertions. □

In our implementation, this LBBD architecture is operationalized within a single-tree *Branch-and-Check* (B&C) framework (Thorsteinsson, 2001): the master problem is not solved iteratively from scratch; instead, candidate integer solutions encountered at search tree nodes trigger callback evaluations of the distance graph, dynamically appending feasibility cuts as lazy constraints. In the following subsections, we formalize the master VTSP formulation (Section 4.1), detail the structure and polynomial-time resolution of the temporal constraint graph (Section 4.2), and describe the logic-based cut generation mechanism (Section 4.3).

4.1 Master problem: Vehicle task assignment problem

The master VTSP is formulated by drawing a conceptual analogy to a machine scheduling problem. In this framework, each vehicle in the fleet is viewed as an individual machine, while each visit to a physical location (origin, hub, or destination) is defined as a discrete task to be processed. By adopting this scheduling perspective, the master VTSP utilizes a task-extended network where node visits are represented as ordered intervals. Thus, for a maximum number of tasks ω, each node is defined by a tuple (n, w), where n is the physical node and w signifies the specific position in the sequence of visits. Therefore, instead of be over a graph $\mathcal{G} = (N, A)$, the master VTSP is formulated over an expanded graph $\mathcal{G}^\omega = (N^\omega, A^\omega)$, where N^ω represents the set of all possible (node, task position) pairs and A^ω comprises arcs connecting these task-extended nodes. Specifically, each arc in A^ω is of the form $((n, w), (m, w + 1))$ where $n \neq m$, representing a transition from task w at physical node n to task $w + 1$ at physical node m. As the depot is the origin and destination of all routes, it is extended for tasks 0, 2 to $\omega + 1$. Figure 1 illustrates the structure of this task-extended network and how physical nodes are duplicated across task layers, for presentation purpose only some arcs are depicted.

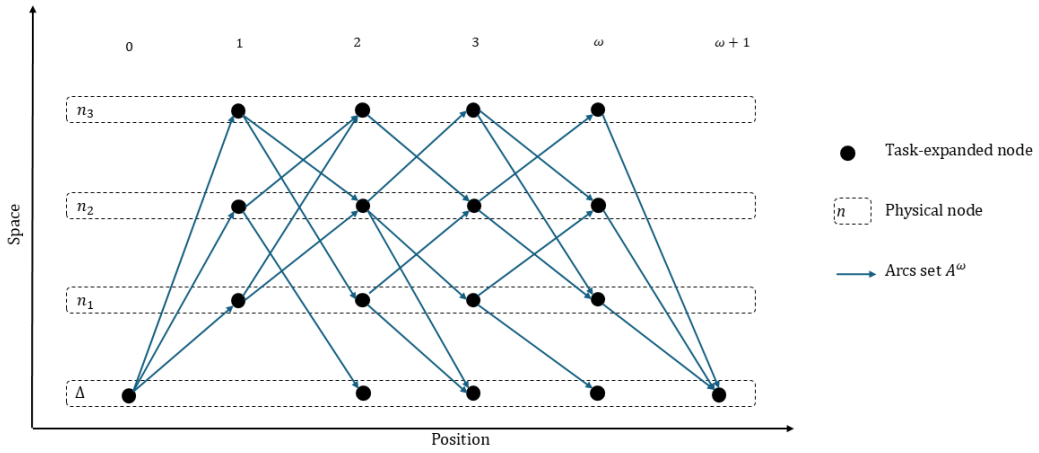

Figure 1: Task-expanded network representation

Over the task-expanded graph, we define the following decision variables. We denote x_a^ℓ as a binary variable that takes a value of 1 if a vehicle ℓ traverses arc $a \in A^\omega$, and 0 otherwise. Correspondingly, $f_a^{k\ell}$ is a binary variable indicating whether commodity k traverses arc $a \in A$ in vehicle ℓ, while y_h is a binary variable denoting the activation of hub $h \in H$.

To simplify the model presentation, we denote by $A_{n\to*}$ and ${A^\omega}_{n\to*}$ the set of outgoing arcs from node $n \in N$ and $n \in N^\omega$ respectively. Similarly, we define $A_{*\to n}$ and ${A^\omega}_{*\to n}$ as the sets of incoming arcs to node $n \in N$ and $n \in N^\omega$, respectively. To help the reader to follow the subsequent formulations, Annex A provides a comprehensive overview of the notation used throughout the paper.

The subsequent mathematical formulation for the master VTSP aims to minimize the total cost, encompassing routing costs, fixed vehicle costs, and hub activation costs. For this we denote by C_h

as the cost associated with activating hub h, C_ℓ as the fixed cost for utilizing vehicle ℓ, and c_a^ℓ as the cost for vehicle ℓ traversing arc $a \in A^\omega$. Thus, The master VTSP is formally defined as:

$$\min \sum_{h\in H} C_h y_h + \sum_{\ell\in L} \sum_{a\in A^\omega_{\Delta\to *}} C_\ell x_a^\ell + \sum_{\ell\in L} \sum_{a\in A^\omega} C_a^\ell x_a^\ell \tag{6}$$

$$\sum_{a\in {A^\omega}_{*\to n}} x_a^\ell = \sum_{a\in {A^\omega}_{n\to *}} x_a^\ell \qquad \forall n \in N^\omega, \forall \ell \in L \tag{7}$$

$$\sum_{a\in {A^\omega}_{\Delta^0\to *}} x_a^\ell \le 1 \qquad \forall \ell \in L \tag{8}$$

$$\sum_{\ell\in L} \sum_{a\in A_{n\to *}} f_a^{k\ell} - \sum_{\ell\in L} \sum_{a\in A_{*\to n}} f_a^{k\ell} = D_k(n) \qquad \forall n \in N, \forall k \in K \tag{9}$$

$$\sum_{a\in A_{n\to *}} f_a^{k\ell} = \sum_{a\in A_{*\to n}} f_a^{k\ell} \qquad \forall n \in N \setminus (H \cup \{o_k, d_k\}), \forall k \in K, \forall \ell \in L \tag{10}$$

$$f_a^{k\ell} \le \sum_{\hat{a}\in A^\omega | \pi(\hat{a})=a} x_{\hat{a}}^\ell \qquad \forall a \in A, \forall k \in K, \forall \ell \in L \tag{11}$$

$$\sum_{\ell\in L} \sum_{a\in A_{*\to h}} f_a^{kl} \le y_h \qquad \forall h \in H, \forall k \in K \tag{12}$$

$$\sum_{a\in A^\omega} \tau_a x_a^\ell \le W_\ell \qquad \forall \ell \in L \tag{13}$$

$$o_k^T + \sum_{a\in A} \sum_{\ell\in L} \tau_a f_a^{k\ell} \le d_k^T \qquad \forall k \in K \tag{14}$$

$$x_a^\ell \in \{0,1\} \qquad \forall a \in A^\omega, \forall \ell \in L \tag{15}$$

$$f_a^{k\ell} \in \{0,1\} \qquad \forall a \in A, \forall k \in K, \forall \ell \in L \tag{16}$$

$$y_h \in \{0,1\} \qquad \forall h \in H \tag{17}$$

Constraints (7) and (8) ensure vehicle flow conservation and that each vehicle is utilized at most once, respectively, while constraints (9) enforce commodity flow conservation throughout the network. Transfer between vehicles is controlled by constraints (10), which ensure that if a commodity enters a node in a vehicle, it must also exit that node in the same vehicle, unless it is its a hub node or the commodity origin or destination. Constraint (11) links commodity flow variables to vehicle routes, ensuring commodities only traverse arcs used by their assigned vehicles. Constraints (12) guarantee that if any commodity utilizes a hub for transfer, that hub must be activated, incurring its associated fixed cost. Constraints (13 ensures that the working time of each vehicle is respected. Constraints (14) Finally, constraints (15) to (17) define the binary domains for all decision variables, ensuring integrality for routing and assignment choices.

A solution to the master VSTP provides vehicle routes (for each vehicle ℓ its sequence of stops is know, commodity flow paths from its origin to its destination, and identifies the hubs utilized for transshipment. As the objective function of the CRO-FDP only evaluates spatial routing, vehicle usage, and hub activation costs, the same as those of the master VSTP, a lower bound for the master VSTP is also a lower bound for the CRO-FDP. To obtain a solution to the CRO-FDP, we must obtain feasible time schedules for the vehicles' visits that respect the commodities' generation and due times, and the precedence between inbound and outbound vehicles when a transshipment occurs at a hub.

4.2 Subproblem: A temporal constraint graph

Once the master VTSP determines an integer solution $(\bar{\mathbf{y}}, \bar{\mathbf{x}}, \bar{\mathbf{f}})$, all spatial decisions are frozen: the hub activation vector $\bar{\mathbf{y}}$ is set, each vehicle route is fixed as an ordered sequence of visits on the

task-extended network, and multi-commodity paths are defined by the active flow variables $\bar{\mathbf{f}}$. Because vehicles are permitted to perform multi-visit tours (i.e., stopping at the same physical location $n \in N$ at distinct sequence positions $w \neq w'$ to accommodate shipments released at different times), an aggregate physical flow variable $f^{kl}_{(u,v)} = 1$ does not inherently identify which specific passage through u and v carries the commodity. If the temporal evaluation arbitrarily attached requests to visit occurrences, it could create artificial timing conflicts (e.g., attempting to serve two non-overlapping requests during the first passage), thereby triggering spurious feasibility cuts that prune valid solutions.

To eliminate this ambiguity, physical commodity flows are deterministically mapped to unique task-sequence occurrences prior to constructing the temporal network. Let $\Omega^\ell = \big((\Delta, 0), (n^l_1, 1), \ldots, (\Delta, \omega+1)\big)$ represent the sequence of task-extended visits performed by vehicle $\ell \in L$. For each commodity k transported by vehicle ℓ between physical endpoints $u \in \{o^P_k\} \cup H$ and $v \in \{d^P_k\} \cup H$, let $\mathcal{W}_\ell(u) = \{w \mid n^l_w = u\}$ and $\mathcal{W}_\ell(v) = \{w \mid n^l_w = v\}$ denote the sets of visit positions where vehicle l calls at u and v, respectively. The boarding position $w_{\text{on}}(k,\ell) \in \mathcal{W}_\ell(u)$ and alighting position $w_{\text{off}}(k,\ell) \in \mathcal{W}_\ell(v)$ are uniquely selected as the unique pair satisfying:

$$w_{\text{on}}(k,l) < w_{\text{off}}(k,l), \tag{18}$$

such that vehicle ℓ continuously carries commodity k across sequence steps $w_{\text{on}}(k,\ell), \ldots, w_{\text{off}}(k,\ell)$. If multiple valid pairs exist in a multi-visit tour, $(w_{\text{on}}, w_{\text{off}})$ is chosen as the canonical occurrence whose implied transit interval is temporally compatible with release time o^T_k and deadline d^T_k. Conversely, if no pair satisfies $w_{\text{on}} < w_{\text{off}}$ (for instance, if the master problem visits destination v before origin u along vehicle l's route), the flow assignment is structurally inverted; this immediately manifests as a topological conflict in the temporal network, producing an unavoidable negative cycle that flags the sequencing failure.

Once occurrence-specific commodity assignments are established, the subproblem evaluates whether there exist arrival times e^l_w and departure times d^l_w at each visited position (n^l_w, w) on the discrete time grid $T = \{0, \delta, \ldots, \eta\delta\}$ that satisfy all operational constraints. Along with a reference temporal origin $t_0 = 0$, every operational requirement compares the difference between exactly two event points, defining a simple temporal network:

- *Travel duration:* $e^\ell_{w+1} - d^\ell_w = \tau_{n^\ell_w, n^\ell_{w+1}}$, requiring the travel time between consecutive stops to be satisfied exactly.
- *Dwell time:* $d^\ell_w - e^\ell_w \geq 0$, ensuring a vehicle does not depart before arriving.
- *Release date:* $d^\ell_{w_{\text{on}}} - t_0 \geq o^T_k$, whenever vehicle ℓ picks up commodity k at origin o^P_k at position $w_{\text{on}}(k,\ell)$.
- *Delivery deadline:* $e^\ell_{w_{\text{off}}} - t_0 \leq d^T_k$, whenever vehicle ℓ delivers commodity k to destination d^P_k at position $w_{\text{off}}(k,\ell)$.
- *Hub transfer precedence:* $d^{\ell'}_{w'_{\text{on}}} - e^\ell_{w_{\text{off}}} \geq 0$, when commodity k is dropped by vehicle ℓ at hub h (position w_{off}) and subsequently collected by vehicle ℓ' (position w'_{on}). This constraint governs vehicle synchronization through unattended staging.
- *Tour duration limit:* $e^\ell_{\omega_{end}} - d^\ell_0 \leq W$, restricting the total elapsed time between leaving and returning to the depot.
- *Horizon bounds:* $d^\ell_0 - t_0 \geq 0$ and $e^l_{\omega_{end}} - t_0 \leq \eta\delta$, anchoring route execution within the planning horizon.

Every temporal requirement can be cast in the canonical potential form $v - u \leq c_{uv}$. Consequently, the temporal subproblem is mapped onto a directed distance graph $G = (V, E)$, where vertices V represent event variables (including the temporal origin t_0) and directed arcs $(u,v) \in E$ with weight c_{uv} represent difference bounds $v - u \leq c_{uv}$. Lower bounds $u - v \geq c$ are rewritten as $v - u \leq -c$ (inducing an arc (u,v) of weight $-c$), while exact travel equalities induce opposing arc pairs with weights τ and $-\tau$. Table 2 summarizes the resulting arc definitions and weights.

Table 2: Arcs and weights of the Temporal Constraint Graph (TCG).

Constraint Type	**Directed Arc** $(u \to v)$	**Weight** (c_{uv})
Travel duration (upper bound)	$d^\ell_w \to e^\ell_{w+1}$	$\tau_{n^\ell_w, n^\ell_{w+1}}$
Travel duration (lower bound)	$e^\ell_{w+1} \to d^\ell_w$	$-\tau_{n^\ell_w, n^\ell_{w+1}}$
Dwell time	$d^\ell_w \to e^\ell_w$	0
Release time of commodity k	$d^\ell_{w_{\text{on}}} \to t_0$	$-o^T_k$
Delivery deadline of commodity k	$t_0 \to e^\ell_{w_{\text{off}}}$	d^T_k
Hub transfer synchronization	$d^{\ell'}_{w'_{\text{on}}} \to e^\ell_{w_{\text{off}}}$	0
Working tour duration limit	$d^\ell_0 \to e^\ell_{\omega+1}$	W
Horizon start anchor	$d^\ell_0 \to t_0$	0
Horizon end anchor	$t_0 \to e^l_{\omega+1}$	$\eta\delta$

By classical potential theory, a candidate plan admits a feasible schedule if and only if G contains no directed cycle of negative total weight. Feasibility detection and schedule extraction are conducted in polynomial time via the Bellman–Ford algorithm:

1. *Negative-Cycle Detection:* To ensure all disconnected components are reachable without missing isolated temporal conflicts, the graph is augmented with a dummy source node s having zero-weight directed arcs (s, v) to all $v \in V$. Executing $|V| - 1$ relaxation passes followed by a cycle-check pass detects whether any arc can still be relaxed. If an arc (u, v) admits further relaxation, tracing predecessor pointers backwards isolates a minimal negative cycle $\mathcal{C}$ proving temporal infeasibility.
2. *Schedule Extraction:* If no negative cycle exists, the candidate spatial plan is temporally feasible. For any arc $u \to v$ with weight c_{uv}, valid event potentials require $t_v - t_u \leq c_{uv}$[cite: 14]. Let $D(t_0, v)$ denote the shortest-path distance from the reference origin t_0 to event vertex v. By the triangle inequality, $D(t_0, v) \leq D(t_0, u) + c_{uv}$, which rearranges to $D(t_0, v) - D(t_0, u) \leq c_{uv}$. Setting each event time to:
$$t_v = D(t_0, v), \quad \forall v \in V, \tag{19}$$
yields the unique earliest operational schedule anchored at $t_0 = 0$. Because all instance parameters (τ, o^T_k, d^T_k, W, and $\eta\delta$) are integer multiples of δ, every shortest path $D(t_0, v)$ is an exact sum of integer weights, ensuring that all event times fall strictly onto the discrete time grid T without requiring post-processing or spatial repair.

Thus, the subproblem resolves global temporal consistency and obtains earliest schedules in $O(|V| \cdot |E|)$ time—requiring only milliseconds for realistic fleet and stop configurations—completely bypassing the need to enumerate or branch over discretized time intervals.

Determining feasibility and extracting an operational schedule reduces to shortest-path computation, implemented via the Bellman-Ford algorithm. To ensure all components are reachable and avoid missing isolated negative cycles, the graph is augmented with a dummy source node with zero-weight outgoing arcs to every vertex. The algorithm executes $|V| - 1$ relaxation passes, followed by a final pass to detect cycle existence; if an arc can still be relaxed, tracing predecessor pointers backwards pinpoints the negative cycle responsible for the conflict. Conversely, if no negative cycle exists, the earliest feasible schedule is obtained directly by setting each variable v to $-D(v, t_0)$, the negated shortest-path distance to the reference node t_0. Because every arc weight (travel durations, time windows, and working limits) is an integer multiple of the discretization parameter δ, all shortest-path distances and resulting schedule events necessarily fall on the discrete time grid $T = \{0, \delta, 2\delta, \ldots, \eta\delta\}$. Thus, the subproblem resolves temporal consistency in $O(|V| \cdot |E|)$ time—requiring only milliseconds for realistic fleet and stop dimensions—without ever having to enumerate or branch over the time horizon.

4.3 Generating cuts from negative cycles

Whenever the temporal subproblem detects a negative cycle, the system of difference constraints is unsolvable, and the offending candidate plan must be excluded from the master VTSP problem. To construct a valid feasibility cut, one must extract from the negative cycle the exact subset of spatial decisions responsible for creating it. This requires distinguishing between instance data and master VTSP problem variables: while the numerical weights on the arcs (such as release times, deadlines, and travel durations) are fixed problem parameters, the *existence* of an arc in the distance graph is conditioned entirely on the spatial configuration chosen by the master VTSP problem. For each arc belonging to the detected negative cycle, we identify the specific master VTSP variables whose activation introduces that arc. As cataloged in Table 3, a travel or dwell arc exists because vehicle ℓ performs that move at position w (variable $x^{\ell}_{\hat{a}}$); working time and horizon arcs depend on vehicle utilization (departure from the depot); a transfer arc requires both the vehicle moves bringing commodities into and out of the hub as well as the flow variables linking them; and crucially, release and due-time arcs depend not only on the vehicle's arrival at the location, but on the commodity flow decisions placing that specific request aboard vehicle ℓ at that position.

Table 3: Decisions of the master problem on which each arc of a negative cycle depends.

Arc in the cycle	Present because	Decisions to name
Travel (either direction) between w and $w+1$	ℓ makes that move	$x^{\ell}_{\hat{a}}$ for $\hat{a} = ((n^{\ell}_{w}, w), (n^{\ell}_{w+1}, w+1))$
Dwell at position w	l is at position w	The $x^{\ell}_{\hat{a}}$ of the move bringing ℓ to position w
Release of k at position w	ℓ visits o^{P}_{k} at w and carries k	$x^{\ell}_{\hat{a}}$ bringing l there, and $f^{k\ell}_{a}$ for arc leaving o^{P}_{k}
Due time of k at position w	ℓ visits d^{P}_{k} at w and carries k	$x^{\ell}_{\hat{a}}$ bringing ℓ there, and $f^{k\ell}_{a}$ for arc entering d^{P}_{k}
Transfer of k at hub h	ℓ at w, ℓ' at w', k handed over	Moves bringing ℓ, ℓ' there; $f^{k\ell}_{a}$ into h and $f^{k\ell'}_{a}$ out of h
Working time, horizon start/end	ℓ is used	The $x^{\ell}_{\hat{a}}$ of the move of ℓ out of the depot

Treating release and due-time arcs as unconditional instance parameters—and thus failing to attribute them to commodity flow decisions—would severely undermine the validity of the cut. For example, consider a vehicle ℓ picking up commodity k at position 1 and delivering k' at position 2, separated by travel time τ. These give rise to a three-arc cycle: a release arc $d^{\ell}_{1} \to t_0$ of weight $-o^{T}_{k}$, a due-time arc $t_0 \to e^{\ell}_{2}$ of weight $d^{T}_{k'}$, and a travel arc $e^{\ell}_{2} \to d^{\ell}_{1}$ of weight $-\tau$, yielding a total weight of $d^{T}_{k'} - o^{T}_{k} - \tau$. If $o^{T}_{k} + \tau > d^{T}_{k'}$, this cycle is strictly negative because commodity k is made available too late to allow on-time delivery of k'. If the cut were to attribute this failure only to the physical move $x^{\ell}_{\hat{a}} \leq 0$, the master problem would be permanently forbidden from using that transit arc, discarding optimal solutions where vehicle ℓ performs the exact same move carrying different, temporally compatible commodities. By instead identifying the set $\mathcal{D}$ of all master decisions supporting the cycle (the move arc alongside the flow assignments placing k and k' aboard ℓ), the conflict is isolated to the specific combination that fails.

Given the set $\mathcal{D}$ of binary master decisions active in the candidate solution ($\bar{v} = 1, \forall v \in \mathcal{D}$), the combinatorial feasibility cut is expressed as:

$$\sum_{v \in \mathcal{D}} v \leq |\mathcal{D}| - 1 \tag{20}$$

Proof of Cut Validity. To prove that inequality (20) is a valid cut for the CRO-FDP, we verify the two fundamental properties of logic-based Benders cuts:

1. *Exclusion of Infeasible Incumbent:* In the current candidate solution, all decisions in $\mathcal{D}$ take the value of 1 by construction, yielding $\sum_{v \in \mathcal{D}} \bar{v} = |\mathcal{D}|$. Hence, inequality (20) is violated and the current time-infeasible candidate is strictly eliminated.

2. *Conservation of Globally Feasible Solutions:* Suppose, for the sake of contradiction, that there exists a globally feasible solution $(\mathbf{y}^*, \mathbf{x}^*, \mathbf{f}^*)$ that violates cut (20). This violation requires $v = 1$ for all $v \in \mathcal{D}$. Because every element in $\mathcal{D}$ explicitly binds the vehicle's task-extended arcs x_a^l and unambiguous commodity boarding/alighting occurrences $(w_{\text{on}}, w_{\text{off}})$, activating all variables in $\mathcal{D}$ forces the subproblem distance graph $G(\mathbf{x}^*, \mathbf{f}^*)$ to contain the exact same directed arcs with identical endpoints and identical parameter weights as cycle $\mathcal{C}$. Consequently, cycle $\mathcal{C}$ is fully induced in $G(\mathbf{x}^*, \mathbf{f}^*)$ with total weight $-\epsilon < 0$. By Proposition 1, this implies that $(\mathbf{y}^*, \mathbf{x}^*, \mathbf{f}^*)$ admits no feasible temporal schedule on grid T, contradicting the assumption that it is globally feasible.

Therefore, cut (20) never prunes any globally feasible solution. □

Finally, the effectiveness of these cuts is governed by structural trade-offs. Although the Bellman-Ford algorithm efficiently detects a negative cycle, it does not necessarily return the cycle with the minimum cardinality of arcs; an alternative cycle elsewhere in the graph might yield a cut involving fewer decisions, which would prune a larger portion of the master solution space. Moreover, while a single vehicle-level cut forbidding the conflicting sequence anywhere along the tour would be mathematically stronger than position-specific cuts, such an aggregation is not valid under the task-extended master formulation: knowing that two stops occur within a route does not imply they are executed consecutively, and enforcing at-most-once visits per node is incompatible with operations where vehicles visit a node multiple times for later-released shipments. Nonetheless, cuts rooted in negative cycles isolate the exact combinatorial conflict responsible for infeasibility, yielding substantial pruning compared to naive no-good plan rejections.

5 The algorithm

The overall solution framework solves the CRO-FDP through a single-tree Branch-and-Check scheme that coordinates the spatial master problem with the temporal subproblem. The procedure begins with an initialization phase that establishes safe operational bounds without eliminating globally optimal solutions.

Because each dispatched vehicle leaves the depot Δ and incurs a strictly positive fixed mobilization cost $C_l > 0$, and since arc traversal costs are non-negative, any optimal solution cannot mobilize more vehicles than the total available upper bound cost permits. Consequently, given a validated primal upper bound UB, an exact upper bound on the number of required vehicles is established as:

$$\hat{L} = \left\lfloor \frac{UB}{C_l} \right\rfloor. \tag{21}$$

Notice that capping the fleet by the number of commodities ($|K|$) is generally invalid: in long-distance transshipment settings where W_ℓ is smaller than the direct origin-to-destination transit, a single commodity may require two or more relay vehicles across intermediate transfer hubs to achieve timely delivery. Hence, bound (21) preserves unrestricted exactness.

Similarly, to define the layer depth of the task-extended network $\mathcal{G}^\omega$, we determine a safe structural bound on the maximum number of physical visits ω that a vehicle can perform within its working time limit W_ℓ. Let:

$$\tau_{\min} = \min_{\substack{m,n \in N \\ m \neq n}} \tau_{mn} > 0 \tag{22}$$

denote the strictly positive minimum travel time between any two distinct physical nodes in N (explicitly excluding zero-duration arcs incident to the dummy depot Δ). Because a vehicle performing

q physical stops executes at least $q-1$ physical transit legs, the total route duration satisfies $(q-1)\tau_{\min} \leq W$. Thus, an exact and unrestrictive upper bound on the number of sequence tasks is:

$$\omega = 1 + \left\lfloor \frac{W}{\tau_{\min}} \right\rfloor. \tag{23}$$

To supply a valid initial upper bound UB and calibrate the fleet limit $\hat{L} = \lfloor UB/C_l \rfloor$, we execute the spatial–temporal grouping procedure described in Appendix B. While solving the reduced instance over aggregated macro-nodes and Abstract Commodities provides a compact routing skeleton, a schedule that is feasible in the aggregate space does not inherently guarantee feasibility in the original network: expanding macro-nodes into individual customer visits introduces additional local transit legs and could potentially violate tight release dates o_k^T, delivery deadlines d_k^T, or hub transfer precedences.

Therefore, the aggregate solution undergoes an explicit disaggregation and feasibility validation step before its cost is admitted as a primal bound. Each Abstract Commodity route is unpacked into an expanded physical tour by solving local TSP sequences across the constituent pickup and delivery sites. The resulting expanded candidate $(\mathbf{x}_{\text{init}}, \mathbf{f}_{\text{init}}, \mathbf{y}_{\text{init}})$ is then submitted to the Temporal Constraint Graph (Section 4.2). If the Bellman–Ford algorithm certifies that no negative cycles exist, all original temporal constraints are respected, and the true operational cost evaluated over the physical network is accepted as the initial upper bound UB. This validated solution serves as a warm start for the branch-and-bound tree. In the rare event that disaggregation induces temporal conflicts, the candidate is discarded, and the master search initializes with $UB = \infty$ (or a trivial direct-delivery upper bound) without restricting the task horizon ω.

Once initialized, the master problem is solved once using branch and bound, maintaining a global search tree rather than restarting the master problem after each feedback loop. During exploration, whenever the tree encounters an integer-feasible spatial candidate (specifying vehicle routes, stop sequences, and commodity flow assignments), the solver triggers an incumbent callback (implemented in Gurobi via `cbLazy` with lazy constraints enabled). Inside the callback, the candidate's spatial assignments are passed to the subproblem, where the directed distance graph is constructed with an auxiliary source node connecting to all event vertices with zero-weight arcs. The Bellman-Ford algorithm is then executed to detect temporal consistency.

Two outcomes are possible from the subproblem evaluation:

1. **Feasible Schedule (No Negative Cycle):** If no negative cycle exists, the earliest feasible schedule is retrieved directly by negating the shortest-path distances to the reference node t_0. Because all arc weights are integer multiples of δ, every arrival and departure event is placed exactly on the discrete time grid T. Combined with the candidate's spatial plan, this yields a fully feasible CRO-FDP solution whose cost matches the candidate's master objective value (since cost is purely spatial). The solver accepts the candidate as a new incumbent, updates the global upper bound UB, and continues the branch-and-bound search.
2. **Infeasible Schedule (Negative Cycle Detected):** If a negative cycle is found, the candidate plan cannot be scheduled in time. The algorithm extracts the single negative cycle identified by Bellman-Ford, maps its constitutive arcs to their supporting master decision variables $\mathcal{D}$ (as cataloged in Table 3), and adds the combinatorial feasibility cut $\sum_{v \in \mathcal{D}} v \leq |\mathcal{D}| - 1$ as a lazy constraint to the master problem. This cut rejects the integer candidate and eliminates the offending spatial configuration from the search tree.

Throughout the single-tree Branch-and-Bound exploration, the master problem continuously supplies a valid global dual lower bound LB via continuous linear relaxations across open search nodes. This lower bound is guaranteed to be valid for the original CRO-FDP because the VTSP master problem represents a spatial relaxation of the global problem, and every lazy cut dynamically appended from the subproblem eliminates exclusively provably time-infeasible configurations without pruning any feasible schedules (as demonstrated in Section 4.3).

Whenever a candidate plan $(\bar{\mathbf{x}}, \bar{\mathbf{f}}, \bar{\mathbf{y}})$ is certified as temporally feasible by the Bellman–Ford subproblem, its evaluated objective value defines a valid primal upper bound UB. Global optimality is certified if the relative optimality gap $(UB - LB)/UB$ falls within a predefined tolerance ϵ_{tol} (or if the branch-and-bound tree is exhaustively explored). Under unconstrained computational time, the single-tree Branch-and-Check algorithm is guaranteed to converge to an exact globally optimal solution in a finite number of iterations: the domain of binary spatial decisions $(\mathbf{x}, \mathbf{f}, \mathbf{y})$ is finite, the temporal oracle is exact and polynomial-time, and each negative-cycle cut eliminates at least the conflicting assignment without recurring.

When the execution reaches the operational computational time limit (e.g., 21,600 seconds), the algorithm halts externally. In this case, rather than certifying full convergence, the method terminates with status `TIME_LIMIT` and returns the best-known integer incumbent S^*, the prevailing global upper bound UB, the best retrieved dual bound LB, and the proven remaining optimality gap. The overall workflow is formalized in Algorithm 1.

Algorithm 1: Branch-and-Check Algorithm for the CRO-FDP.

Input: Network instance $\mathcal{G} = (N, A)$, commodity requests K, travel durations τ, fleet cost C_l, traversal cost c_a, hub costs C_h, shift limit W, discretization step δ, relative tolerance ϵ_{tol}, time limit $T_{\max}$.
Output: Termination status, best feasible solution S^*, primal bound UB, dual bound LB, remaining gap.

`// Step 1: Initialization and Safe Bounding (Appendix B)`
$(UB, S_{\text{init}}, \omega, \hat{L}) \leftarrow$ ValidatedGroupingHeuristic(N, K, W);
$S^* \leftarrow S_{\text{init}}$;
`// Step 2: Single-Tree Master Problem Formulation`
Initialize Master Problem $\mathcal{MP}$ on task-extended network A^ω with vehicle limit $\hat{L}$;
Set solver parameter `LazyConstraints` $\leftarrow 1$, time limit $\leftarrow T_{\max}$, and tolerance $\leftarrow \epsilon_{\text{tol}}$;
`// Step 3: Incumbent Callback Routine`
Function `CallbackOnIntegerCandidate(`$\bar{\mathbf{x}}, \bar{\mathbf{f}}, \bar{\mathbf{y}}$`):`
  Construct directed distance graph $G = (V, E)$ from candidate $(\bar{\mathbf{x}}, \bar{\mathbf{f}}, \bar{\mathbf{y}})$;
  Add dummy source s with zero-weight directed arcs (s, v) to all $v \in V$;
  (hasCycle, $\mathcal{C}$) $\leftarrow$ BellmanFord(G, s);
  if *not hasCycle* **then**
    `// Candidate admits a feasible schedule on grid` T
    Extract earliest schedule: $t_v \leftarrow D(t_0, v), \quad \forall v \in V$;
    cost $\leftarrow$ ObjectiveValue$(\bar{\mathbf{x}}, \bar{\mathbf{f}}, \bar{\mathbf{y}})$;
    if *cost* $< UB$ **then**
      $UB \leftarrow$ cost;
      $S^* \leftarrow (\bar{\mathbf{x}}, \bar{\mathbf{f}}, \bar{\mathbf{y}}, \mathbf{t})$;
      Accept candidate as new global incumbent;
    end
  end
  else
    `// Temporally infeasible: extract cycle and add lazy cut`
    Identify master decisions $\mathcal{D}$ supporting cycle $\mathcal{C}$ (Table 3);
    Add lazy constraint: $\sum_{v \in \mathcal{D}} v \leq |\mathcal{D}| - 1$ to $\mathcal{MP}$;
    Reject candidate solution;
  end
`// Step 4: Solve Single-Tree Search and Retrieve Global State`
Solve $\mathcal{MP}$ with `CallbackOnIntegerCandidate` enabled;
$LB \leftarrow$ GetBestDualBound$(\mathcal{MP})$;
status $\leftarrow$ GetTerminationStatus$(\mathcal{MP})$;
Gap $\leftarrow (UB - LB)/UB$;
return *status*, $S^*, UB, LB,$ *Gap*;

6 Computational experiments and results

This experimental analysis evaluates and compares the performance of the newly proposed decomposition algorithm using the Vehicle Task Sequencing Problem (VTSP) as master problem, against two benchmarks: the Interval Relaxation (IR) method introduced by Ocampo-Giraldo et al. (2026), and the monolithic mixed-integer programming formulation over a time-extended network (MIP-TE). The primary objective is to benchmark the effectiveness of the proposed approach in terms of solution

quality, scalability, and computational efficiency across a diverse testbed of 100 benchmark instances categorized by size into Small (s, 27 instances), Medium (m, 47 instances), and Large (l, 26 instances). All experiments were conducted under a maximum CPU runtime limit of 21,600 seconds (6 hours) per execution. In our comparative evaluation, any instance for which a method returned a valid feasible integer solution within the allocated computational budget without closing the optimality gap is accounted for as a solved (suboptimal) instance, whereas an instance is classified as optimal only when global optimality was proven within the time limit.

Table 4 summarizes the aggregate performance metrics across the evaluated methods, reporting the number of instances proven optimal, the total solved instances (both optimal and suboptimal), the average remaining optimality gap, the mean CPU time in seconds, and the corresponding percentage success rates.

Table 4: Overall computational performance comparison among VTSP, IR, and MIP-TE across instance sizes.

Method	Size	Total	Optimal	Solved	Avg. Gap (%)	Avg. Time (s)	% Opt.	% Solved
VTSP	s	27	27	27	0.00	253.99	100.00	100.00
	m	47	32	47	3.47	8724.55	68.09	100.00
	l	26	2	26	14.15	20367.21	7.69	100.00
IR	s	27	20	27	1.59	3408.09	74.07	100.00
	m	47	5	47	11.70	14584.97	10.64	100.00
	l	26	2	26	25.30	21581.69	7.69	100.00
MIP-TE	s	27	17	27	2.95	9062.94	62.96	100.00
	m	47	5	39	14.92	19716.68	10.64	82.98
	l	26	0	8	24.62	21600.26	0.00	30.77

The numerical results highlight a clear performance shift governed by instance complexity and the underlying model structure. For small instances (s), the VTSP decomposition consistently outperforms all benchmarks, solving 100% of the instances to proven optimality in an average runtime of only 253.99 seconds, compared to 3,408.09 seconds for IR (74.07% optimal) and 9,062.94 seconds for MIP-TE (62.96% optimal). This represents more than a 13-fold speedup over IR and a 35-fold speedup over MIP-TE, eliminating the optimality gap entirely (0.00%).

This advantage becomes even more prominent on medium-scale instances (m), where the curse of dimensionality inherent to time-indexed representations begins to severely degrade solver efficiency. VTSP proves global optimality for 32 out of 47 instances (68.09%) with an average gap of 3.47% and an average solution time of 8,724.55 seconds. In contrast, both IR and MIP-TE manage to prove optimality in only 5 instances (10.64%), yielding substantially larger average gaps of 11.70% and 14.92%, while requiring 14,584.97 and 19,716.68 seconds, respectively. Moreover, MIP-TE fails to find any integer-feasible solution for 8 medium instances within the 6-hour limit.

On the largest and most challenging instances (l), where the combinatorial complexity of route synchronization and cross-docking hubs restricts the ability of any exact method to close the search tree within 21,600 seconds, VTSP demonstrates superior primal and dual bounding behavior. While all methods experience timeouts across the vast majority of runs, VTSP successfully obtains feasible integer solutions for all 26 instances and cuts the average optimality gap nearly in half compared to IR (14.15% versus 25.30%). Furthermore, as illustrated by the gap distributions, the median gap achieved by VTSP remains around 12.5%, with its upper quartile bounded near 21%, whereas the interquartile ranges for IR and MIP-TE stretch well past 30% to 45%. The monolithic MIP-TE model collapses computationally on this class, failing to locate a feasible solution in 18 out of 26 instances (a 69.23% failure rate) and averaging an optimality gap of 24.62% only over the few instances where feasibility was attained.

In summary, the experimental outcomes demonstrate the structural superiority of projecting out the continuous-time discretization from the master problem: formulating the routing problem as a

VTSP allows the branch-and-bound solver to explore spatial sequencing with high efficiency, while the temporal constraint graph rapidly validates schedules in polynomial time. As a result, the proposed approach delivers significantly faster convergence on small-to-medium instances and provides consistently tighter, more dependable bounds on large-scale problems.

7 Conclusion

The Collaborative Routing Orchestration for Organic Food Distribution Problem (CRO-FDP) addresses a critical challenge in sustainable regional logistics by coordinating independent producers, commercial clients, and food banks through shared cross-docking hubs. The core mathematical difficulty of this network stems from the tight inter-vehicle temporal dependencies created at shared hubs, which operate with temporary staging buffers but lack long-term cold storage. When routing, hub selection, and high-resolution continuous-time synchronization are solved simultaneously, standard mathematical programming formulations suffer from severe combinatorial explosions.

To overcome this methodological hurdle, this paper introduced an exact decomposition framework rooted in Logic-Based Benders Decomposition (LBBD), implemented as a single-tree Branch-and-Check algorithm. By completely projecting out the temporal dimension from the master problem, the spatial decisions—vehicle stop sequences, commodity routing paths, and hub activation—are modeled as a Vehicle Task Sequencing Problem (VTSP) over a task-extended network. The master problem carries the entire operational objective and supplies a rigorous dual lower bound. In turn, the temporal subproblem evaluates the feasibility of schedules in polynomial time through negative-cycle detection in a directed distance graph. Whenever temporal conflicts arise, the identified negative cycle provides a certificate of infeasibility that is translated into a logic-based Benders cut, eliminating only the conflicting combination of spatial decisions without pruning any feasible configurations. Furthermore, we distinguished between heuristic bounding and exact exploration by developing an initialization routine based on spatial clustering and bin packing (Appendix B); this aggregates transportation requests into Abstract Commodities to supply an initial primal upper bound and calibrate safe network parameters without compromising global exactness.

The computational experiments conducted on 100 realistic benchmark instances demonstrate a significant performance shift over both the monolithic time-extended model (MIP-TE) and the previously established Interval Relaxation (IR) matheuristic. For medium-sized instances, the proposed LBBD approach increases the percentage of proven optimal solutions from 10.6% to 68.1% while drastically decreasing the average optimality gap from 11.7% to 3.5%. On large-scale instances—where commercial solvers running MIP-TE fail to locate a single feasible solution in nearly 70% of runs—the LBBD approach achieves a 100% feasibility rate and slashes the average optimality gap from 25.3% down to 14.2%.

These methodological advances confirm that decoupling spatial sequencing from temporal graph constraints provides superior dual bounds, reliable primal convergence, and enhanced scalability.

A Notation summary

Table 5: Summary of sets, parameters, and decision variables used in the LBBD framework.

Category	Symbol	Description
Sets and Graphs	N, A	Physical nodes and directed arcs in network $\mathcal{G} = (N, A)$.
	$\mathcal{O}, \mathcal{D}, H, \{\Delta\}$	Origin nodes, destination nodes, candidate transfer hubs, and dummy depot.
	N^ω, A^ω	Task-extended nodes (n, w) and arcs $((n, w), (m, w + 1))$ in $\mathcal{G}^\omega$.
	K, L	Set of transport requests (commodities) and homogeneous fleet.
	T	Discrete time grid $\{0, \delta, 2\delta, \ldots, \eta\delta\}$.
	$G = (V, E)$	Temporal Constraint Graph (TCG) distance graph.
	$\mathcal{C}, \mathcal{D}$	Detected negative cycle and its supporting active master decisions.
Parameters	o_k^P, d_k^P	Origin and destination physical locations of commodity k.
	o_k^T, d_k^T	Release availability time and strict delivery deadline of commodity k.
	τ_{mn}	Travel time along physical arc (m, n) (including stop service).
	W_l, C_l	Maximum vehicle tour duration and fixed vehicle dispatch cost.
	$C_h, c^l_{\pi(a)}$	Fixed activation cost of hub h and distance traversal cost of vehicle l.
	δ, η	Time discretization step and total intervals in planning horizon.
	$\omega, \hat{L}$	Safe sequence task limit $(1 + \lfloor W/\tau_{\min} \rfloor)$ and fleet cap $(\lfloor UB/C_l \rfloor)$.
	$\tau_{\min}$	Minimum physical transit duration among distinct physical nodes $(m \neq n)$.
Mappings	$\pi(\hat{a})$	Spatial projection mapping task arc $\hat{a}$ to physical arc $(n, m) \in A$.
	$A_{n\to*}, A_{*\to n}$	Outgoing and incoming physical arcs at node $n \in N$.
	$A^\omega_{(n,w)\to*}, A^\omega_{*\to(n,w)}$	Outgoing and incoming task arcs at task node $(n, w) \in N^\omega$.
	$w_{\text{on}}(k, l), w_{\text{off}}(k, l)$	Task visit positions where vehicle l picks up and drops commodity k.
Master Variables	$y_h \in \{0, 1\}$	Equal to 1 if candidate transfer hub $h \in H$ is activated, 0 otherwise.
	$x^l_{\hat{a}} \in \{0, 1\}$	Equal to 1 if vehicle l traverses task arc $\hat{a} \in A^\omega$, 0 otherwise.
	$f^{kl}_a \in \{0, 1\}$	Equal to 1 if commodity k traverses physical arc a in vehicle l, 0 otherwise.
Subproblem Variables	$e^l_w, d^l_w \geq 0$	Arrival and departure time of vehicle l at/from sequence position w.
	t_0, t_v	Reference temporal origin $(t_0 = 0)$ and schedule potential for event $v \in V$.

B The initial solution

To establish a high-quality initial primal bound UB and calibrate search parameters without compromising global exactness, we solve an aggregated instance via a constructive spatial–temporal grouping procedure. Because physical vehicle carrying capacity is non-binding in this regional distribution setting, the procedure aggregates transportation requests strictly based on geographic proximity and temporal compatibility.

The heuristic operates in three sequential phases:

1. **Spatial Clustering (Macro-Nodes):** Physical nodes are partitioned by service role into origin sites $\mathcal{O}$ and destination sites $\mathcal{D}$. Within each set, nodes located within a spatial travel time threshold τ_{cluster} (e.g., 60 minutes) are clustered into macro-origins and macro-destinations using hierarchical clustering.
2. **Request Partitioning:** The full demand set K is partitioned into subsets $S \subset K$ of requests that share the exact same macro-origin and macro-destination pair.
3. **Temporal Bin-Packing of Requests into Abstract Commodities:** For each subset S, requests are sorted chronologically by their release dates o_k^T (with ties broken by due dates d_k^T). We then apply a greedy one-dimensional bin-packing logic where a "bin" represents an *Abstract Commodity* $B \subseteq S$ to be serviced within a single collective tour segment. A candidate request k is merged into an active bin B (forming $B_{\text{test}} = B \cup \{k\}$) if and only if two temporal conditions are satisfied:

Mutual Time-Window Overlap: The collective pickup availability window $[\max_{j \in B_{\text{test}}} o_j^T, \min_{j \in B_{\text{test}}} d_j^T]$ remains non-empty and temporally valid.

Tour Duration Feasibility: The duration of the shortest local tour visiting all individual physical sites in B_{test} plus handling and service times does not exceed the maximum vehicle shift duration W_ℓ.

If the merge violates either condition, bin B is closed as an Abstract Commodity, and a new bin is initialized with request k.

For each finalized Abstract Commodity $B \in M$, its collective release date is set to $\max_{k \in B} o_k^T$, its delivery deadline to $\min_{k \in B} d_k^T$, and its service duration to the optimal TSP sequence traversing all constituent sites. The resulting reduced-size instance is solved with a commercial solver.

Once the aggregate MIP formulation is solved with solver, the resulting solution must be certified on the original physical network $\mathcal{G} = (N, A)$. For each vehicle route and each macro-node visited, the constituent physical locations are sequenced in shortest-path order. The reconstructed physical visit sequence and commodity assignments are then mapped into the Temporal Constraint Graph $G = (V, E)$ defined in Section 4.2. Running the Bellman–Ford algorithm verifies whether individual arrival and departure times on the discrete grid T strictly satisfy every release date o_k^T, due date d_k^T, and transfer precedence. Only upon passing this temporal validation is the actual network traversal cost recorded as the certified primal upper bound UB for master problem initialization.

Algorithm 2: Temporal Bin-Packing for Abstract Commodities

Input: Original request set K, spatial clustering threshold τ_{cluster}, vehicle shift limit W.
Output: Set of merged Abstract Commodities M.

$M \leftarrow \emptyset$;
// Phase 1: Spatial Clustering
Cluster physical origins $\mathcal{O}$ and destinations $\mathcal{D}$ into macro-nodes based on travel time $\leq \tau_{\text{cluster}}$;
// Phase 2: Request Partitioning
Partition K into subsets $\mathcal{S}$, where each $S \in \mathcal{S}$ groups requests sharing the same macro-origin and macro-destination;
// Phase 3: Greedy Temporal Bin-Packing
foreach *subset* $S \in \mathcal{S}$ **do**
  Sort requests in S in ascending order of release dates o_k^T (ties broken by d_k^T);
  $B \leftarrow \emptyset$; // Initialize current temporal bin
  foreach *request* $k \in S$ **do**
    $B_{\text{test}} \leftarrow B \cup \{k\}$;
    Compute effective pickup window: $[o_{\max}, d_{\min}] \leftarrow [\max_{j \in B_{\text{test}}} o_j^T, \min_{j \in B_{\text{test}}} d_j^T]$;
    Compute minimum local tour duration $\text{Dur}(B_{\text{test}})$ covering all stops in B_{test};
    if $o_{\max} < d_{\min}$ ***and*** $Dur(B_{test}) \leq W$ **then**
      $B \leftarrow B_{\text{test}}$; // Merge accepted into current bin
    end
    else
      if $B \neq \emptyset$ **then**
        $M \leftarrow M \cup \{B\}$; // Close and store Abstract Commodity
      end
      $B \leftarrow \{k\}$; // Open new bin with request k
    end
  end
  if $B \neq \emptyset$ **then**
    $M \leftarrow M \cup \{B\}$; // Close last bin in subset
  end
end
return M;

References

Alkaabneh, F., Diabat, A., and Gao, H. O. (2020). Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. Computers & Operations Research, 113:104751.

Baez, Y. P., Sequeira, M., and Hilletofth, P. (2020). Local and organic food distribution systems: Towards a future agenda. Operations and Supply Chain Management, 13:336–348.

Bosona, T. G. and Gebresenbet, G. (2011). Cluster building and logistics network integration of local food supply chain. Biosystems engineering, 108(4):293–302.

Çalık, H., Oulamara, A., Prodhon, C., and Salhi, S. (2021). The electric location-routing problem with heterogeneous fleet: Formulation and benders decomposition approach. Computers & Operations Research, 131:105251.

Castro, J. A. O., Cabrera, J. P. O., and Jaimes, W. A. (2019). Last mile logistics in mega-cities for perishable fruits. Journal of Industrial Engineering and Management, 12:318–318.

Cleophas, C., Cottrill, C., Ehmke, J. F., and Tierney, K. (2019). Collaborative urban transportation: Recent advances in theory and practice. European Journal of Operational Research, 273(3):801–816.

Contardo, C., Hemmelmayr, V., and Crainic, T. G. (2012). Lower and upper bounds for the two-echelon capacitated location-routing problem. Computers & operations research, 39(12):3185–3199.

Contreras, I., Cordeau, J.-F., and Laporte, G. (2011). Benders decomposition for large-scale uncapacitated hub location. Operations research, 59(6):1477–1490.

Crainic, T. G., Feliu, J. G., Ricciardi, N., Semet, F., and Van Woensel, T. (2023). Operations research for planning and managing city logistics systems. In Handbook on city logistics and urban freight, pages 190–223. Edward Elgar Publishing.

Dereli, B. U., Hiermann, G., and Schiffer, M. (2025). Solving large-scale two-echelon location routing problems in city logistics. arXiv preprint arXiv:2502.08403.

Fachini, R. F. and Armentano, V. A. (2020). Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. Computers & Industrial Engineering, 148:106641.

Fragkogios, A., Qiu, Y., Saharidis, G. K., and Pardalos, P. M. (2024). An accelerated benders decomposition algorithm for the solution of the multi-trip time-dependent vehicle routing problem with time windows. European Journal of Operational Research, 317(2):500–514.

Gharehyakheh, A., Krejci, C. C., Cantu, J., and Rogers, K. J. (2020). A multi-objective model for sustainable perishable food distribution considering the impact of temperature on vehicle emissions and product shelf life. Sustainability, 12(16):6668.

Grangier, P., Gendreau, M., Lehuédé, F., and Rousseau, L.-M. (2016). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European journal of operational research, 254(1):80–91.

Hooker, J. N. and Ottosson, G. (2003). Logic-based Benders decomposition. Mathematical Programming, 96(1):33–60.

Lyu, X. (2020). Studies on Collaborative Transportation in Less than Truckload Transportation. PhD thesis, Université de Technologie de Troyes; Université Jiaotong (Xi'an, Chine).

Mahéo, A., Rossit, D. G., and Kilby, P. (2023). Solving the integrated bin allocation and collection routing problem for municipal solid waste: a benders decomposition approach. Annals of Operations Research, 322(1):441–465.

Mirhedayatian, S. M., Crainic, T. G., Guajardo, M., and Wallace, S. W. (2021). A two-echelon location-routing problem with synchronisation. Journal of the Operational Research Society, 72:145–160.

Mohamed, I. B., Klibi, W., Sadykov, R., Şen, H., and Vanderbeck, F. (2023). The two-echelon stochastic multi-period capacitated location-routing problem. European journal of operational research, 306(2):645–667.

Ocampo-Giraldo, D., Anaya-Arenas, A. M., Klibi, W., and Ortmann, J. (2026). Collaborative routing orchestration for organic food distribution. Transportation Research Part E: Logistics and Transportation Review, 214:105060.

Rahmani, Y. (2015). The multi-product location-routing problem with pickup and delivery. Université de Lorraine.

Sadykov, R. (2019). Modern branch-cut-and-price. PhD thesis, Université de Bordeaux.

Santini, A., Schneider, M., Vidal, T., and Vigo, D. (2023). Decomposition strategies for vehicle routing heuristics. INFORMS Journal on Computing, 35(3):543–559.

Soares, R., Marques, A., Amorim, P., and Parragh, S. N. (2024). Synchronisation in vehicle routing: classification schema, modelling framework and literature review. European Journal of Operational Research, 313(3):817–840.

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In Walsh, T., editor, Principles and Practice of Constraint Programming — CP 2001, volume 2239 of Lecture Notes in Computer Science, pages 16–30, Berlin, Heidelberg. Springer.

Winkenbach, M., Kleindorfer, P. R., and Spinler, S. (2016). Enabling urban logistics services at la poste through multi-echelon location-routing. Transportation Science, 50(2):520–540.

Zhang, C., Yong, R., Jiang, X., Zhang, Y., Song, M., and Cheng, L. (2025). Benders decomposition for charging station location-routing problem with time-dependent travel times: a space-time-electricity network perspective. Energy, 333:137310.

Zhang, Z., Luo, Z., Baldacci, R., and Lim, A. (2021). A benders decomposition approach for the multivehicle production routing problem with order-up-to-level policy. Transportation Science, 55(1):160–178.