

Scaling Industrial Logistics: Tackling Multi-Batching Problems via Sequential Solving

Emmanuelle Dietz^{*}, Guillaume Povéda[†], Karl Henning^{*} and Clara Buire[†]

Airbus Central R&T, Hamburg, Germany^{*}, Airbus Central R&T, Toulouse, France[†]

32nd International Conference on Principles and Practice of Constraint Programming 2026,
Lisbon, Portugal

A Global Logistics Problem

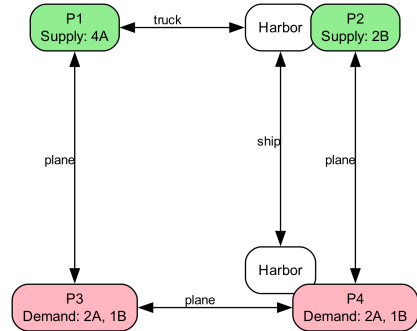


- 35 aircraft parts
- 50 production locations
- 16 transport resources
- 5,610 possible routes

- ① How to choose routes for parts?
- ② How to find packings on each route?
- ③ How to optimize overall transport costs?

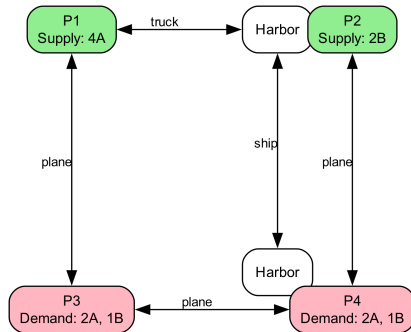
An Example

- Four production sites (P1-P4) and two harbors
- Three transport resources: plane (cap: 7), ship (cap: 50), truck (cap: 1).
- Two types of parts: Part A (size 1), Part B (size 5).
- Production sites have demands and supplies



An Example

- Four production sites (P1-P4) and two harbors
- Three transport resources: plane (cap: 7), ship (cap: 50), truck (cap: 1).
- Two types of parts: Part A (size 1), Part B (size 5).
- Production sites have demands and supplies



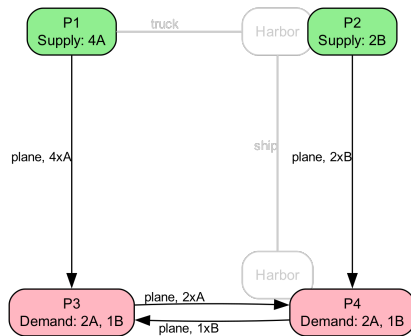
Goal

- 1 Find transport routes for parts to satisfy all demands
- 2 Find packings for each route
- 3 Optimize solution with respect to transport costs

Step 1: Find Transport Routes

A Possible Routing Solution

- Load 4A from P1 to P3 with the plane
- Load 2A from P3 to P4 with the plane
- Load 2B from P2 to P4 with the plane
- Load 1B from P4 to P3 with the plane



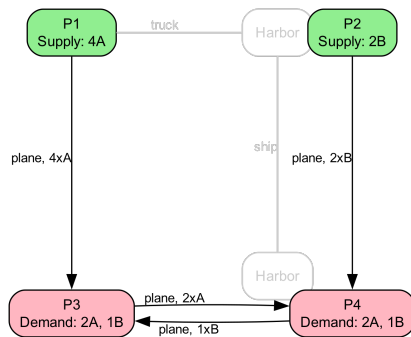
Step 1: Find Transport Routes

A Possible Routing Solution

- Load 4A from P1 to P3 with the plane
- Load 2A from P3 to P4 with the plane
- Load 2B from P2 to P4 with the plane
- Load 1B from P4 to P3 with the plane

Goal

- 1 Find transport routes for parts to satisfy all demands ✓
- 2 Find packings for each route
- 3 Optimize solution with respect to transport costs

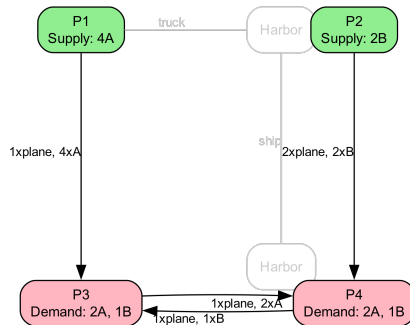


Step 2: Find Packings per Route

A Possible Packing Solution

Recall capacities: plane is 7, ship is 50, truck is 1.
sizes: A is 1, B is 5.

- 1 plane with 4A (P1 $\rightarrow$ P3)
- 1 plane with 2A (P3 $\rightarrow$ P4)
- 2 planes with 2B (P2 $\rightarrow$ P4)
- 1 plane with 1B (P4 $\rightarrow$ P3)

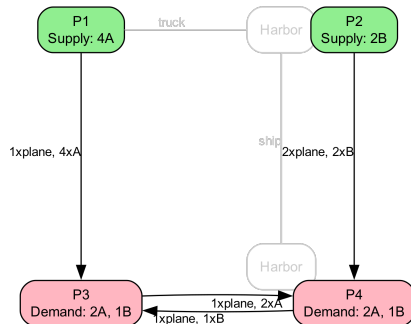


Step 2: Find Packings per Route

A Possible Packing Solution

Recall capacities: plane is 7, ship is 50, truck is 1.
sizes: A is 1, B is 5.

- 1 plane with 4A (P1 $\rightarrow$ P3)
- 1 plane with 2A (P3 $\rightarrow$ P4)
- 2 planes with 2B (P2 $\rightarrow$ P4)
- 1 plane with 1B (P4 $\rightarrow$ P3)



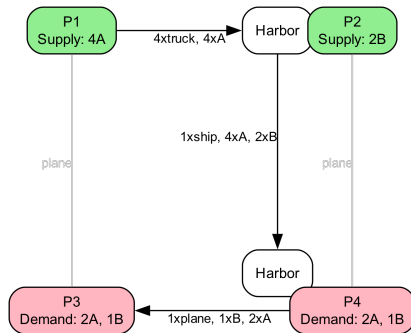
Observation

- Only single-batch packings on routes

Single-batching vs Multi-batching

Alternative Packing Solution

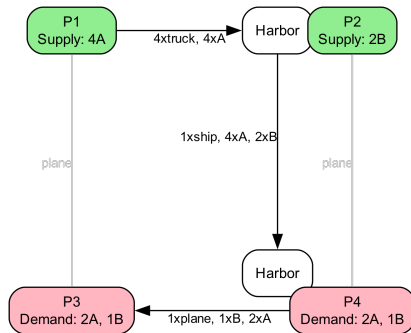
- 4 trucks with 4A ($P1 \rightarrow H1$)
- 1 ship with 4A and 2B ($H1 \rightarrow H3$)
- 1 plane with 1B and 2A ($P4 \rightarrow P3$)



Single-batching vs Multi-batching

Alternative Packing Solution

- 4 trucks with 4A (P1 → H1)
- 1 ship with 4A and 2B (H1 → H3)
- 1 plane with 1B and 2A (P4 → P3)



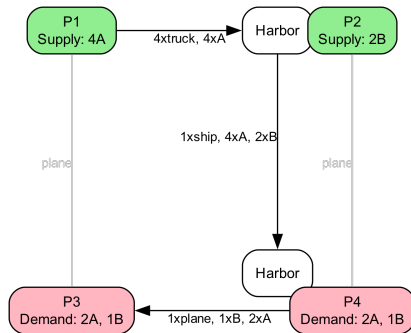
Goal

- 1 Find transport routes for parts to satisfy all demands (min-cost network flow problem) ✓
- 2 Find packings for each route (bin packing problem) ✓
- 3 Optimize solution with respect to transport costs

Single-batching vs Multi-batching

Alternative Packing Solution

- 4 trucks with 4A (P1 → H1)
- 1 ship with 4A and 2B (H1 → H3)
- 1 plane with 1B and 2A (P4 → P3)



Goal

- 1 Find transport routes for parts to satisfy all demands (min-cost network flow problem) ✓
- 2 Find packings for each route (bin packing problem) ✓
- 3 Optimize solution with respect to transport costs

★ **Industrial use case**

Similar Problems from Literature

Problem	Description	Limitation wrt Multi-batching	Preferred Approaches
Multicommodity Flow (Ahuja et al., 1993)	Optimal flows	Ignores discrete items in flow	LP
Unsplittable Flow (Kleinberg, 1996)	Handles whole items	Ignores packing, no path splitting	MILP
Bin Packing (BPP) (Martello & Toth, 1990)	Optimal packings	Static problem, no routing decisions	CP, MILP, ASP
Capacitated VRP (Pavlikov et al., 2023)	Capacity constraints	Predefined supply/ demand pairs, no path splitting	Heuristics
Multi-batching	Flow with discrete packing		★ Our contribution

Exact Flow Formulation

Idea Model individual trips, enforcing exact packing constraints and simultaneously routing

Exact Flow Formulation

Decision Variables

Given upper bound K_r , for route $r \in \mathcal{R}$, trip $k \in \{1, \dots, K_r\}$, and product $p \in \mathcal{P}$:

- $\alpha_{r,k,p} \in \mathbb{N}$: quantity of p loaded on k th trip on route r
- $u_{r,k} \in \{0, 1\}$: binary indicator if trip k is used on route r
- $f_{r,p} \in \mathbb{N}$: total flow of product p on route r

Exact Flow Formulation

Decision Variables

Given upper bound K_r , for route $r \in \mathcal{R}$, trip $k \in \{1, \dots, K_r\}$, and product $p \in \mathcal{P}$:

- $\alpha_{r,k,p} \in \mathbb{N}$: quantity of p loaded on k th trip on route r
- $u_{r,k} \in \{0, 1\}$: binary indicator if trip k is used on route r
- $f_{r,p} \in \mathbb{N}$: total flow of product p on route r

Constraints

For route $r \in \mathcal{R}$ and product $p \in \mathcal{P}$:

$$f_{r,p} = \sum_{k=1}^{K_r} \alpha_{r,k,p} \quad (\text{Flow aggregation})$$

For product $p \in \mathcal{P}$ and $\ell \in \mathcal{L}$:

$$\sum_{r \in \mathcal{R}_\ell^{\text{in}}} f_{r,p} - \sum_{r \in \mathcal{R}_\ell^{\text{out}}} f_{r,p} = d_{\ell,p} \quad (\text{Flow conservation})$$

For route $r \in \mathcal{R}$ and $k \in \{1, \dots, K_r\}$:

$$\sum_{p \in \mathcal{P}} (\alpha_{r,k,p} \times v_p) \leq (u_{r,k} \times Q_r) \quad (\text{Capacity})$$

where v_p is the size of p and Q_r capacity on r

Exact Flow Formulation

Decision Variables

Given upper bound K_r , for route $r \in \mathcal{R}$, trip $k \in \{1, \dots, K_r\}$, and product $p \in \mathcal{P}$:

- $\alpha_{r,k,p} \in \mathbb{N}$: quantity of p loaded on k th trip on route r
- $u_{r,k} \in \{0, 1\}$: binary indicator if trip k is used on route r
- $f_{r,p} \in \mathbb{N}$: total flow of product p on route r

Objective

$$\min \sum_{r \in \mathcal{R}} (c_r \times \left(\sum_{k=1}^{K_r} u_{r,k} \right)) \quad (\text{Minimize cost})$$

where c_r is the cost per trip on route $r \in \mathcal{R}$

Constraints

For route $r \in \mathcal{R}$ and product $p \in \mathcal{P}$:

$$f_{r,p} = \sum_{k=1}^{K_r} \alpha_{r,k,p} \quad (\text{Flow aggregation})$$

For product $p \in \mathcal{P}$ and $\ell \in \mathcal{L}$:

$$\sum_{r \in \mathcal{R}_\ell^{\text{in}}} f_{r,p} - \sum_{r \in \mathcal{R}_\ell^{\text{out}}} f_{r,p} = d_{\ell,p} \quad (\text{Flow conservation})$$

For route $r \in \mathcal{R}$ and $k \in \{1, \dots, K_r\}$:

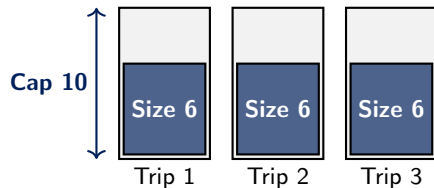
$$\sum_{p \in \mathcal{P}} (\alpha_{r,k,p} \times v_p) \leq (u_{r,k} \times Q_r) \quad (\text{Capacity})$$

where v_p is the size of p and Q_r capacity on r

Exact Flow Formulation: Scalability

Example: Route $P1 \rightarrow P3$ (Cap=10)

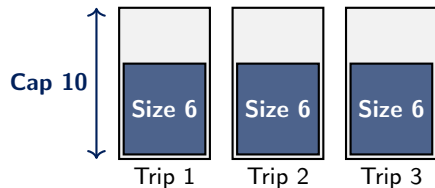
Flow: 3 items of size 6 = 18 volume units



Exact Flow Formulation: Scalability

Example: Route P1→P3 (Cap=10)

Flow: 3 items of size 6 = 18 volume units



To find this exact packing while simultaneously routing, for **every potential trip** we must instantiate variables.

Decision Space Formulation:

$$|\mathcal{R}| \times K_r \times |\mathcal{P}|$$

The Industrial Use Case

- **5,610** possible routes
- Up to **50** trips per route
- **35** distinct products

Result: $\approx$ 10 Million variables.

Two-Step Approach - Step 1: Aggregated Flow

Core Concept: Replace per-trip tracking with a total volume.

Model Formulation

New variable: $n_r \in \mathbb{N}$ = total # trips on route r .

$$\sum_p f_{r,p} \cdot v_p \leq n_r \cdot Q_r \quad (\text{Global capacity})$$

$$\min \sum_r c_r \cdot n_r \quad (\text{Minimize trips})$$

Instead of managing individual discrete trips, we enforce that the **sum of volumes** for all products routed on r fits within the total capacity provided by n_r trips.

Why is this an approximation?

- It treats discrete products as divisible entities (LP relaxation).
- It estimates the required number of trips purely based on raw volume:

$$n_r = \left\lceil \frac{\text{Total Volume}}{Q_r} \right\rceil$$

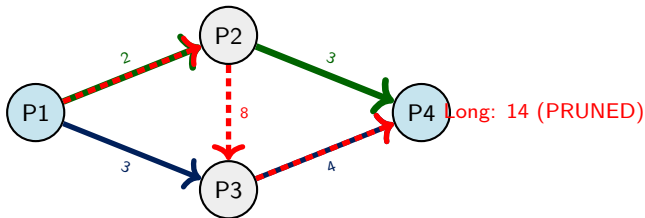
- **Interest:** This provides a very fast **lower bound** for the solver, but completely ignores physical packing constraints and item indivisibility.

Scaling Technique: Shortest-Path Pruning

Strategy: For each origin-destination pair, compute shortest path and keep only routes within $\tau \times$ shortest cost

Example: Paths from P1 to P4

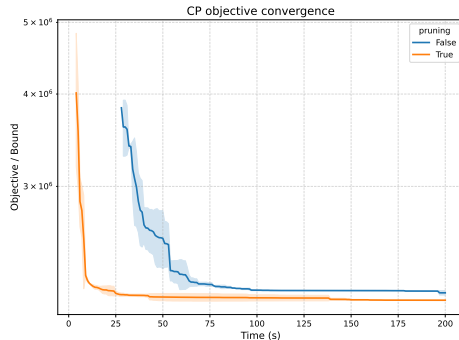
Shortest: 5 (KEPT)



Alternative: 7 (KEPT)

With $\tau = 1.5$: $\Rightarrow$ 5,610 routes down to 1,849 (67% reduction)

Impact on Convergence



Two-Step Approach - Step 2: Bin Packing Post-Processing

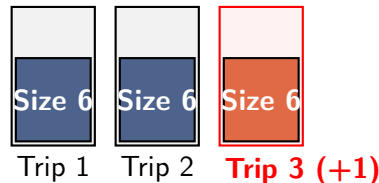
Core Concept: Repair the floating flows into feasible discrete packings.

Example Recap (Route P1→P3):

- Capacity (Q_r): 10
- Step 1 Flow: 3 items of size 6

The Gap:

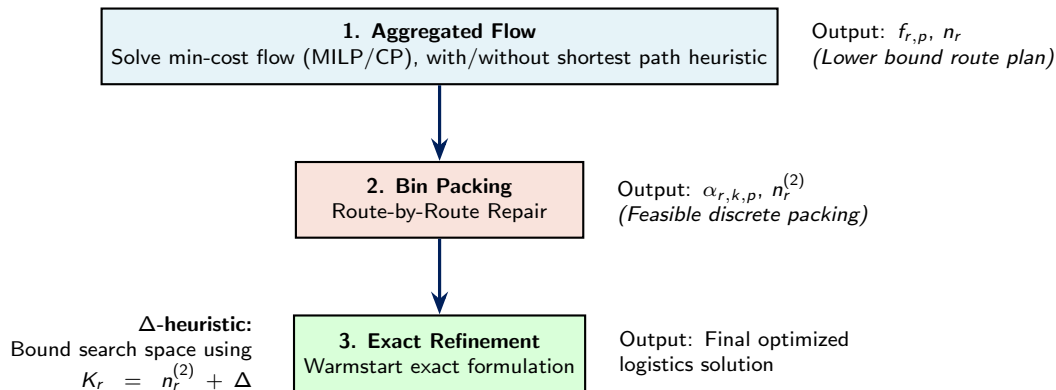
- **Step 1:** Volume = 18 $\Rightarrow n_r = \lceil 18/10 \rceil = 2$ trips
- **Step 2:** Items of size 6 cannot share a bin (capacity 10) $\Rightarrow$ Need 3 trips



Packing Strategy:

Solve bin packing per route using CP, ASP, or greedy heuristics.

Sequential Solving Pipeline Overview



By decoupling routing from packing, we reduce complexity while maintaining near-optimal solution quality.

Evaluation: Impact of Pruning & Refinement

Instance: 50 locations, 5,610 routes, 35 products. **Timeout:** 500s.

Steps	Flow Solver	Pruning	Packing	Refinement	Mean Cost
1	MILP ^{max}	No	–	–	2,200,596
2	CASP	No	RbR-Greedy	–	2,671,101
	MILP	No	Global-CP	–	2,103,689
	MILP	No	RbR-Greedy	–	2,138,310
	MILP	Yes	RbR-Greedy	–	2,123,164
	CP	No	Global-CP	–	2,114,727
	CP	Yes	RbR-CP	–	2,108,942
3	MILP	No	RbR-Greedy	CP ^Δ	2,102,043
	MILP	Yes	RbR-Greedy	CP ^Δ	2,112,065
	CP	No	RbR-Greedy	MILP ^Δ	2,133,188
	CP	Yes	RbR-Greedy	MILP ^Δ	2,108,145

Key Observations

- **Flow solver:** MILP > CP > CASP
- **Packing:** Global-CP best, RbR-Greedy faster
- **Pruning:** Speeds up CP, may degrade quality slightly

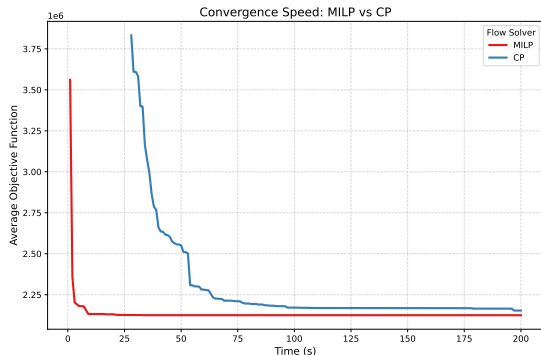
Best Configuration

- 3-step: MILP + RbR-Greedy + CP^Δ
- No pruning for optimal quality
- Cost: **2.10M** (5% better than 1-step)

Step 1: MILP vs CP Performance

Question: Which solver works best for aggregated flow formulation?

Convergence Speed



Why MILP fits better for Step 1

- **Flow conservation** is naturally linear (LP structure)
- **Aggregated capacity** is a simple linear inequality
- **No discrete packing** constraints in Step 1
- MILP solvers exploit LP relaxation efficiently

Result: MILP achieves best objective (2.10M) in 10s
CP even with 200s fails to reach similar quality

Available Resources

• Problem Implementation

- Multi-batching problem class
- Solution representation
- Objective function computation

• Industrial Instance

- 50 locations, 5,610 routes, 35 products
- Realistic demands and capacities
- Publicly shared benchmark

• Solution Checker

- Validates feasibility
- Computes objective value
- Verifies all constraints

Solver Implementations

Paper approaches + additional solvers:

- **Modeling:** MILP, CP, CASP, generic Minizinc one.
- **Two-step:** Aggregated flow + packing repair (Greedy, Global-CP, Route-by-Route BPP)
- **Three-step:** + Exact refinement with Δ -heuristic
- **Heuristics:** Shortest-path pruning

GitHub:

`github.com/airbus/discrete-optimization`
multibatching

Main Results

- **Sequential solving is mandatory** and outperforms direct exact formulations at industrial scale.
- The Three-step approach ($MILP + greedy + CP/MILP^{\Delta K_r}$) achieves the best operational cost.

Ways Forward

- **Extend problem:** Integrate dynamicity of the problem, storage costs, tied-up capital, lead times, 3D packing.
- **Deployment:** Integration with Airbus digital decision-support systems.

Thank you! Any Questions?