

Constraint Programming for Mixed-model Assembly Line Scheduling with Complex Industrial Constraints

Guillaume Pováda¹ Javier Buil Tejero² Tamara Borreguero Sanchidrian³

¹Airbus SAS, Toulouse, France

²ALTEN GmbH, Manching, Germany

³Airbus DS, Getafe, Spain

CP 2026 – Lisbon, Portugal

AIRBUS

ANITI

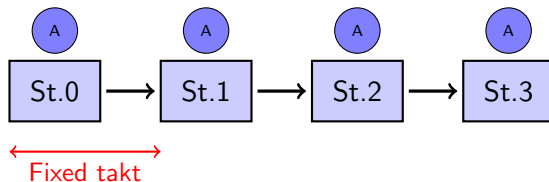
Outline

- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution
- 4 Computational Results
- 5 Software & Benchmarks
- 6 Conclusion

- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution
- 4 Computational Results
- 5 Software & Benchmarks
- 6 Conclusion

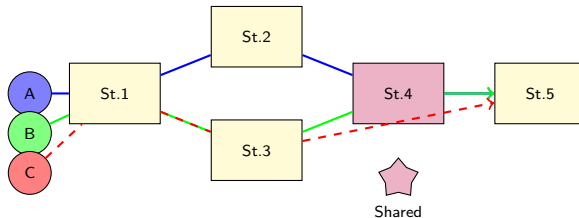
From Paced Lines to Flexible Assembly

Traditional : Linear Paced Line



Same product, fixed sequence, predictable rhythm

Today's Challenge: Mixed-Model Assembly



Aircraft Assembly Problem

Scale & Complexity:

- 6-month horizon, ~50 products
- ~ 500 "meta" tasks to schedule
- ~ 30 Station, ~ 10 material and workforce resources pool, each of them with calendars (not necessarily synchronized calendars)

Objectives (conflicting):

- Minimize tardiness: each product has a desired deadline.
- Minimize earliness (or Just in Time) cost
- Minimize resource usage, makespan ...

Dual Purpose

Operational: Re-schedule on supply chain delays
ramp-up scenarios

Design: Validate factory layouts under

- 1 Industrial Context
- 2 Problem Formulation**
- 3 Algorithmic Contribution
- 4 Computational Results
- 5 Software & Benchmarks
- 6 Conclusion

① Calendar-based Preemption

- Tasks pause during unavailability (shifts, weekends)
- Effective duration depends on *when* task starts

② Group-based Resource Blocking

- Station reserved for entire task group
- No interleaving with other products

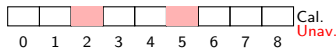
③ Generic Resource Blocking ("Bridging")

- Resource blocked during transitions
- Models material handling (jigs, transport)

④ (not in this talk), Multimode, generalized precedence, time-windows etc.

Calendar Preemption

Problem: Task, nominal duration 4



Different starts $\Rightarrow$ different durations:

Start=0:  $D = 5$

The bar for Start=0 consists of 5 segments: blue (0-1), red (2), blue (3-4), red (5), and white (6-8).

Start=1:  $D = 6$

The bar for Start=1 consists of 6 segments: white (0), green (1-2), red (3), green (4-5), red (6), and white (7-8).

Prerequisites:

S_i : start time, $D_{i,m}$: duration, $\mathcal{T}_{i,m,d}$: valid starts for duration d

1. Pre-compute duration function:

$$\Phi_{i,m}(0) = 5, \Phi_{i,m}(1) = 6, \Phi_{i,m}(3) = 5, \dots$$

2. Binary indicators $\delta_{i,m,d}$:

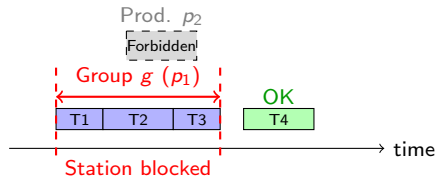
- $\delta_{i,m,d} = 1$ iff duration d is selected
- Links task start time to valid start time domain

3. CP constraints:

- Unique duration: $\sum_{d \in \Delta_{i,m}} \delta_{i,m,d} = 1$
- Duration binding: $\delta_{i,m,d} = 1 \Rightarrow D_{i,m} = d$
- Start binding: $\delta_{i,m,d} = 1 \Rightarrow S_i \in \mathcal{T}_{i,m,d}$

Resource Blocking Constraints

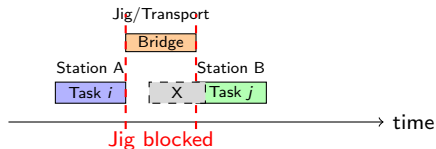
Group Blocking: Station reserved



Constraint: $(g, k, q) \in \mathcal{BG}$

- q units of resource k blocked
- From first to last task in group
- Master interval I_g spans group

Generic Blocking (Bridging):



Constraint: $(X_{src}, X_{dst}, k, q) \in \mathcal{B}$

- q units blocked from $\text{end}(X_{src})$ to $\text{start}(X_{dst})$
- Models material handling, setup
- Interval I_{bridge} with demand q

Variables

- Each task i in mode m : optional interval $l_{i,m}$
- Mode selection: exactly one mode per task
- Indicators $\delta_{i,m,d}$ link start time to duration

Resource Constraints: For each resource k , aggregate intervals:

- Task execution intervals
- Group blocking intervals
- Generic blocking (bridging) intervals

Then apply global constraints:

- **Disjunctive:** NoOverlap for unary resources
- **Cumulative:** Cumulative for multi-capacity resources

- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution**
- 4 Computational Results
- 5 Software & Benchmarks
- 6 Conclusion

Direct solve result

Monolithic CP Model: Solve entire 6-month horizon at once

Simple Instances (S1-S3):

- 1-2 products, 8-25 tasks
- Direct CP: **100% success**
- Optimal solutions found

Hard Instances (H1-H6):

- 50 products, 396-534 tasks
- 284-392 blocking constraints
- Direct CP: **0% success**
- **Cannot find feasible solution!**

Why Direct CP Fails?

- Blocking constraints create tight coupling
- Calendar interactions multiply variables
- Some ablation study show that the resource blocking is the main bottleneck.

Idea

Assembly has natural *phases*

⇒ Can we exploit this structure?

Sequential Meta-Algorithm

Main Idea: Decompose into sequence of smaller sub-problems

① Atom decomposition

- Build dependency graph: edges between task sharing "hard constraint" i.e synchronisation or blocking resource.
- Connected components = **atoms** (tightly coupled tasks)

② Topological ordering

- Order atoms using classical precedence constraints
- Group into N batches (balance size vs. flexibility)

③ Sequential solving

- For each batch $k = 1$ to N :
- Fix previous batches, optimize current batch
- Use warmstart for final monolithic model

Result

Always feasible — no global optimality guarantee

Atom Decomposition Example

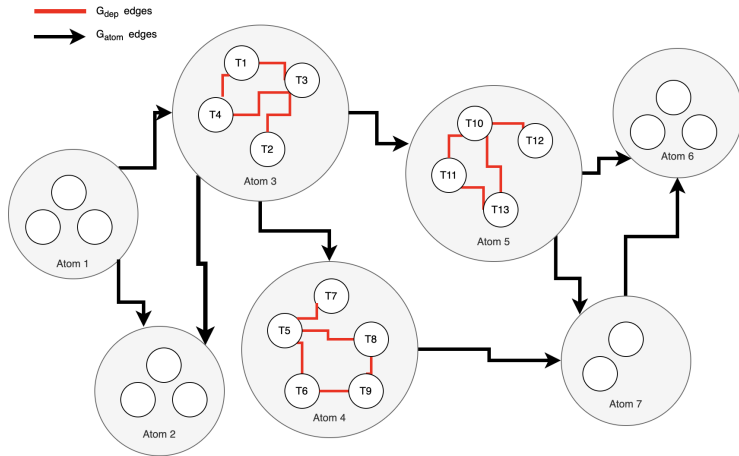


Figure: Atoms decomposition and batch aggregation

Sequential Solving: Iteration Process

Example from previous slide:

- Batch 1: Atoms 1, 3, 2
- Batch 2: Atoms 4, 5
- Batch 3: Atoms 6, 7

Iteration k=1:

- Include tasks from **Batch 1**
- No fixed decisions yet
- Optimize with time budget

Then iterate...

- Fix Batch 1, solve Batch 2
- Fix Batches 1 & 2, solve Batch 3

Solving

Batch 1 (A1,A3,A2)

Future

Batch 2 (A4,A5)

Future

Batch 3 (A6,A7)

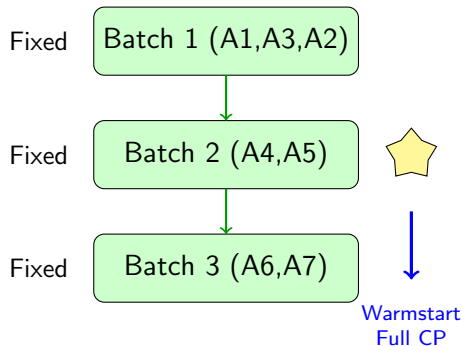
Sequential Solving: Iteration Process

Example from previous slide:

- Batch 1: Atoms 1, 3, 2
- Batch 2: Atoms 4, 5
- Batch 3: Atoms 6, 7

Final: Warmstart Monolithic

- Sequential solution = feasible starting point
- Used as **warmstart** for full CP model
- Allows further optimization if time permits



- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution
- 4 Computational Results**
- 5 Software & Benchmarks
- 6 Conclusion

Experimental Setup

Real Industrial Instances:

Set	Prod.	Tasks	Prec.	Block.
S1-S3	1-2	8-25	13-31	1-18
H1-H6	50	396-534	542-728	284-392

Challenge

50 products, 400-500 tasks, 300-400 blocking constraints

Benchmarks: Artificial generator released

Solver Setup:

- **Solver:** OR-Tools CP-SAT
- **Workers:** {1, 8, 16}
- **Time:** 1000s per objective
- **Hardware:** ARM 2.42 GHz, 32GB

Objective Orders:

- **Order A:** Tardiness only
- **Order B:** JIT only
- **Orders C/D:** Lexicographic combinations

Results: Success Rate

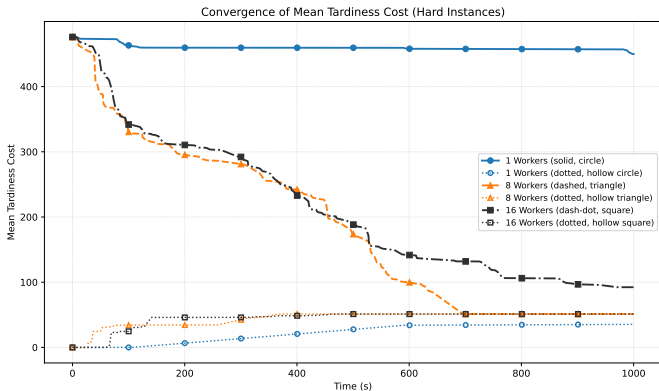
Method	Simple	Hard
Direct CP	100%	0%
Sequential CP	100%	100%

Table: Success rate

Trade-off

- + Always feasible
- + Scalable
- + Fast (minutes vs timeout)
- No global optimality

Solver Performance



Parallelism Impact:

- Best: 8 workers
- Diminishing returns at 16
- CP-SAT parallel search effective

Figure: Evolution of tardiness with parallel workers

Schedule Visualization

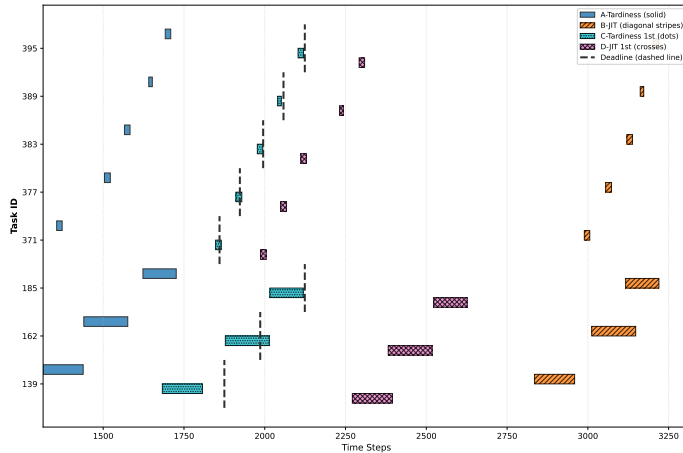


Figure: Gantt charts comparing objective orders (4 products excerpt)

- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution
- 4 Computational Results
- 5 Software & Benchmarks**
- 6 Conclusion

Open-Source: discrete-optimization

Python Library:

- Generic Problem/Solution/Solver API
- Multiple backends: CP, MILP, heuristics
- Solver pipelines & hybrid approaches
- Benchmarking framework

Backends:

- ortools: cpsat, mathopt, routing
- didp,
- gurobi,
- minizinc, cpmPy, optalcp

Repository

`github.com/airbus/
discrete-optimization`
This paper : **flex_scheduling**
submodule.

Benchmarks:

- Industrial instances
- Instance generator
- Reproducible scripts

- 1 Industrial Context
- 2 Problem Formulation
- 3 Algorithmic Contribution
- 4 Computational Results
- 5 Software & Benchmarks
- 6 Conclusion

Contributions:

- ① **Problem:** Extended RCPSP with calendar preemption + resource blocking
- ② **CP:** Very good framework to model the problem, but direct solve was not enough
- ③ **Algorithm:** Sequential meta-algorithm
 - Atom decomposition + topological ordering
 - 100% success on industrial instances
 - Scales where direct CP fails
- ④ **Software:** Open-source library + benchmarks
- ⑤ **Future works**
 - custom LNS for this problem
 - HITL optim
 - Alternative multiobjective methods
 - Generalize the sequential decomposition to other problems
 - Try native preemptive cp model
 - Logic benders decomposition for the resource blocking constraint ?
 - ... ?

Thank you!

Questions?

Guillaume Pováda

guillaume.poveda@airbus.com

Paper: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CP.2026.46>