

Modeling and Explaining an Industrial Workforce Allocation Problem

Ignace Bleukx¹

Ryma Boumazouza^{2,3}

Tias Guns¹

Nadine Laage²

Guillaume Poveda²

¹KU Leuven, Belgium

²Airbus, Toulouse

³IRT Saint-Exupery, France

Sister conference track

Best Application Paper Award CP 2025

Modeling and Explaining an Industrial Workforce Allocation Problem

Ignace Bleukx¹

Ryma Boumazouza^{2,3}

Tias Guns¹

Nadine Laage²

Guillaume Poveda²

¹KU Leuven, Belgium

²Airbus, Toulouse

³IRT Saint-Exupery, France

Sister conference track

Best Application Paper Award CP 2025

The Challenge: Workforce Allocation at Airbus



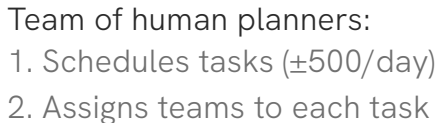
Airbus manufacturing sites
Toulouse/Hamburg/...



Assigning worker teams to logistical tasks
e.g., transporting aircraft parts

Transportation "tasks"
Equipment, duration, deadlines ...

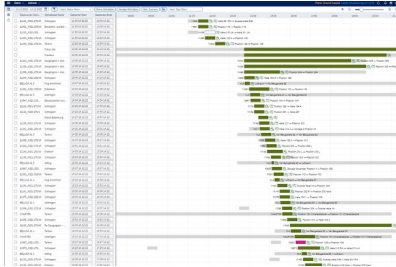
AIRBUS



Working team qualifications + calendars,
Manufacturing deadlines,
Task precedences,
High-level logistics

...

Three main planning tasks



1. Schedule tasks (based on order book)
2. Allocate tasks to worker teams
3. Monitor and resolve disruptions

Contribution 1:
CP-Modeling of allocation + scheduling problem

Goal: find optimal allocation/schedule*

Constraints: based on flexible-jobshop problems
~reduces to graph coloring for pure allocation



Model

+



Solve

* Several modeling tricks to ensure fast runtime!

Goal: find optimal allocation/schedule*

Constraints: based on flexible-jobshop problems
~reduces to graph coloring for pure allocation



Model

+



Solve

Objective 1: ensure efficient schedule
Minimize nb of teams required

* Several modeling tricks to ensure fast runtime!

Goal: find optimal allocation/schedule*

Constraints: based on flexible-jobshop problems
~reduces to graph coloring for pure allocation



Model

+



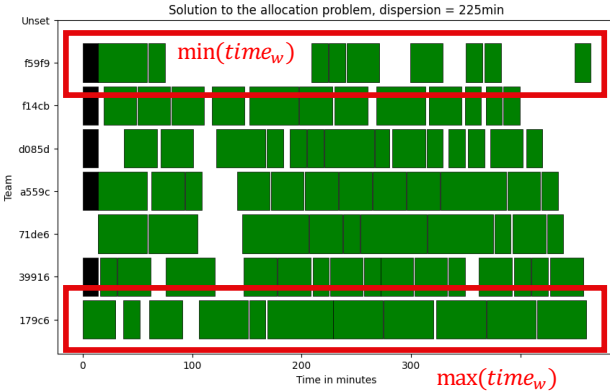
Solve

Objective 1: ensure efficient schedule
Minimize nb of teams required

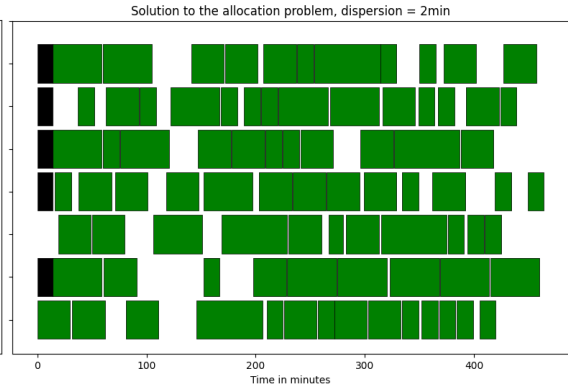
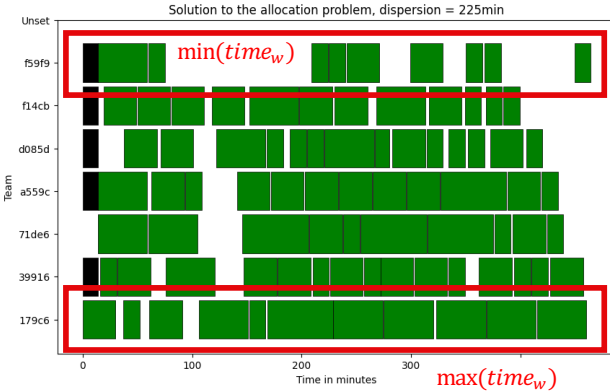
Objective 2: ensure *fair* schedule
Minimize *time-dispersion*
(max time worked – min time worked)

* Several modeling tricks to ensure fast runtime!

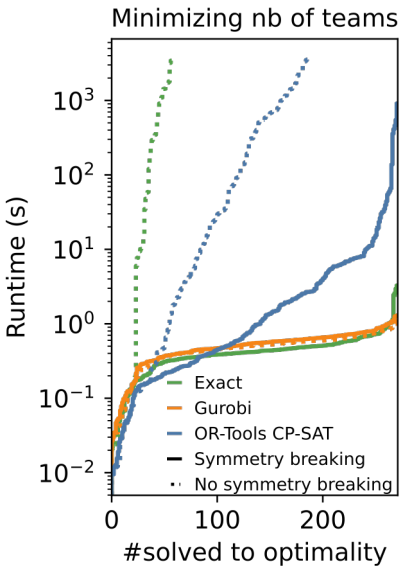
Example solutions



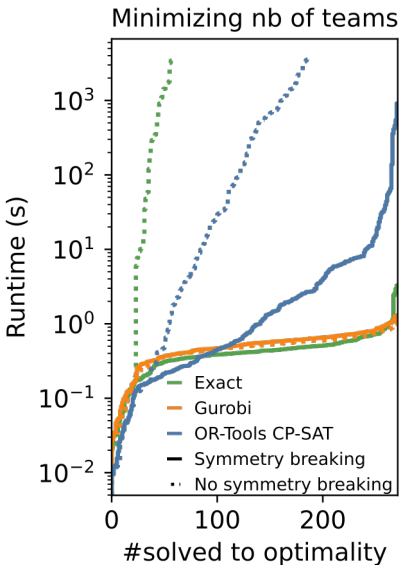
Example solutions



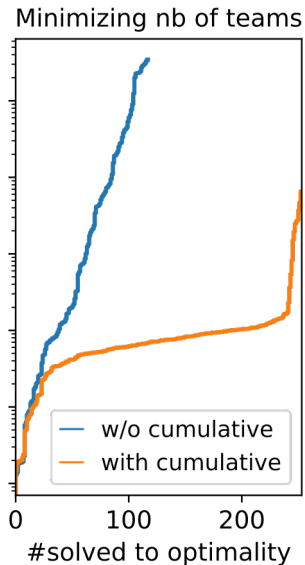
Allocation-only



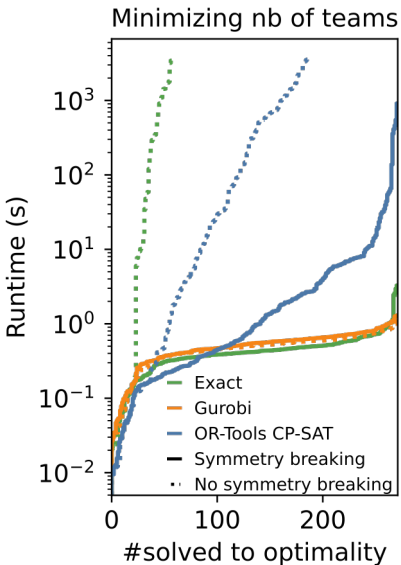
Allocation-only



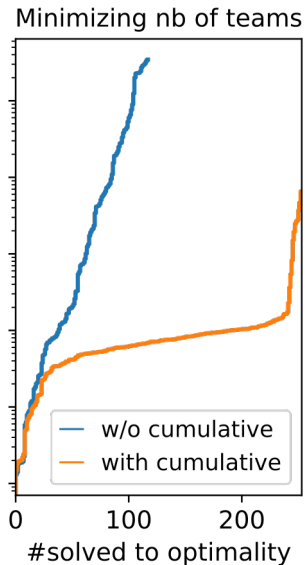
Alloc + scheduling



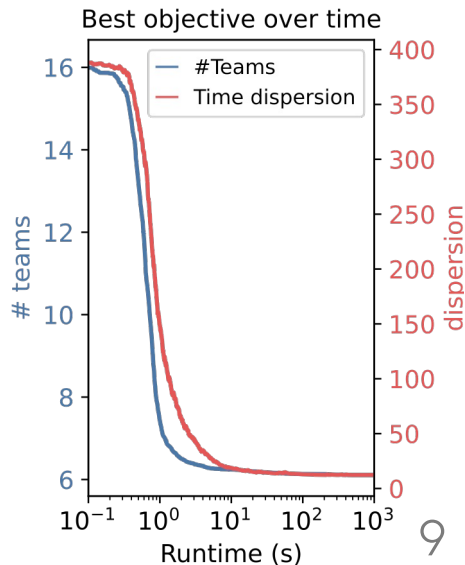
Allocation-only



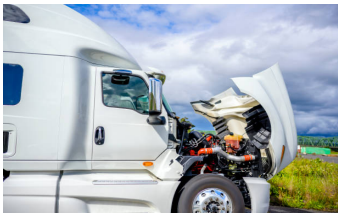
Alloc + scheduling



Anytime objective



Contribution 2: Explanations for disruptions



Unforeseen Disruptions

Unforeseen events are common
delays, weather, team unavailability

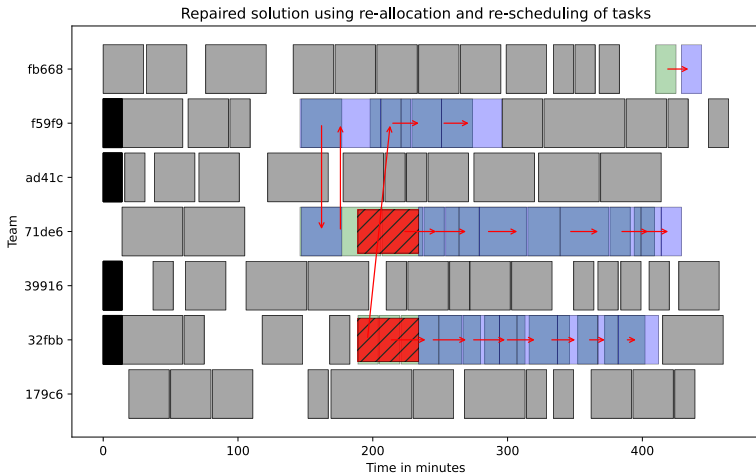
Invalidate the precomputed schedule
also for manually-crafted schedules



Planners have a limited window to react
Reduce “wait and see” time of worker-teams



Fine-grained repair of schedules



Optimize:

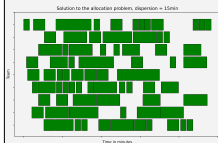
1. #tasks allocated
2. #tasks re-allocated
3. #tasks re-scheduled
4. Max-shift of tasks
5. Avg-shift of tasks

Modeling and Explaining an Industrial Workforce Allocation Problem



Ignace Bleukx, Ryma Boumazouza, Tias Guns, Nadine Laage, Guillaume Poveda

Use-case: Workforce allocation at Airbus manufacturing sites



Logistical tasks in manufacturing plant (e.g., part transportation) must be **scheduled and assigned** to specialized worker teams

Such that the resulting schedule is

1. Efficient: uses a **minimal** number of worker teams
2. Fair: each team is allocated a similar **amount of work**

Modeling with CPMpy

CP-model is based on **flexible-jobshop** problems
Each task is allocated to exactly 1 team

$$\sum_{w \in W} x_{t,w} = 1$$

Ensure team is compatible with task

$$\neg \text{Compat}_{t,w} \Rightarrow \neg x_{t,w}$$

Assign task "clusters" to the same team

$$\text{AllEqual}(\{x_{t,w} \mid t \in g\})$$

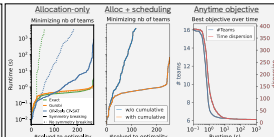
Enforce precedences

$$s_t + d_t \leq s_{t'}$$

Assign overlapping tasks to different teams

$$\text{NoOverlap}(\{\text{OptInterval}(s_t, d_t, x_{t,w}) \mid t \in T\} \\ \{\text{Interval}(c_s, c_d) \mid (c_s, c_d) \in \text{Calendar}_w\})$$

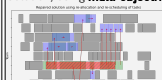
Redundant constraint: model pool of teams as cumulative resource (= pool of identical teams)



Explaining & Resolving disruptions

Unforeseen disruptions can invalidate schedule
E.g., technical breakdown, adverse weather...

"Repair" model while staying close to original solution, using **multi-objective** optimization



Optimize:

1. #tasks allocated
2. #tasks re-allocated
3. #tasks re-scheduled
4. Max shift of tasks

Poster at 16h30
Panel 2.10-2.12