Integrated crew management for rail freight

Héloïse Gachet, Frédéric Meunier, Juliette Pouzet

20 november 2024

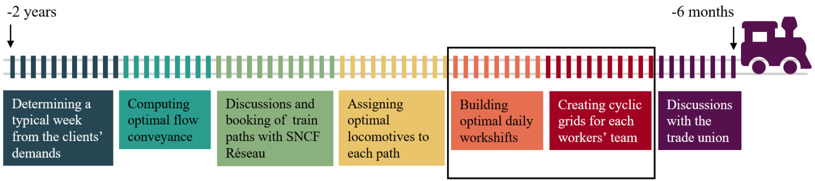
SNCF

CERMICS
École nationale des ponts et chaussées

Rail freight in France

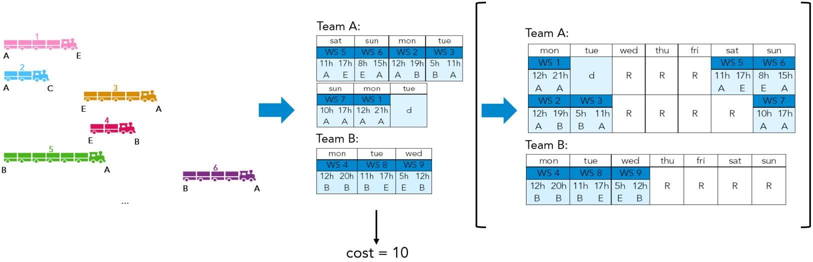
- 18% of rail freight in Europe (10% in France)
- Between 1800 and 2000 trains per week
- Many differences with passengers transportation...
  - Priority goes to passengers
  - Trains mostly at night
  - Client satisfaction: delivery on time
  - Key goal: reducing operational costs

Resource planning



Input: Trains on a typical week

Output: Covering of trains by "blocks" with minimum cost, each block assigned to a team (each block then placed in a grid)

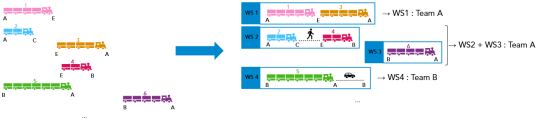


Heuristic: sequential approach

1. Building shifts from trains

   Input: Trains on a typical week

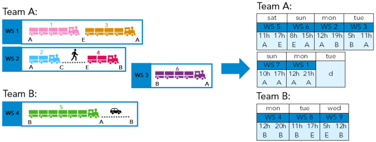
   Output: Covering of trains by shifts with minimum cost, each shift assigned to a team



2. Building grids from shifts

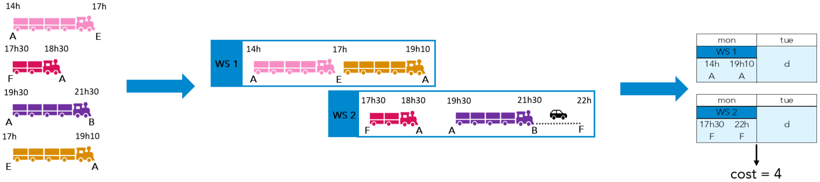
   Input: Shifts, each assigned to a team *(output above)*

   Output: For each team, covering of shifts by blocks with minimum cost

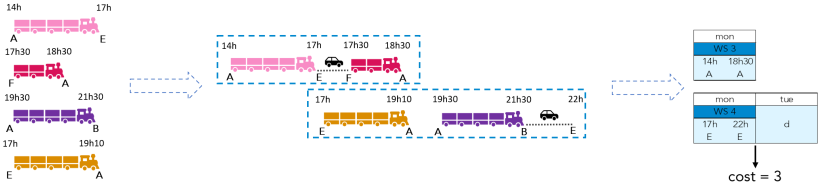


Sub-optimal approach

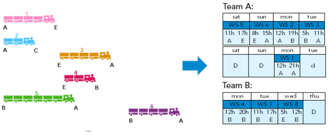
Sequential approach:



Optimal approach:



Model



$$\text{Min} \quad \sum_{i\in I}\sum_{b\in B_i} c_b x_{b,i}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{\substack{b\in B_i\\ t\in b}} x_{b,i} \geqslant 1 \quad \forall t \in T$$

$$x_{b,i} \in \{0,1\} \qquad \forall i \in I, \forall b \in B_i$$

B_i = set of feasible "blocks" for the team i

$\rightarrow$ combinatorial explosion

National input:

1 800 trains $\rightarrow$ 200 000 shifts $\rightarrow \sim 10^{25}$ blocks

Column Generation: standard resolution methodology of a linear program when the number of variables is large

Column Generation

Generation of variables throughout the resolution $\rightarrow$ solving the pricing sub-problem

Key: Solving the pricing sub-problem quickly

For our problem:
Pricing sub-problem = Shortest path with constraints (Resource Constrained Shortest Path)
$\rightarrow$ NP-hard problem

Shortest path with constraints

Input: Graph with vertices o and d, cost function and feasibility function on each path
Ouput: Feasible $o - d$ path with minimum cost

Feasibility is expressed as: "resources" accumulated along the path are under a given threshold

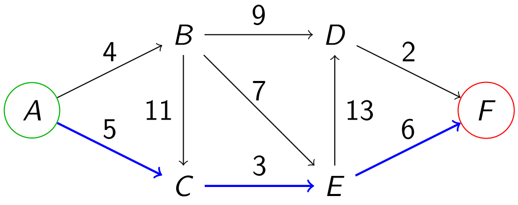
ex.: shortest $o - d$ path under energy budget
$\rightarrow$ *resources: accumulated arc energy*
minimum-cost block under shifts and block feasibility constraints
$\rightarrow$ *resources: accumulated shift range, number of days in block, etc.*

Shortest path

Usual shortest path

Input: Graph with vertices o and d, non-negative cost on each arc

Ouput: $o - d$ path with minimum cost



$\rightarrow$ Dijkstra's algorithm

$\rightarrow$ A^* search algorithm

A^* search algorithm

Principle: Enumeration algorithm with bounds to discard paths

- Bound b_v under-estimating cost of shortest path from any vertex v to d
- Discard paths P from o to v with "estimated cost" $c(P) + b_v$ greater than one of an explored $o - d$ path

$\rightarrow$ with $b_v = 0$: Dijkstra's algorithm

ex.: shortest route on a map
$\rightarrow$ $b_v =$ *distance as the crow flies*

Shortest path with constraints

Resolution: Enumeration algorithm using

- Key: order of paths processing
- Bound: under-estimate of the resources and cost to reach d
- [not in A^*] Dominance: comparison of resources for paths with same cost

Different algorithms[1]:

- Generalized A^*
  Key = "estimated cost", discard paths using "estimated cost"
- Label dominance
  Key = cost of path, discard paths using dominance
- Label correcting
  Key = "estimated cost", discard paths using "estimated cost" and dominance

[1] synthetized by A.Parmentier in Algorithms for Non-Linear and Stochastic Resource Constrained Shortest Paths, 2017.

For our problem

- Vertices are trains
- Arcs can be of different types, mainly:
    - succession of trains within shift
    - change of shifts between trains
- Resource contains several indicators, as:
    - block indicators (number of days, etc.)
    - previous shift indicators (range, driving duration, etc.)
    - current shift indicators

Results

	Sequential approach		Our approach				
	Objective	Time	Obj. lower bound	Obj. upper bound[2]	Total time	Nb iterations	Nb variables
Instance 195 trains	176	2s	149.9	**151 (-14%)**	15min31	8	128 527
Instance 265 trains	261	3s	227.3	**229 (-12%)**	28min13	11	175 716
Instance 400 trains	402	5s	344.2	**347 (-13%)**	32min46	9	108 150

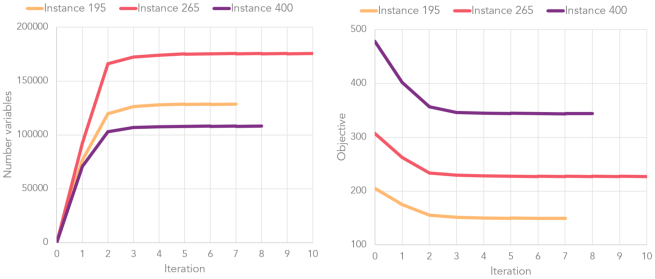


Figure: Column Generation indicators per iteration

[2] solving MILP with CG output variables

Current limitations

- Overall computation time
  $\rightarrow$ Preprocessing time: building pricing graph
  $\rightarrow$ Pricing sub-problem time
- Extend results to a national instance

Thank you!