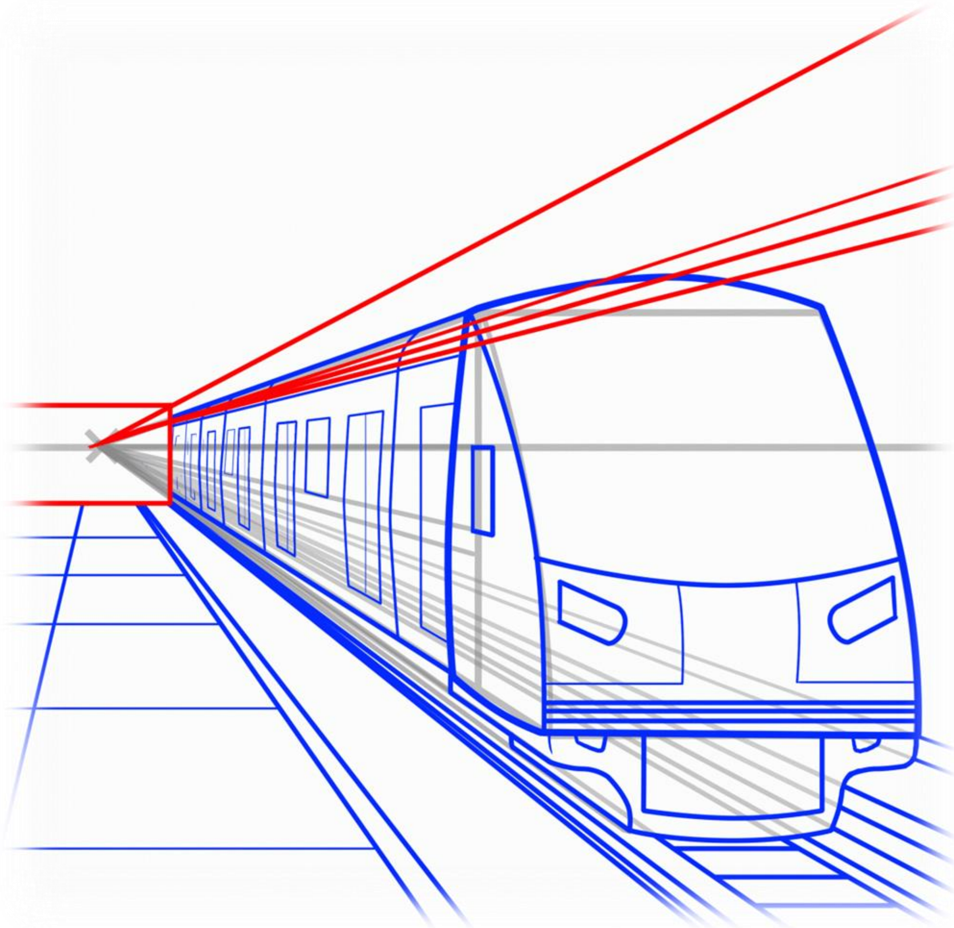
A hybrid model for robust crew scheduling in rapid transit networks

Manuel Fuentes González
Luis Cadarso Morga
Ángel Marín Gracia

Technical University of Madrid
2018 INFORMS Annual Meeting Phoenix
November 4, 2018

Outline

1. Overview
2. Mathematical model
3. Solution approach
4. Case study
5. Conclusions



Outline

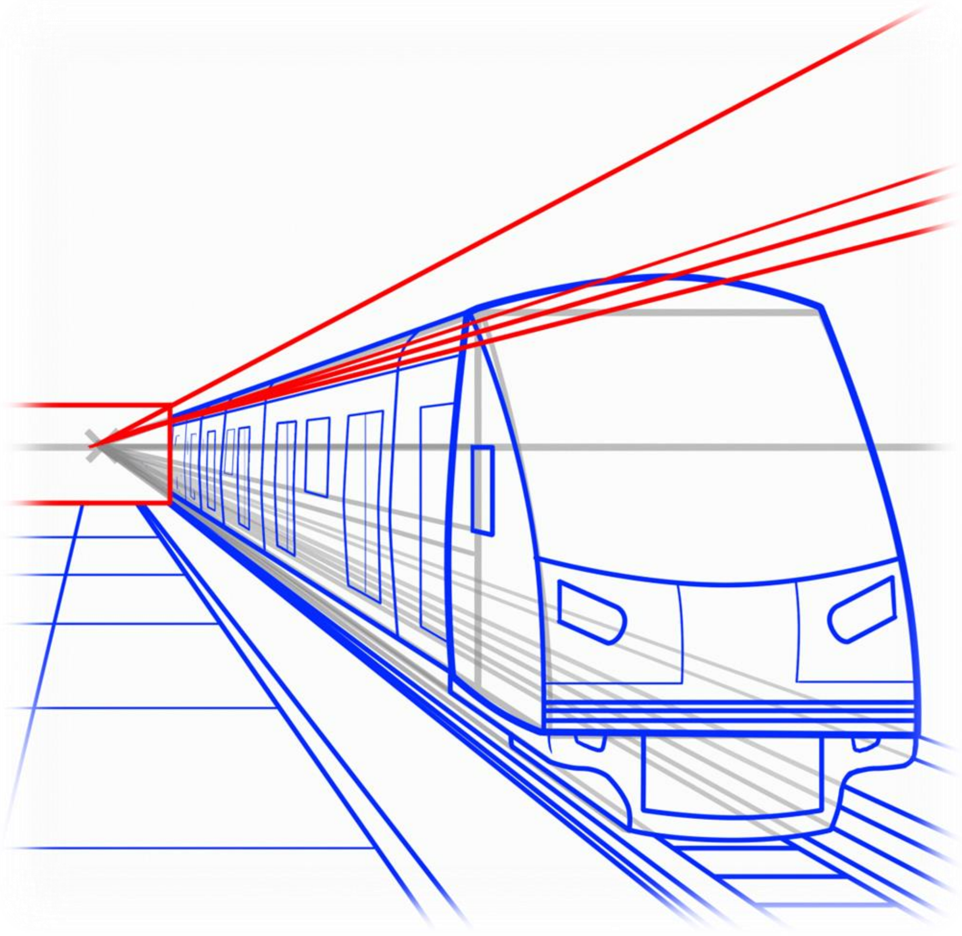
1. Overview

2. Mathematical model

3. Solution approach

4. Case study

5. Conclusions



Railway Transport Operator Planning

Typical railway transport operator planning process:

SCHEDULE PLANNING

↓

CIRCULATIONS
- ROLLING STOCK
  ↓
- VEHICLE ROUTING & MAINTENANCE

↓

CREW PLANNING
- CREW SCHEDULING
  ↓
- CREW ROSTERING

Basic Definitions

- **Task:** shortest single operation to be performed by a crew-member.
- **Deadheads:** tasks which are not performed by a crew-member, but used for their placement in the right station and time.
- **Duty:** sequence of tasks to be performed by a single crew-member within a working shift.
- **Roster:** sequence of duties to be performed by a crew member within a medium-term period, such a month.

Crew Planning

The problem is often divided into two subproblems solved sequentially:

1) **Crew Scheduling Problem (CSP):** assigning crew resources to certain tasks derived from the schedule planning problem, taking into account the circulations problem solution.

2) **Crew Rostering Problem (CRP):** determining rosters of crew members based in CSP solutions.

Rapid Transit Networks

- Particularities
  - Tasks typically short and frequencies high, leading to combinatory complexity.
  - Duties composed by many tasks.
  - Daily basis.
  - Simple cost structure, which allows linear modelling.

- They can be exploited by different approaches to the traditional one based in Set Covering or Set Partitioning.

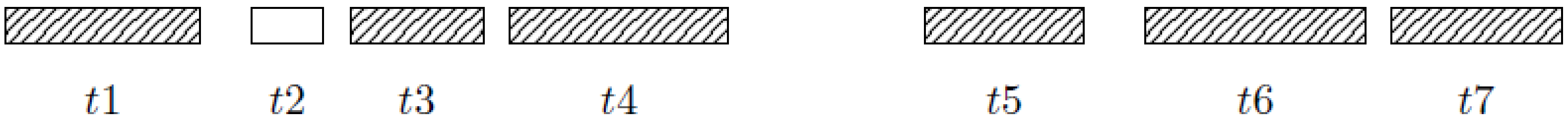
Motivation & Contributions

- **Set Partitioning/Covering Problem**
  - Choosing from a set of pre-computed duties.
  - It can deal with non-linear cost structures.
  - Usually solved by Column Generation **(**Shortest Path, etc.)

- **The presented formulation**
  - Hybrid, taking concepts from SPP/SCP and sequencing.
  - It takes advantage of the special particularities of the applied problem.
  - More similar to previous planning stages problems than SPP/SCP approaches.
  - An ad-hoc decomposition strategy: personnel-temporal based clustering.
  - Fix & Relax algorithm to speed up the solving process.

Crew Scheduling Problem (CSP)

Objectives



Goal: to minimize overall operational cost when grouping tasks into duties while keeping robustness of solutions.

Secondary objectives:

- Minimize presence of deadheads
- Minimize out-of-base overnight rests

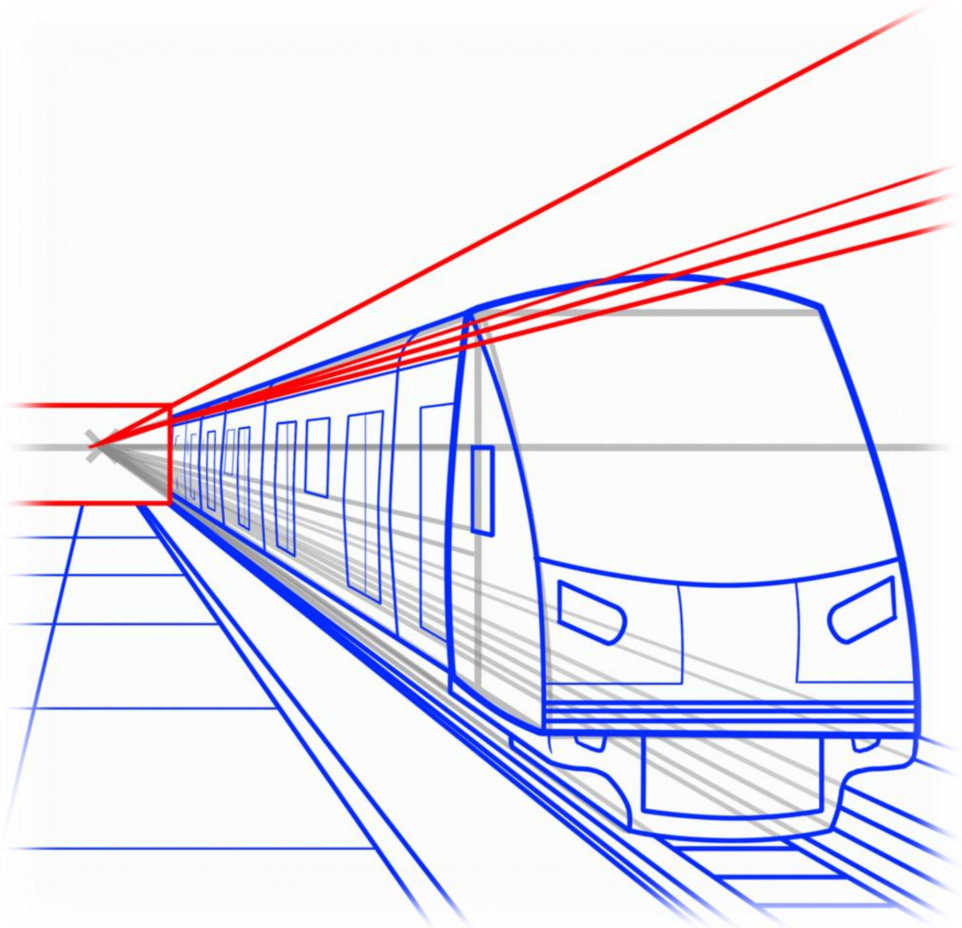
Crew Scheduling Problem (CSP)

Constraints:

- Physical constraints (spatial and temporal)
- Operator’s constraints (brief and debrief times, maximum connection times)
- Legal/Union’s constraints (duty duration, rest periods)

Outline

1. Overview
2. Mathematical model
3. Solution approach
4. Case study
5. Conclusions



Notations

Sets

- T is the set of tasks indexed by i, j or k.
- D is the set of potential duties indexed by d. $|D|$ is the maximum number of available drivers.

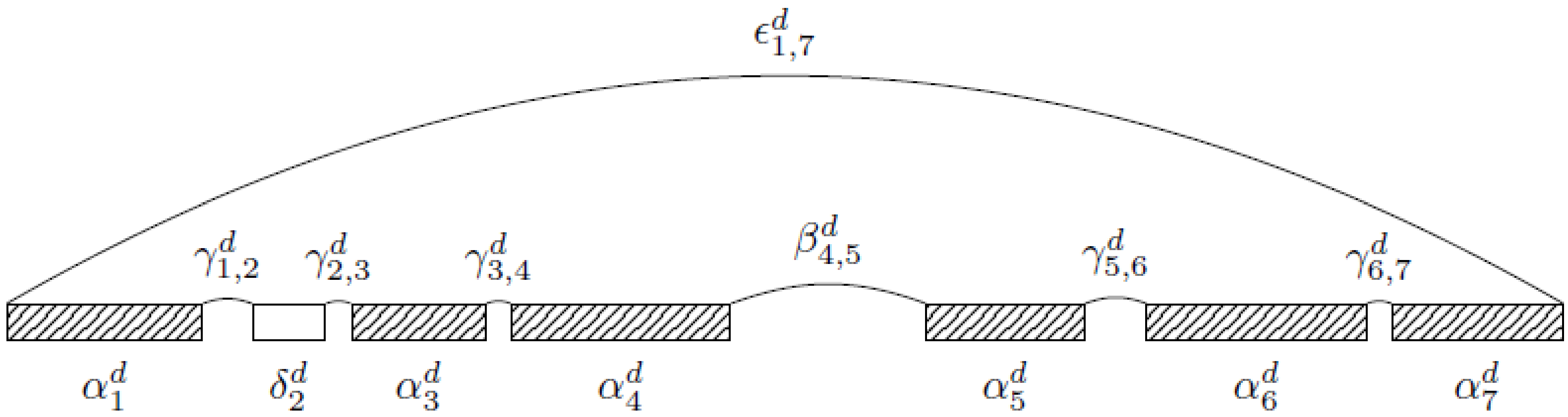
Notations

Variables

- $\alpha_i^d \in \{0,1\}$, defined for $i \in T$, $d \in D$, is 1 if task i is performed within duty d; 0, otherwise.
- $\delta_i^d \in \{0,1\}$, defined for $i \in T$, $d \in D$, is 1 task i is a deadhead within duty d; 0, otherwise.
- $\gamma_{i,j}^d \in \{0,1\}$, defined for $i, j \in T$, $d \in D$, is 1 if tasks i, j are consecutively performed within duty d; 0, otherwise.
- $\beta_{i,j}^d \in \{0,1\}$, defined for $i, j \in T$, $d \in D$, is 1 if tasks i, j are consecutively performed within duty d, considering the time between them as a legal break; 0, otherwise.
- $\epsilon_{i,j}^d \in \{0,1\}$, defined for $i, j \in T$, $d \in D$, is 1 if tasks i, j are the first and last tasks, respectively, within duty d; 0, otherwise.

Notations

Variables in a generic duty:



Model Formulation

Objective function

$$\min z = \sum_{i \in T} \sum_{j \in T} \sum_{d \in D} a^{i,j} e^{i,j} \epsilon_{i,j}^{d} + \sum_{i \in T} \sum_{d \in D} b^{i} \delta_{i}^{d} + \sum_{i \in T} u^{i} (1 - \sum_{d \in D} \alpha_{i}^{d})$$

- First term determines the total cost of the duties in the solution
  - It considers implicitly the cost of overnight rests, traveling to/from base, extra working hours, etc.
- Second term ensures the number of deadheads gets minimized.
- Last term penalizes the tasks which remain unassigned in the solution.

Model Formulation

Task covering and duty constraints

$$\sum_{d \in D} \alpha_i^d \leq 1 \qquad \forall i \in T$$

$$\alpha_i^d + \delta_i^d \leq 1 \qquad \forall i \in T; d \in D$$

$$\sum_{i \in T} \sum_{j \in T} e^{i,j} \epsilon_{i,j}^d \leq 1 \qquad \forall d \in D$$

Task sequencing constraints

$$\sum_{i \in T} c^{i,j} \gamma_{i,j}^d + \sum_{i \in T} d^{i,j} \beta_{i,j}^d + \sum_{k \in T} e^{j,k} \epsilon_{j,k}^d = \alpha_j^d + \delta_j^d \qquad \forall j \in T, d \in D$$

$$\sum_{k \in T} c^{j,k} \gamma_{j,k}^d + \sum_{k \in T} d^{j,k} \beta_{j,k}^d + \sum_{i \in T} e^{i,j} \epsilon_{i,j}^d = \alpha_j^d + \delta_j^d \qquad \forall j \in T, d \in D$$

Model Formulation

Break constraints

$$\sum_{i\in T}\sum_{j\in T}\beta_{i,j}^{d} = \sum_{i\in T}\sum_{j\in T}\epsilon_{i,j}^{d} \qquad \forall d\in D$$

$$\sum_{i\in T}\sum_{j\in T}\beta_{i,j}^{d}\cdot et_i - \sum_{i\in T}\sum_{j\in T}\epsilon_{i,j}^{d}\cdot st_i \le w \qquad \forall d\in D$$

$$\sum_{i\in T}\sum_{j\in T}\epsilon_{i,j}^{d}\cdot et_j - \sum_{i\in T}\sum_{j\in T}\beta_{i,j}^{d}\cdot st_j \le w \qquad \forall d\in D$$

Model Formulation

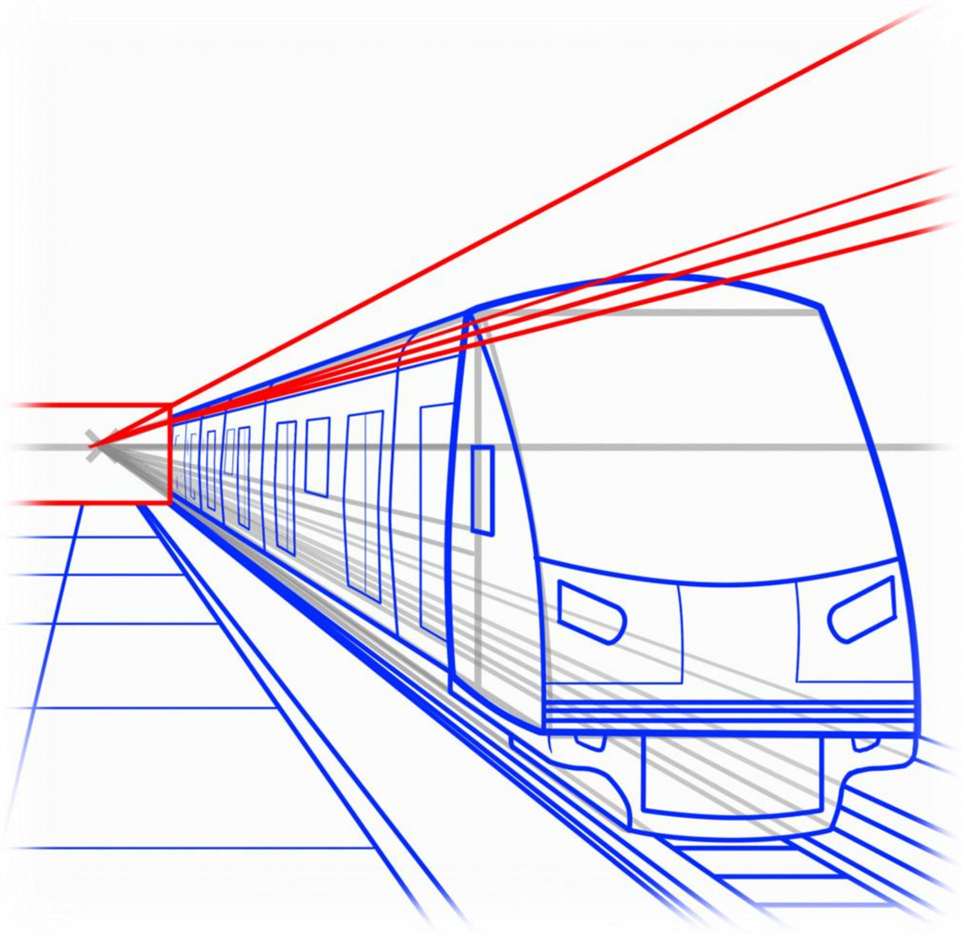
Robustness

Controlled by:

- Discarding critical connections by increasing the minimum time allowable for direct or break connections or by decreasing the maximum time for the extreme connections.
- Penalizing critical connections in the objective function.
- Penalizing crew-member movements in deadheading.

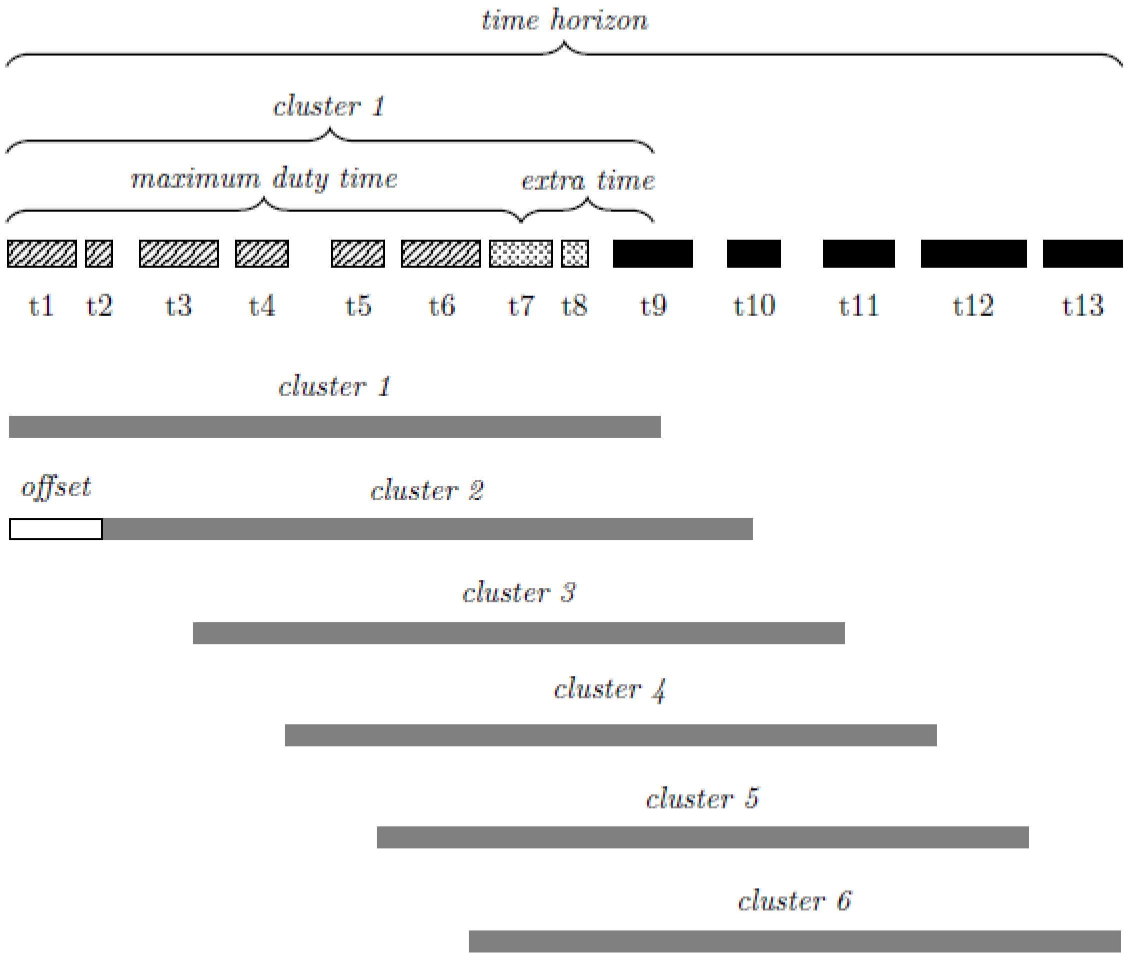
Outline

1. Overview
2. Mathematical model
3. Solution approach
4. Case study
5. Conclusions



Clustering

Clustering



Fix & Relax

- Solving a sequential series of mixed-integer or integer programming subproblems, such that the variables in each one are partitioned in three subsets.
  - The values of the variables in the first subset are fixed
  - The 0-1 variables in the second subset are kept integer
  - The integrality of the 0-1 variables in the third subset is relaxed

Fix & Relax

Algorithm 1 FRA

Step 0: (Initialization)
Set $i := 1$ for task 1.

Step 1: (On solving subproblem for task i)
Solve the model, see next. Output: $\hat{X}, z^i$.
The FRA reduced model of the original one (either clustered or unclustered) indexed with i is the same original model where the constraint system (21) is appended.

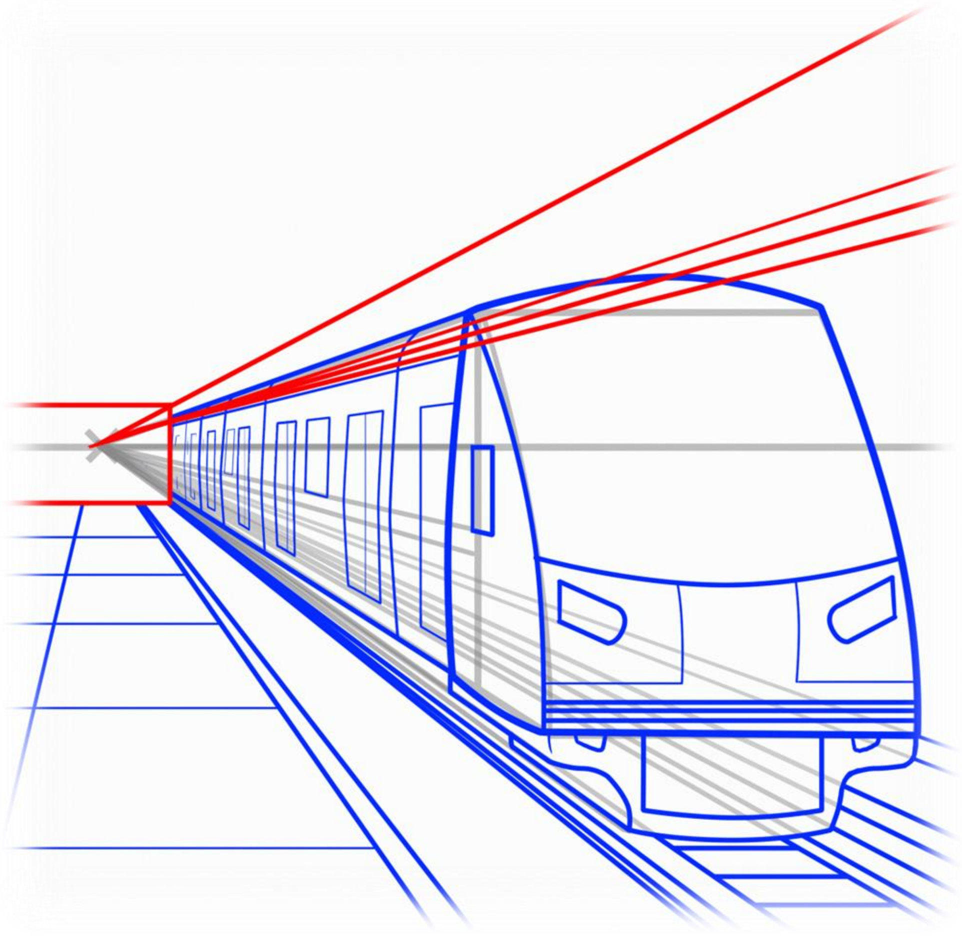
$$\begin{array}{ll} x_j = \hat{x}_j & \forall j \in \mathcal{A}^i : j \geq 1 \\ x_i \in \{0,1\} & \\ x_j \in [0,1] & \forall j \in \mathcal{B}^i : j \leq |T| \end{array}$$

where $\hat{x}_j$ are the values of the variables vectors x retrieved from the solution in the related FRA reduced model. $\mathcal{A}^i$ and $\mathcal{B}^i$ are the subset of tasks, which contain all the tasks before and after task i in the ordered list of tasks, respectively. Note that vector of variables x_i refers to $\{\alpha_i^d, \delta_i^d, \gamma_{i,j}^d, \beta_{i,j}^d, \epsilon_{i,j}^d\}$.

Step 2: (Testing the stopping criterion)
If $i = 1$ then compute the FRA lower bound of the solution value of the original model as z^1.
If $i = |T|$ or the integrality of the variables is satisfied then compute the FRA solution value of the original model as $\overline{z} := z^T$ and STOP.
Update level $i := i + 1$ and go to Step 1.

Outline

1. Overview
2. Mathematical model
3. Solution approach
4. Case study
5. Conclusions



Case study

- A daily schedule of the C-5 line of the Madrid rapid transit network operated by RENFE.
- Linear line with more than 50 train units dedicated.
- 23 stations and 4 depot stations.
- Covered by over 70 drivers.

HW & SW

- Server with an Intel Xeon E5-2620 @ 2.00GHz and 8x8GiB DIMM DDR3 1333 MHz, running under Ubuntu 14.04.3 (Linux 3.13.0).
- Models implemented in Python 3.6.3.
- Commercial solvers CPLEX 12.7.1 and Gurobi 7.5.1.
- Processor usage up to 8 threads per run.

Sensitivity Analysis on Clustering

Table 1: Sensitivity analysis results for clustering parameters

Parameter	60%	80%	100%	120%	140%
ϕ	83.597	83.468	83.448	83.443	83.463
θ	83.477	83.453	83.448	83.478	83.500
ψ	84.068	83.712	83.448	83.421	83.407

- Parameters:
    - Extra time (20% of maximum duty duration)
    - Offset time (15% of maximum duty duration)
    - Potential duties (increment of 30% of the expected number of duties)

Sensitivity Analysis on Connections Generation

Table 2: Sensitivity analysis results for connections generation parameters.

Parameter	80%	90%	100%	110%	120%
ξ	84.179	83.003	82.997	82.997	82.997
η	84.140	83.623	82.997	82.720	82.335
ζ	81.379	81.468	82.997	119.269	381.678

- Parameters:
  - Maximum direct connections time (45 minutes)
  - Maximum break connections time (75 minutes)
  - Minimum extreme connections time (405 minutes)

Cases

Table 3: Set of parameters for the case studies.

Parameter	Case 1	Case 2
ξ	45	60
η	75	90
ζ	405	360
ϕ	0.20	0.20
θ	0.15	0.15
ψ	0.30	0.30

- The model is solved by three different methods:
    - Branch and Bound
    - Heuristics
    - Fix and Relax

Clustering Impact

Table 4: Model size and linear relaxation.

	Case 1		Case 2	
Item	Unclustered	Clustered	Unclustered	Clustered
columns	707,226	261,957	1,125,726	406,342
rows	100,102	50,796	100,102	50,796
non-zeros	3,092,488	1,056,035	5,277,594	1,784,587
L.R. of	82.997	83.448	81.150	81.179

- Clustering:
  - Great size reductions.
  - High fidelity of linear relaxation.

Fix-And-Relax Algorithm Performance

Table 5: Solution approach comparison for case study 1.

Method	Solver	I.S.	L.B.	O.G.(%)	R.G.(%)	T.
B&B	C	85.7678	83.4479	2.70	3.23	43,200
	G	153.7762	83.4577	45.73	46.03	43,200
Heur	C	86.9474	83.4478	4.02	4.54	43,200
	G	106.8227	83.4553	21.87	22.30	43,200
FRA	C	87.0756	83.4476	4.17	4.68	2,348
	G	84.7095	83.4476	1.49	2.02	7,284

Fix-And-Relax Algorithm Performance

Table 6: Solution approach comparison for case study 2.

Method	Solver	I.S.	L.B.	O.G.(%)	R.G.(%)	T.
B&B	C	81.5948	81.1866	0.50	0.54	43,200
	G	163.4840	81.1795	50.34	50.36	43,200
Heur	C	83.0953	81.1795	2.31	2.34	43,200
	G	115.4085	81.1795	29.66	29.68	43,200
FRA	C	81.6159	81.1795	0.53	0.57	2,563
	G	81.5499	81.1795	0.45	0.49	21,176

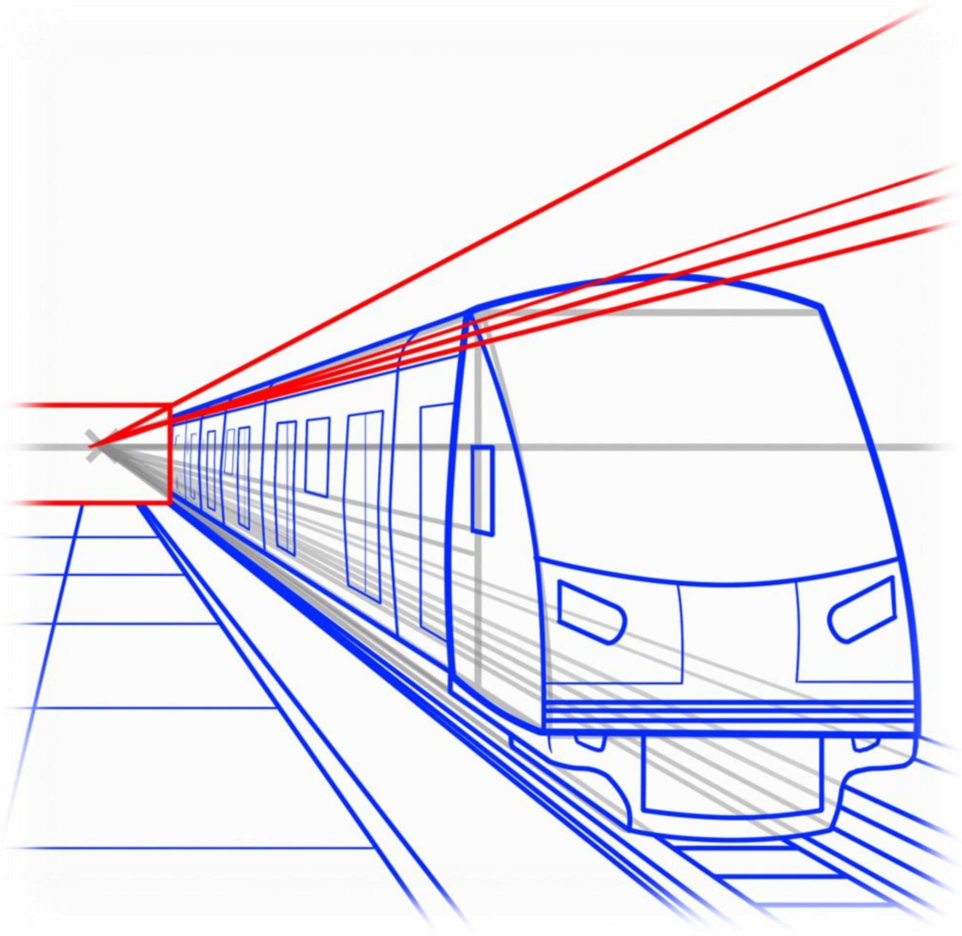
Impact of Robustness

Table 7: Influence of considering robustness in case 1.

	I.S.	L.R. of	R.G.	D.	$\mu(\%)$
Rob.	84.7095	82.9974	2.02	73	66.48
Non-rob.	81.3144	80.0400	1.57	71	68.35

Outline

1. Overview
2. Mathematical model
3. Solution approach
4. Case study
5. Conclusions



Conclusions

- Solutions of high quality can be found with this model.
- The formulation is friendly with previous planning stages.
- It can be applied to Metro, Bus, etc.
- With standard methods, it takes 12 hours or more to obtain best solutions.
- With the ad-hoc Fix & Relax matheuristic, best solutions can be found ~17x faster (in less than 1 hour).

THANK YOU

ANY QUESTIONS?

Manuel Fuentes
Manuel.fuentes@urjc.es