Railway Traffic Management

The Meet & Pass Problem

Pedro A. Afonso & Carlos F. Bispo
Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisbon, Portugal

Outline

- Introduction
- Overview
- The approach
- Results
- Conclusions

Introduction

- Railway companies aim to achieve regular and reliable train services.
- Daily schedules are produced offline to meet such objectives.
- During execution, events may disturb the original schedule.
- Such disturbances are more dramatic in the context of single track lines for outbound and inbound trains.

Introduction

- For such lines, the original schedule accounts for Meet and Pass points at sidings or stations.
  - Two trains ***meet*** while traveling in opposite directions;
  - A faster train needs to ***pass*** a slower train ahead.
- A disturbance during the schedule execution may compromise one or many such points.
- Whenever that happens, an alternative has to be produced in real time.

Introduction

- A conflict is said to occur whenever two trains are bound to share the same track segment.
  - By definition of a track segment, at most one train may travel it;
  - Meet conflict – If two trains approach each other on a single track segment, traveling in opposite directions;
  - Pass conflict – If a faster train catches a slower train traveling in the same direction on the same track segment.

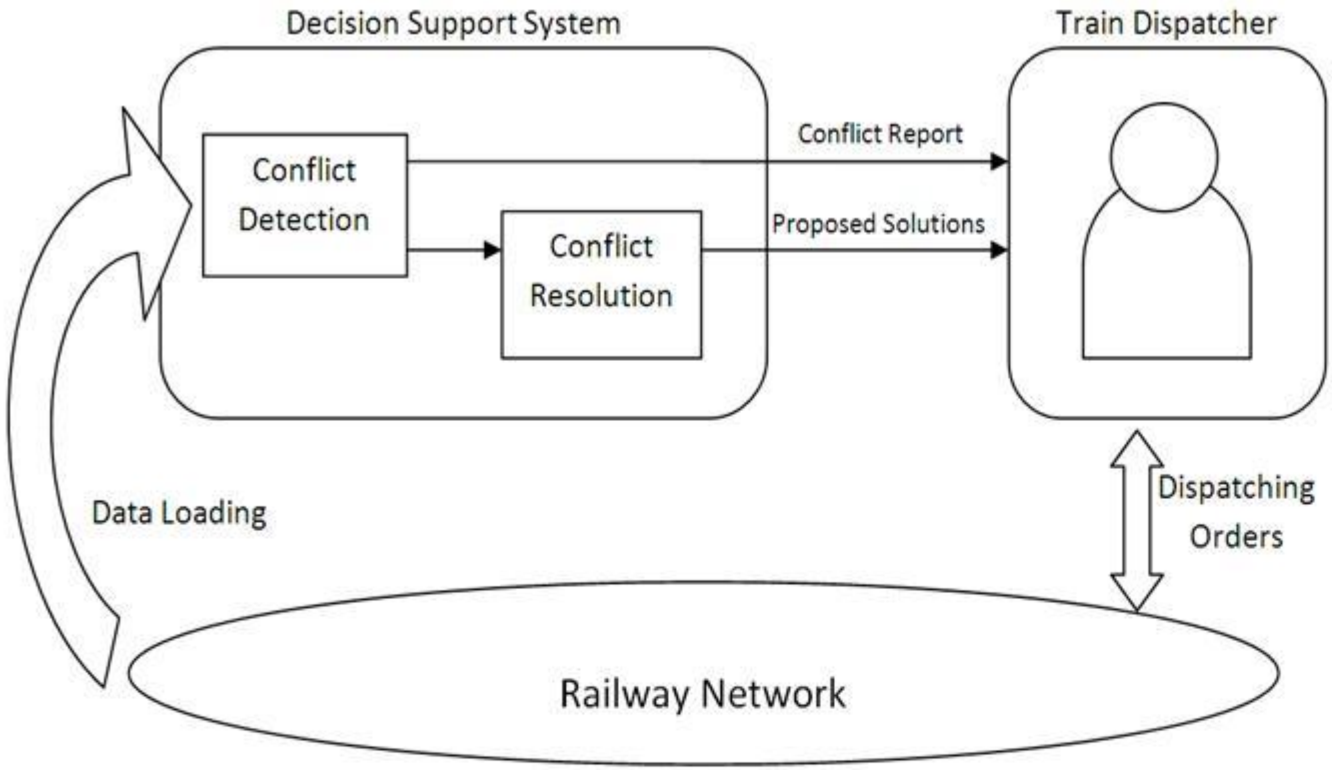
Introduction

- Typpically, conflicts are solved by human operators.
- Decisions have to be produced in a timely manner, instructing each of the conflicting trains with what to do.
- A given conflict resolution may induce future conflicts.
- Train priorities is usually the criterion to decide on each conflict.

Introduction

- The quality of these decisions may not be the best.
  - Operators rely on experience;
  - Operators have no decision support system;
  - Operators may only forecast the impact of their decisions on a relatively small time frame with very simple graphic applications.
- Therefore, there is an opportunity to develop a Decision Support System.

Overview



Decision Support System
Train Dispatcher
Conflict Report
Conflict Detection
Proposed Solutions
Conflict Resolution
Data Loading
Dispatching Orders
Railway Network

The approach

- The focus of this presentation will be on the "Conflict Resolution" box.
  - Formulating a mathematical model;
  - Charaterizing the conflict detection problem;
  - Identifying conflicts and their solutions;
  - Implementation

The approach – Mathematical formulation

- Objective function

$$\min Z = \sum_{i=1}^{n} w_i \ \max\{0, (a_i^m - \alpha_i^m)\}$$

- Subject to:
  - Free running time constraints

$$r_i^k \geq \tau_i^k, \forall i \in I, k = 1, 2, \ldots, m-1$$

  - Consecutive departure and arrival constraints

$$f_i^k \geq s_i^k + \tau_i^k, \forall i \in I, k = 1, 2, \ldots, m-1$$

The approach – Mathematical formulation

- Subject to (continued):
  - Minumum dwell time constraints

$$s_i^u \geq \omega_i^u, \forall i \in I, u = 1, 2, \dots, m$$

  - Headway constraints on arrival times to stations

$$a_i^u \geq a_{i'}^u + g_u \oplus a_{i'}^u \geq a_i^u + g_u, \forall i, i' \in I, i \neq i', u \in U$$

  - Meet condition

$$d_i^{u+1} \geq a_{i'}^{u+1} + g_u \oplus d_{i'}^u \geq a_i^u + g_u, \forall u \in U, i \in I_i, i' \in I_o$$

The approach – Mathematical formulation

- Subject to (continued):
  - Pass condition

$$\left(d_i^u \le d_{i'}^u + h^k \wedge a_i^{u+1} \le a_{i'}^{u+1} + h^k\right)$$

$$\oplus$$

$$\left(d_{i'}^u \le d_i^u + h^k \wedge a_{i'}^{u+1} \le a_i^{u+1} + h^k\right),$$

$$\forall u \in U, \{i, i'\} \in I_o$$

  - Meetpoint capacity limits*

$$S^u \le C^u, \forall u \in U$$

The approach – Conflict detection

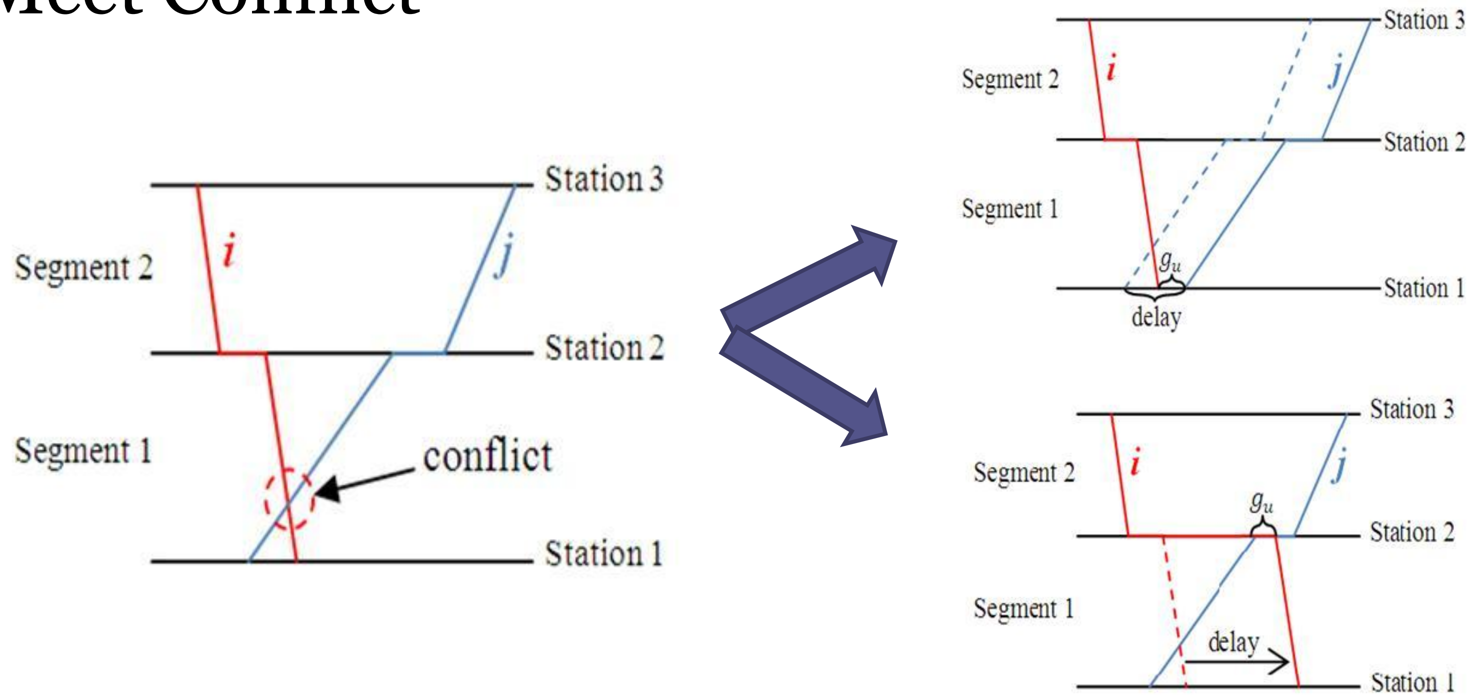
- Theorem 1
  - If train i does not collide with train i+1 and train i+1 does not collide with train i+2, then train i cannot collide with train i+2.
- Theorem 2
  - If there is a conflict between train i and train p, with $p \geq i+2$, there there is also a conflict between trains i and p-1 or between trains p-1 and p.

The approach – Conflict detection

- Corollary
  - In order to conclude about the existence, or non-existence of conflicts, in a given track segment, it is only necessary to check for conflicts between consecutive trains, in terms of their entering order.
- Concluding:
  - The conflict detection is linear, instead of quadratic, in the number of trains.

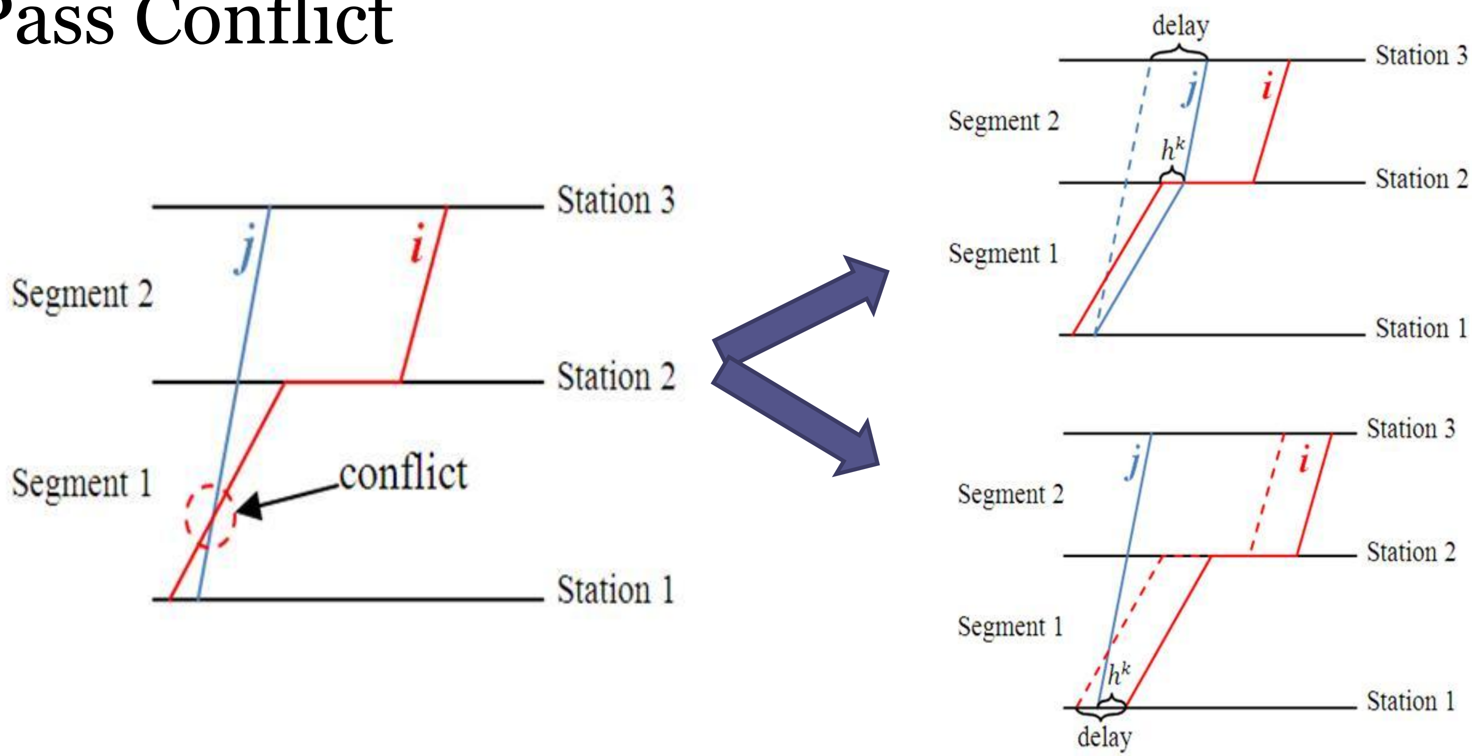
The approach – Conflict resolution

- Meet Conflict



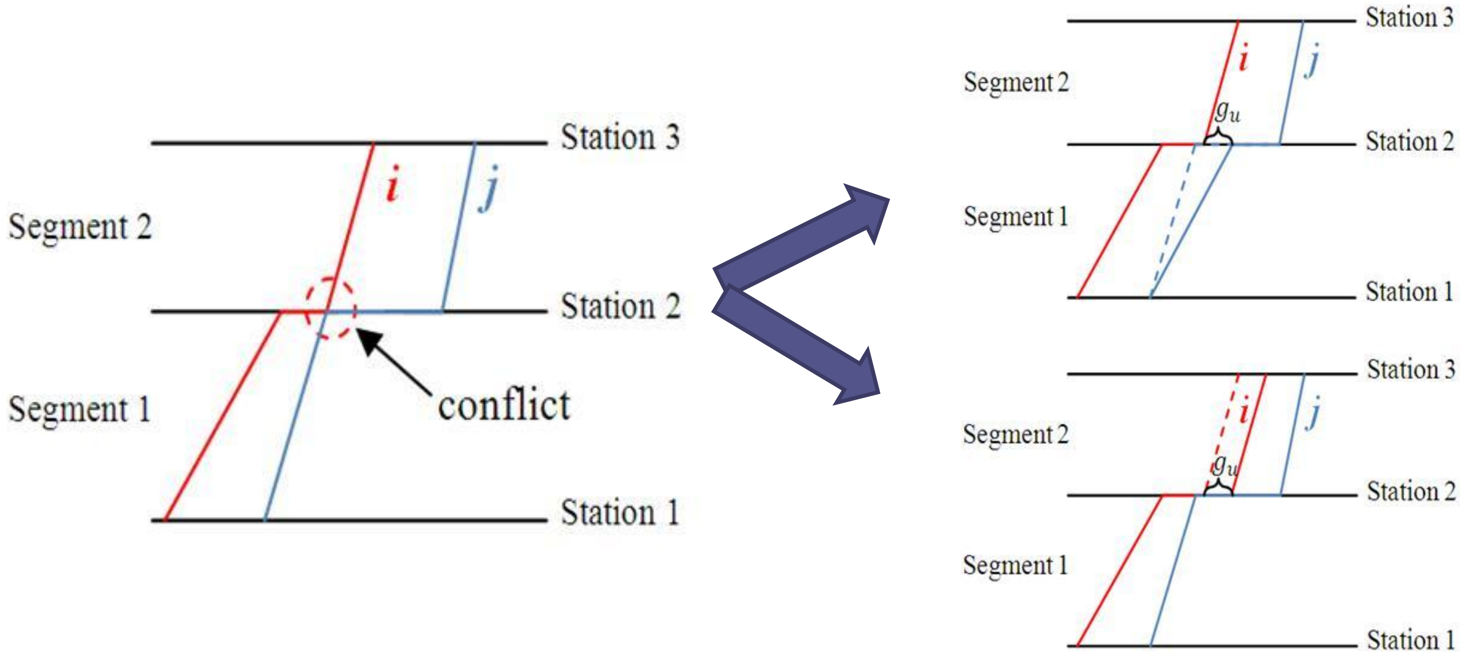
The approach – Conflict resolution

- Pass Conflict



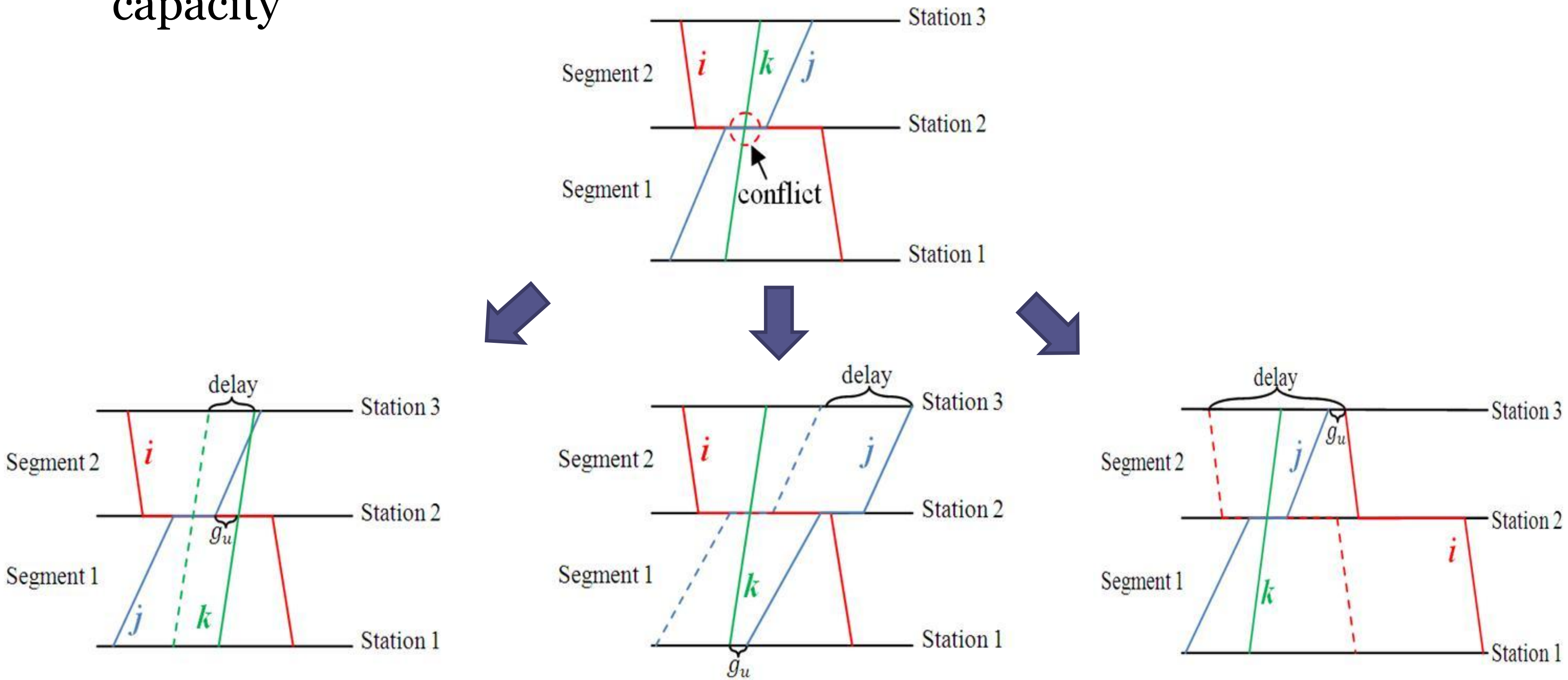
The approach – Conflict resolution

- Safety intervals at stations Conflict

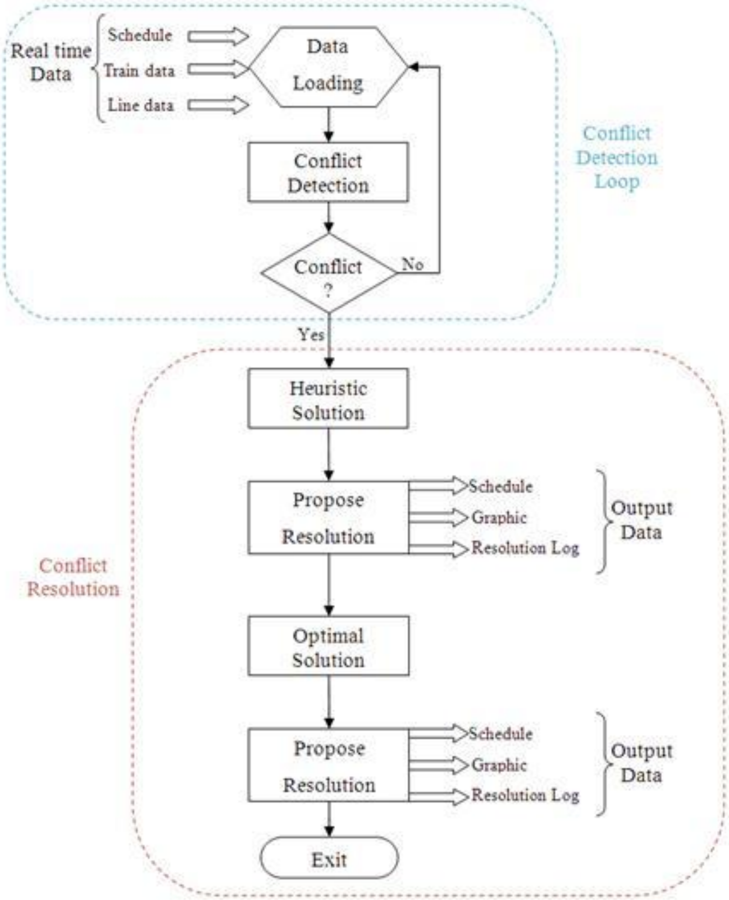


The approach – Conflict resolution

- **Capacity Conflict** – example for 3 trains with a 2 train capacity



The approach – Implementation

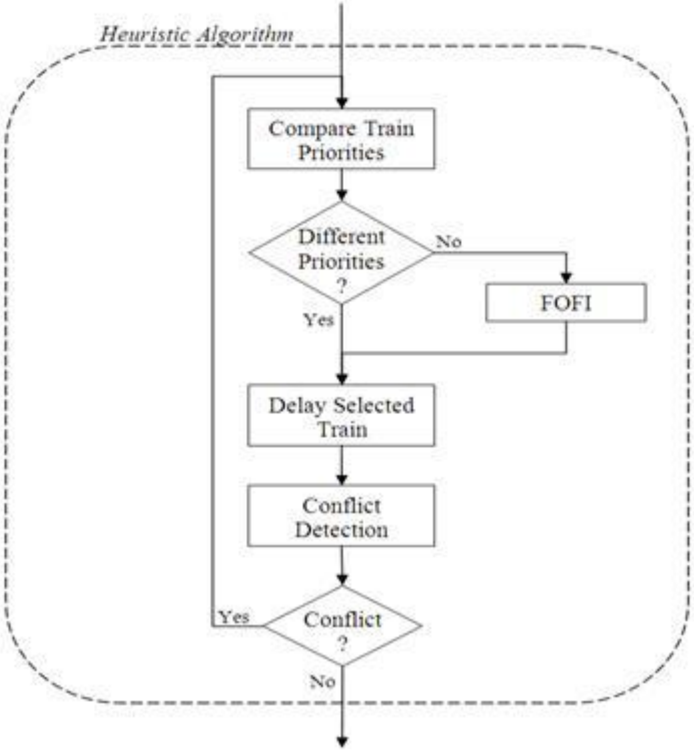


The approach – Implementation

Heuristic Solution

- Applies train priorities, as a human controller would do, until it produces a conflict free schedule.
- When trains have the same priority, uses the FOFI dispatching rule.
  - First Out First Serve;
  - Also known as First Leave First Serve.

Flowchart

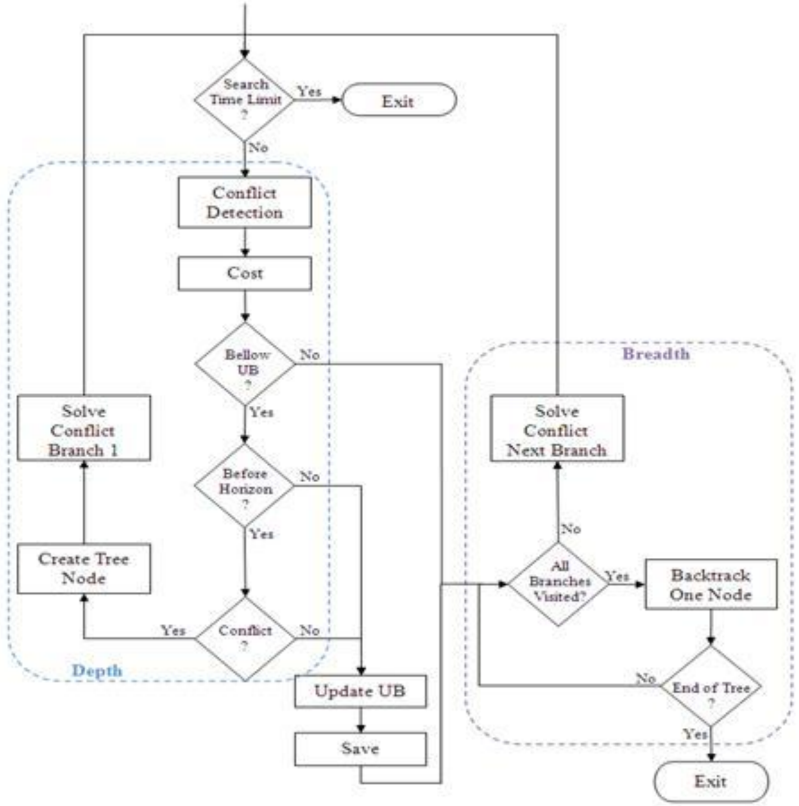


The approach – Implementation

Search-based solution

- Performs a Depth First Search.
  - Produces complete conflict free schedules fast.
- Improves over them, using best known schedule to bound the search.
  - Branch-and-bound.
- Has a computation time budget, after which it returns the best solution found.
- May produce optimal schedules.

Flowchart



The approach – Implementation

- The first time a conflict is detected, priority is given to produce one first conflict free schedule.
  - This is not much different from what human operators do.
- After that, knowing how much time is has until the first decision has to be enforced, uses that time to improve over the first solution.
  - When time expires, either it produces a better solution or the first is executed.

Results

- The final package possesses a series of parameters, which affect its performance.
  - Initial schedule
    - offline schedule for the day;
  - Time horizon
    - no conflicts up to the horizon;
  - Number of solutions
    - solutions presented to the human operator as alternatives;
  - Cost function
    - metric used to evaluate the effects of delays for trains;
  - Maximum search time
    - computational budget;
  - Upper bound
    - heuristic schedule

Results

Initial Schedule

- Evaluating performance as a function of the initial schedule's complexity and number of conflicts.

Input Schedules	Trains	Meetpoints	Initial Conflicts	CPU Time (s)		Weighted Tardiness		Limits	
				Heuristic Solution	Optimal Solution	Heuristic Solution	Optimal Solution	Time Horizon	Upper Bound
1	6	3	8	2,63	2,58	37,35	23,80	79	429
2	12	6	11	2,52	3,97	101,55	44,30	10	353
3	12	24	22	3,01	944,89	46,54	43,72	62	141694
4	20	24	38	2,73	1800*	52,91	52,14	330	250591
5	40	24	240	6,09	1800*	321,9	382,27	669	205350

Number of solutions

- Number of solutions provided to the human dispatcher as alternative solutions for the same conflict.

Input Schedules	Number of Solutions	CPU Time (s) Optimal Solution
3	1	27,88
	5	30,78
	10	32,79
4	1	29,67
	5	33,39
	10	35,16

Results

Time horizon

- How far in time does the search provide a conflict free schedule.
- How does that affect performan-ce.

Input Schedules	Time Horizon (h)	CPU Time (s)		Weighted Tardiness	
		Heuristic Solution	Optimal Solution	Heuristic Solution	Optimal Solution
3	2	2,62	2,44	1,754	1,754
	5	2,81	2,74	9,822	9,822
	10	3,72	27,88	25,30	24,61
	24	3,01	944,89	46,54	43,72
4	2	2,90	2,42	1,762	1,762
	5	2,63	2,73	11,734	11,734
	10	3,09	29,67	29,53	26,72
	24	2,73	1800*	52,91	52,14
5	2	3,17	2,9	0,438	0,438
	5	2,93	1800*	70,42	64,31
	10	3,53	1800*	152,64	186,80
	24	6,09	1800*	321,90	382,27

Results

Cost function

- Sets of weights for the weighted tardiness function

	Priority 1	Priority 2	Priority 3
Set 1	0.7	0.2	0.1
Set 2	0.6	0.3	0.1
Set 3	0.5	0.4	0.1
Set 4	0.5	0.3	0.2

Comparison

Input Schedules	Cost Function	Sets of Weights	Solution Cost	
			Optimal	Heuristic
4	Weighted Tardiness	Set 1	23,97	24,66
		Set 2	29,97	31,27
		Set 3	31,76	37,88
		Set 4	27,23	30,00
	Total tardiness	–	69,47	94,52
5	Weighted Tardiness	Set 1	182,07	169,92
		Set 2	170,94	174,50
		Set 3	158,93	179,14
		Set 4	161,52	209,23
	Total tardiness	–	433,75	771,02

Results

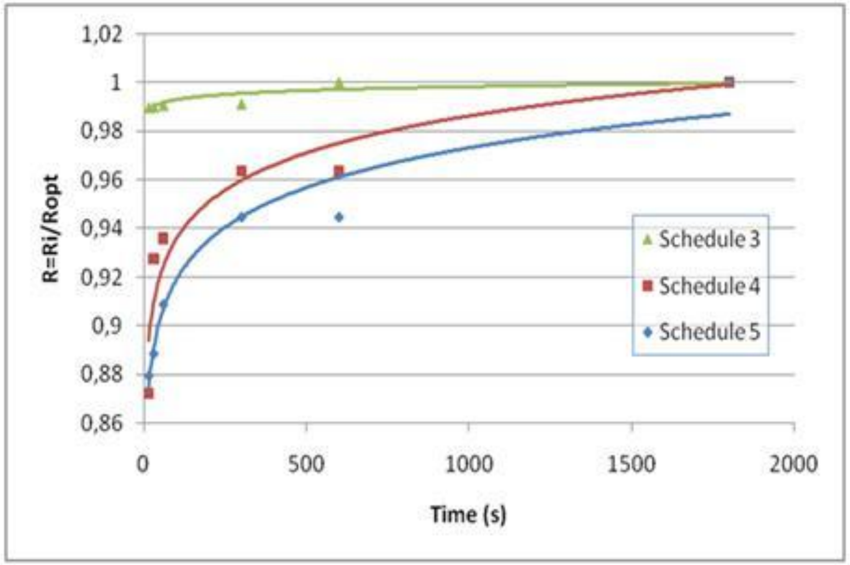
Maximum search time

- Performance achieved with progressively larger computational budgets.

Input Schedules	Maximum Search Time (s)					
	15	30	60	300	600	1800
3	49,73	49,22	48,12	46,29	46,29	43,72*
4	59,79	56,22	55,72	54,12	54,12	52,14
5	386,22	386,12	385,84	385,62	382,27	382,27

Maximum search time

- Ratio* to optimal solution



Results

Initial upper bound

- Effect of using the heuristic solution as starting upper bound for the search.

Input Schedules	Initial Upper Bound	Maximum Search Time (s)					
		15	30	60	300	600	1800
3	Infinite	49,73	49,22	48,12	46,29	46,29	43,72 (944s)
	Heuristic Solution = 46,54	46,54	46,54	46,54	46,54	46,54	43,72 (903s)
4	Infinite	59,79	56,22	55,72	54,12	54,12	52,14
	Heuristic Solution = 52,91	52,91	52,91	52,91	52,91	52,91	51,95

Conclusions

- Presented a Decision Support System for Railway Traffic Management.
  - Combines what human operators do with a complementary search engine;
  - Provides more than a solution to be chosen;
  - Takes advantage of the time to the next conflict to improve over a first heuristic solution;
  - Always produces a solution fast.

Conclusions

- Caracterized the computational complexity for conflict detection.
- Conflicts addressed
  - The meet conflict;
  - The pass conflict;
  - Safety intervals conflict;
  - Capacity conflict;
- Presented a series of numerical results to evaluate main features.

Conclusions

- Future work
  - Need to address networks of lines, instead of a single line;
  - Move from single track to multiple tracks;
    - There are no meet points in multiple tracks.
    - But there may be connecting trains that need to be synchronized at given stations

Thank You

谢谢您