Delay-Robust Event-Scheduling
In Memoriam of A. Caprara

A. Caprara[1] L. Galli[2] S. Stiller[3] P. Toth[1]

[1]University of Bologna

[2]University of Pisa

[3]Technische Universität Berlin

17th Combinatorial Optimization Workshop
January 7-11, 2013
Aussois, France

Outline

Outline

Outline

Outline

Outline

Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.

Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.

Pros

- no knowledge of the underlying distribuition is required

Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.

Pros

- no knowledge of the underlying distribuition is required

Cons

Strict robustness is generally overconservative, because:

- solutions must cope with every likely scenarios without any recovery

Recoverable Robustness

Informally speaking, a solution to an optimization problem is called *recovery robust* if it can be adjusted to all likely scenarios by limited recovery action. Thus a recovery-robust solution provides a service guarantee (Liebchen *et al.* 2009 [3], Stiller 2008 [4]).

Robust Network Buffering

We are interested in the special case in which the recovery problem is a delay (d_e) propagation in some directed graph $N = (\mathscr{E}, A)$, which is buffered on the arcs against disturbances on the nodes $e \in \mathscr{E}$.

Details in Liebchen *et al.* 2009 [3], Stiller 2008 [4].

A framework

- A set $\mathscr{E}$ of *events* to be scheduled.
- Denoting by F the set of *feasible* schedulings of events $\mathscr{E}$, i.e., the feasible region, a *nominal problem* of the form $\min\{c(x) : x \in F\}$.
- A set $\mathscr{S}$ of possible *scenarios*, where each scenario $s \in \mathscr{S}$ is defined by the external *disturbance* $\delta_e^s \geq 0$ assigned to each event $e \in \mathscr{E}$.
- A *delay-propagation network* $N = (\mathscr{E}, A)$, with $(e', e) \in A$ if a delay $d_{e'}$ on event e' may propagate to a delay d_e on event e.
- A delay $d_{e'}$ on event e' may propagate to a delay d_e on event e according to the relation $d_e \geq d_{e'} - b((e', e), x)$, where $b((e', e), x)$ is a *buffer time* function.

Mathematical model

The delay-robust problem can be formulated as:

$$\min c(x) + D, \tag{1}$$

subject to

$$x \in F, \tag{2}$$

$$D \geq \sum_{e \in \mathscr{E}} d_e^s, \qquad s \in \mathscr{S}, \tag{3}$$

$$d_e^s \geq \delta_e^s, \qquad e \in \mathscr{E},\ s \in \mathscr{S}, \tag{4}$$

$$d_e^s \geq d_{e'}^s - f_{(e',e)}, \qquad (e',e) \in A,\ s \in \mathscr{S}, \tag{5}$$

$$f_{(e',e)} = b((e',e),x), \qquad (e',e) \in A. \tag{6}$$

Scenario set

$$\{\delta^s \in \mathbb{R}_+^{|\mathscr{E}|} : \|\delta^s\|_1 \leq \Delta\} \tag{7}$$

These are *uncountably* many scenarios, which can however be handled easily thanks to the following observation.

Proposition

For the delay-robust problem (1)-(6)*, the compact scenario set defined by* (7) *is equivalent to the finite scenario set S defined by the vectors*

$$\{\delta^s \in \{0, \Delta\}^{|\mathscr{E}|} : \|\delta^s\|_1 = \Delta\},$$

which contains only $|\mathscr{E}|$ scenarios.

Quadratic buffer time function

In the most natural case, the buffer time $b((e', e), x)$ only depends on the way in which events e' and e are scheduled in x, and is independent of the scheduling of the other events:

$$f_{(e',e)} = \sum_{h' \in \mathscr{H}_{e'}} \sum_{h \in \mathscr{H}_e} \varphi(e', h', e, h)\; x_{e',h'}\; x_{e,h}, \qquad (e', e) \in A.$$

These quadratic constraints can be linearised in a few ways.

Projecting out delay variables

One may directly optimise over the feasible region obtained by projecting out the d_e^s variables (and removing the constraints (3)-(5)).

Let $\sigma_{\overline{f}}(s,\bar{e})$ be the value of the shortest path from s to $\bar{e} \in \mathscr{E}$ on the delay propagation network $N = (\mathscr{E}, A)$ with arc lengths $\overline{f}_{(e',e)}$ for $(e', e) \in A$. Moreover, let $P_{\overline{f}}(s,\bar{e}) \subseteq A$ denote such a shortest path.

Lemma

Given buffer time values $\overline{f}_{(e',e)}$, an external disturbance equal to Δ on event $s \in \mathscr{E}$ (and null for the other events) propagates to a delay of value $\max\{0, \Delta - \sigma_{\overline{f}}(s,\bar{e})\}$ on event $\bar{e} \in \mathscr{E}$.

Projecting out delay variables (cont.)

A direct consequence is:

Lemma

Given buffer time values $\overline{f}_{(e',e)}$ *and maximum cumulative delay* $\overline{D}$*, for every scenario* $s \in \mathscr{S}$ *there exist values of the* d_e^s *variables satisfying* (3)-(5) *if and only if* $\sum_{e \in \mathscr{E}} \max\{0, \Delta - \sigma_{\overline{f}}(s,e)\} \leq \overline{D}$*. Otherwise, for the considered scenario* s*, letting* $\mathscr{E}_{\overline{f}}(s) \subseteq \mathscr{E}$ *be the set of events such that* $\Delta - \sigma_{\overline{f}}(s,e) > 0$*, a valid inequality that is violated by* $\overline{f}_{(e',e)}$ *and* $\overline{D}$ *is*

$$\sum_{e \in \mathscr{E}_{\overline{f}}(s)} \sum_{a \in P_{\overline{f}}(s,e)} f_a \geq \Delta |\mathscr{E}_{\overline{f}}(s)| - D.$$

Problem definition

The objective of train platforming is assigning trains to platforms in a railway station.

Problem definition

The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:

Problem definition

The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:

- for a specific railway station

Problem definition

The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:

- for a specific railway station
- after the timetable has been defined

In and Out

In and Out

Input

- Train schedule : arrival, departure times, directions and allowed shifts
- Railway station topology: platforms, paths and directions

In and Out

Input

- Train schedule : arrival, departure times, directions and allowed shifts
- Railway station topology: platforms, paths and directions

Output

- Assign each train a platform and two paths for arrival and departure s.t. no operational constraint is violated

Train schedule

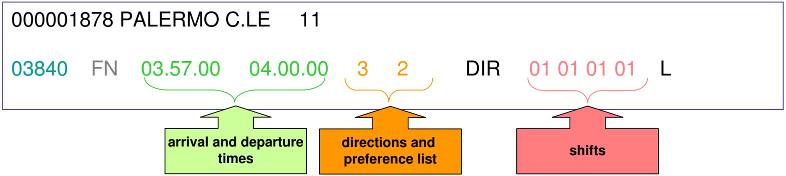

Figure: Train Schedule

The train schedule of a railway station contains info on arrival and departure times, directions and allowed shifts of each train passing through it.

Railway station topology

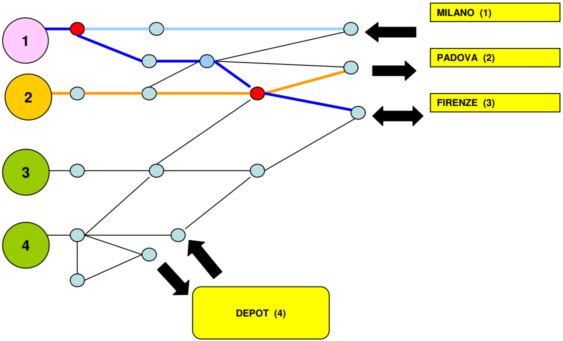

Figure: Topology

The topology of a railway station includes platforms, paths and directions.

Resources and operational constraints

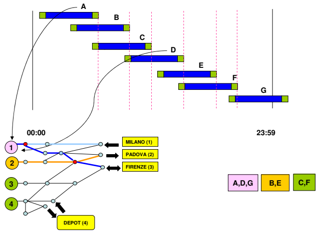

Figure: Platform and path.

Platform conflicts are forbidden, path conflicts are allowed to some extent.

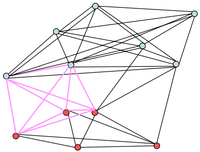

Figure: Incompatibility graph.

A pattern P for a train t is a 5-tuple defining: platform, arrival/departure paths and shifts. Operational constraints can be expressed using an incompatibility graph among patterns.

TPP deterministic model

$$\min \sum_{t \in T} \sum_{P \in \mathscr{P}_t} c_{t,P}\ x_{t,P}$$

s.t.

$$\sum_{P \in \mathscr{P}_t} x_{t,P} = 1, \qquad t \in T$$

$$\sum_{(t,P) \in K} x_{t,P} \leq 1, \qquad K \in \mathscr{K}$$

$$x_{t,P} \in \{0,1\}, \qquad t \in T,\ P \in \mathscr{P}_t$$

Details in Caprara *et al.* 2011 [1].

Delay propagation network

The platforming gives rise to a network in which the delay caused by disturbances propagates.

Delay propagation network

The platforming gives rise to a network in which the delay caused by disturbances propagates.

We consider two type of disturbances: (i) the train arrives late at the station, and (ii) the platform operations take longer than required.

Delay propagation network

The platforming gives rise to a network in which the delay caused by disturbances propagates.

We consider two type of disturbances: (i) the train arrives late at the station, and (ii) the platform operations take longer than required.

Hence, the following two events are associated with each train $t \in T$ and define $\mathscr{E}$:

- *arrival* a_t: the train occupies a given arrival path at a given instant
- *departure* p_t: the train frees its platform at a given instant

Example

Assume that in the nominal solution trains t' and t stop at the same platform, event $p_{t'}$ is scheduled at 10:00, and event a_t at 10:04 with 3 minutes to travel along the arrival path, so that t occupies the platform at 10:07. Moreover, assume that the headway time that must elapse between t' freeing the platform and t occupying it is 2 minutes and that the 3-minute travel time for t along its arrival path is fixed. If the departure path of t' and the arrival path of t are compatible, the buffer time $f_{(p_{t'},a_t)}$ is equal to 5 minutes, as a delay of up to 5 minutes on $p_{t'}$ does not affect a_t, whereas a larger one does.

Delay-robust mathematical model

$$\min \sum_{t \in T} \sum_{P \in \mathscr{P}_t} c_{t,P} \; x_{t,P} + D$$

subject to

$$\sum_{P \in \mathscr{P}_t} x_{t,P} = 1, \qquad t \in T,$$

$$\sum_{(t,P) \in K} x_{t,P} \leq 1, \qquad K \in \mathscr{K},$$

$$x_{t,P} \in \{0,1\}, \qquad t \in T, \; P \in \mathscr{P}_t,$$

$$f_{(e',e)} = \sum_{P_1 \in \mathscr{P}_{t_1}} \sum_{P_2 \in \mathscr{P}_{t_2}} c_{P_1,P_2,a} \; x_{t_1,P_1} \; x_{t_2,P_2}, \qquad (e',e) \in A,$$

$$D \geq \sum_{e \in \mathscr{E}} d_e^s, \qquad s \in \mathscr{S},$$

$$d_e^s \geq \delta_e^s, \qquad e \in \mathscr{E}, \; s \in \mathscr{S},$$

$$d_e^s \geq d_{e'}^s - f_{(e',e)}, \qquad (e',e) \in A, \; s \in \mathscr{S}.$$

Buffers linking constraints

A straightforward link between the buffer value of a given arc $a \in A(N)$ associated with train pair $(t_1, t_2) \in T^2$ and the choice of patterns for the given pair of trains is the following:

$$\sum_{P_1 \in \mathscr{P}_{t_1}} \sum_{P_2 \in \mathscr{P}_{t_2}} c_{P_1,P_2,a} \; x_{t_1,P_1} \; x_{t_2,P_2}$$

where c_{a,P_1,P_2} is a constant associated to arc a and to the corresponding choice of patterns (P_1, P_2) for trains (t_1, t_2).

Buffers linking constraints

$$f_a \leq \sum_{P_1 \in \mathscr{P}_{t_1}} \alpha^a_{P_1} x_{t_1,P_1} + \sum_{P_2 \in \mathscr{P}_{t_2}} \beta^a_{P_2} x_{t_2,P_2} - \gamma^a, \qquad a \in A(N),\ (\alpha,\beta,\gamma) \in \mathscr{F}_a \quad (8)$$

Following Caprara *et al.* 2011 [1], the separation of Constraints (8) is done by a sort of polyhedral brute force, given that, for each pair of trains t_1, t_2, and for each arc $a \in A(N)$ the number of vertices in $Q_{t_1,t_2,a}$ is small. Specifically, $Q_{t_1,t_2,a}$ has $|\mathscr{P}_{t_1}||\mathscr{P}_{t_2}|$ vertices and lies in $\mathbb{R}^{|\mathscr{P}_{t_1}|+|\mathscr{P}_{t_2}|+1}$, so we can separate over it by solving an LP with $|\mathscr{P}_{t_1}||\mathscr{P}_{t_2}|$ variables and $|\mathscr{P}_{t_1}|+|\mathscr{P}_{t_2}|+1$ constraints.

Computational results: Palermo C.Le.

	[1] Solution		Delay-Robust LP		Delay-Robust Solution			
instance	D	time	D	time	D	%gap	%reduction	time
PA I	1112	5	451.88	131	791	43%	29%	9788
PA II	694	5	252.00	49	584	57%	16%	3892
PA III	756	5	285.00	47	593	52%	22%	5411
PA IV	525	5	207.00	48	459	55%	13%	1417

Table: Results for the real-world instances of Rete Ferroviaria Italiana.

Computational results: Genova P.Principe

	[1] Solution		Delay-Robust LP		Delay-Robust Solution			
instance	D	time	D	time	D	%gap	%reduction	time
GE I	561	5	193.11	30	453	57%	19%	147
GE II	1252	5	388.00	201	810	52%	35%	30417
GE III	1073	5	297.95	118	865	66%	19%	1086
GE IV	758	5	231.00	54	498	54%	34%	274

Table: Results for the real-world instances of Rete Ferroviaria Italiana.

Computational results: Bari C.Le.

instance	[1] Solution		Delay-Robust LP		Delay-Robust Solution			
	D	time	D	time	D	%gap	%reduction	time
BA I	770	5	341.00	68	592	42%	23%	1095
BA II	959	5	508.08	118	959	47%	0%	5
BA III	711	5	293.91	115	651	55%	8%	525
BA IV	733	5	308.40	103	686	55%	6%	2431
BA V	786	5	245.26	165	734	67%	7%	3204
BA VI	819	5	312.00	51	819	62%	0%	5
BA VII	1025	5	315.18	77	743	58%	27%	4126
BA VIII	714	5	240.00	45	580	59%	19%	535

Table: Results for the real-world instances of Rete Ferroviaria Italiana.

Computational results: Milano C.Le.

	[1] Solution		Delay-Robust LP		Delay-Robust Solution			
instance	*D*	time	*D*	time	*D*	%gap	%reduction	time
MI I	480	5	354.00	105	476	26%	1%	201
MI II	785	5	449.28	330	686	35%	13%	2415
MI III	1102	6	382.88	334	822	53%	25%	938
MI IV	1067	5	721.00	839	951	24%	11%	2547
MI V	1002	8	290.76	288	565	49%	44%	2503
MI VI	898	5	415.13	655	816	49%	9%	618

Table: Results for the real-world instances of Rete Ferroviaria Italiana.

Questions

Thank you! 

References

- A. Caprara, L. Galli and P. Toth (2011). Solution of the Train Platforming Problem. *Tran. Sci.*, 45(2), 246–257.
- A. Caprara, L. Galli and P. Toth (2012). Delay-Robust Event-Scheduling *Submitted*.
- C. Liebchen, M. Lübbecke, R.H. Möhring, and S.Stiller (2009). The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications. In R.K. Ahuja, R.H. Möhring and C.D. Zaroliagis (eds.), *Robust and On-Line Large Scale Optimization*. Lecture Notes in Computer Science 5868, Springer-Verlag, 1–27.
- S. Stiller (2008). *Extending Concepts of Reliability* PhD thesis, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany.