Delay Management on a Path

Sven Jäger

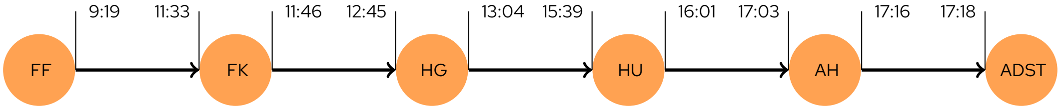
OR 2023
August 31, 2023, Hamburg

RPTU
OPT Optimization Research Group

Motivation

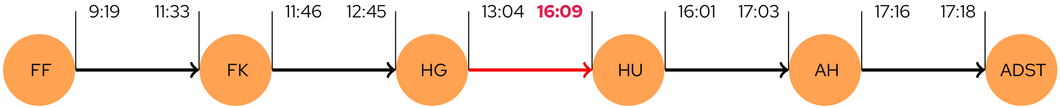
Frankfurt(Main)Hbf	Di, 29.08.23	ab 09:19 16	RE 30 (4154)	
Kassel Hbf	Di, 29.08.23	an 11:33 7		
Umsteigezeit 13 Min.				
Kassel Hbf	Di, 29.08.23	ab 11:46 11	RB 83 (24014)	
Göttingen	Di, 29.08.23	an 12:45 5		
Umsteigezeit 19 Min.				
Göttingen	Di, 29.08.23	ab 13:04 6	ME RE2 (82828)	
Uelzen	Di, 29.08.23	an 15:39 103		
Umsteigezeit 22 Min.				
Uelzen	Di, 29.08.23	ab 16:01 103	ME RE3 (82128)	
Hamburg Hbf	Di, 29.08.23	an 17:03 13A-C		
Umsteigezeit 13 Min.				
Hamburg Hbf (S-Bahn)	Di, 29.08.23	ab 17:16 1	S 3	
Hamburg Dammtor	Di, 29.08.23	an 17:18 1		

The Problem



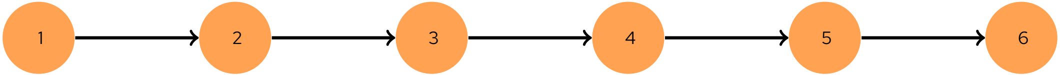
RPTU

The Problem



9:19
11:33
11:46
12:45
13:04
16:09
16:01
17:03
17:16
17:18
FF
FK
HG
HU
AH
ADST

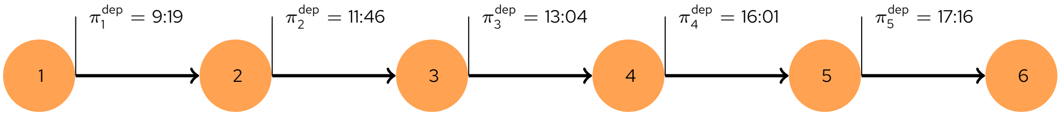
The Problem



Given: › Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$

RPTU

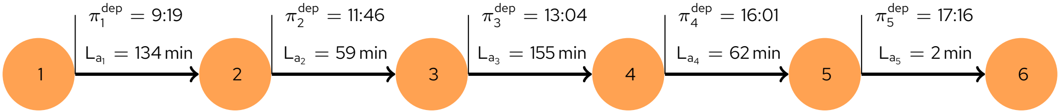
The Problem



Given:

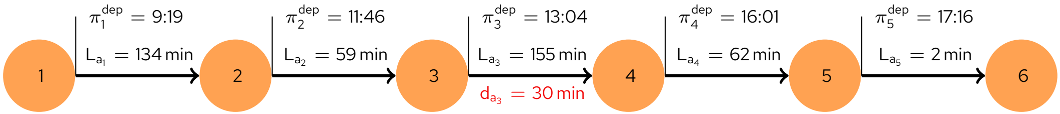
- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$

The Problem



Given:
- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$
- normal driving durations $L_a \geq 0$ of arcs $a \in A$,

The Problem

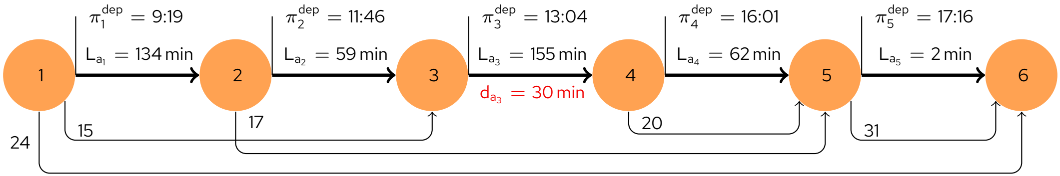


Given:

- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$
- normal driving durations $L_a \geq 0$ of arcs $a \in A$,
- source delays $d_a \geq 0$ on the arcs $a \in A$,

RPTU

The Problem

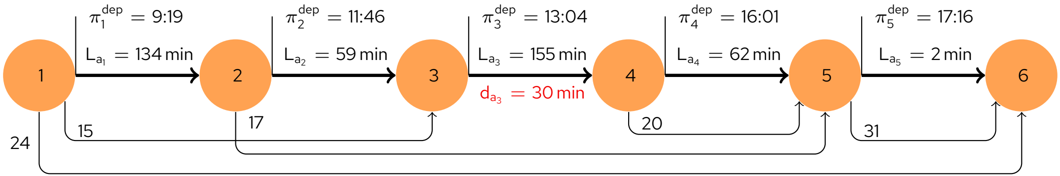


Given:

- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$
- normal driving durations $L_a \geq 0$ of arcs $a \in A$,
- source delays $d_a \geq 0$ on the arcs $a \in A$,
- numbers $w_{i,j} \geq 0$ of passengers traveling from v_i to v_j for $1 \leq i < j \leq m + 1$,

RPTU

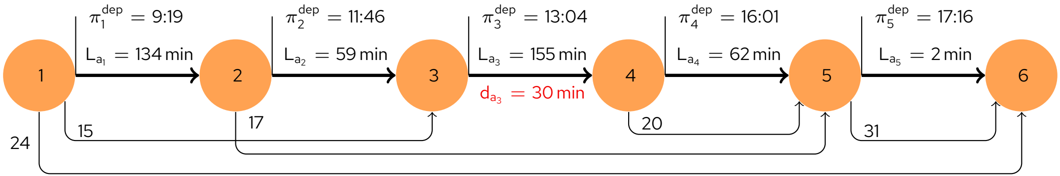
The Problem



Given:

- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$
- normal driving durations $L_a \geq 0$ of arcs $a \in A$,
- source delays $d_a \geq 0$ on the arcs $a \in A$,
- numbers $w_{i,j} \geq 0$ of passengers traveling from v_i to v_j for $1 \leq i < j \leq m + 1$,
- Delay T incurred by missing a connection (e.g. period length).

The Problem



Given:
- Directed path (V, A) on nodes $V = \{v_1, \ldots, v_{m+1}\}$ with arcs $a_i = (v_i, v_{i+1})$
- planned departure times π_i^{dep} for $i = 1, \ldots, m$
- normal driving durations $L_a \geq 0$ of arcs $a \in A$,
- source delays $d_a \geq 0$ on the arcs $a \in A$,
- numbers $w_{i,j} \geq 0$ of passengers traveling from v_i to v_j for $1 \leq i < j \leq m + 1$,
- Delay T incurred by missing a connection (e.g. period length).

Task: Find new departure times $x_i^{dep} \geq \pi_i^{dep}$ for $i = 1, \ldots, m$ such that the total delay of all passengers is minimized.

Assumptions

- For all $i \in \{1, \dots, m\}$ the arrival time at node v_{i+1} in the disposition timetable will be

$$x_{i+1}^{arr} = x_i^{dep} + L_{a_i} + d_{a_i}.$$

RPTU

Assumptions

- For all $i \in \{1, \ldots, m\}$ the arrival time at node v_{i+1} in the disposition timetable will be
$$x_{i+1}^{arr} = x_i^{dep} + L_{a_i} + d_{a_i}.$$
- If $x_i^{arr} \leq x_i^{dep}$, then the transfer at v_i is **maintained**. Otherwise, it is **missed**.

RPTU

Assumptions

- For all $i \in \{1, \ldots, m\}$ the arrival time at node v_{i+1} in the disposition timetable will be

$$x^{arr}_{i+1} = x^{dep}_i + L_{a_i} + d_{a_i}.$$

- If $x^{arr}_i \leq x^{dep}_i$, then the transfer at v_i is **maintained**. Otherwise, it is **missed**.
- For $1 \leq i < j \leq m + 1$ the origin-destination pair (i, j) is **maintained** if the transfers at all v_k, $k \in \{i + 1, \ldots, j - 1\}$, are maintained. Otherwise, it is **dropped**.

Assumptions

- For all $i \in \{1, \dots, m\}$ the arrival time at node v_{i+1} in the disposition timetable will be

$$x^{arr}_{i+1} = x^{dep}_{i} + L_{a_i} + d_{a_i}.$$

- If $x^{arr}_{i} \leq x^{dep}_{i}$, then the transfer at v_i is **maintained**. Otherwise, it is **missed**.
- For $1 \leq i < j \leq m+1$ the origin-destination pair (i, j) is **maintained** if the transfers at all v_k, $k \in \{i+1, \dots, j-1\}$, are maintained. Otherwise, it is **dropped**.

- The objective function is

$$\sum_{\substack{1 \leq i < j \leq m+1 \\ (i,j) \text{ maintained}}} w_{i,j} \cdot (x^{arr}_{j} - \pi^{dep}_{j-1} - L_{a_{j-1}}) + \sum_{\substack{1 \leq i < j \leq m+1 \\ (i,j) \text{ dropped}}} w_{i,j} \cdot T.$$

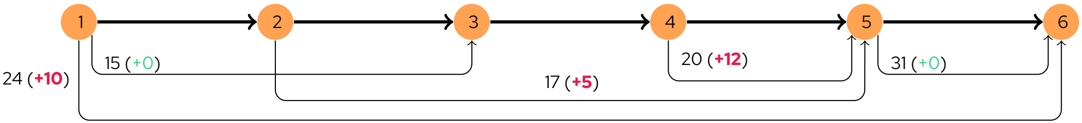
Related Work

Gatto et al., 2004

› Motivation: Passengers arrive by feeder trains that may be delayed.

RPTU

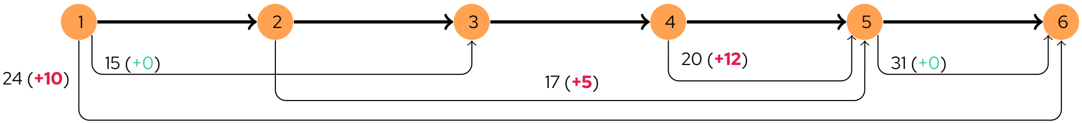
Related Work



Gatto et al., 2004

- Motivation: Passengers arrive by feeder trains that may be delayed.
- Each origin-destination pair has a given source delay. The planned trains do not get additional delays.

Related Work

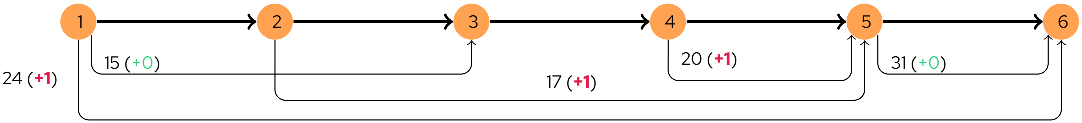


Gatto et al., 2004

- Motivation: Passengers arrive by feeder trains that may be delayed.
- Each origin-destination pair has a given source delay. The planned trains do not get additional delays.
- There are no slack times, i.e., if a train a_{i+1} waits for a train a_i, then it will get exactly the same delay.

RPTU

Related Work

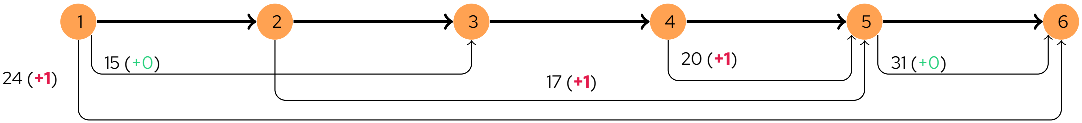


Gatto et al., 2004

- Motivation: Passengers arrive by feeder trains that may be delayed.
- Each origin-destination pair has a given source delay. The planned trains do not get additional delays.
- There are no slack times, i.e., if a train a_{i+1} waits for a train a_i, then it will get exactly the same delay.
- If all passengers have delay in $\{0, 1\}$, the problem can be solved in polynomial time.

RPTU

Related Work



Gatto et al., 2004

- Motivation: Passengers arrive by feeder trains that may be delayed.
- Each origin-destination pair has a given source delay. The planned trains do not get additional delays.
- There are no slack times, i.e., if a train a_{i+1} waits for a train a_i, then it will get exactly the same delay.
- If all passengers have delay in $\{0, 1\}$, the problem can be solved in polynomial time.

Gatto et al., 2005

- If there are slack times, the problem becomes NP-hard.

RPTU

Main Result

Theorem

The delay management problem on a line with delays occurring on the driving arcs can be solved in time $O(m^2)$, even if there are arbitrary delays and slack times at the transfer stations.

RPTU

Main Result

Theorem

The delay management problem on a line with delays occurring on the driving arcs can be solved in time $O(m^2)$, even if there are arbitrary delays and slack times at the transfer stations.

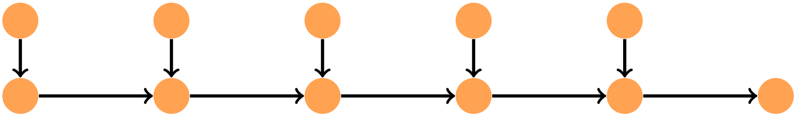
- The dynamic program can be generalized to out-trees.

Main Result

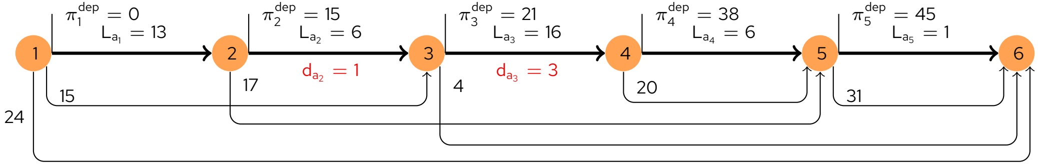
Theorem

The delay management problem on a line with delays occurring on the driving arcs can be solved in time $O(m^2)$, even if there are arbitrary delays and slack times at the transfer stations.

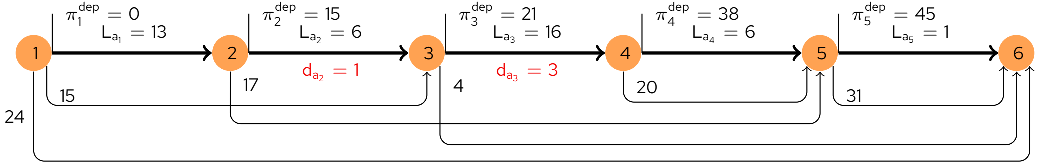
- The dynamic program can be generalized to out-trees.
- The hardness proof of Gatto et al. shows that the problem is hard for in-trees of the form



RPTU

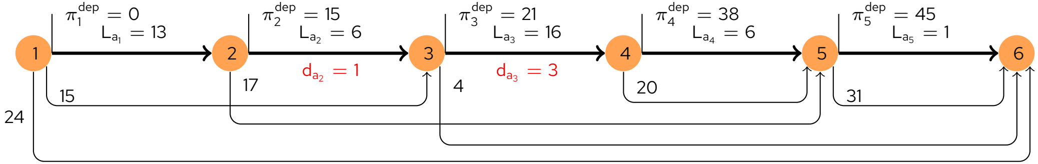


- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \dots, \ell-1$ are maintained.

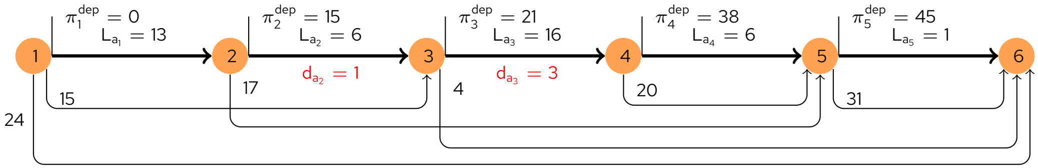


- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k + 1, \ldots, \ell - 1$ are maintained.
- Then $z[1, 1]$ is the optimal objective value.

RPTU



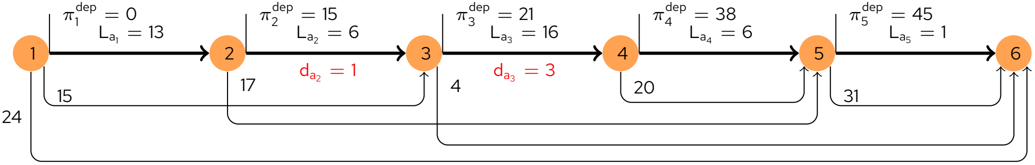
- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \ldots, \ell-1$ are maintained.
- Then $z[1, 1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1, \ldots, m$.



- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \ldots, \ell-1$ are maintained.
- Then $z[1, 1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1, \ldots, m$.

$$z[\ell, \ell] = \underbrace{w_{\ell,\ell+1} \cdot d_{a_\ell}}_{\text{dest. } \ell+1} + \underbrace{z[\ell, \ell+1]}_{\text{dest. } > \ell+1} \qquad \text{for } \ell = 1, \ldots, m$$

RPTU



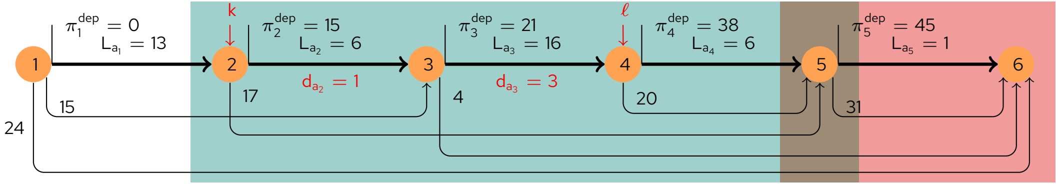
- For $1 \leq k \leq \ell \leq m$ let $z[k,\ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \ldots, \ell-1$ are maintained.
- Then $z[1,1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1, \ldots, m$.

$$z[\ell,\ell] = \underbrace{w_{\ell,\ell+1} \cdot d_{a_\ell}}_{\text{dest. } \ell+1} + \underbrace{z[\ell,\ell+1]}_{\text{dest. } >\ell+1} \qquad \text{for } \ell = 1, \ldots, m$$

$$z[k,\ell] = \min\left\{ \overbrace{\underbrace{\sum_{k \leq i < \ell < j} w_{i,j} \cdot T}_{\text{origin } <\ell} + \underbrace{z[\ell,\ell]}_{\text{origin } \geq \ell}}^{\text{don't wait at } \ell},\ \overbrace{\underbrace{\sum_{i=k}^{\ell} w_{i,\ell+1} \cdot D[k,\ldots,\ell]}_{\text{dest. } \ell+1} + \underbrace{z[k,\ell+1]}_{\text{dest. } >\ell+1}}^{\text{wait at } \ell} \right\} \qquad \text{for } 1 \leq k < \ell \leq m,$$

where $D[k,\ldots,\ell]$ is the delay of train ℓ if transfers are maintained at stations $k+1, \ldots, \ell$.

RPTU

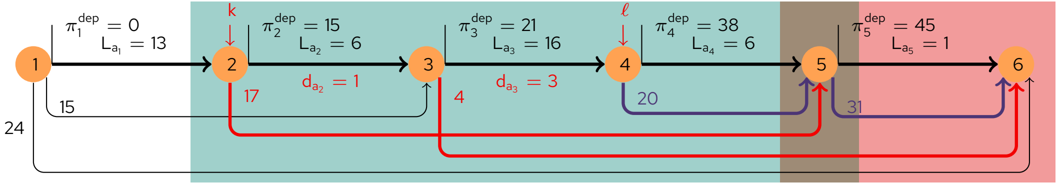


- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \ldots, \ell-1$ are maintained.
- Then $z[1, 1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1, \ldots, m$.

$$z[\ell, \ell] = \underbrace{w_{\ell,\ell+1} \cdot d_{a_\ell}}_{\text{dest. } \ell+1} + \underbrace{z[\ell, \ell+1]}_{\text{dest. } >\ell+1} \qquad \text{for } \ell = 1, \ldots, m$$

$$z[k, \ell] = \min \Bigg\{ \overbrace{\underbrace{\sum_{k \leq i < \ell < j} w_{i,j} \cdot T}_{\text{origin } <\ell} + \underbrace{z[\ell, \ell]}_{\text{origin } \geq \ell}}^{\text{don't wait at } \ell}, \overbrace{\underbrace{\sum_{i=k}^{\ell} w_{i,\ell+1} \cdot D[k, \ldots, \ell]}_{\text{dest. } \ell+1} + \underbrace{z[k, \ell+1]}_{\text{dest. } >\ell+1}}^{\text{wait at } \ell} \Bigg\} \qquad \text{for } 1 \leq k < \ell \leq m,$$

where $D[k, \ldots, \ell]$ is the delay of train ℓ if transfers are maintained at stations $k+1, \ldots, \ell$.



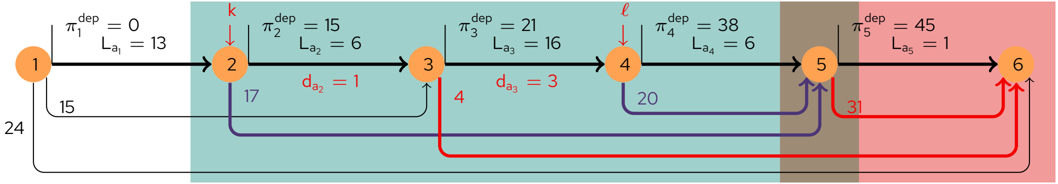
- For $1 \le k \le \ell \le m$ let $z[k,\ell]$ be the smallest possible total delay of passengers with origin $\ge k$ and destination $> \ell$ under the condition that the transfers at stations $k+1,\dots,\ell-1$ are maintained.
- Then $z[1,1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1,\dots,m$.

$$z[\ell,\ell] = \underbrace{w_{\ell,\ell+1} \cdot d_{a_\ell}}_{\text{dest. } \ell+1} + \underbrace{z[\ell,\ell+1]}_{\text{dest. } >\ell+1} \qquad \text{for } \ell = 1,\dots,m$$

$$z[k,\ell] = \min\left\{ \overbrace{\underbrace{\sum_{k \le i < \ell < j} w_{i,j} \cdot T}_{\text{origin } <\ell} + \underbrace{z[\ell,\ell]}_{\text{origin } \ge \ell}}^{\text{don't wait at } \ell},\ \overbrace{\underbrace{\sum_{i=k}^{\ell} w_{i,\ell+1} \cdot D[k,\dots,\ell]}_{\text{dest. } \ell+1} + \underbrace{z[k,\ell+1]}_{\text{dest. } >\ell+1}}^{\text{wait at } \ell} \right\} \qquad \text{for } 1 \le k < \ell \le m,$$

where $D[k,\dots,\ell]$ is the delay of train ℓ if transfers are maintained at stations $k+1,\dots,\ell$.

RPTU



- For $1 \leq k \leq \ell \leq m$ let $z[k, \ell]$ be the smallest possible total delay of passengers with origin $\geq k$ and destination $> \ell$ under the condition that the transfers at stations $k+1, \dots, \ell-1$ are maintained.
- Then $z[1, 1]$ is the optimal objective value.
- $z[k, m+1] = 0$ for $k = 1, \dots, m$.

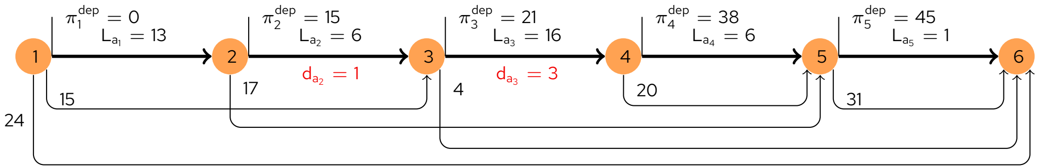
$$z[\ell, \ell] = \underbrace{w_{\ell,\ell+1} \cdot d_{a_\ell}}_{\text{dest. } \ell+1} + \underbrace{z[\ell, \ell+1]}_{\text{dest. } >\ell+1} \qquad \text{for } \ell = 1, \dots, m$$

$$z[k, \ell] = \min \left\{ \overbrace{\underbrace{\sum_{k \leq i < \ell < j} w_{i,j} \cdot T}_{\text{origin } <\ell} + \underbrace{z[\ell, \ell]}_{\text{origin } \geq \ell}}^{\text{don't wait at } \ell}, \overbrace{\underbrace{\sum_{i=k}^{\ell} w_{i,\ell+1} \cdot D[k, \dots, \ell]}_{\text{dest. } \ell+1} + \underbrace{z[k, \ell+1]}_{\text{dest. } >\ell+1}}^{\text{wait at } \ell} \right\} \qquad \text{for } 1 \leq k < \ell \leq m,$$

where $D[k, \dots, \ell]$ is the delay of train ℓ if transfers are maintained at stations $k+1, \dots, \ell$.

Example

$T = 6$

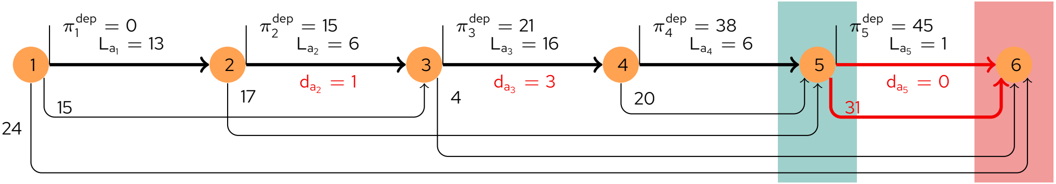


	1	2	3	4	5	6
1						0
2						0
3						0
4						0
5						0

RPTU

Example

$T = 6$

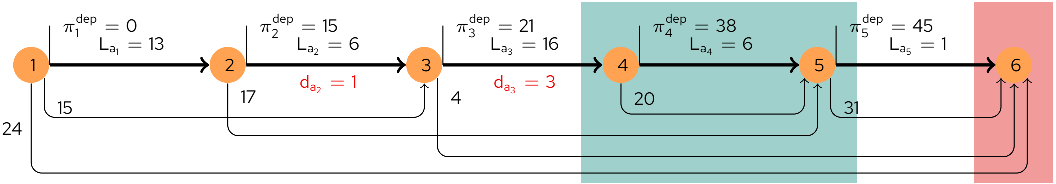


	1	2	3	4	5	6
1						0
2						0
3						0
4						0
5					**0**	0

RPTU

Example

$T = 6$

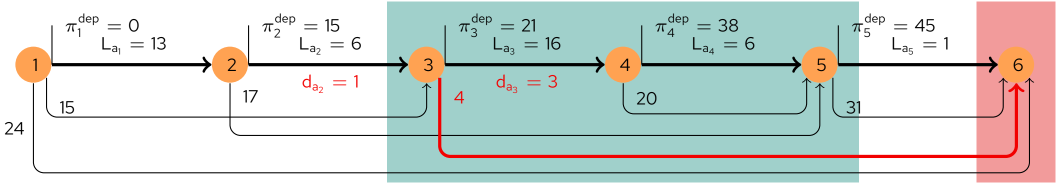


	1	2	3	4	5	6
1						0
2						0
3						0
4					**0**	0
5					0	0

RPTU

Example

$T = 6$



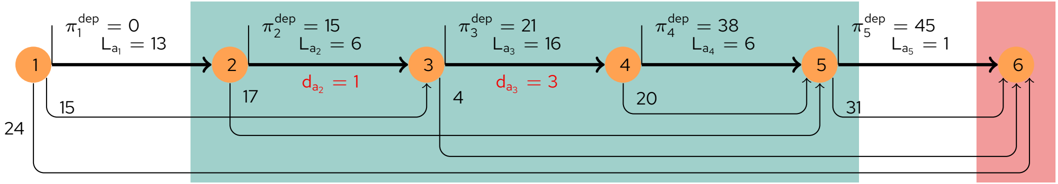
	1	2	3	4	5	6
1						0
2						0
3					**24**	0
4					0	0
5					0	0

- Don't wait at 5: $4 \cdot 6 + 0 = 24$
- Wait at 5: $35 \cdot 1 + 0 = 35$

RPTU

Example

$T = 6$



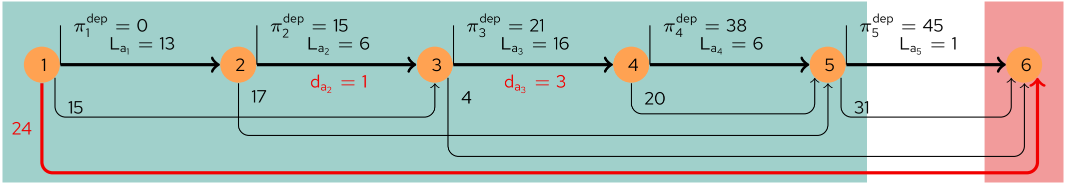
	1	2	3	4	5	6
1						0
2					**24**	0
3					24	0
4					0	0
5					0	0

- Don't wait at 5: $4 \cdot 6 + 0 = 24$
- Wait at 5: $35 \cdot 2 + 0 = 70$

RPTU

Example

$T = 6$

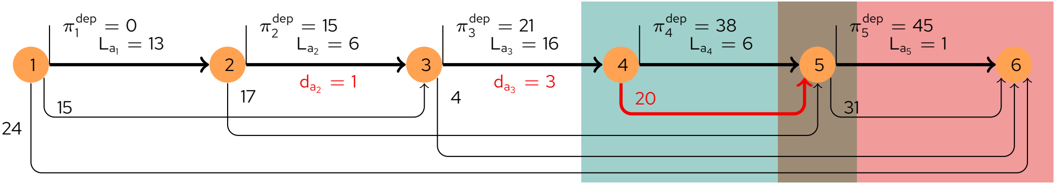


	1	2	3	4	5	6
1					**118**	0
2					24	0
3					24	0
4					0	0
5					0	0

- Don't wait at 5: $28 \cdot 6 + 0 = 168$
- Wait at 5: $59 \cdot 2 + 0 = 118$

Example

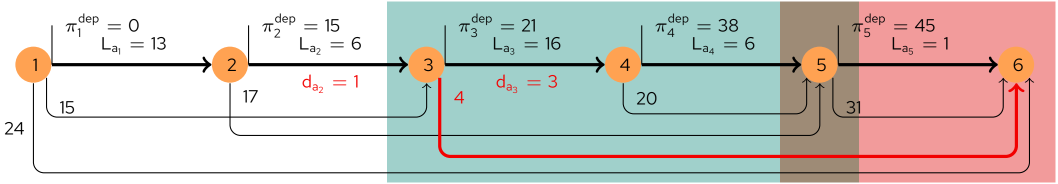
$T = 6$



	1	2	3	4	5	6
1					118	0
2					24	0
3					24	0
4				**0**	0	0
5					0	0

Example

$T = 6$



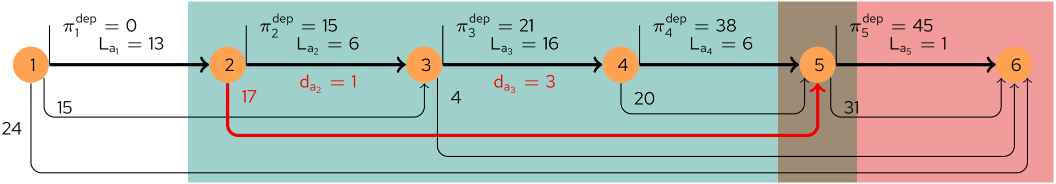
	1	2	3	4	5	6
1					118	0
2					24	0
3				**24**	24	0
4				0	0	0
5					0	0

- Don't wait at 4: $4 \cdot 6 + 0 = 24$
- Wait at 4: $20 \cdot 2 + 24 = 64$

RPTU

Example

$T = 6$



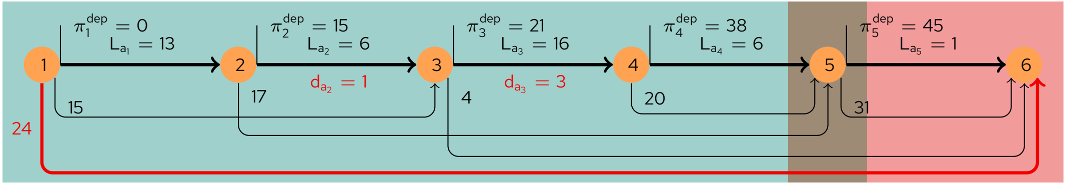
	1	2	3	4	5	6
1					118	0
2				**126**	24	0
3				24	24	0
4				0	0	0
5					0	0

- Don't wait at 4: $21 \cdot 6 + 0 = 126$
- Wait at 4: $37 \cdot 3 + 24 = 135$

RPTU

Example

$T = 6$



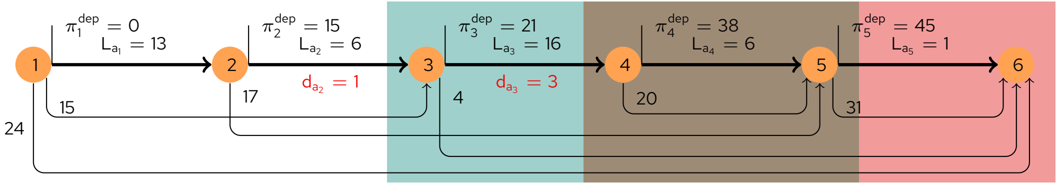
	1	2	3	4	5	6
1				**229**	118	0
2				126	24	0
3				24	24	0
4				0	0	0
5					0	0

- Don't wait at 4: $45 \cdot 6 + 0 = 270$
- Wait at 4: $37 \cdot 3 + 118 = 229$

RPTU

Example

$T = 6$

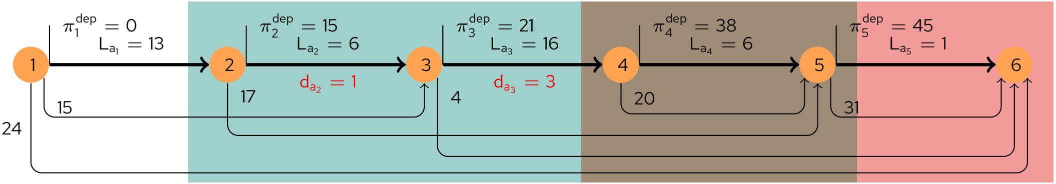


	1	2	3	4	5	6
1				229	118	0
2				126	24	0
3			**24**	24	24	0
4				0	0	0
5					0	0

RPTU

Example

$T = 6$

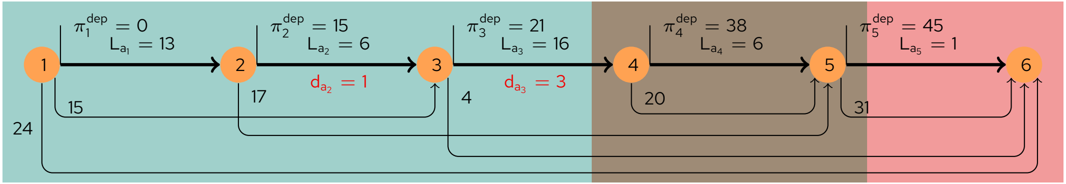


	1	2	3	4	5	6
1				229	118	0
2			**126**	126	24	0
3			24	24	24	0
4				0	0	0
5					0	0

- Don't wait at 3: $17 \cdot 6 + 24 = 126$
- Wait at 3: $0 \cdot 4 + 126 = 126$

Example

$T = 6$



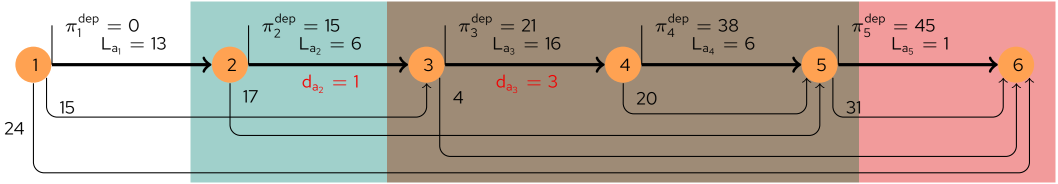
	1	2	3	4	5	6
1			**229**	229	118	0
2			126	126	24	0
3			24	24	24	0
4				0	0	0
5					0	0

- Don't wait at 3: $41 \cdot 6 + 24 = 270$
- Wait at 3: $0 \cdot 4 + 229 = 229$

RPTU

Example

$T = 6$

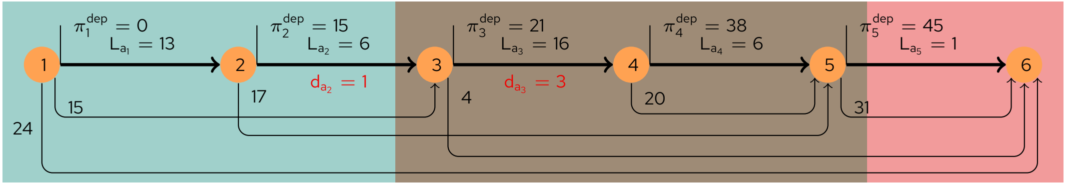


	1	2	3	4	5	6
1			229	229	118	0
2		**126**	126	126	24	0
3			24	24	24	0
4				0	0	0
5					0	0

RPTU

Example

$T = 6$



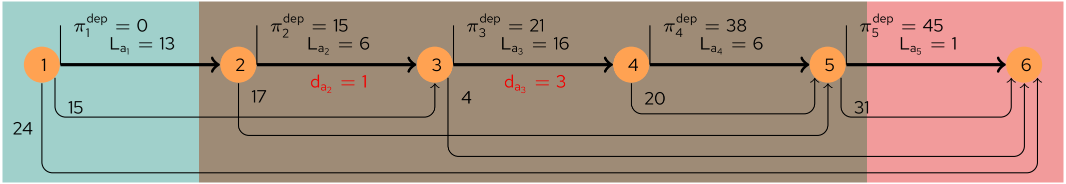
	1	2	3	4	5	6
1		**244**	229	229	118	0
2		126	126	126	24	0
3			24	24	24	0
4				0	0	0
5					0	0

- Don't wait at 2: $39 \cdot 6 + 126 = 360$
- Wait at 2: $15 \cdot 1 + 229 = 244$

RPTU

Example

$T = 6$



	1	2	3	4	5	6
1	**244**	244	229	229	118	0
2		126	126	126	24	0
3			24	24	24	0
4				0	0	0
5					0	0

RPTU

Possible Future Work

- Approximation algorithms for more complicated networks
- Fixed-parameter tractability
- Online delay management

RPTU

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.

RPTU

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.
- The delays of all passengers boarding at station v_i become known when the train arrives at v_i.

RPTU

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.
- The delays of all passengers boarding at station v_i become known when the train arrives at v_i.
- In this case, a solution specifies a single station where the train waits and from which it keeps the delay.

RPTU

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.
- The delays of all passengers boarding at station v_i become known when the train arrives at v_i.
- In this case, a solution specifies a single station where the train waits and from which it keeps the delay.
- For this setting they give a 2-competitive online algorithm.

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.
- The delays of all passengers boarding at station v_i become known when the train arrives at v_i.
- In this case, a solution specifies a single station where the train waits and from which it keeps the delay.
- For this setting they give a 2-competitive online algorithm.
- Krumke, Thielen, and Zeck gave a lower bound of 1.837.

Future Work: Online Setting

Previous Work

- Gatto et al. (2004) consider a single line without slacks and passengers arriving with delays from $\{0, 1\}$.
- The delays of all passengers boarding at station v_i become known when the train arrives at v_i.
- In this case, a solution specifies a single station where the train waits and from which it keeps the delay.
- For this setting they give a 2-competitive online algorithm.
- Krumke, Thielen, and Zeck gave a lower bound of 1.837.

What about our setting?

- Arbitrary delays
- Slack times

RPTU

Future Work: Online Setting

When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

RPTU

Future Work: Online Setting

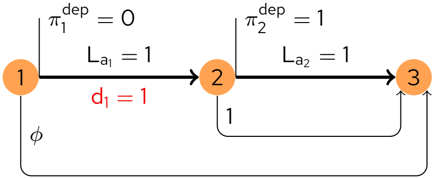
When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

Example (A)

$T = 2$



RPTU

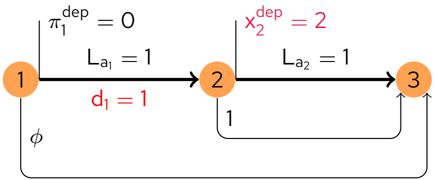
Future Work: Online Setting

When does the delay become known?

A The delay of a train a_i becomes known when the train departs.
B The delay of a train a_i becomes known when the train arrives.

Example (A)

$T = 2$



RPTU

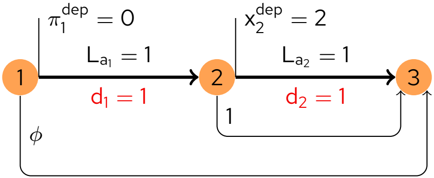
Future Work: Online Setting

When does the delay become known?

A The delay of a train a_i becomes known when the train departs.
B The delay of a train a_i becomes known when the train arrives.

Example (A)

$T = 2$



- Total delay: $(\phi + 1) \cdot 2$
- Without waiting: $\phi \cdot 2 + 1$
- Competitive ratio: $\sqrt{5} - 1 \approx 1.236$.

RPTU

Future Work: Online Setting

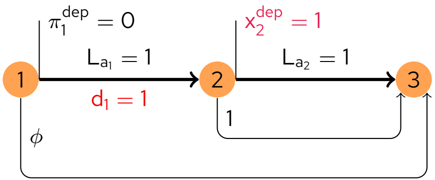
When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

Example (A)

$T = 2$



RPTU

Future Work: Online Setting

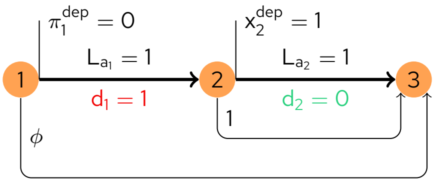
When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

Example (A)

$T = 2$



- Total delay: $\phi \cdot 2$
- With waiting: $\phi + 1$
- Competitive ratio: $\sqrt{5} - 1 \approx 1.236$.

RPTU

Future Work: Online Setting

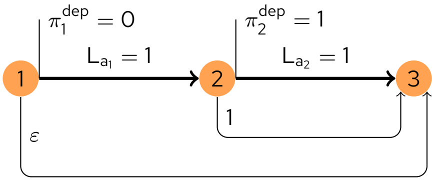
When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

Example (B)

$T = 2$



RPTU

Future Work: Online Setting

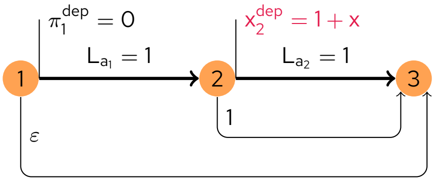
When does the delay become known?

A The delay of a train a_i becomes known when the train departs.

B The delay of a train a_i becomes known when the train arrives.

Example (B)

$T = 2$



RPTU

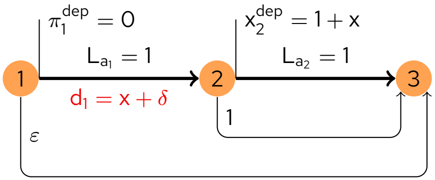
Future Work: Online Setting

When does the delay become known?

A The delay of a train a_i becomes known when the train departs.
B The delay of a train a_i becomes known when the train arrives.

Example (B)

$T = 2$



RPTU

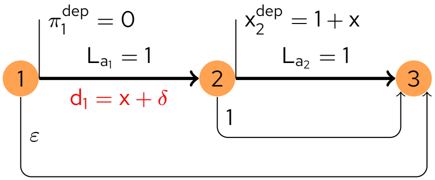
Future Work: Online Setting

When does the delay become known?

A The delay of a train a_i becomes known when the train departs.
B The delay of a train a_i becomes known when the train arrives.

Example (B)

$T = 2$



- Total delay: $\varepsilon \cdot 2 + 1 \cdot x$,
- Waiting until arrival: $(1 + \varepsilon) \cdot (x + \delta)$,
- Without waiting: $\varepsilon \cdot 2$,
- Competitive ratio: $\geq 1 + \frac{1}{1+\varepsilon} \xrightarrow{\varepsilon \to 0} 2$.

Summary

1. The delay management problem on out-trees with source delays on the arcs can be solved in polynomial time by a dynamic program (while it is NP-hard for in-trees).

RPTU

Summary

1. The delay management problem on out-trees with source delays on the arcs can be solved in polynomial time by a dynamic program (while it is NP-hard for in-trees).
2. There are different interesting online models (with simple lower bounds).

RPTU

Thank you!

References

- M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer: Railway delay management: Exploring the algorithmic complexity. SWAT 2004
- M. Gatto, R. Jacob, L. Peeters, and P. Widmayer: Online delay management on a single train line. Dagstuhl Workshp Railway Optimization 2004
- M. Gatto, R. Jacob, L. Peeters, and A. Schöbel: The computational complexity of delay management. WG 2005