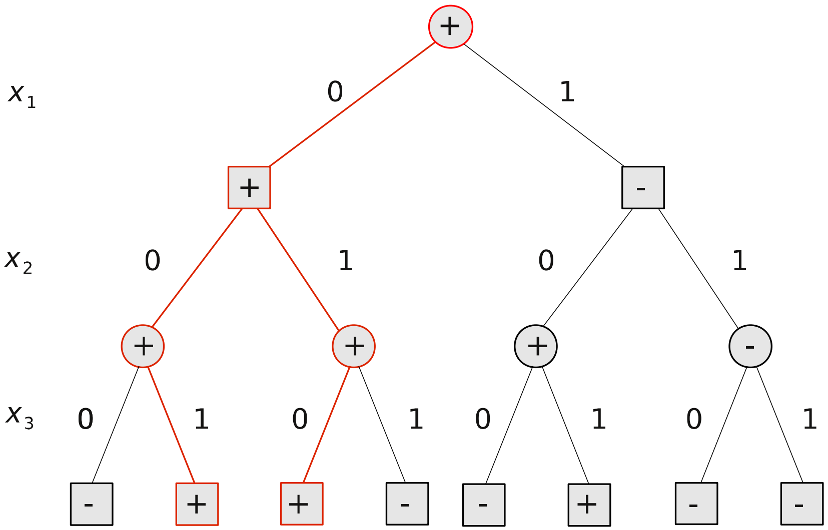
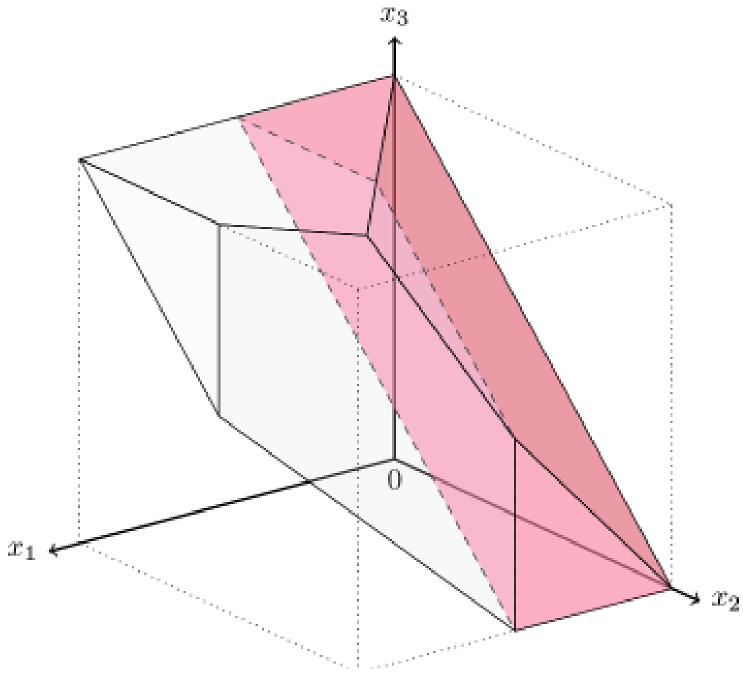
Quantified Combinatorial Optimization

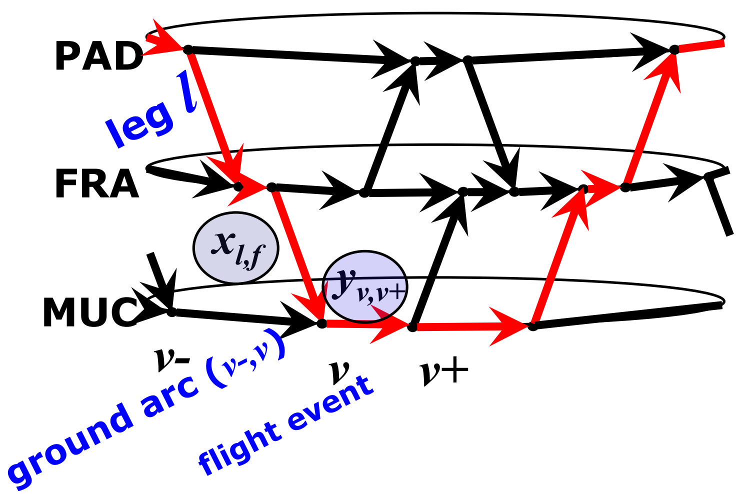
Thorsten Ederer, Ulf Lorenz, Thomas Opfer





Motivation

Some years ago: Fleet Assignment and disruptions



$x_{l,f}=1$ ⇔ leg l is assigned to subfleet f

$y_{v,v+}$: aircraft on ground

maximize $\sum_{l\in\mathcal{L}}\sum_{f\in\mathcal{F}} p_{l,f}\cdot x_{l,f}$

subject to

$$\sum_{f\in\mathcal{F}} x_{l,f} = 1 \quad \forall\, l\in\mathcal{L}$$

$$\sum_{l_f^{arr}=v} x_{l,f} - \sum_{l_f^{dep}=v} x_{l,f} + y_{v^-,v} - y_{v,v^+} = 0 \quad \forall\, v\in V$$

$$\sum_{l_f\in E_{F0}^f} x_{l,f} + \sum_{(v,v^+)\in E_{G0}^f} y_{v,v^+} \leq size_f \quad \forall f\in\mathcal{F}$$

$$x_{l,f} \in \{0,1\} \quad \forall\, l\in\mathcal{L}; \forall\, f\in\mathcal{F}$$

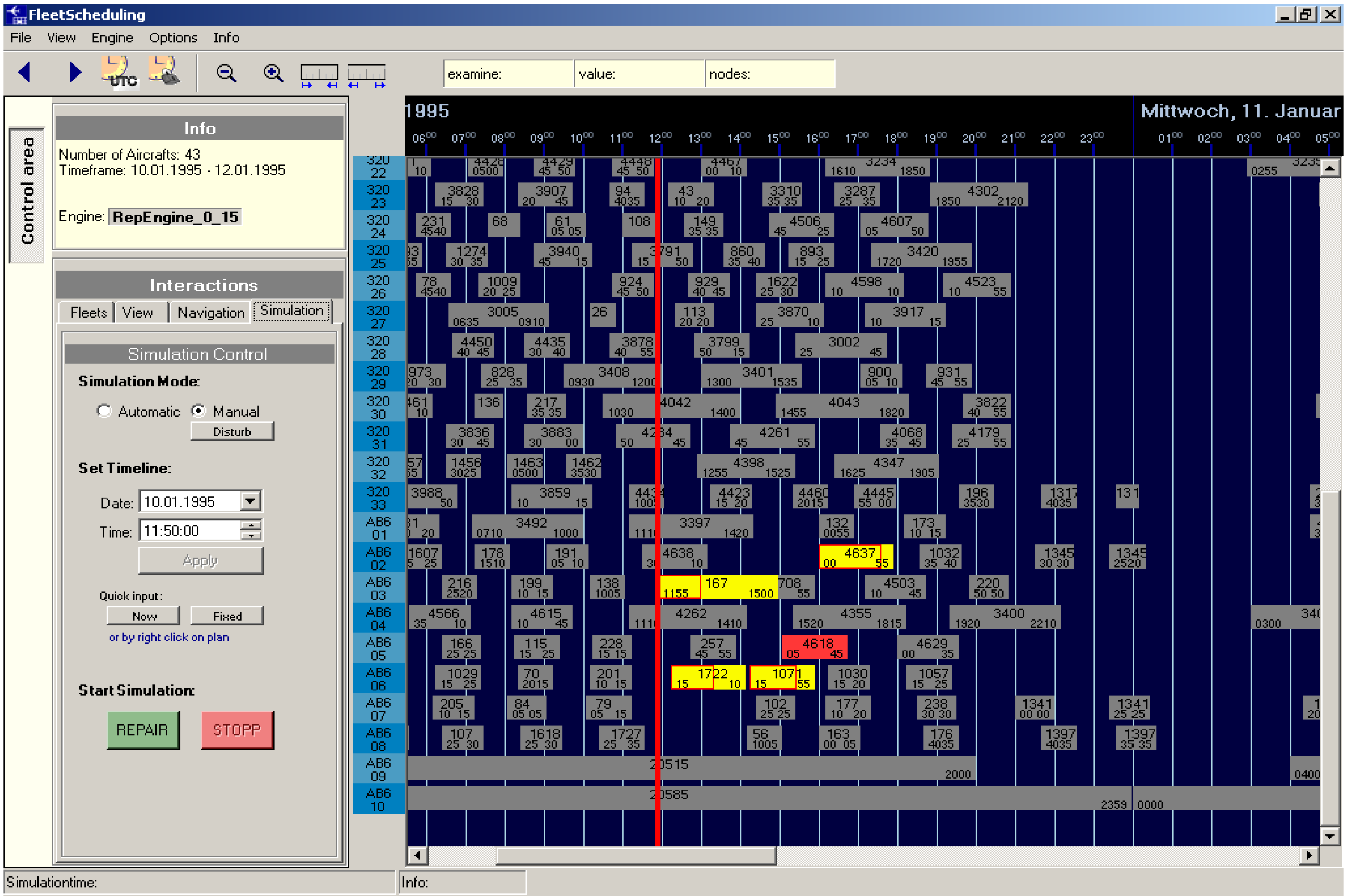
$$y_{v,v^+} \geq 0 \quad \forall\,(v,v^+)\in E_G$$

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Lufthansa Systems

Motivation

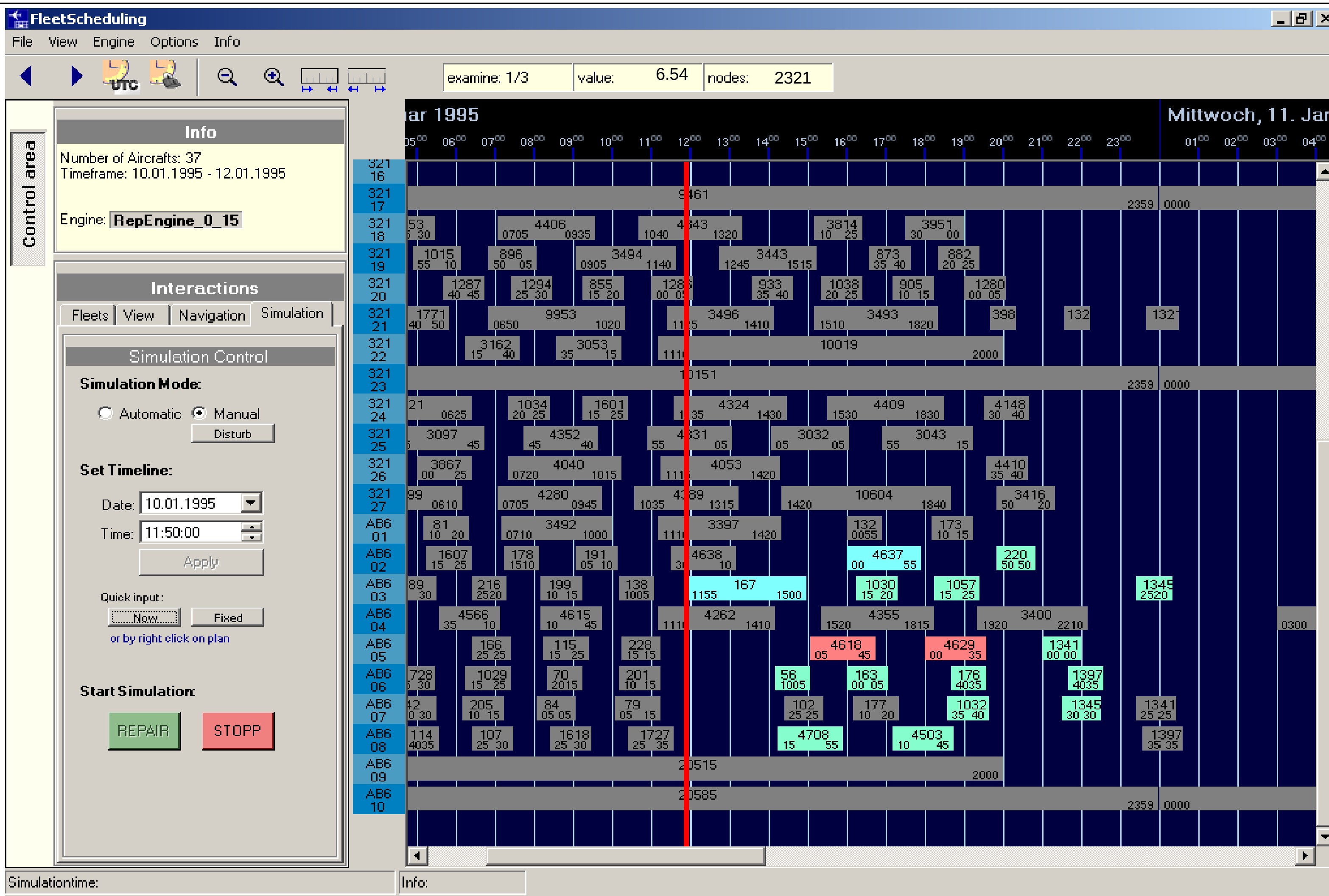
TECHNISCHE UNIVERSITÄT DARMSTADT



FLUID SYSTEM TECHNIK

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT



FLUID SYSTEM TECHNIK

Motivation

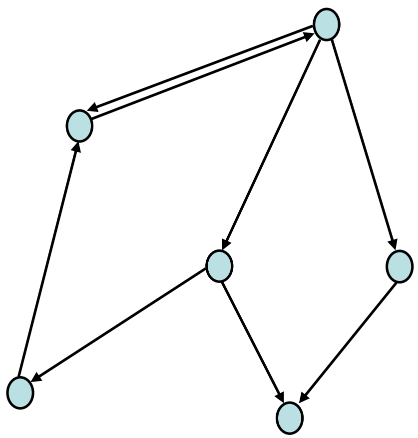
TECHNISCHE UNIVERSITÄT DARMSTADT

Some years ago: Railway delay management and robust planning

given: **plan** **nodes (stations), arcs (tracks)**

12:45 Hamburg – Berlin 15:45
13:00 Hamburg – Kiel 14:30
...

12:45 Berlin – München 15:45
13:00 Berlin – Darmstadt 16:30
...



Technische Universität Berlin

further attributes of the model:
arcs have capacities, FIFO ...
- passengers follow routes (paths), weights and desired start times
- online decision problem: Should we wait for the connecting train?
- minimal travel times at arcs are uncertain

wanted: plan and policy which minimizes the sum of all travel times of all passengers.

FLUID SYSTEM TECHNIK

Motivation

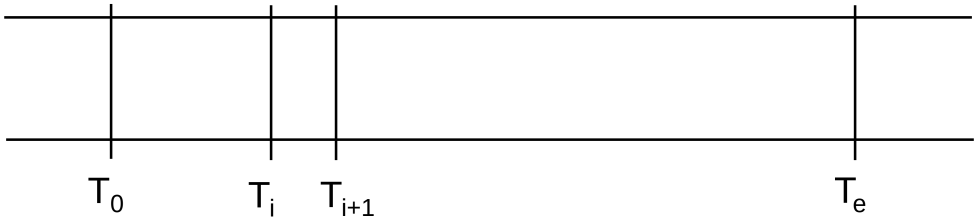
Optimization model „Game against Nature":

1. Player 1: The company — optimizes its plan.

2. Player 2: Nature — „rolls dice".

3. Game: from time T_0 to T_e. After each time step T_i, Nature determines some data. Then the company reacts with a recovery action.

4. The game is a multi-level decision process.

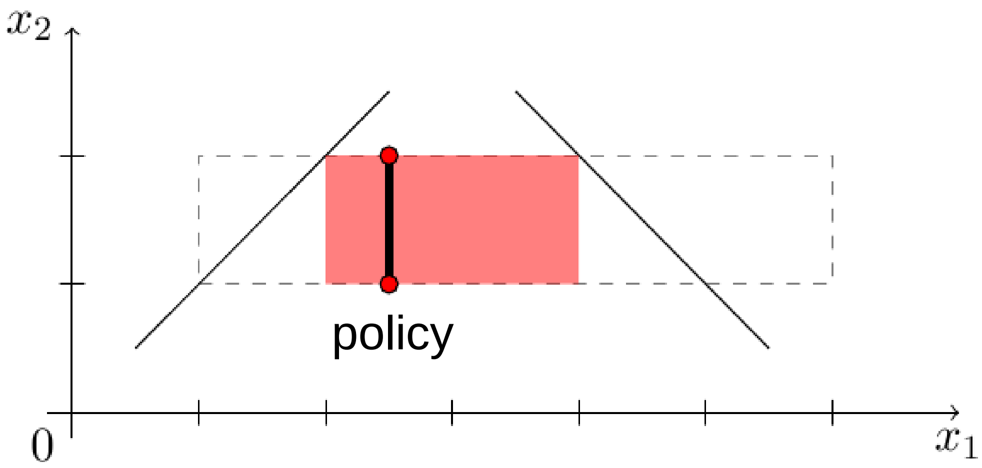


Quantified Linear Programs [1)]

$$\exists x_1 \in [0,1]\ \forall x_2 \in [0,1]\ \exists x_3 \in [0,1]:$$

$$\begin{pmatrix} 0 & -1 & -1 \\ -1 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ 3 \end{pmatrix}$$

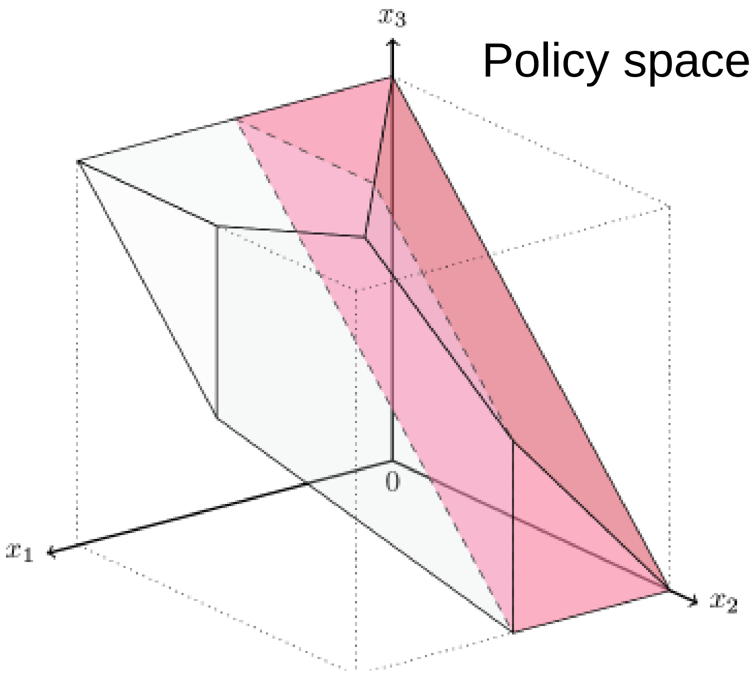
- The union of all games of all winning policies for the existential player forms a polytope.



$$\exists x_1 \in [1,6]\ \forall x_2 \in [1,2] : x_1 + x_2 \leq 6 \ \wedge \ x_2 - x_1 \leq 0$$

1) K. Subramani. Analyzing selected quantified integer programs. Springer, LNAI 3097, pages 342–356, 2004.

Quantified Linear Programs



Quantified Linear Program (QLP)

- continuous variables
- polyhedral solution space
- relaxation of QIP

- There is a winning policy for the existential player against the universal player
  if and only if
  there is a wining policy against the universal player when he is limited to {0,1}
- If the number of quantifier-alternations is constant the vertices of the polytope of winning policies can be described with polynomially many bits.

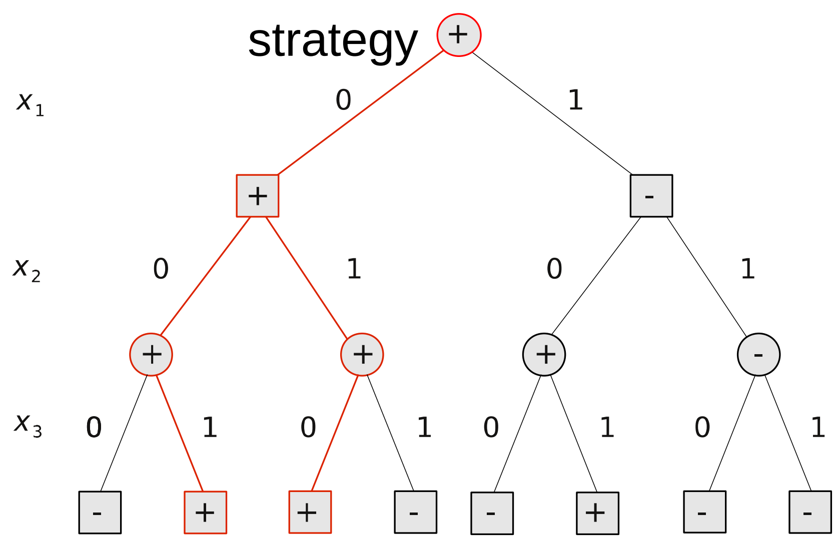
Quantified Linear Integer Programs

$$\exists x_1 \in [0,1]\ \forall x_2 \in [0,1]\ \exists x_3 \in [0,1]\ \text{, x integer:}$$

$$\begin{pmatrix} 0 & -1 & -1 \\ -1 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ 3 \end{pmatrix}$$

Quantified Integer Program (QIP)

- integral variables
- can be visualized as a game-tree, strategy
- PSPACE-complete



A minmax-objective can be added:
E.g.: The existential player tries to minimize $3x_1 + 2x_2 - x_4$ against all odds.

Quantified Jobshop

TECHNISCHE UNIVERSITÄT DARMSTADT

Machines
Papers 1, 2, 3

Table 1: Jobshop model notation.

J	set of jobs
M	set of machines
T	set of tasks, $T \subseteq J \times M$
O	taskorder, $O \subseteq T \times T$
$s_{j,m}$	start time (integer) of task (j,m)
$d_{j,m}$	duration of task (j,m)
$\delta_{j,m}$	additional duration of task (j,m) in case of delay
$e_{j,m}$	earliness of task (j,m), i.e., $e = \max\{d^1 - d^2, 0\}$
$\overline{e}$	mean earliness
m	makespan
$r_{u,j,m}$	indicator of unary encoding of retarded task
$\tilde{r}_b$	indicator of binary encoding retarded task
w	wrapping indicator for binary to unary translation

Table 2: Jobshop tasks.

job	machine	duration	extra
Paper1	Blue	45	5
Paper1	Yellow	10	0
Paper2	Blue	20	5
Paper2	Green	10	10
Paper2	Yellow	34	0
Paper3	Blue	12	0
Paper3	Green	17	0
Paper3	Yellow	28	20

Table 3: Jobshop order.

prior task	later task
Paper1 Blue	Paper1 Yellow
Paper2 Green	Paper2 Blue
Paper2 Blue	Paper2 Yellow
Paper3 Yellow	Paper3 Blue
Paper3 Blue	Paper3 Green

Quantified Jobshop

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\min \quad m^2 + k \cdot \bar{e} + \frac{1}{M} \cdot m^1 \qquad \text{s.t.} \qquad \exists\, s^1\, y^1\, m^1\ \forall\, \tilde{r}\ \exists\, r\, w,\, s^2\, y^2\, m^2,\, e : \tag{1}$$

$\forall\, \tilde{r}$ → bin. encoding of scenario index

4 → 01000

$$s^1_{j,m} + d_{j,m} \le m^1 \qquad \forall\, (j,m) \in T \tag{2}$$

$$s^1_{i,m} + d_{i,m} \le s^1_{j,n} \qquad \forall\, (i,m,j,n) \in O \tag{3}$$

$$s^1_{i,m} + d_{i,m} \le s^1_{j,m} + M \cdot (1 - y^1_{i,m,j}) \qquad \forall\, (i,m) \in T,\ (j,m) \in T \tag{4}$$

$$s^1_{j,m} + d_{j,m} \le s^1_{i,m} + M \cdot y^1_{i,m,j} \qquad \forall\, (i,m) \in T,\ (j,m) \in T \tag{5}$$

$$\sum_{(u,j,m) \in U} u \cdot r_{j,m} = \sum_{b \in B} 2^b \cdot \tilde{r}_b - |T| \cdot w \quad \wedge \quad \sum_{\substack{u \in U \\ (j,m) = T_u}} r_{j,m} \le 1 \tag{6}$$

$$s^2_{j,m} + d_{j,m} + \delta_{j,m} \cdot r_{j,m} \le m^2 \qquad \forall\, (j,m) \in T \tag{7}$$

$$s^2_{i,m} + d_{i,m} + \delta_{i,m} \cdot r_{i,m} \le s^2_{j,n} \qquad \forall\, (i,m,j,n) \in O \tag{8}$$

$$s^2_{i,m} + d_{i,m} + \delta_{i,m} \cdot r_{i,m} \le s^2_{j,m} + M \cdot (1 - y^2_{i,m,j}) \qquad \forall\, (i,m) \in T,\ (j,m) \in T \tag{9}$$

$$s^2_{j,m} + d_{j,m} + \delta_{i,m} \cdot r_{i,m} \le s^2_{i,m} + M \cdot y^2_{i,m,j} \qquad \forall\, (i,m) \in T,\ (j,m) \in T \tag{10}$$

$$e_{i,m} \ge s^1_{i,m} - s^2_{i,m} \qquad \forall\, (i,m) \in T \tag{11}$$

$$e_{i,m} \ge 0 \tag{12}$$

$$\bar{e} = \frac{1}{|T|} \cdot \sum_{(i,m) \in T} e_{i,m} \tag{13}$$

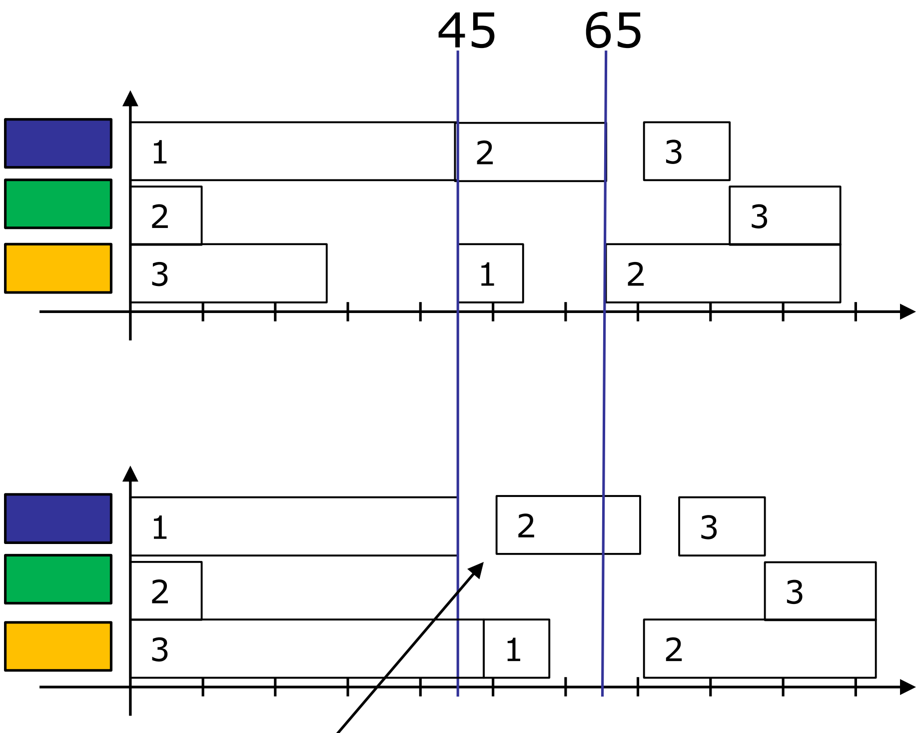
FLUID SYSTEM TECHNIK

Quantified Jobshop

TECHNISCHE UNIVERSITÄT DARMSTADT

Table 4: Solution of the Jobshop Example.

sc.	1B	1Y	2G	2B	2Y	3Y	3B	3G	m.s.
	first stage solution								
	0	45	0	45	65	0	70	82	99
1	0	50	0	50	70	0	70	82	104
2	0	45	0	45	65	0	65	82	99
3	0	45	0	45	70	0	70	82	104
4	0	45	0	45	65	0	65	82	99
5	0	45	0	45	65	0	65	82	99
6	0	45	0	45	65	0	65	82	99
7	0	45	0	45	65	0	70	82	99
8	0	48	0	50	70	0	75	87	104



Some space, i.e. this is not
The worst case scenario.

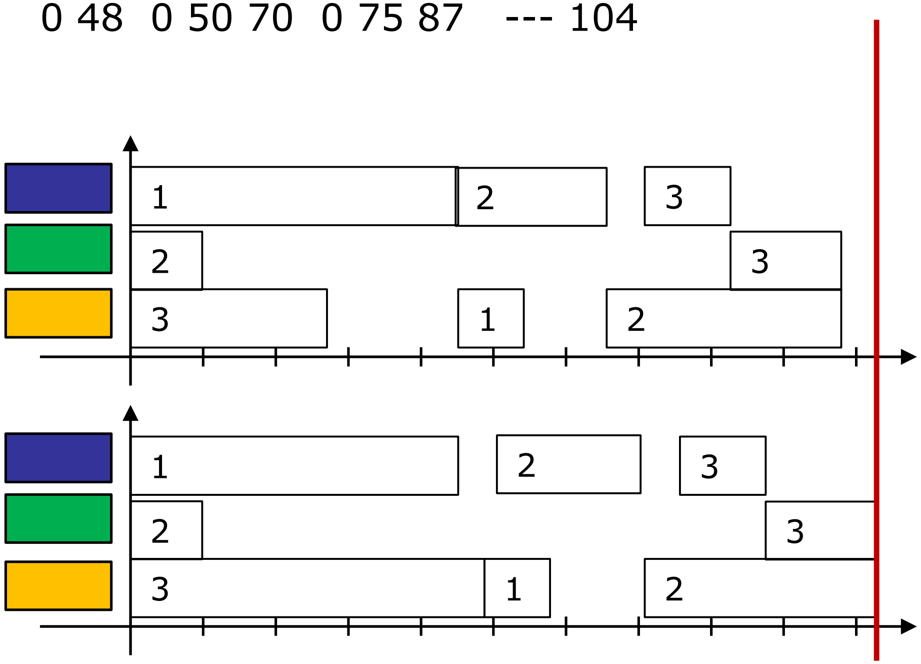
FLUID SYSTEM TECHNIK

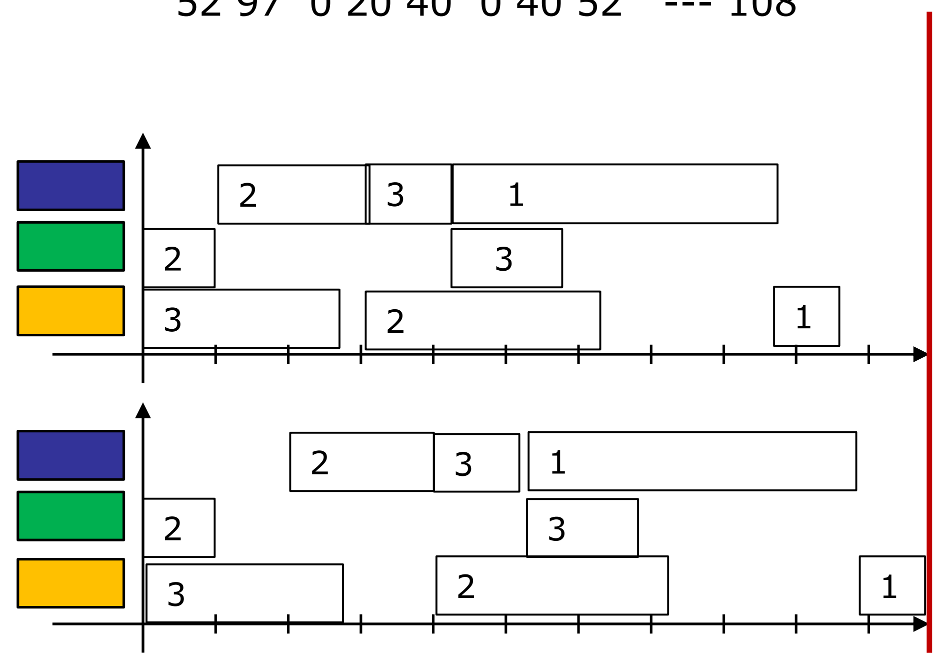
Quantified Jobshop

TECHNISCHE UNIVERSITÄT DARMSTADT

Optimal first-stage solution
0 45 0 45 65 0 70 82 --- 99
Scenario 8:
0 48 0 50 70 0 75 87 --- 104

Optimal deterministic solution
42 87 0 10 30 0 30 42 --- 97
Scenario 4:
52 97 0 20 40 0 40 52 --- 108





No scenario is worse: 104

no awareness: 108

FLUID SYSTEM TECHNIK

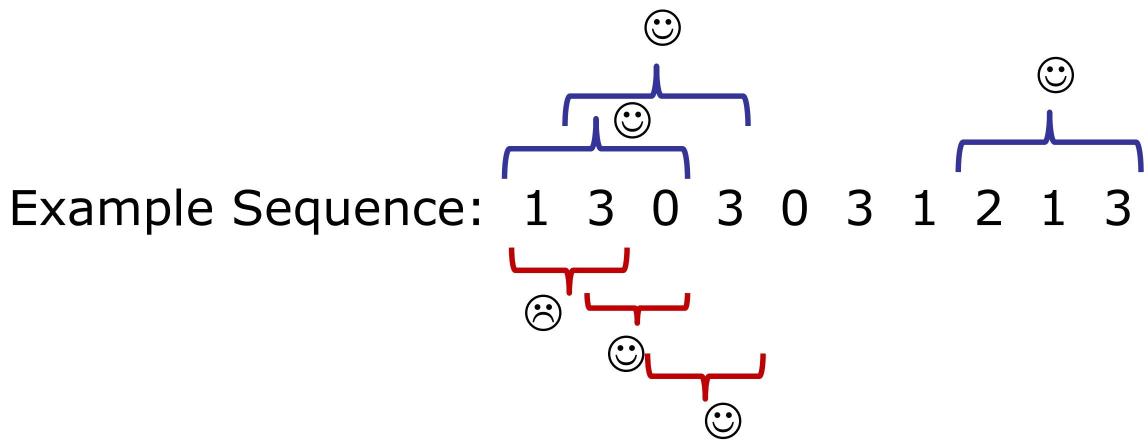
Quantified Carsequencing

Options: 0: basic 1: big wheels 2: no roof

Table 5: Car Instance.

opt	r	s
1	1	2
2	2	3

class	cars	opt 1	opt 2
0	2	0	0
1	3	0	1
2	1	1	0
3	4	1	1



Minimize the violated rules

Quantified Carsequencing

Options: 0: basic 1: big wheels 2: no roof

Table 5: Car Instance.

opt	r	s
1	1	2
2	2	3

class	cars	opt 1	opt 2
0	2	0	0
1	3	0	1
2	1	1	0
3	4	1	1

Example Sequence: 1 3 0 3 0 3 1 2 1 3

Disruption:

Example Sequence: 1 0 3 0 3 1 3 2 1 3

Recover-Action:

Example Sequence: 1 0 3 0 3 1 3 1 3 2

Quantified Carsequencing

Options: 0: basic 1: big wheels 2: no roof

Table 6: Notation of the car sequencing model.

O	set of options
C	set of classes, $C \subseteq \mathcal{P}(O)$
T	set of timesteps ($\lvert T \rvert$ equals number of models)
T^k	set of intervals (by first timestep) for option k , $T^k = \{1, \ldots, \lvert T \rvert - s_k + 1\}$
$r_k : s_k$	at most r_k out of s_k successively sequenced models may require option k
D_c	demand of models of class c
$A_{k,c}$	indicator, if models of class c require option k
$x_{t,c}$	indicator, if a model of class c is produced at timestep t
y_{k,t_0}	indicator, if the sequencing rule $r_k : s_k$ beginning at timestep t_0 is satisfied
S	set of stages (duplicates of the original variables after each move)
σ	stage index: 1 pre-scheduling, 2 state after delay, 3 re-scheduling
F	ordered set of possible delays, $F = \{(t,t') \in T \times T,\, t < t'\}$
U	unary encoding vector of F
m_u	indicator of unary encoding for the delay from timestep t to t', $(t,t') \in U$
B	binary encoding of possible delays
$\widetilde{\mathrm{m}}_b$	indicator of binary encoding for the delay, $b \in B$
w	wrapping indicator for binary to unary translation

Quantified Carsequencing

$$\min \sum_{k \in O} \sum_{t_0 \in T^k} y^1_{k,t_0} \qquad \text{s.t.} \qquad \exists\, x^1\, y^1\, \forall\, \tilde{m}\, \exists\, m\, w,\, x^2\, y^2,\, a,\, x^3\, y^3 : \tag{14}$$

$$\sum_{t \in T} x^s_{t,c} = D_c \qquad \forall\, c \in C,\, s \in S \tag{15}$$

$$\sum_{c \in C} x^s_{t,c} = 1 \qquad \forall\, t \in T,\, s \in S \tag{16}$$

$$\sum_{t=t_0}^{t_0+s_k} \sum_{c \in C} A_{k,c} \cdot x^s_{t,c} \leq r_k + M \cdot y^s_{k,t_0} \qquad \forall\, k \in O,\, t_0 \in T^k,\, s \in S \tag{17}$$

$$\sum_{u \in U} u \cdot m_u = \sum_{b \in B} 2^b \cdot \tilde{m}_b - |F| \cdot w \quad \wedge \quad \sum_{u \in U} m_u \leq 1 \tag{18}$$

$$|x^2_{t,c} - x^1_{t,c}| \leq \sum_{\substack{u \in U \\ (t_i,t_j)=F_u \\ t_i < t < t_j}} m_u \quad \forall\, c \in C,\, t \in T \tag{19}$$

$$\textit{further stage-connecting constraints} \ldots \tag{20}$$

Quantified Carsequencing

Table 7: Solution of the Car Sequencing Example.

Scenario	Mal.	Ans.	Production Sequence	Scenario	Mal.	Ans.	Production Sequence
first stage solution:			1, 3, 0, 3, 0, 3, 1, 2, 1, 3	*first stage solution:*			1, 3, 0, 3, 0, 3, 1, 2, 1, 3
1	–	(4, 5)	1, 3, 0, 3, 3, 0, 1, 2, 1, 3	24	(2, 8)	–	1, 3, 3, 0, 3, 1, 2, 1, 0, 3
2	(0, 1)	(2, 7)	3, 1, 3, 0, 3, 1, 2, 0, 1, 3	25	(2, 9)	–	1, 3, 3, 0, 3, 1, 2, 1, 3, 0
3	(0, 2)	–	3, 0, 1, 3, 0, 3, 1, 2, 1, 3	26	(3, 4)	–	1, 3, 0, 0, 3, 3, 1, 2, 1, 3
4	(0, 3)	–	3, 0, 3, 1, 0, 3, 1, 2, 1, 3	27	(3, 5)	–	1, 3, 0, 0, 3, 3, 1, 2, 1, 3
5	(0, 4)	–	3, 0, 3, 0, 1, 3, 1, 2, 1, 3	28	(3, 6)	–	1, 3, 0, 0, 3, 1, 3, 2, 1, 3
6	(0, 5)	(6, 8)	3, 0, 3, 0, 3, 1, 2, 1, 1, 3	29	(3, 7)	–	1, 3, 0, 0, 3, 1, 2, 3, 1, 3
7	(0, 6)	(7, 8)	3, 0, 3, 0, 3, 1, 1, 1, 2, 3	30	(3, 8)	–	1, 3, 0, 0, 3, 1, 2, 1, 3, 3
8	(0, 7)	(8, 9)	3, 0, 3, 0, 3, 1, 2, 1, 3, 1	31	(3, 9)	–	1, 3, 0, 0, 3, 1, 2, 1, 3, 3
9	(0, 8)	–	3, 0, 3, 0, 3, 1, 2, 1, 1, 3	32	(4, 5)	–	1, 3, 0, 3, 3, 0, 1, 2, 1, 3
10	(0, 9)	–	3, 0, 3, 0, 3, 1, 2, 1, 3, 1	33	(4, 6)	–	1, 3, 0, 3, 3, 1, 0, 2, 1, 3
11	(1, 2)	–	1, 0, 3, 3, 0, 3, 1, 2, 1, 3	34	(4, 7)	(8, 9)	1, 3, 0, 3, 3, 1, 2, 0, 3, 1
12	(1, 3)	–	1, 0, 3, 3, 0, 3, 1, 2, 1, 3	35	(4, 8)	–	1, 3, 0, 3, 3, 1, 2, 1, 0, 3
13	(1, 4)	–	1, 0, 3, 0, 3, 3, 1, 2, 1, 3	36	(4, 9)	–	1, 3, 0, 3, 3, 1, 2, 1, 3, 0
14	(1, 5)	–	1, 0, 3, 0, 3, 3, 1, 2, 1, 3	37	(5, 6)	(7, 9)	1, 3, 0, 3, 0, 1, 3, 1, 3, 2
15	(1, 6)	–	1, 0, 3, 0, 3, 1, 3, 2, 1, 3	38	(5, 7)	–	1, 3, 0, 3, 0, 1, 2, 3, 1, 3
16	(1, 7)	–	1, 0, 3, 0, 3, 1, 2, 3, 1, 3	39	(5, 8)	–	1, 3, 0, 3, 0, 1, 2, 1, 3, 3
17	(1, 8)	–	1, 0, 3, 0, 3, 1, 2, 1, 3, 3	40	(5, 9)	–	1, 3, 0, 3, 0, 1, 2, 1, 3, 3
18	(1, 9)	–	1, 0, 3, 0, 3, 1, 2, 1, 3, 3	41	(6, 7)	(8, 9)	1, 3, 0, 3, 0, 3, 2, 1, 3, 1
19	(2, 3)	(4, 9)	1, 3, 3, 0, 3, 1, 2, 1, 3, 0	42	(6, 8)	–	1, 3, 0, 3, 0, 3, 2, 1, 1, 3
20	(2, 4)	–	1, 3, 3, 0, 0, 3, 1, 2, 1, 3	43	(6, 9)	–	1, 3, 0, 3, 0, 3, 2, 1, 3, 1
21	(2, 5)	–	1, 3, 3, 0, 3, 0, 1, 2, 1, 3	44	(7, 8)	–	1, 3, 0, 3, 0, 3, 1, 1, 2, 3
22	(2, 6)	–	1, 3, 3, 0, 3, 1, 0, 2, 1, 3	45	(7, 9)	–	1, 3, 0, 3, 0, 3, 1, 1, 3, 2
23	(2, 7)	–	1, 3, 3, 0, 3, 1, 2, 0, 1, 3	46	(8, 9)	–	1, 3, 0, 3, 0, 3, 1, 2, 3, 1

P	Set of items
M	Set of machines
I	Set of tasks
T	Number of periods (1…T)
M_i	Machine needed by task i
P_i	Item produced by task i
I_i	$= \{ j \in I : M_j = M_i, j \neq i \}$. Set of tasks that concur with task i for the same machine.
S_p	Nonnegative holding costs for one unit of item p one period in inventory.
B_p	Nonnegative cost for backlogging one unit of item p for one period
O_p	Stockout cost for not fulfilling the (external) demand of one unit of item p
$X_{i,j}$	Nonnegative setup costs for switching from task i to j (on machine $M_i = M_j$)
Q_i	Production cost for one unit of item P_i with task i
$D_{p,t}$	External demand for item p in period t
$A_{p,q}$	Gozinto-factor. Zero if item q is not immediate successor of item p. Otherwise: quantity of item p directly needed to produce one item of q.
$\Delta_{p,i}$	(Integral) number of periods for „transporting" item p to machine M_i (for task i)
$C_{i,t}$	(Maximum) production quantity of item P_i with task i during period t
$T_{i,j}$	(Fractional) number of periods for switching from task i to j
$R_{i,j}$	$= \lfloor T_{i,j} \rfloor$. Number of periods excusively used for switching from task i to j
$R^f_{i,j}$	$= T_{i,j} - R_{i,j}$. Fraction of the $(R_{i,j}+1)$th period after switching from task i to j used for switching

$q_{i,t}$	Production quantity of item P_i by task i in period t
$s_{p,t}$	Inventory for item p at end of period t
$b_{p,t}$	Backlog of item p at end of period t
$o_{p,t}$	Shortfall quantity of item p in period t (no backlog)
$x_{i,j,t}$	Binary variable which indicates whether a setup change from task i to j occurs in (at the beginning of) period t ($x_{i,j,t}$=1) or not ($x_{i,j,t}$=0)
$y_{i,t}$	Binary variable which indicates whether machine M_i is set up for task i at the beginning of (during) period t ($y_{i,t}$=1) or not ($y_{i,t}$=0)

Thank you For your Attention!

minimize $$\sum_{t=1}^{T}\Big(\sum_{p\in P}(S_p s_{p,t}+B_p b_{p,t}+O_p o_{p,t})+\sum_{\substack{i\in I\\ j\in I_i}} X_{i,j} x_{i,j,t}+\sum_{i\in I} Q_i q_{i,t}\Big)$$

$$s_{p,t-1}+\sum_{i:P_i=p} q_{i,t}-D_{p,t}-\sum_{\substack{q,\\ j:P_j=q}} A_{q,p}\, q_{j,t+\Delta_{p,j}}+b_{p,t}+o_{p,t}-b_{p,t-1}=s_{p,t}, \forall p\in P, t=1\ldots T$$

$$b_{p,t}+o_{p,t}-b_{p,t-1}\le D_{p,t}, \forall p\in P, t=1\ldots T$$

$$\sum_{i:M_i=m} y_{i,t}=1, \forall m\in M, t=1\ldots T$$

$$y_{i,t-1}+y_{i,t}-1\le x_{i,j,t}, \forall i\in I, j\in I_i, t=1\ldots T$$

$$C_{i,t}\Big(y_{i,t}-\sum_{\substack{j\in I_i\\ t-R_{j,i}<t'\le t}} x_{j,i,t'}-\sum_{j\in I_i} R^f_{j,i}\, x_{j,i,t-R_{j,i}}\Big)\ge q_{i,t}, \forall i\in I, t=1\ldots T$$

$$q_{i,t}\ge 0; s_{p,t}, b_{p,t}, o_{p,t}\ge 0; x_{i,j,t}, y_{i,t}\in\{0,1\}; \forall i\in I, \forall p\in P, t=1\ldots T$$

	$y_{i,t}=0$	$y_{i,t}=1$
$y_{i,t-1}=1$	$x_{i,j,t}$ darf 0 sein	$x_{i,j,t}$ = 1!
$y_{i,t-1}=0$	$x_{i,j,t}$ darf 0 sein	$x_{i,j,t}$ darf 0 sein