Linear and dynamic programming for constraints

Hadrien Cambazard
G-SCOP, Université Grenoble Alpes

Outline

1. **Reduced-costs based filtering**
    - Linear Programming duality
    - First example: *AtMostNValue*
        - Filtering the upper bound of a 0/1 variable
        - Filtering the lower bound of a 0/1 variable
    - General principles
    - Second example
    - Assignment, Cumulative, Bin-packing, …
2. **Dynamic programming filtering algorithms**
    - Linear equation, WeightedCircuit
    - General principles
    - Other relationships of DP and CP
3. **Illustration with a real-life application**

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

Reduced cost based filtering

- Linear Programming duality [Linear Programming, Chvatal, 2003]
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \; 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Linear Programming duality

$$\begin{array}{lrcl} & \text{Min } z = \quad 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

$$z = 5x + 6y \geq \overbrace{2x + 3y}^{c_1} \geq 10$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

$$z = \boxed{5}x + 6y \geq \overbrace{\boxed{2}x + 3y}^{c_1} \geq 10$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq 10 \\
(c_2) & & x + y & \geq 5 \\
 & & x, y & \geq 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{2}x + \boxed{3}y}^{c_1} \geq 10$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{2}x + \boxed{3}y}^{c_1} \geq 10$$

And x, y positive

Linear Programming duality

$$
\begin{array}{llcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq & 10 \\
(c_2) & & x + y & \geq & 5 \\
 & & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 alone:

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{2}x + \boxed{3}y}^{c_1} \geq 10$$

So $z^* \geq 10$

And x, y positive

Linear Programming duality

$$\begin{array}{llcl} & \text{Min } z = \; 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 :

… so $z^* \geq 10$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq & 10 \\
(c_2) & & x + y & \geq & 5 \\
 & & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = 5x + 6y \geq 3x + 4y \geq 10 + 5 = 15$$

Linear Programming duality

$$
\begin{array}{llcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq & 10 \\
(c_2) & & x + y & \geq & 5 \\
 & & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = 5x + 6y \geq \overbrace{3x + 4y}^{c_1 + c_2} \geq 10 + 5 = 15$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \;\; 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = 5x + \boxed{6}y \geq \overbrace{3x + \boxed{4}y}^{c_1 + c_2} \geq 10 + 5 = 15$$

$\geq$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq \; 10 \\
(c_2) & & x + y & \geq \; 5 \\
 & & x, y & \geq \; 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{3}x + \boxed{4}y}^{c_1 + c_2} \geq 10 + 5 = 15$$

Linear Programming duality

$$\begin{array}{lrcl}
 & \text{Min } z = & 5x + 6y & \\
(c_1) & & 2x + 3y & \geq & 10 \\
(c_2) & & x + y & \geq & 5 \\
 & & x, y & \geq & 0
\end{array}$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{3}x + \boxed{4}y}^{c_1 + c_2} \geq 10 + 5 = 15$$

And x, y positive

Linear Programming duality

$$\begin{array}{lrcl} & \text{Min } z = \quad 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 : … so $z^* \geq 10$

$$z = \boxed{5}x + \boxed{6}y \geq \overbrace{\boxed{3}x + \boxed{4}y}^{c_1 + c_2} \geq 10 + 5 = 15 \quad \text{so} \quad z^* \geq 15$$

And x, y positive

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \; 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

Using c_1 and c_2 :

… so $z^* \geq 10$

… so $z^* \geq 15$

$$z = \boxed{5}x + \boxed{6}y \geq \underbrace{\boxed{5}x + \boxed{6}y}_{c_1 + 3c_2} \geq 10 + 15 = 25$$ so $z^* \geq 25$

And x, y positive

Linear Programming duality

$$\begin{array}{lrcl} & \text{Min } z = & 5x+6y & & \\ (c_1) & & 2x+3y & \geq & 10 \\ (c_2) & & x+y & \geq & 5 \\ & & x,y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

c_1	implies	$z^* \geq 10$
$c_1 + c_2$	implies	$z^* \geq 15$
$c_1 + 3c_2$	implies	$z^* \geq 25$

Linear Programming duality

$$\begin{array}{lrcl} & \text{Min } z = \quad 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

c_1	implies	$z^* \geq 10$
$c_1 + c_2$	implies	$z^* \geq 15$
$c_1 + 3c_2$	implies	$z^* \geq 25$

Is there a gap left ?

Linear Programming duality

$$\begin{array}{lrcl} & \text{Min } z = \quad 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

c_1	implies	$z^* \geq 10$
$c_1 + c_2$	implies	$z^* \geq 15$
$c_1 + 3c_2$	implies	$z^* \geq 25$

Is there a gap left ?   No

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \ 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

$$
\begin{array}{lll}
c_1 & \text{implies} & z^* \geq 10 \\
c_1 + c_2 & \text{implies} & z^* \geq 15 \\
c_1 + 3c_2 & \text{implies} & z^* \geq 25
\end{array}
$$

Is there a gap left ? No

$$(x, y) = (5, 0) \quad \text{is feasible so} \quad z^* \leq 25$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

What lower bound can you derive from the constraints ?

c_1	implies	$z^* \geq 10$
$c_1 + c_2$	implies	$z^* \geq 15$
$c_1 + 3c_2$	implies	$z^* \geq 25$

Goal: a linear combination of the right hand sides

- that bounds the objective from below
- and which is maximum

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \; 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
$$

Goal: a linear combination of the right hand sides:

- that bounds the objective from below
- and which leads to the maximum bound

Linear Programming duality

$$\begin{array}{lrcl}
 & \text{Min } z = 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}$$

$$\boxed{\begin{matrix}\lambda_1 \\ \lambda_2\end{matrix}}$$

Goal: a linear combination of the right hand sides:

- that bounds the objective from below
- and which leads to the maximum bound

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
\qquad
\boxed{\begin{array}{c}\lambda_1 \\ \lambda_2\end{array}}
$$

Goal: a linear combination of the right hand sides:

- that bounds the objective from below
- and which leads to the maximum bound

$$
\begin{array}{rcl}
\text{Max } w = \quad 10\lambda_1 + 5\lambda_2 & & \\
2\lambda_1 + \lambda_2 & \leq & 5 \\
3\lambda_1 + \lambda_2 & \leq & 6 \\
\lambda_1, \lambda_2 & \geq & 0
\end{array}
$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
\qquad
\boxed{\begin{array}{c}\lambda_1 \\ \lambda_2\end{array}}
$$

Goal: a linear combination of the right hand sides:

- that bounds the objective from below
- and which leads to the maximum bound

$$
\begin{array}{rcl}
\text{Max } w = \quad 10\lambda_1 + 5\lambda_2 & & \\
2\lambda_1 + \lambda_2 & \leq & 5 \\
3\lambda_1 + \lambda_2 & \leq & 6 \\
\lambda_1, \lambda_2 & \geq & 0
\end{array}
$$

Linear Programming duality

$$
\begin{array}{lrcl}
 & \text{Min } z = \quad 5x + 6y & & \\
(c_1) & 2x + 3y & \geq & 10 \\
(c_2) & x + y & \geq & 5 \\
 & x, y & \geq & 0
\end{array}
\qquad
\boxed{\begin{array}{c}\lambda_1 \\ \lambda_2\end{array}}
$$

Goal: a linear combination of the right hand sides:

- that bounds the objective from below
- and which leads to the maximum bound

$$
\begin{array}{rcl}
\boxed{\text{Max } w = \quad 10\lambda_1 + 5\lambda_2} & & \\
2\lambda_1 + \lambda_2 & \leq & 5 \\
3\lambda_1 + \lambda_2 & \leq & 6 \\
\lambda_1, \lambda_2 & \geq & 0
\end{array}
$$

Linear Programming duality

$$\begin{array}{llcl} & \text{Min } z = \; 5x + 6y & & \\ (c_1) & 2x + 3y & \geq & 10 \\ (c_2) & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array}$$

What lower bound can you derive from the constraints ?

$$\begin{array}{lcl} \text{Max } w = \; 10\lambda_1 + 5\lambda_2 & & \\ 2\lambda_1 + \lambda_2 & \leq & 5 \\ 3\lambda_1 + \lambda_2 & \leq & 6 \\ \lambda_1, \lambda_2 & \geq & 0 \end{array}$$

Any **feasible solution** of the dual gives a lower bound

$c_1 + c_2$ is $(\lambda_1, \lambda_2) = (1, 1)$ which gives $w = 15$

$c_1 + 3c_2$ is $(\lambda_1, \lambda_2) = (1, 3)$ which gives $w = 25$

Linear Programming duality

(P)

$$\begin{array}{rrcl} \text{Min } z = & 5x + 6y & & \\ & 2x + 3y & \geq & 10 \\ & x + y & \geq & 5 \\ & x, y & \geq & 0 \end{array} \qquad \boxed{\begin{array}{c} \lambda_1 \\ \lambda_2 \end{array}}$$

What lower bound can you derive from the constraints ?

(D)

$$\begin{array}{rrcl} \text{Max } w = & 10\lambda_1 + 5\lambda_2 & & \\ & 2\lambda_1 + \lambda_2 & \leq & 5 \\ & 3\lambda_1 + \lambda_2 & \leq & 6 \\ & \lambda_1, \lambda_2 & \geq & 0 \end{array} \qquad \boxed{\begin{array}{c} x \\ y \end{array}}$$

The dual of the dual is the primal

Linear Programming duality

(P)

$$\begin{aligned} \text{Min } z = \quad & \sum_{i=1}^{n} c_i x_i \\ & \sum_{i=1}^{n} a_{ij} x_i \quad \geq \quad b_j \quad \forall j = 1, \ldots, m \\ & x_i \quad \geq \quad 0 \quad \forall i = 1, \ldots, n \end{aligned}$$

(D)

$$\begin{aligned} \text{Max } w = \quad & \sum_{j=1}^{m} b_j \lambda_j \\ & \sum_{j=1}^{m} a_{ij} \lambda_j \quad \leq \quad c_i \quad \forall i = 1, \ldots, n \\ & \lambda_j \quad \geq \quad 0 \quad \forall j = 1, \ldots, m \end{aligned}$$

- View the dual as the problem of the best linear combination of the constraints
- Any feasible solution of the dual gives a lower bound

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

$$\begin{aligned}
D(X_1) &= \{1,2,3,4,5,6\} \\
D(X_2) &= \{2,4\} \\
D(X_3) &= \{1,2\} \\
D(X_4) &= \{1,2,3\} \\
D(X_5) &= \{4,5\} \\
D(X_6) &= \{4,5\} \\
D(N) &= \{1,2\}
\end{aligned}$$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

$$\begin{array}{lcl} D(X_1) & = & \{1, 2, 3, 4, 5, 6\} \\ D(X_2) & = & \{2, 4\} \\ D(X_3) & = & \{1, 2\} \\ D(X_4) & = & \{1, 2, 3\} \\ D(X_5) & = & \{4, 5\} \\ D(X_6) & = & \{4, 5\} \\ D(N) & = & \{1, 2\} \end{array}$$

A solution: $\textsc{AtMostNValue}([2, 2, 2, 2, 4, 4], 2)$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

$$\begin{array}{lcl} D(X_1) & = & \{1,2,3,4,5,6\} \\ D(X_2) & = & \{2,4\} \\ D(X_3) & = & \{1,2\} \\ D(X_4) & = & \{1,2,3\} \\ D(X_5) & = & \{4,5\} \\ D(X_6) & = & \{4,5\} \\ D(N) & = & \{1,2\} \end{array}$$

$$\begin{array}{lcl} D(X_1) & = & \{1,2,\cancel{3},4,5,\cancel{6}\} \\ D(X_2) & = & \{2,4\} \\ D(X_3) & = & \{1,2\} \\ D(X_4) & = & \{1,2,\cancel{3}\} \\ D(X_5) & = & \{4,5\} \\ D(X_6) & = & \{4,5\} \\ D(N) & = & \{\cancel{1},2\} \end{array}$$

A solution: $\textsc{AtMostNValue}([2,2,2,2,4,4], 2)$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

$$\begin{aligned}
D(X_1) &= \{1,2,3,4,5,6\} \\
D(X_2) &= \{2,4\} \\
D(X_3) &= \boxed{\{1,2\}} \\
D(X_4) &= \{1,2,3\} \\
D(X_5) &= \boxed{\{4,5\}} \\
D(X_6) &= \{4,5\} \\
D(N) &= \{1,2\}
\end{aligned}
\qquad
\begin{aligned}
D(X_1) &= \{1,2,\cancel{3},4,5,\cancel{6}\} \\
D(X_2) &= \{2,4\} \\
D(X_3) &= \{1,2\} \\
D(X_4) &= \{1,2,\cancel{3}\} \\
D(X_5) &= \{4,5\} \\
D(X_6) &= \{4,5\} \\
D(N) &= \{\cancel{1},2\}
\end{aligned}$$

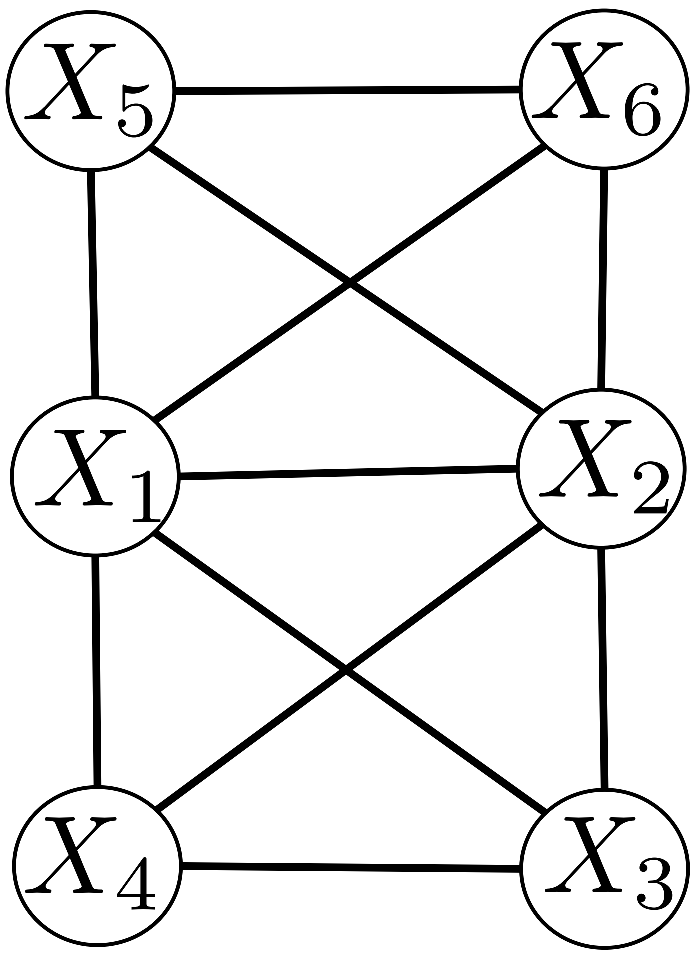
$$\boxed{D(X_3) \cap D(X_5) = \emptyset}$$

A solution: $\textsc{AtMostNValue}([2,2,2,2,4,4], 2)$

AtMostNValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N



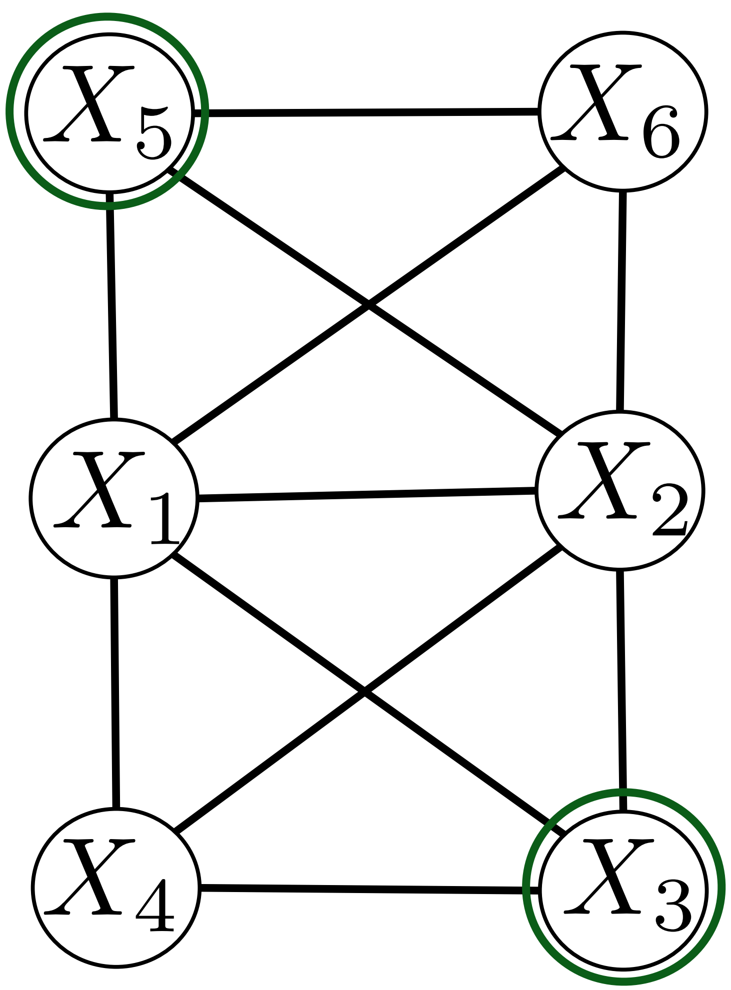
Intersection graph of the domains

$D(X_1)$ = {1, 2, 3, 4, 5, 6}
$D(X_2)$ = {2, 4}
$D(X_3)$ = {1, 2}
$D(X_4)$ = {1, 2, 3}
$D(X_5)$ = {4, 5}
$D(X_6)$ = {4, 5}
$D(N)$ = {1, 2}

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N



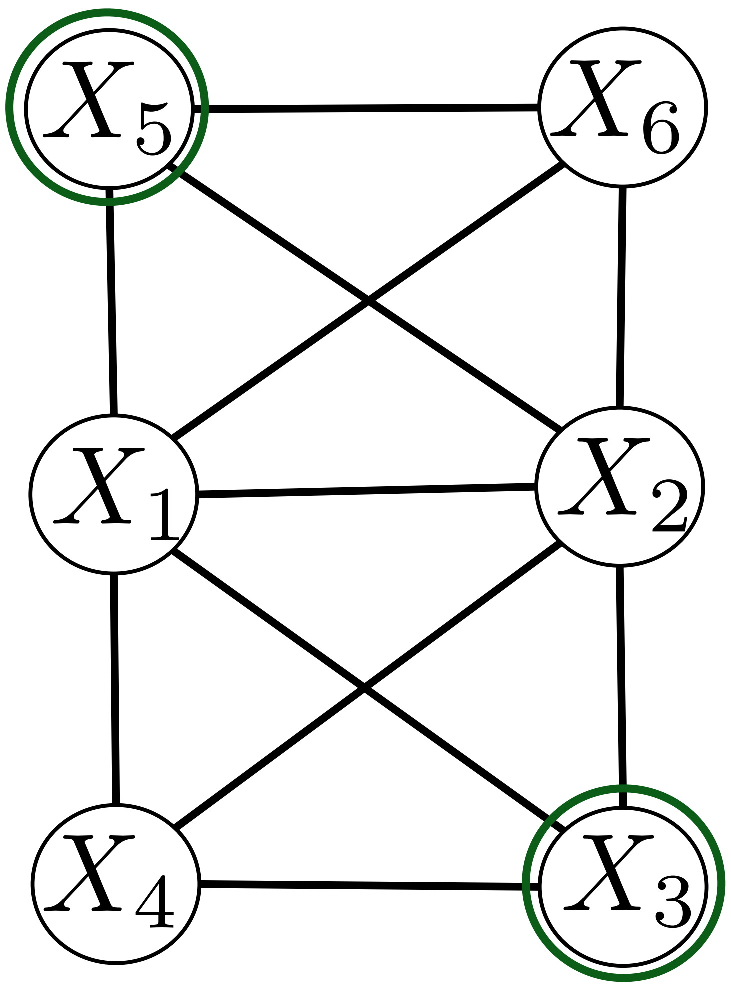
$D(X_1)$ = {1, 2, 3, 4, 5, 6}
$D(X_2)$ = {2, 4}
$D(X_3)$ = {1, 2}
$D(X_4)$ = {1, 2, 3}
$D(X_5)$ = {4, 5}
$D(X_6)$ = {4, 5}
$D(N)$ = {1, 2}

A support of the lower bound of N= an independent set

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N



$D(X_1)$ = {1, 2, 3, 4, 5, 6}
$D(X_2)$ = {2, 4}
$D(X_3)$ = {1, 2}
$D(X_4)$ = {1, 2, 3}
$D(X_5)$ = {4, 5}
$D(X_6)$ = {4, 5}
$D(N)$ = {1, 2}

Remove all values except {1,2,4,5} since $D(X_5) \cup D(X_3) = \{1, 2, 4, 5\}$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

- Enforcing Generalized-Arc-Consistency is NP-Hard
- Filtering algorithm can be based on:
  - Greedy computation of independent sets
  - Cost-based filtering with Lagrangian relaxation
  - LP Reduced-costs

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

- Enforcing Generalized-Arc-Consistency is NP-Hard
- Filtering algorithm can be based on:
  - Greedy computation of independent sets

    [Hebrard et al. 2006]
  - Cost-based filtering with Lagrangian relaxation
  - LP Reduced-costs

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_6], N)$$

Enforce the number of distinct values appearing in the set X to be at most N

- Enforcing Generalized-Arc-Consistency is NP-Hard
- Filtering algorithm can be based on:
  - Greedy computation of independent sets

    [Hebrard et al. 2006]
  - Cost-based filtering with Lagrangian relaxation

    [Cambazard et al. 2015]
  - LP Reduced-costs

At**Most**NValue

- However we cannot express reasonings on mandatory values

At**Most**NValue

- However we cannot express reasonings on mandatory values

Example: $\textsc{AtMostNValue}([X_1, X_2, X_3], N)$

$$
\begin{array}{lcl}
D(X_1) & = & \{1,2\} \\
D(X_2) & = & \{2,3\} \\
D(X_3) & = & \{2,4\} \\
D(N) & = & \{2\}
\end{array}
$$

At**Most**NValue

- However we cannot express reasonings on mandatory values

Example: $\textsc{AtMostNValue}([X_1, X_2, X_3], N)$

$$
\begin{aligned}
D(X_1) &= \{1, 2\} \\
D(X_2) &= \{2, 3\} \\
D(X_3) &= \{2, 4\} \\
D(N) &= \{2\}
\end{aligned}
$$

How to propagate the fact that value 2 is mandatory ?

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_n], [Y_1, \ldots, Y_m], N)$$

$$Y_j \in \{0, 1\} : \text{ value j occurs at least once}$$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_n], [Y_1, \ldots, Y_m], N)$$

$$Y_j \in \{0, 1\} : \text{ value j occurs at least once}$$

Express reasonings on mandatory values

$$\begin{array}{lcl} D(X_1) & = & \{1,2\} \\ D(X_2) & = & \{2,3\} \\ D(X_3) & = & \{2,4\} \\ D(N) & = & \{2\} \end{array} \qquad \begin{array}{lcl} D(Y_1) & = & \{0,1\} \\ D(Y_2) & = & \{0,1\} \\ D(Y_3) & = & \{0,1\} \\ D(Y_4) & = & \{0,1\} \end{array}$$

At**Most**NValue

$$\text{AtMostNValue}([X_1, \ldots, X_n], [Y_1, \ldots, Y_m], N)$$

$$Y_j \in \{0, 1\} : \text{ value j occurs at least once}$$

➡ Express reasonings on mandatory values

$$\begin{array}{lcl} D(X_1) & = & \{1,2\} \\ D(X_2) & = & \{2,3\} \\ D(X_3) & = & \{2,4\} \\ D(N) & = & \{2\} \end{array} \qquad \begin{array}{lcl} D(Y_1) & = & \{0,1\} \\ D(Y_2) & = & \{\cancel{0},1\} \\ D(Y_3) & = & \{0,1\} \\ D(Y_4) & = & \{0,1\} \end{array}$$

At**Most**NValue

$$\textsc{AtMostNValue}([X_1, \ldots, X_n], [Y_1, \ldots, Y_m], N)$$

$$Y_j \in \{0, 1\} : \text{ value j occurs at least once}$$

➡ Express reasonings on mandatory values

$$\begin{array}{lclclcl} D(X_1) & = & \{1,2\} & & D(Y_1) & = & \{0,1\} \\ D(X_2) & = & \{2,3\} & & D(Y_2) & = & \{\cancel{0},1\} \\ D(X_3) & = & \{2,4\} & & D(Y_3) & = & \{0,1\} \\ D(N) & = & \{2\} & & D(Y_4) & = & \{0,1\} \end{array}$$

➡ $Y_2 = 1$

Note that domains of X cannot be filtered…

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{1, 2\} \quad D(X_2) = \{2, 3\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{1, 2\}$$

Reduced cost based filtering

Consider the following example:

$D(X_1) = \{1, 2\} \;\; D(X_2) = \{2, 3\} \;\; D(X_3) = \{4, 5\} \;\; D(N) = \{\cancel{1}, 2\}$

Reduced cost based filtering

Consider the following example:

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

The exact lower bound of N can be computed with the following MIP:

$$\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & & \\
& y_1 & +y_2 & & & & \geq & 1 & \text{(Domain of } X_1\text{)} \\
& & y_2 & +y_3 & & & \geq & 1 & \text{(Domain of } X_2\text{)} \\
& & & & y_4 & +y_5 & \geq & 1 & \text{(Domain of } X_3\text{)} \\
& y_i & & & & & \in & \{0,1\} &
\end{array}$$

$y_i \in \{0, 1\}$: do we use value i ?

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Consider the linear relaxation:

$$\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & y_i & & & & & \geq & 0
\end{array}$$

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Consider the linear relaxation:

$$\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & y_i & & & & & \geq & 0
\end{array}$$

Notice that we don't need to state $y_i \leq 1$

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Consider the linear relaxation:

$$\begin{array}{lcccccl}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & \\
& y_1 & +y_2 & & & & \geq 1 \\
& & y_2 & +y_3 & & & \geq 1 \\
& & & & y_4 & +y_5 & \geq 1 \\
& y_i & & & & & \geq 0
\end{array}$$

Notice that we don't need to state $y_i \leq 1$

First of all, we get $z^* = 2$

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Consider the linear relaxation:

$$\begin{array}{lccccccc}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & y_i & & & & & \geq & 0
\end{array}$$

Notice that we don't need to state $y_i \leq 1$

First of all, we get $z^* = 2$

Reduced cost based filtering

Consider the following example:

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Consider the linear relaxation:

$$\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& y_i & & & & & \geq & 0
\end{array}$$

Notice that we don't need to state $y_i \leq 1$

First of all, we get $z^* = 2$

$$y^* = (0, \overset{y_2^*}{1}, 0, \overset{y_4^*}{1}, 0)$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & y_i & & & & & \geq & 0
\end{array}
$$

(D)

$$
\begin{array}{llllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & & \\
 & \lambda_1 & & & \leq & 1 \\
 & \lambda_1 & +\lambda_2 & & \leq & 1 \\
 & & \lambda_2 & & \leq & 1 \\
 & & & \lambda_3 & \leq & 1 \\
 & & & \lambda_3 & \leq & 1 \\
 & \lambda_j & & & \geq & 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & & \\
& y_1 & +y_2 & & & & \geq & 1 & (\lambda_1) \\
& & y_2 & +y_3 & & & \geq & 1 & (\lambda_2) \\
& & & & y_4 & +y_5 & \geq & 1 & (\lambda_3) \\
& y_i & & & & & \geq & 0 &
\end{array}
$$

(D)

$$
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & & & \\
& \lambda_1 & & & \leq & 1 & (y_1) \\
& \lambda_1 & +\lambda_2 & & \leq & 1 & (y_2) \\
& & \lambda_2 & & \leq & 1 & (y_3) \\
& & & \lambda_3 & \leq & 1 & (y_4) \\
& & & \lambda_3 & \leq & 1 & (y_5) \\
& \lambda_j & & & \geq & 0 &
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & & \\
& y_1 & +y_2 & & & & \geq & 1 & (\lambda_1) \\
& & y_2 & +y_3 & & & \geq & 1 & (\lambda_2) \\
& & & & y_4 & +y_5 & \geq & 1 & (\lambda_3) \\
& y_i & & & & & \geq & 0 &
\end{array}
$$

(D)

$$
\begin{array}{llllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & & \\
& \lambda_1 & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \leq & 1 \\
& & \lambda_2 & & \leq & 1 \\
& & & \lambda_3 & \leq & 1 \\
& & & \lambda_3 & \leq & 1 \\
& \lambda_j & & & \geq & 0
\end{array}
$$

	y^*
(y_1)	(0)
(y_2)	(1)
(y_3)	(0)
(y_4)	(1)
(y_5)	(0)

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$

(P)

$$
\begin{array}{lcccccrl}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& y_i & & & & & \geq & 0
\end{array}
$$

$$
\boxed{\begin{array}{cc}
 & \lambda^* \\
(\lambda_1) & (0) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1)
\end{array}}
$$

(D)

$$
\begin{array}{lcccrl}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & & \\
& \lambda_1 & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \leq & 1 \\
& & \lambda_2 & & \leq & 1 \\
& & & \lambda_3 & \leq & 1 \\
& & & \lambda_3 & \leq & 1 \\
& \lambda_j & & & \geq & 0
\end{array}
$$

$$
\boxed{\begin{array}{cc}
 & y^* \\
(y_1) & (0) \\
(y_2) & (1) \\
(y_3) & (0) \\
(y_4) & (1) \\
(y_5) & (0)
\end{array}}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to filter value 1 from X_1 :

(P)

$$
\begin{array}{lcccccl}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & \\
& y_1 & +y_2 & & & & \geq 1 \\
& & y_2 & +y_3 & & & \geq 1 \\
& & & & y_4 & +y_5 & \geq 1 \\
& & & & & & \\
& y_i & & & & & \geq 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to filter value 1 from X_1 :

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& \boxed{y_1} & & & & & \boxed{\geq} & \boxed{1} \\
& y_i & & & & & \geq & 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to filter value 1 from X_1 :

$$
\text{(P)} \quad
\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& \boxed{y_1} & & & & & \geq & 1 \\
& y_i & & & & & \geq & 0
\end{array}
$$

$$
\boxed{
\begin{array}{ll}
& \lambda^* \\
(\lambda_1) & (0) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\gamma) & (?)
\end{array}
}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to filter value 1 from X_1 :

(P)

$$
\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& \boxed{y_1} & & & & & \geq & 1 \\
& y_i & & & & & \geq & 0
\end{array}
$$

$$
\boxed{
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (0) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\gamma) & (?)
\end{array}
}
$$

(D)

$$
\begin{array}{llllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\gamma} & & \\
& \lambda_1 & & & \boxed{+\gamma} & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to filter value 1 from X_1 :

(P)

$$
\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& \boxed{y_1} & & & & & \boxed{\geq} & \boxed{1} \\
& y_i & & & & & \geq & 0
\end{array}
$$

$$
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (0) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\gamma) & (?)
\end{array}
$$

(D)

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\gamma} & & \\
& \lambda_1 & & & \boxed{+\gamma} & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

We can build a dual solution by setting γ greedily to $(1 - \lambda_1^*)$

Note that we are not solving the LP again

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

$\lambda^* = (0, 1, 1)$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

$$\lambda^* = (0, 1, 1)$$

We can build a **feasible dual** solution by setting γ to $(1 - \lambda_1^*)$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}$$

$$\lambda^* = (0, 1, 1)$$

We can build a **feasible dual** solution by setting γ to $(1 - \lambda_1^*)$

Thus $z^* + (1 - \lambda_1^*)$ is a lower bound of the modified problem

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}$$

$$\lambda^* = (0, 1, 1)$$

We can build a **feasible dual** solution by setting γ to $(1 - \lambda_1^*)$

Thus $z^* + (1 - \lambda_1^*)$ is a lower bound of the modified problem

Upper bound of N

So $z^* + (1 - \lambda_1^*) > 2 \implies y_1 \neq 1 \ (X_1 \neq 1)$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

$\lambda^* = (0, 1, 1)$

We can build a **feasible dual** solution by setting γ to $(1 - \lambda_1^*)$

Thus $z^* + (1 - \lambda_1^*)$ is a lower bound of the modified problem

Upper bound of N

So $\quad z^* + \underbrace{(1 - \lambda_1^*)}_{\text{Reduced cost of } y_1} > 2 \implies y_1 \neq 1 \;\; (X_1 \neq 1)$

Reduced cost of y_1 (Slack of the dual constraint)

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

$$\lambda^* = (0, 1, 1)$$

So $z^* + rc(y_1) > \overline{z} \implies y_1 \neq 1 \ (X_1 \neq 1)$

Reduced cost of y_1 : $rc(y_1) = (1 - \lambda_1^*) = 1$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}$$

$$\lambda^* = (0, 1, 1)$$

So $z^* + rc(y_1) > \overline{z} \implies y_1 \neq 1 \ (X_1 \neq 1)$

Reduced cost of y_1 : $rc(y_1) = (1 - \lambda_1^*) = 1$

Reduced cost of y_3 : $rc(y_3) = (1 - \lambda_2^*) = (1 - 1) = 0$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

$$\lambda^* = (0, 1, 1)$$

So $z^* + rc(y_1) > \overline{z} \implies y_1 \neq 1 \ (X_1 \neq 1)$

Reduced cost of y_1 : $rc(y_1) = (1 - \lambda_1^*) = 1$

Reduced cost of y_3 : $rc(y_3) = (1 - \lambda_2^*) = (1 - 1) = 0$

We cannot filter value 3 using this dual solution

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

But consider

$$\lambda^* = (1, 0, 1)$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

But consider

$$\lambda^* = (1, 0, 1)$$

Reduced cost of $y_1 : rc(y_1) = (1 - \lambda_1^*) = 0$

Reduced cost based filtering

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

$$\begin{array}{lccccll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}$$

But consider

$$\lambda^* = (1, 0, 1)$$

Reduced cost of $y_1 : rc(y_1) = (1 - \lambda_1^*) = 0$

Reduced cost of $y_3 : rc(y_3) = (1 - \lambda_2^*) = 1$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

$$
\begin{array}{lcccccc}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\gamma & & \\
& \lambda_1 & & & +\gamma & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j, & \gamma & & & \geq & 0
\end{array}
$$

But consider

$$\lambda^* = (1, 0, 1)$$

Reduced cost of $y_1 : rc(y_1) = (1 - \lambda_1^*) = 0$

Reduced cost of $y_3 : rc(y_3) = (1 - \lambda_2^*) = 1$

Value 3 is now filtered but value 1 is not filtered anymore

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

- We are filtering the upper bound of y_1 or y_3

$$z^* + rc(y_i) > \overline{z} \implies y_i \neq 1$$

Reduced cost based filtering

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

- We are filtering the upper bound of y_1 or y_3

$$z^* + rc(y_i) > \overline{z} \implies y_i \neq 1$$

- But if y_i is in the optimal LP solution (the basis), its reduced cost is 0

- This is due to the complementary slackness theorem:

  Either the variable is 0, or the slack of the dual constraint (i.e. the reduced cost) is 0, or both

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

- We are filtering the upper bound of y_1 or y_3

$$z^* + rc(y_i) > \overline{z} \implies y_i \neq 1$$

- But if y_i is in the optimal LP solution (the basis), its reduced cost is 0

- This is due to the complementary slackness theorem:

  Either the variable is 0, or the slack of the dual constraint (i.e. the reduced cost) is 0, or both

- How to filter the lower bound of y_i?

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

Reduced cost based filtering

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

Let's try to prove that value 2 is mandatory i.e. filter the lower bound of y_2 : $y_2 \neq 0$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory i.e. filter the lower bound of y_2 : $y_2 \neq 0$

Filter Upper bound $y_1 \neq 1$

1. Solve the **original** LP optimally
2. Use the optimal dual solution, to build **a feasible** dual solution to the problem that **would** include $y_1 \geq 1$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory i.e. filter the lower bound of y_2 : $y_2 \neq 0$

Filter Upper bound $y_1 \neq 1$

1. Solve the **original** LP optimally
2. Use the optimal dual solution, to build **a feasible** dual solution to the problem that **would** include $y_1 \geq 1$

Filter Lower bound $y_2 \neq 0$

1. **Include** in the original LP the constraint $y_2 \leq 1$
2. Solve the **modified** problem and perform **sensibility analysis** on the right hand side of $y_2 \leq 1$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

$$
(P) \quad \begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& & \boxed{y_2} & & & & \boxed{\leq} & \boxed{1} \\
& y_i & & & & & \geq & 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

$$
(P) \quad
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& & \boxed{y_2} & & & & \boxed{\leq} & \boxed{1} \\
& y_i & & & & & \geq & 0
\end{array}
$$

Note that the upper-bound constraint is now added **before** solving the LP

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

$$
(P) \quad
\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & & \boxed{y_2} & & & & \leq & 1 \\
 & y_i & & & & & \geq & 0
\end{array}
$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

Note that the upper-bound constraint is now added **before** solving the LP

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& & \boxed{y_2} & & & & \boxed{\leq} & \boxed{1} \\
& y_i & & & & & \geq & 0
\end{array}
$$

$$
\begin{array}{ll}
& \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

(D)

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\theta} & & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \boxed{\theta} & & & & \boxed{\leq} & \boxed{0}
\end{array}
$$

Note that the upper-bound constraint is now added **before** solving the LP

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(P)

$$
\begin{array}{llllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & & \boxed{y_2} & & & & \leq & \boxed{1-\epsilon} \\
 & y_i & & & & & \geq & 0
\end{array}
$$

$$
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

(D)

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\theta} & & \\
 & \lambda_1 & & & & \leq & 1 \\
 & \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
 & & \lambda_2 & & & \leq & 1 \\
 & & & \lambda_3 & & \leq & 1 \\
 & & & \lambda_3 & & \leq & 1 \\
 & \lambda_j & & & & \geq & 0 \\
 & \boxed{\theta} & & & & \leq & 0
\end{array}
$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(P)

$$\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& & \boxed{y_2} & & & & \leq & \boxed{1-\epsilon} \\
& y_i & & & & & \geq & 0
\end{array}$$

$$\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}$$

(D)

$$\begin{array}{lllllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\theta} & \boxed{-\epsilon\theta} & & \\
& \lambda_1 & & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & & \leq & 1 \\
& & \lambda_2 & & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& \lambda_j & & & & & \geq & 0 \\
& \boxed{\theta} & & & & & \leq & 0
\end{array}$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
 & y_1 & +y_2 & & & & \geq & 1 \\
 & & y_2 & +y_3 & & & \geq & 1 \\
 & & & & y_4 & +y_5 & \geq & 1 \\
 & & \boxed{y_2} & & & & \leq & \boxed{1-\epsilon} \\
 & y_i & & & & & \geq & 0
\end{array}
$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

(D)

$$
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\theta} & \boxed{-\epsilon\theta} & & \\
 & \lambda_1 & & & & \leq & 1 \\
 & \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
 & & \lambda_2 & & & \leq & 1 \\
 & & & \lambda_3 & & \leq & 1 \\
 & & & \lambda_3 & & \leq & 1 \\
 & \lambda_j & & & & \geq & 0 \\
 & \boxed{\theta} & & & & \leq & 0
\end{array}
$$

Decreasing the upper-bound by ϵ increases the objective of **at least** $-\epsilon\theta^*$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(P)

$$
\begin{array}{lllllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & +y_5 & & \\
& y_1 & +y_2 & & & & \geq & 1 \\
& & y_2 & +y_3 & & & \geq & 1 \\
& & & & y_4 & +y_5 & \geq & 1 \\
& & \boxed{y_2} & & & & \leq & \boxed{1-\epsilon} \\
& y_i & & & & & \geq & 0
\end{array}
$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

(D)

$$
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & \boxed{+\theta} & \boxed{-\epsilon\theta} & & \\
& \lambda_1 & & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & & \leq & 1 \\
& & \lambda_2 & & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& \lambda_j & & & & & \geq & 0 \\
& \boxed{\theta} & & & & & \leq & 0
\end{array}
$$

Decreasing the upper-bound by ϵ increases the objective of **at least** $-\epsilon\theta^*$

Feasibility of the dual solution is not affected by the change !

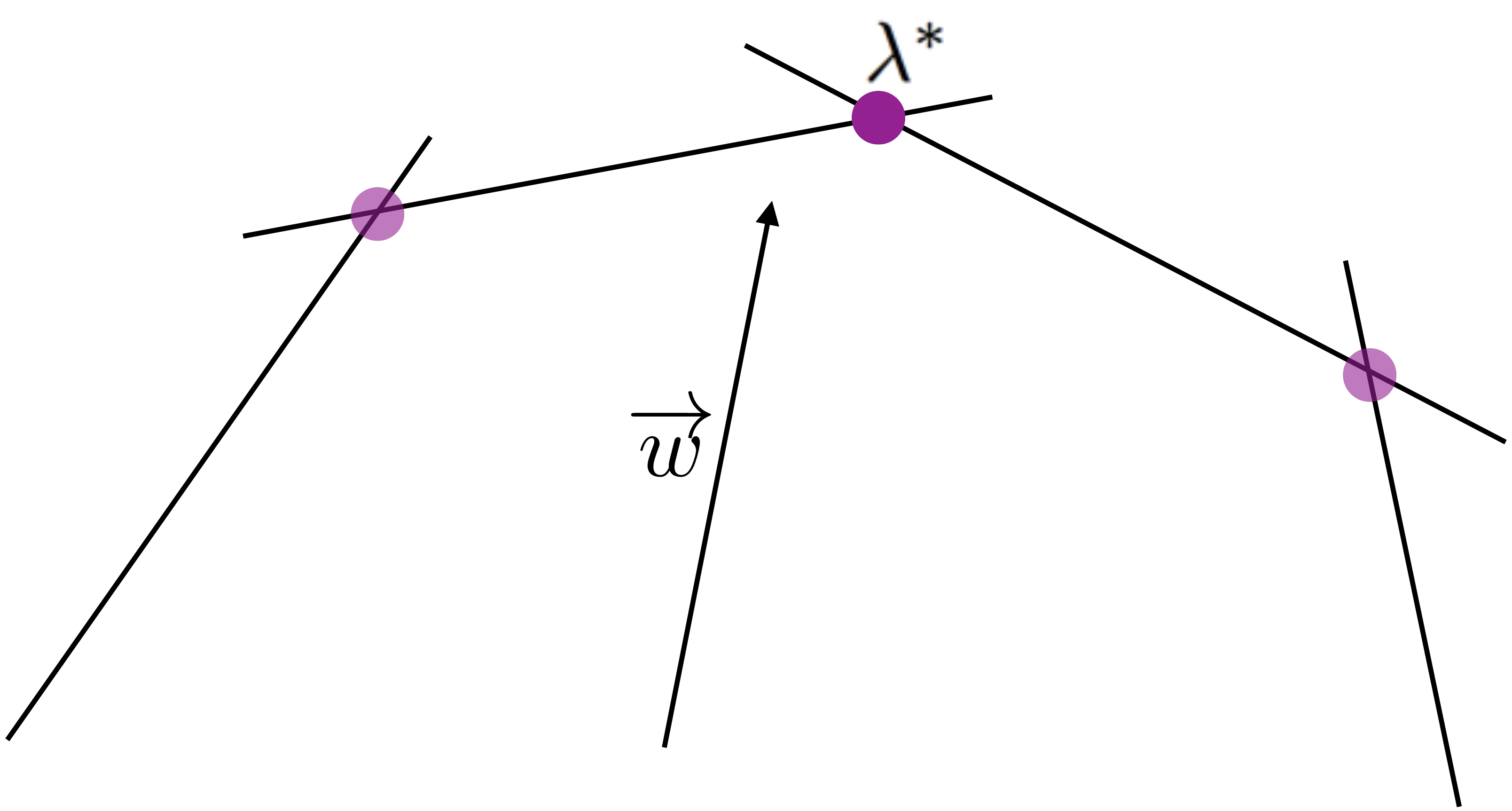
Reduced cost based filtering

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

Reduced cost based filtering

$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \le 1$$

$$\boxed{\begin{array}{ll} & \lambda^* \\ (\lambda_1) & (1) \\ (\lambda_2) & (1) \\ (\lambda_3) & (1) \\ (\theta) & (-1) \end{array}}$$

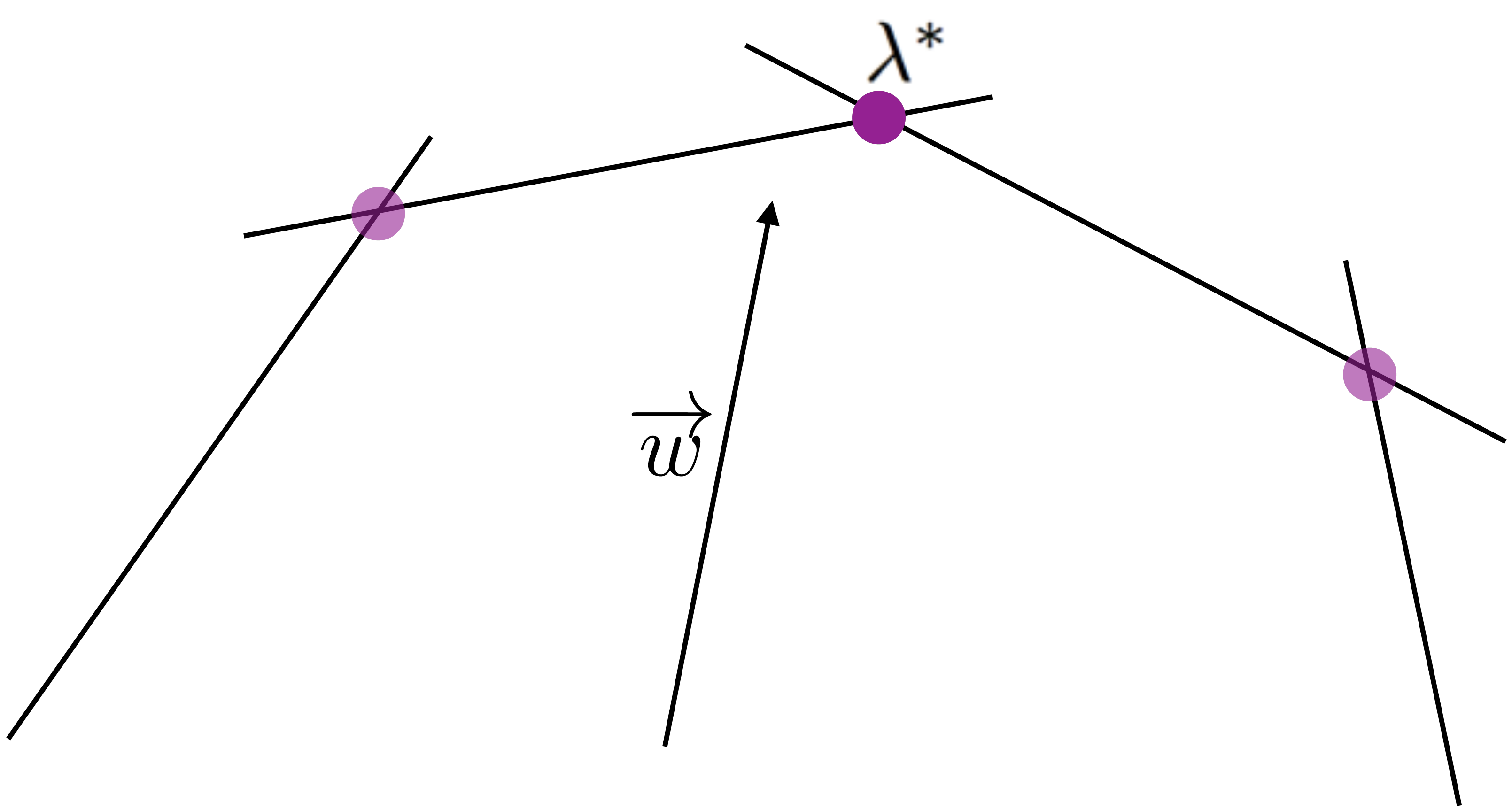


Reduced cost based filtering

$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \leq 1$$

$$\text{Max } w' = \lambda_1 + \lambda_2 + \lambda_3 + (1-\epsilon)\theta \qquad y_2 \leq \boxed{(1-\epsilon)}$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

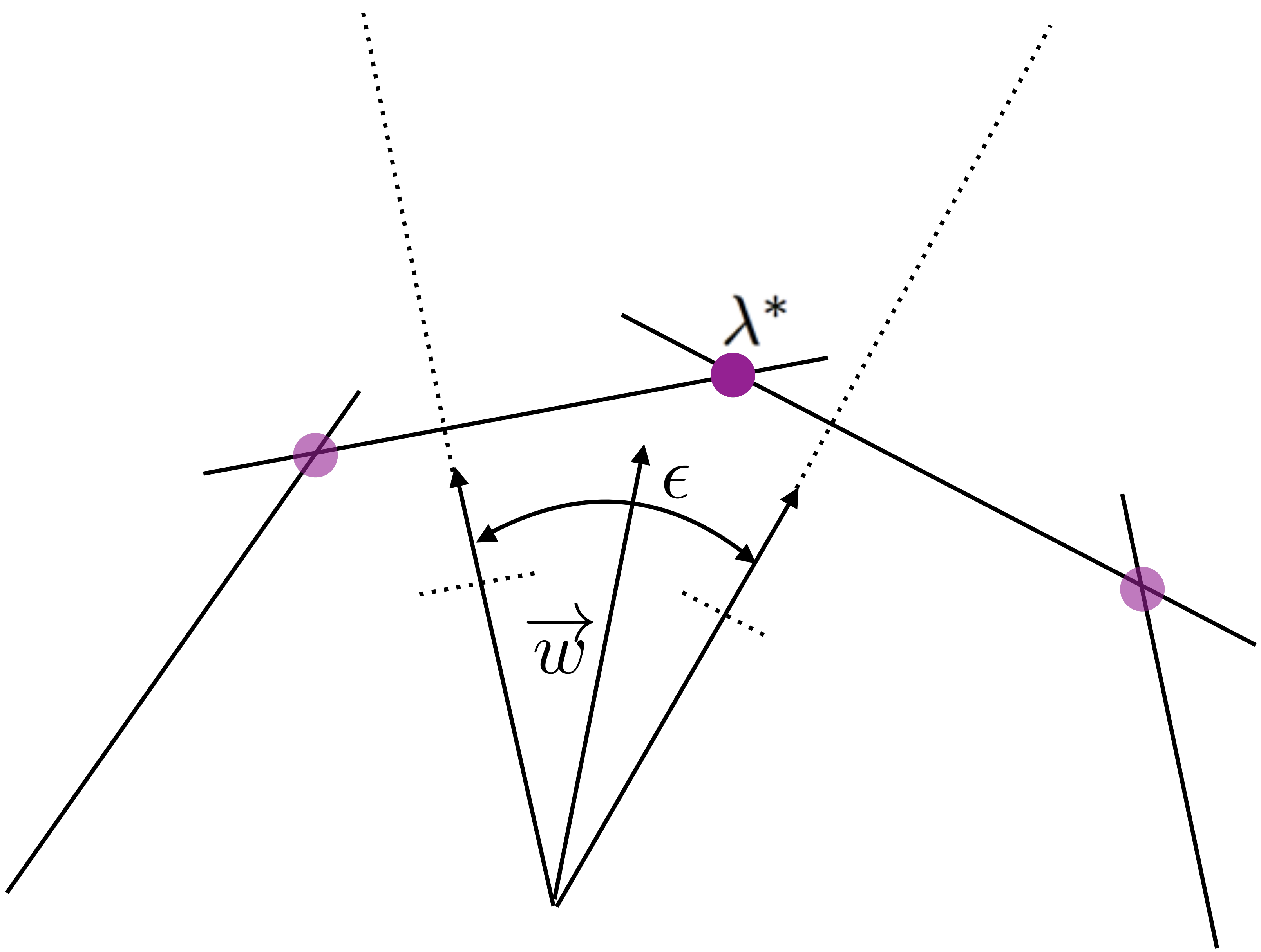


Reduced cost based filtering

$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \leq 1$$

$$\text{Max } w' = \lambda_1 + \lambda_2 + \lambda_3 + (1-\epsilon)\theta \qquad y_2 \leq \boxed{(1-\epsilon)}$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)



Reduced cost based filtering

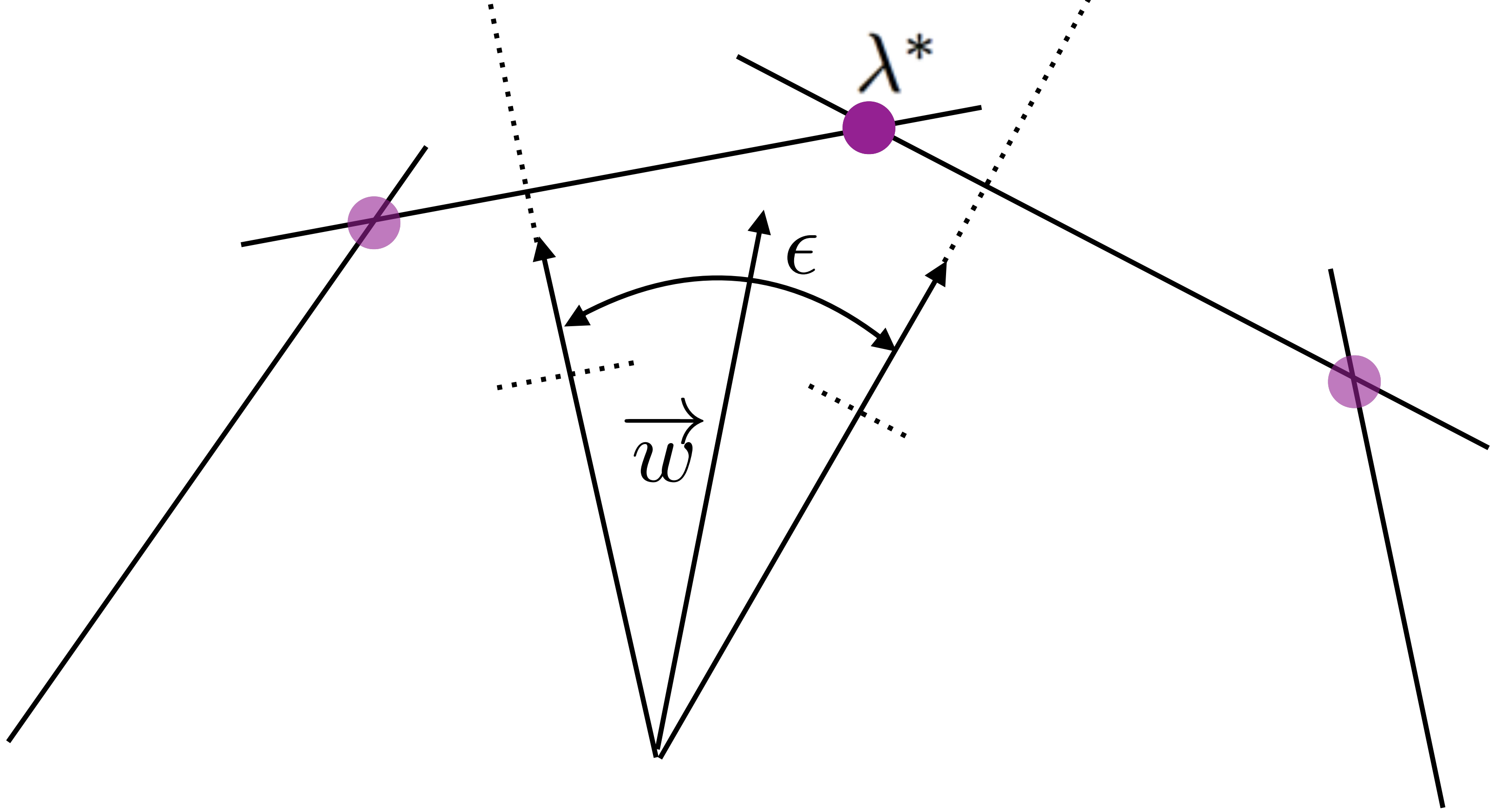
$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \leq 1$$

$$\text{Max } w' = \lambda_1 + \lambda_2 + \lambda_3 + (1-\epsilon)\theta \qquad y_2 \leq \boxed{(1-\epsilon)}$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

$$w'^* = w^* - \epsilon\theta^*$$

exact increase



Reduced cost based filtering

$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \leq 1$$

$$\text{Max } w' = \lambda_1 + \lambda_2 + \lambda_3 + (1-\epsilon)\theta \qquad y_2 \leq \boxed{(1-\epsilon)}$$

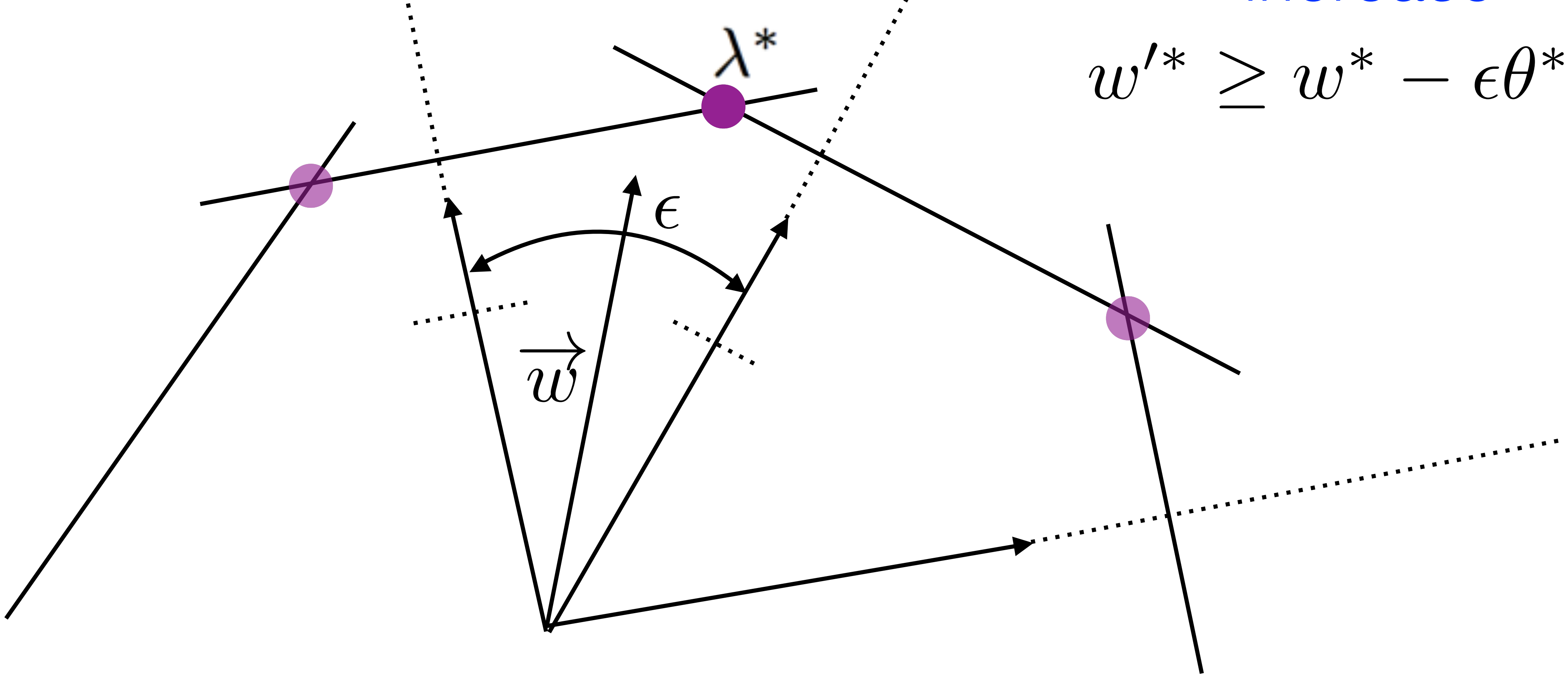
	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

$$w'^* = w^* - \epsilon\theta^*$$

exact increase

lower bound of the increase

$$w'^* \geq w^* - \epsilon\theta^*$$



Reduced cost based filtering

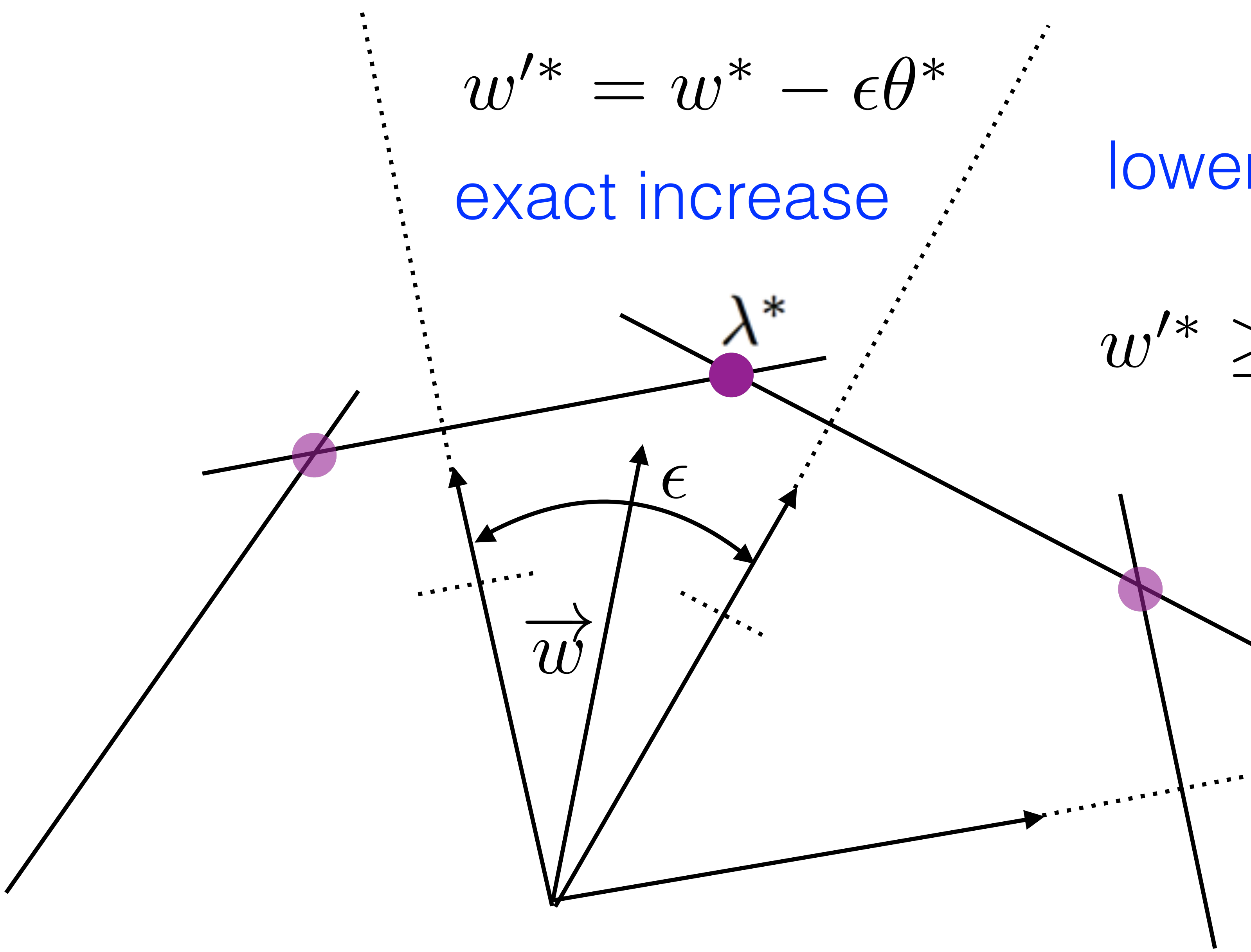
$$\text{Max } w = \lambda_1 + \lambda_2 + \lambda_3 + \theta \qquad y_2 \leq 1$$

$$\text{Max } w' = \lambda_1 + \lambda_2 + \lambda_3 + (1-\epsilon)\theta \qquad y_2 \leq \boxed{(1-\epsilon)}$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

$$w'^* = w^* - \epsilon\theta^*$$

exact increase



lower bound of the increase

$$w'^* \geq w^* - \epsilon\theta^*$$

Decreasing the upper-bound by ϵ increases the objective of **at least** $-\epsilon\theta^*$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \theta & & & & \leq & 0
\end{array}$$

$$\boxed{\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}}$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \theta & & & & \leq & 0
\end{array}
$$

$$
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

So, (by sensitivity analysis) if we forbid value 2 i.e. if we set the upper bound of y_2 to 0, the increase is at least of $-\theta^*$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \boxed{\theta} & & & & \leq & 0
\end{array}
$$

$$
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

So, (by sensitivity analysis) if we forbid value 2 i.e. if we set the upper bound of y_2 to 0, the increase is at least of $-\theta^*$

$$z^* - \theta^* = 2 - (-1) > \overline{z} = 2 \implies y_2 \neq 0 \ (Y_2 = 1)$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$
\begin{array}{lllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \theta & & & & \leq & 0
\end{array}
$$

$$
\begin{array}{ll}
& \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

If we ignore θ and compute the reduced cost of y_2 :

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

$$
(D) \quad
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & & \\
& \lambda_1 & & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & & \leq & 1 \\
& & \lambda_2 & & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& & & \lambda_3 & & & \leq & 1 \\
& \lambda_j & & & & & \geq & 0 \\
& \theta & & & & & \leq & 0
\end{array}
$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

If we ignore θ and compute the reduced cost of y_2 :

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let’s try to prove that value 2 is mandatory :

(D)

$$
\begin{array}{lccccccc}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & & \\
 & \lambda_1 & & & & & \leq & 1 \\
 & \lambda_1 & +\lambda_2 & & \boxed{+\theta} & & \leq & 1 \\
 & & \lambda_2 & & & & \leq & 1 \\
 & & & \lambda_3 & & & \leq & 1 \\
 & & & \lambda_3 & & & \leq & 1 \\
 & \lambda_j & & & & & \geq & 0 \\
 & \theta & & & & & \leq & 0
\end{array}
$$

$$
\begin{array}{ll}
 & \lambda^* \\
(\lambda_1) & (1) \\
(\lambda_2) & (1) \\
(\lambda_3) & (1) \\
(\theta) & (-1)
\end{array}
$$

If we ignore θ and compute the reduced cost of y_2 :

$$rc(y_2) = 1 - \lambda_1^* - \lambda_2^* = -1$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$\begin{array}{llllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} \\
& \lambda_1 & & & & \leq \; 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq \; 1 \\
& & \lambda_2 & & & \leq \; 1 \\
& & & \lambda_3 & & \leq \; 1 \\
& & & \lambda_3 & & \leq \; 1 \\
& \lambda_j & & & & \geq \; 0 \\
& \boxed{\theta & & & & \leq \; 0}
\end{array}$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

If we ignore θ and compute the reduced cost of y_2 :

$$rc(y_2) = 1 - \lambda_1^* - \lambda_2^* = -1$$

And the filtering rule can be seen as :

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

Let's try to prove that value 2 is mandatory :

(D)

$$
\begin{array}{llllllll}
\text{Max } w = & \lambda_1 & +\lambda_2 & +\lambda_3 & +\theta & \boxed{-\epsilon\theta} & & \\
& \lambda_1 & & & & \leq & 1 \\
& \lambda_1 & +\lambda_2 & & \boxed{+\theta} & \leq & 1 \\
& & \lambda_2 & & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& & & \lambda_3 & & \leq & 1 \\
& \lambda_j & & & & \geq & 0 \\
& \theta & & & & \leq & 0
\end{array}
$$

	λ^*
(λ_1)	(1)
(λ_2)	(1)
(λ_3)	(1)
(θ)	(-1)

If we ignore θ and compute the reduced cost of y_2 :

$$rc(y_2) = 1 - \lambda_1^* - \lambda_2^* = -1$$

And the filtering rule can be seen as :

$$z^* - rc(y_2) > \overline{z} \implies y_2 \neq 0 \ (Y_2 = 1)$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

- To filter the lower bound of y_2 ?

We include the upper bound constraints in the LP: $y_i \leq 1$

And compute the reduced cost by ignoring the dual variables of these constraints

$$z^* - rc(y_i) > \overline{z} \implies y_i \neq 0$$

Reduced cost based filtering

$D(X_1) = \{\cancel{1}, 2\}$ $D(X_2) = \{2, \cancel{3}\}$ $D(X_3) = \{4, 5\}$ $D(N) = \{\cancel{1}, 2\}$

- To filter the lower bound of y_2 ?

  We include the upper bound constraints in the LP: $y_i \leq 1$

  And compute the reduced cost by ignoring the dual variables of these constraints

$$z^* - rc(y_i) > \overline{z} \implies y_i \neq 0$$

- To filter the upper bound of y_1 or y_3

$$z^* + rc(y_i) > \overline{z} \implies y_i \neq 1$$

  But if y_i is in the optimal LP solution (the basis), its reduced cost is 0 (complementary slackness)

Reduced cost based filtering

$$D(X_1) = \{\cancel{1}, 2\} \quad D(X_2) = \{2, \cancel{3}\} \quad D(X_3) = \{4, 5\} \quad D(N) = \{\cancel{1}, 2\}$$

- To filter the lower bound of y_2 ?

$$z^* - rc(y_i) > \overline{z} \implies y_i \neq 0$$

- To filter the upper bound of y_1 or y_3

$$z^* + rc(y_i) > \overline{z} \implies y_i \neq 1$$

In any case, the reduced cost can be interpreted as a lower bound of the variation of the objective function per unit of change of the variable

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

General principles

$$\text{(P)} \quad \begin{array}{lcll} \text{Min } z = & \displaystyle\sum_{i=1}^{n} c_i x_i & & \\ & \displaystyle\sum_{i=1}^{n} a_{ij} x_i & \geq & b_j \quad \forall j = 1, \ldots, m \\ & x_i & \geq & 0 \quad \forall i = 1, \ldots, n \end{array}$$

$$\text{(D)} \quad \begin{array}{lcll} \text{Max } w = & \displaystyle\sum_{j=1}^{m} b_j \lambda_j & & \\ & \displaystyle\sum_{j=1}^{m} a_{ij} \lambda_j & \leq & c_i \quad \forall i = 1, \ldots, n \\ & \lambda_j & \geq & 0 \quad \forall j = 1, \ldots, m \end{array}$$

Consider one variable

$$x_k \in [\underline{x_k}, \overline{x_k}]$$

and suppose the LP is solved with **the simplex algorithm handling bounds** directly

General principles

$$\text{(D)} \quad \begin{array}{lll} \text{Max } w = & \sum_{j=1}^{m} b_j \lambda_j & \\ & \sum_{j=1}^{m} a_{ij}\lambda_j \leq c_i & \forall i = 1, \ldots, n \\ & \lambda_j \geq 0 & \forall j = 1, \ldots, m \end{array}$$

$$x_k \in [\underline{x_k}, \overline{x_k}]$$

Proposition 1 (Reduced cost) *Let x^* and λ^* be a pair of primal and dual feasible solutions of (P) and (D), satisfying the complementary slackness conditions. The reduced cost of variable x_k is denoted $rc(x_k)$ and defined as :*

$$rc(x_k) = c_k - (\sum_{j=1}^{m} a_{kj}\lambda_j^*)$$

1. *If $x_k^* = \underline{x_k}$ then $rc(x_k) \geq 0$*
2. *If $x_k^* = \overline{x_k}$ then $rc(x_k) \leq 0$*
3. *If $\underline{x_k} < x_k^* < \overline{x_k}$ then $rc(x_k) = 0$*

General principles

Proposition 1 (Reduced cost) *Let x^* and λ^* be a pair of primal and dual feasible solutions of (P) and (D), satisfying the complementary slackness conditions. The reduced cost of variable x_k is denoted $rc(x_k)$ and defined as :*

$$rc(x_k) = c_k - (\sum_{j=1}^{m} a_{kj}\lambda_j^*)$$

1. *If $x_k^* = \underline{x_k}$ then $rc(x_k) \geq 0$*
2. *If $x_k^* = \overline{x_k}$ then $rc(x_k) \leq 0$*
3. *If $\underline{x_k} < x_k^* < \overline{x_k}$ then $rc(x_k) = 0$*

Upper bound

$$\boxed{\text{If } rc(x_k) > 0 \text{ then } x_k \leq \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \text{ in any solution of cost less than } \overline{z}}$$

Lower bound

$$\boxed{\text{If } rc(x_k) < 0 \text{ then } x_k \geq \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \text{ in any solution of cost less than } \overline{z}}$$

General principles

Upper bound

$$\boxed{\textit{If } rc(x_k) > 0 \textit{ then } x_k \leq \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \textit{ in any solution of cost less than } \overline{z}}$$

Lower bound

$$\boxed{\textit{If } rc(x_k) < 0 \textit{ then } x_k \geq \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \textit{ in any solution of cost less than } \overline{z}}$$

- In any case, the reduced cost can be interpreted as **a lower bound** of the **increase of the objective** per unit of change of x_k

General principles

Upper bound

$$\boxed{\textit{If } rc(x_k) > 0 \textit{ then } x_k \leq \left\lfloor \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rfloor \textit{ in any solution of cost less than } \overline{z}}$$

Lower bound

$$\boxed{\textit{If } rc(x_k) < 0 \textit{ then } x_k \geq \left\lceil \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rceil \textit{ in any solution of cost less than } \overline{z}}$$

- In any case, the reduced cost can be interpreted as **a lower bound** of the **increase of the objective** per unit of change of x_k

General principles

Upper bound

$$\boxed{If\ rc(x_k) > 0\ then\ x_k \leq \left\lfloor \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rfloor in\ any\ solution\ of\ cost\ less\ than\ \overline{z}}$$

Lower bound

$$\boxed{If\ rc(x_k) < 0\ then\ x_k \geq \left\lceil \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rceil in\ any\ solution\ of\ cost\ less\ than\ \overline{z}}$$

- In any case, the reduced cost can be interpreted as **a lower bound** of the **increase of the objective** per unit of change of x_k
- Floor and ceil if x are integers in the original problem

General principles

Upper bound

$$\boxed{\mathit{If}\ rc(x_k) > 0\ \mathit{then}\ x_k \leq \left\lfloor \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rfloor \mathit{in\ any\ solution\ of\ cost\ less\ than}\ \overline{z}}$$

Lower bound

$$\boxed{\mathit{If}\ rc(x_k) < 0\ \mathit{then}\ x_k \geq \left\lceil \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rceil \mathit{in\ any\ solution\ of\ cost\ less\ than}\ \overline{z}}$$

- In any case, the reduced cost can be interpreted as **a lower bound** of the **increase of the objective** per unit of change of x_k
- Floor and ceil if x are integers in the original problem
- Technique referred to as **Variable Fixing**

General principles

Upper bound

$$\boxed{\textit{If } rc(x_k) > 0 \textit{ then } x_k \leq \left\lfloor \underline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rfloor \textit{ in any solution of cost less than } \overline{z}}$$

Lower bound

$$\boxed{\textit{If } rc(x_k) < 0 \textit{ then } x_k \geq \left\lceil \overline{x_k} + \frac{(\overline{z} - z^*)}{rc(x_k)} \right\rceil \textit{ in any solution of cost less than } \overline{z}}$$

- In any case, the reduced cost can be interpreted as **a lower bound** of the **increase of the objective** per unit of change of x_k

- Floor and ceil if x are integers in the original problem

- Technique referred to as **Variable Fixing**

  [Nemhauser and Wolsey. Integer and Combinatorial Optimization. 1988] ?

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

At**Most**NValue

At**Most**NValue

$$D(X_1) = \{1, 2\},\ \ D(X_2) = \{2, 3\},\ \ D(X_3) = \{2, 4\},\ \ D(N) = \{2\}$$
$$D(Y_1) = \{0, 1\},\ \ D(Y_2) = \{\cancel{0}, 1\},\ \ D(Y_3) = \{2, 4\},\ \ D(Y_4) = \{0, 1\}$$

At**Most**NValue

$$D(X_1) = \{1,2\},\ \ D(X_2) = \{2,3\},\ \ D(X_3) = \{2,4\},\ \ D(N) = \{2\}$$
$$D(Y_1) = \{0,1\},\ \ D(Y_2) = \{\cancel{0},1\},\ \ D(Y_3) = \{2,4\},\ \ D(Y_4) = \{0,1\}$$

$$\begin{array}{lccccll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & & \\
& y_1 & +y_2 & & & \geq & 1 \\
& & y_2 & +y_3 & & \geq & 1 \\
& & y_2 & & +y_4 & \geq & 1 \\
& y_i & & & & \geq & 0
\end{array}$$

with $y_2 \in [0,1]$

At**Most**NValue

$$D(X_1) = \{1, 2\},\ \ D(X_2) = \{2, 3\},\ \ D(X_3) = \{2, 4\},\ \ D(N) = \{2\}$$
$$D(Y_1) = \{0, 1\},\ \ D(Y_2) = \{\cancel{0}, 1\},\ \ D(Y_3) = \{2, 4\},\ \ D(Y_4) = \{0, 1\}$$

$$\begin{array}{lllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & & \\
& y_1 & +y_2 & & & \geq & 1 \\
& & y_2 & +y_3 & & \geq & 1 \\
& & y_2 & & +y_4 & \geq & 1 \\
& y_i & & & & \geq & 0
\end{array}$$

with $y_2 \in [0, 1]$

$$y^* = (0, 1, 0, 0)$$

At**Most**NValue

$$D(X_1) = \{1,2\},\ D(X_2) = \{2,3\},\ D(X_3) = \{2,4\},\ D(N) = \{2\}$$
$$D(Y_1) = \{0,1\},\ D(Y_2) = \{\cancel{0,} 1\},\ D(Y_3) = \{2,4\},\ D(Y_4) = \{0,1\}$$

$$\begin{array}{lllllll} \text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & & \\ & y_1 & +y_2 & & & \geq & 1 \\ & & y_2 & +y_3 & & \geq & 1 \\ & & y_2 & & +y_4 & \geq & 1 \\ & y_i & & & & \geq & 0 \end{array}$$

with $y_2 \in [0,1]$

$$y^* = (0,1,0,0)$$
$$\lambda^* = (1,1,1)$$

At**Most**NValue

$$D(X_1) = \{1,2\},\ \ D(X_2) = \{2,3\},\ \ D(X_3) = \{2,4\},\ \ D(N) = \{2\}$$
$$D(Y_1) = \{0,1\},\ \ D(Y_2) = \{\cancel{0},1\},\ \ D(Y_3) = \{2,4\},\ \ D(Y_4) = \{0,1\}$$

$$\begin{array}{lllllll} \text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & & \\ & y_1 & +y_2 & & & \geq & 1 \\ & & y_2 & +y_3 & & \geq & 1 \\ & & y_2 & & +y_4 & \geq & 1 \\ & y_i & & & & \geq & 0 \end{array}$$

with $y_2 \in [0,1]$

$$y^* = (0,1,0,0)$$
$$\lambda^* = (1,1,1)$$

$$y_1^* = \underline{y_1} \text{ and } rc(y_1) = 1 - \lambda_1^* = 0$$
$$y_2^* = \overline{y_2} \text{ and } rc(y_2) = 1 - \lambda_1^* - \lambda_2^* - \lambda_3^* = -2$$
$$y_3^* = \underline{y_2} \text{ and } rc(y_3) = 1 - \lambda_2^* = 0$$
$$y_4^* = \underline{y_3} \text{ and } rc(y_4) = 1 - \lambda_3^* = 0$$

At**Most**NValue

$$D(X_1) = \{1, 2\},\ D(X_2) = \{2, 3\},\ D(X_3) = \{2, 4\},\ D(N) = \{2\}$$
$$D(Y_1) = \{0, 1\},\ D(Y_2) = \{\cancel{0}, 1\},\ D(Y_3) = \{2, 4\},\ D(Y_4) = \{0, 1\}$$

$$\begin{array}{lllllll}
\text{Min } z = & y_1 & +y_2 & +y_3 & +y_4 & & \\
 & y_1 & +y_2 & & & \geq & 1 \\
 & & y_2 & +y_3 & & \geq & 1 \\
 & & y_2 & & +y_4 & \geq & 1 \\
 & y_i & & & & \geq & 0
\end{array}$$

with $y_2 \in [0, 1]$

$$y^* = (0, 1, 0, 0)$$
$$\lambda^* = (1, 1, 1)$$

$$y_1^* = \underline{y_1} \text{ and } rc(y_1) = 1 - \lambda_1^* = 0$$
$$y_2^* = \overline{y_2} \text{ and } rc(y_2) = 1 - \lambda_1^* - \lambda_2^* - \lambda_3^* = -2$$
$$y_3^* = \underline{y_2} \text{ and } rc(y_3) = 1 - \lambda_2^* = 0$$
$$y_4^* = \underline{y_3} \text{ and } rc(y_4) = 1 - \lambda_3^* = 0$$

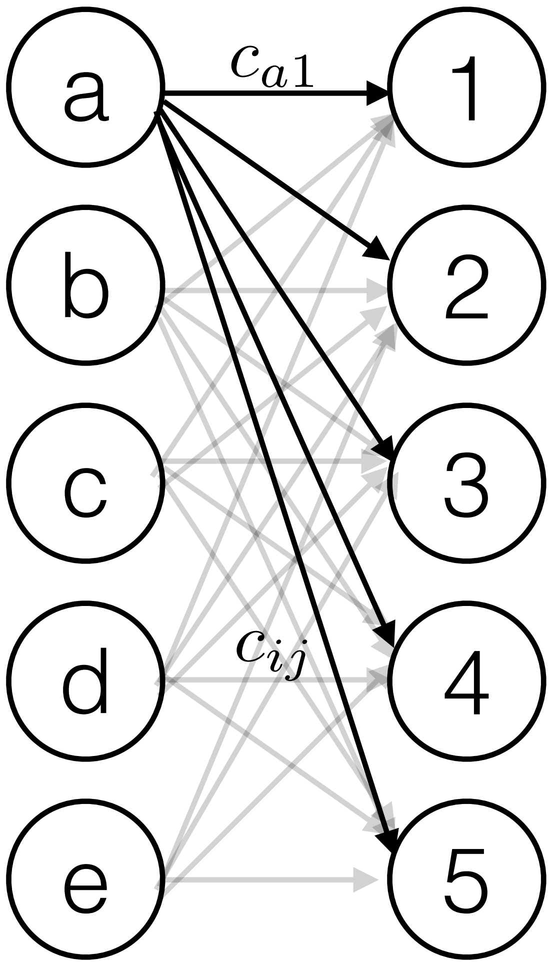
$$y_2 \geq \lceil \overline{y_2} + \frac{(z^* - z)}{rc(y_2)} \rceil = \lceil 1 + \frac{2-1}{-2} \rceil = \lceil 0.5 \rceil = 1$$

Reduced cost based filtering

- Linear Programming duality
- First example: At**Most**NValue
- Filtering the upper bound of a 0/1 variable
- Filtering the lower bound of a 0/1 variable
- General principles
- Second example
- Assignment, Cumulative, Bin-packing, …

LP relaxations used for global constraints

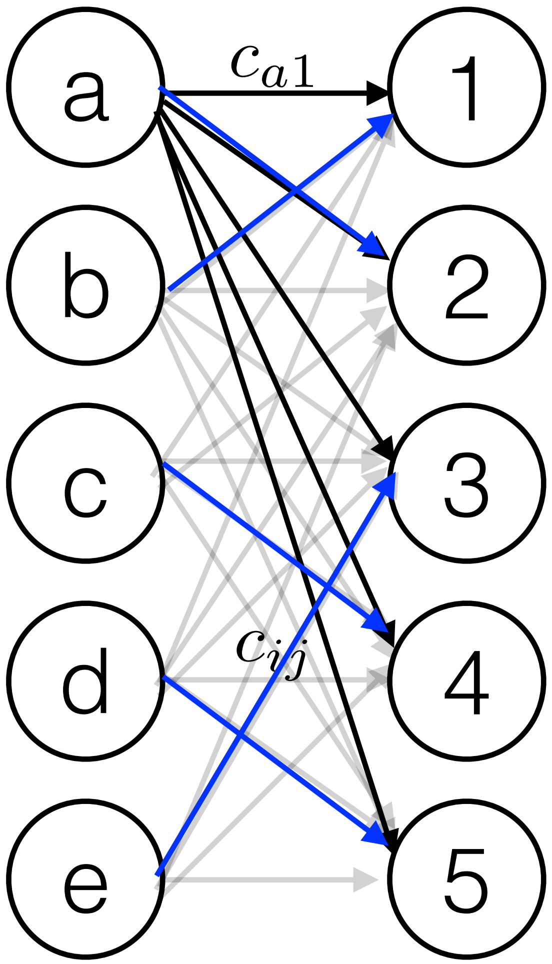
- **Assignment problem** (used as a lower bound for TSP)

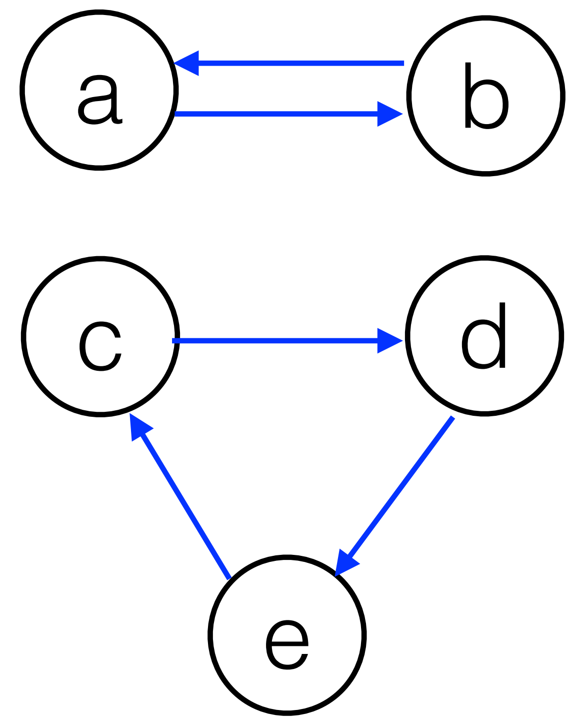


$$\text{Min} \sum_{i,j} x_{ij} c_{ij}$$

LP relaxations used for global constraints

- **Assignment problem** (used as a lower bound for TSP)

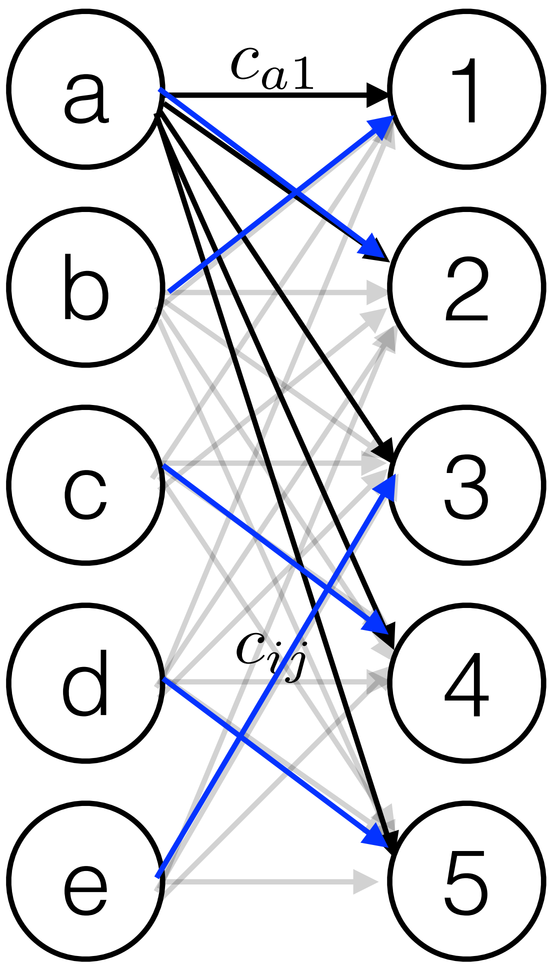




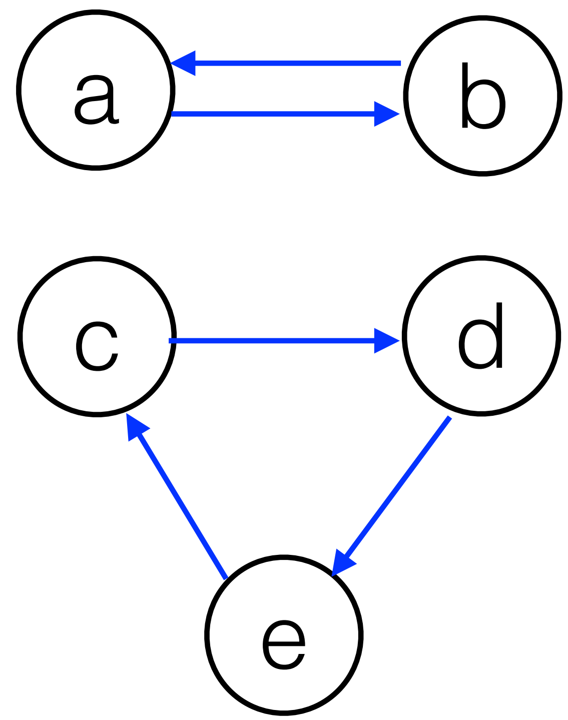
$$\text{Min} \sum_{i,j} x_{ij} c_{ij}$$

LP relaxations used for global constraints

- **Assignment problem** (used as a lower bound for TSP)



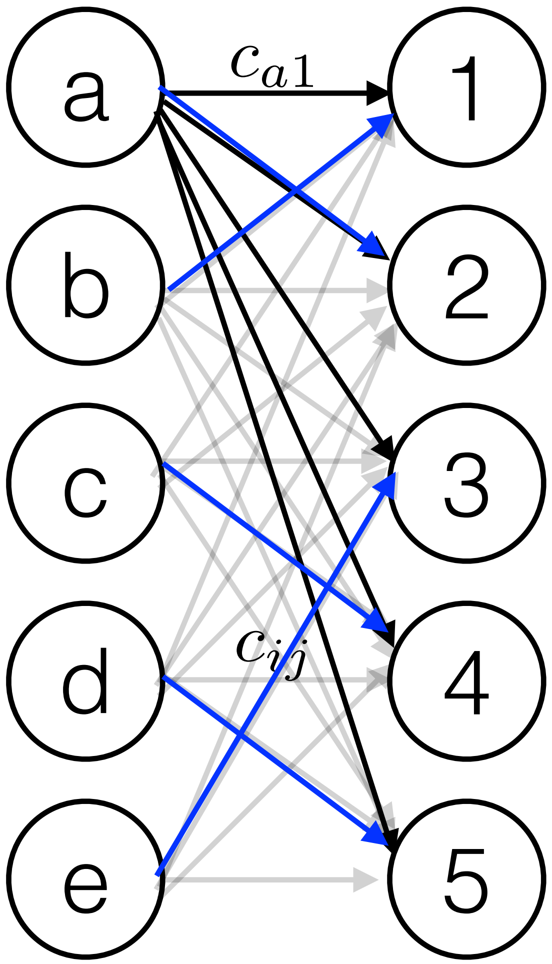
$$\text{Min} \sum_{i,j} x_{ij} c_{ij}$$



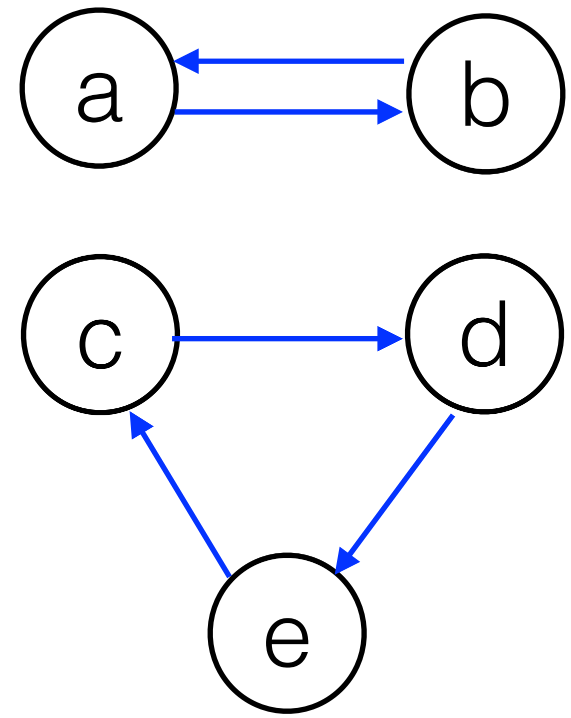
Used as a relaxation for TSP (relax connectivity but keep degree 2 constraints)

LP relaxations used for global constraints

- **Assignment problem** (used as a lower bound for TSP)



$$\text{Min} \sum_{i,j} x_{ij} c_{ij}$$

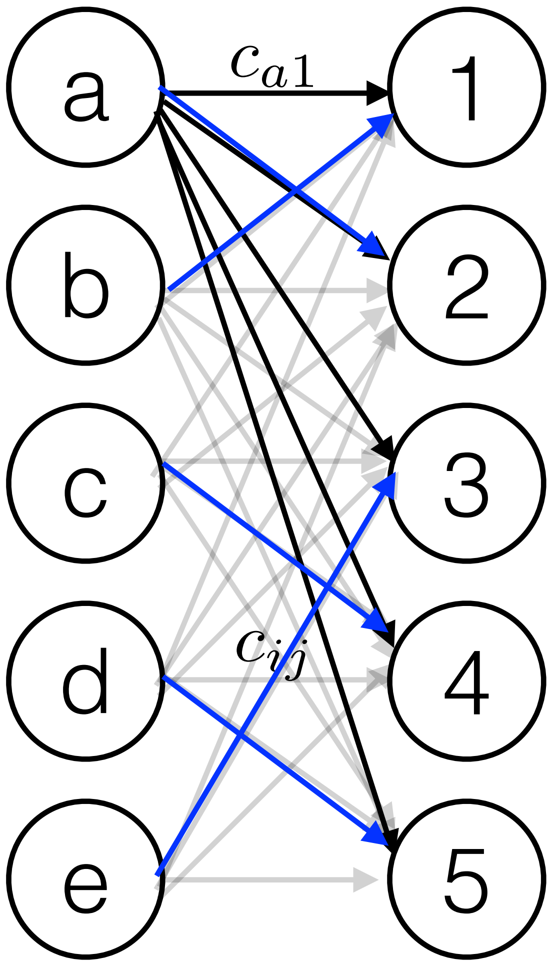


Used as a relaxation for TSP (relax connectivity but keep degree 2 constraints)

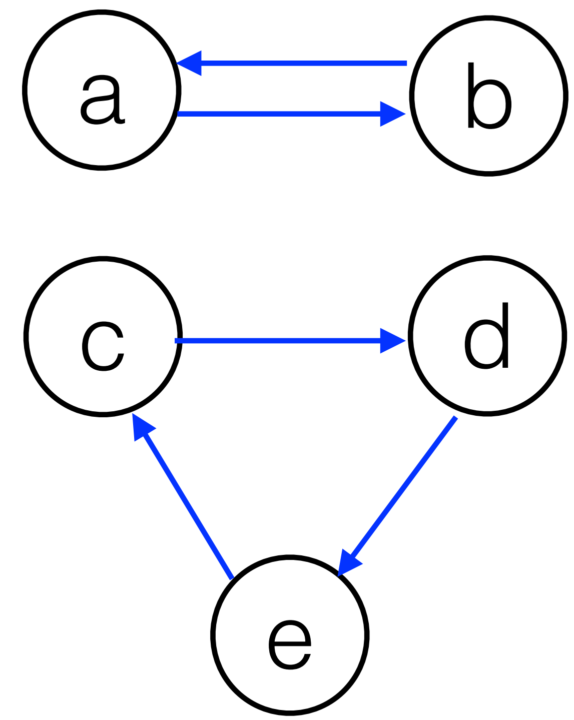
[Milano and al. 2006]

LP relaxations used for global constraints

- **Assignment problem** (used as a lower bound for TSP)



$$\text{Min} \sum_{i,j} x_{ij} c_{ij}$$



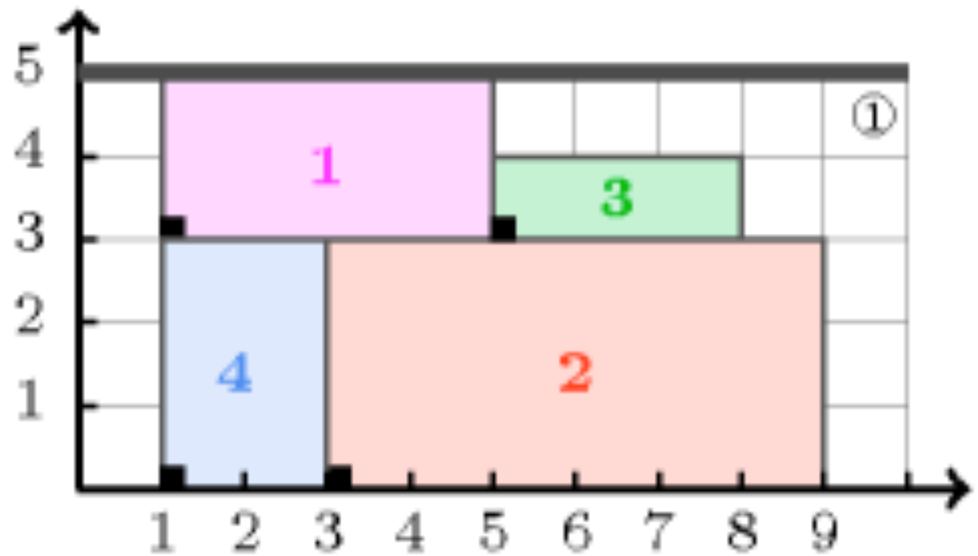
Used as a relaxation for TSP (relax connectivity but keep degree 2 constraints)

[Milano and al. 2006]

- **Global cardinality with costs** (ref ? folklore ?)

LP relaxations used for global constraints

- **Cumulative** (LP formulation with cutting planes)

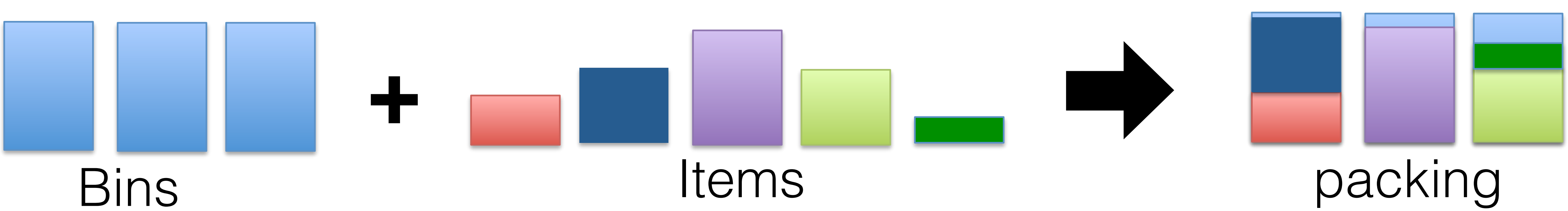


[Hooker. 2002]

(Picture from the *global constraint catalog*)

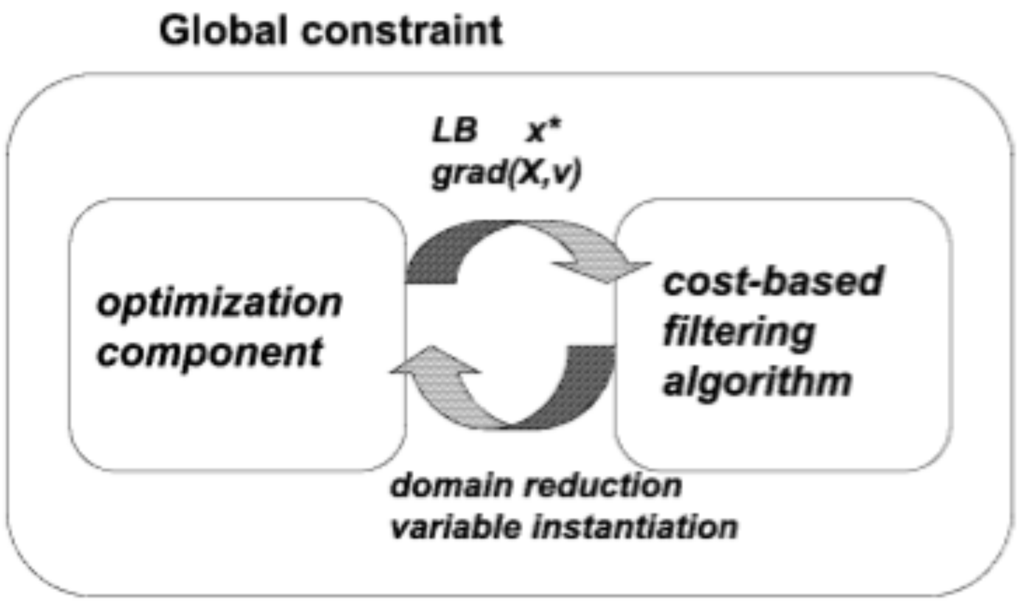
- **Bin-Packing** (Arc-flow formulation …)

[Valério de Carvalho 1999] [Cambazard. 2010]



LP relaxations used for global constraints

- Linear relaxation of global constraints
  [Refalo, 2000]: Linear formulation of Constraint Programming models and Hybrid Solvers
  ★ AllDifferent
  ★ Element
  ★ Among
  ★ Cycle



(Picture from [Foccaci, 2002])

- Cost-based filtering
  [Focacci, Lodi, Milano. 2002]: Embedding relaxations in global constraints for solving TSP and TSPTW

Outline

1. **Reduced-costs based filtering**
    - Linear Programming duality
    - First example: *AtMostNValue*
        - Filtering the upper bound of a 0/1 variable
        - Filtering the lower bound of a 0/1 variable
    - General principles
    - Second example
    - Assignment, Cumulative, Bin-packing, …
2. **Dynamic programming filtering algorithms**
    - Linear equation, WeightedCircuit
    - General principles
    - Other relationships of DP and CP
3. **Illustration with a real-life application**

Dynamic programming for global constraints

- Linear equation
- General principles
- Regular and variants
- WeightedCircuit
- Table constraint and MDD domains ?

Linear equation

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, 3, 4\}$$
$$D(x_2) = \{0, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, 4\}$$

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, 3, 4\}$$
$$D(x_2) = \{0, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, 4\}$$

Q: Give the arc-consistent domains

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, \cancel{3}, 4\}$$
$$D(x_2) = \{0, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, 4\}$$

Q: Give the arc-consistent domains

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{0, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, 4\}$$

Q: Give the arc-consistent domains

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, 4\}$$

Q: Give the arc-consistent domains

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, 2, \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, \boxed{2,} \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

$\overline{x_1}$?

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, \boxed{2,} \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

$$\overline{x_1} \ ?$$

Lower bound for the rest of the expression

$$3\overline{x_1} + \boxed{(-2\overline{x_2} + 4\underline{x_3})} \leq 7$$

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, \boxed{2}, \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

$$\overline{x_1} \ ?$$

Lower bound for the rest of the expression

$$3\overline{x_1} + \boxed{(-2\overline{x_2} + 4\underline{x_3})} \leq 7$$

$$3\overline{x_1} + (-8 + 8) \leq 7$$

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, \boxed{2,} \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

$$\overline{x_1} \ ?$$

Lower bound for the rest of the expression

$$3\overline{x_1} + \boxed{(-2\overline{x_2} + 4\underline{x_3})} \leq 7$$

$$3\overline{x_1} + (-8 + 8) \leq 7$$

$$\overline{x_1} \leq \lfloor \tfrac{7}{3} \rfloor = 2$$

Linear equation

- Let's start with linear inequalities first and enforce GAC:

$$3x_1 - 2x_2 + 4x_3 \leq 7$$

$$D(x_1) = \{0, 1, \boxed{2,} \cancel{3}, \cancel{4}\}$$
$$D(x_2) = \{\cancel{0}, 1, 2, 3, 4\}$$
$$D(x_3) = \{2, 3, \cancel{4}\}$$

Q: Give the arc-consistent domains

$\overline{x_1}$?

Lower bound for the rest of the expression

$$3\overline{x_1} + \boxed{(-2\overline{x_2} + 4\underline{x_3})} \leq 7$$

$$3\overline{x_1} + (-8 + 8) \leq 7$$

$$\overline{x_1} \leq \lfloor \tfrac{7}{3} \rfloor = \boxed{2}$$

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Update the upper bound of variables with a positive coefficient $(k < n_1)$

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Update the upper bound of variables with a positive coefficient

$(k < n_1)$

Lower bound for the rest of the expression

$$\overline{x_k} \leftarrow \left\lfloor \frac{c - \boxed{(\sum_{i=1 \land i \neq k}^{n_1-1} a_i \underline{x_i} - \sum_{i=n_1}^{n} b_i \overline{x_i})}}{a_k} \right\rfloor$$

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Update the upper bound of variables with a positive coefficient

$(k < n_1)$

Lower bound for the rest of the expression

$$\overline{x_k} \leftarrow \lfloor \frac{c - \boxed{(\sum_{i=1 \wedge i \neq k}^{n_1-1} a_i \underline{x_i} - \sum_{i=n_1}^{n} b_i \overline{x_i})}}{a_k} \rfloor$$

- Update the upper bound of variables with a negative coefficient

$(k \geq n_1)$

$$\underline{x_k} \leftarrow \lceil \frac{(\sum_{i=1}^{n_1-1} a_i \underline{x_i} - \sum_{i=n_1 \wedge i \neq k}^{n} b_i \overline{x_i}) - c}{b_k} \rceil$$

Linear equation

[Laurière, 1978]

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Update the upper bound of variables with a positive coefficient

$(k < n_1)$

Lower bound for the rest of the expression

$$\overline{x_k} \leftarrow \left\lfloor \frac{c - \boxed{\left(\sum_{i=1 \wedge i \neq k}^{n_1-1} a_i \underline{x_i} - \sum_{i=n_1}^{n} b_i \overline{x_i}\right)}}{a_k} \right\rfloor$$

- Update the upper bound of variables with a negative coefficient

$(k \geq n_1)$

$$\underline{x_k} \leftarrow \left\lceil \frac{\left(\sum_{i=1}^{n_1-1} a_i \underline{x_i} - \sum_{i=n_1 \wedge i \neq k}^{n} b_i \overline{x_i}\right) - c}{b_k} \right\rceil$$

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Is a fixed point needed between the two rules ?

- Does that achieve BC or GAC ?

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Is a fixed point needed between the two rules ?

  No, the rules and updates are not on the same bounds

- Does that achieve BC or GAC ?

Linear equation

$$\sum_{i=1}^{n_1-1} a_i x_i - \sum_{i=n_1}^{n} b_i x_i \leq c$$

Suppose for sake of simplicity: $\forall i \;\; a_i, b_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

- Is a fixed point needed between the two rules ?

  No, the rules and updates are not on the same bounds

- Does that achieve BC or GAC ?

  Only bounds are updated but all remaining values have a support so it achieves GAC

Linear equation

Linear equation

- Consider now: $2x_1 + 3x_2 + 4x_3 = 7$

$$D(x_1) = \{0, 1, 2\}$$
$$D(x_2) = \{0, 1\}$$
$$D(x_3) = \{0, 1\}$$

Linear equation

- Consider now: $2x_1 + 3x_2 + 4x_3 = 7$

$$D(x_1) = \{0, 1, 2\}$$
$$D(x_2) = \{0, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Linear equation

- Consider now: $$2x_1 + 3x_2 + 4x_3 = 7$$

$$D(x_1) = \{0, \cancel{1}, 2\}$$

$$D(x_2) = \{0, 1\}$$

$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Linear equation

- Consider now: $2x_1 + 3x_2 + 4x_3 = 7$

$$D(x_1) = \{0, \cancel{1}, 2\}$$
$$D(x_2) = \{\cancel{0}, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Linear equation

- Consider now: $$2x_1 + 3x_2 + 4x_3 = 7$$

$$D(x_1) = \{0, \cancel{1}, 2\}$$
$$D(x_2) = \{\cancel{0}, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Q: How does a CP solver usually filters that constraint ?

Linear equation

- Consider now: $$2x_1 + 3x_2 + 4x_3 = 7$$

$$D(x_1) = \{0, \cancel{1}, 2\}$$
$$D(x_2) = \{\cancel{0}, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Q: How does a CP solver usually filters that constraint ?

Q: What values are removed in the example with this technique ?

Linear equation

- Consider now: $2x_1 + 3x_2 + 4x_3 = 7$

$$D(x_1) = \{0, \cancel{1}, 2\}$$
$$D(x_2) = \{\cancel{0}, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Q: How does a CP solver usually filters that constraint ?

Apply previous filtering algorithm for both (until fixed-point) :

$$2x_1 + 3x_2 + 4x_3 \geq 7$$
$$2x_1 + 3x_2 + 4x_3 \leq 7$$

Q: What values are removed in the example with this technique ?

Linear equation

- Consider now: $$2x_1 + 3x_2 + 4x_3 = 7$$

$$D(x_1) = \{0, \cancel{1}, 2\}$$
$$D(x_2) = \{\cancel{0}, 1\}$$
$$D(x_3) = \{0, 1\}$$

Q: Give the arc-consistent domains

Q: How does a CP solver usually filters that constraint ?

Apply previous filtering algorithm for both (until fixed-point) :

$$2x_1 + 3x_2 + 4x_3 \geq 7$$
$$2x_1 + 3x_2 + 4x_3 \leq 7$$

Q: What values are removed in the example with this technique ?

None

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

Q: What is the complexity of achieving GAC ?

Q: What is the complexity of achieving BC ?

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

Q: What is the complexity of achieving GAC ?

- Consider only {0,1} domains
- It is as hard as subset sum: « *given an integer **k** and a set **S** of integers, is there a subset of **S** that sums to **k** ?* »

Q: What is the complexity of achieving BC ?

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

Q: What is the complexity of achieving GAC ?

- Consider only {0,1} domains
- It is as hard as subset sum: « *given an integer **k** and a set **S** of integers, is there a subset of **S** that sums to **k** ?* »

Q: What is the complexity of achieving BC ?

- BC and GAC are the same on {0,1} domains…
- So BC is just as hard

Linear equation

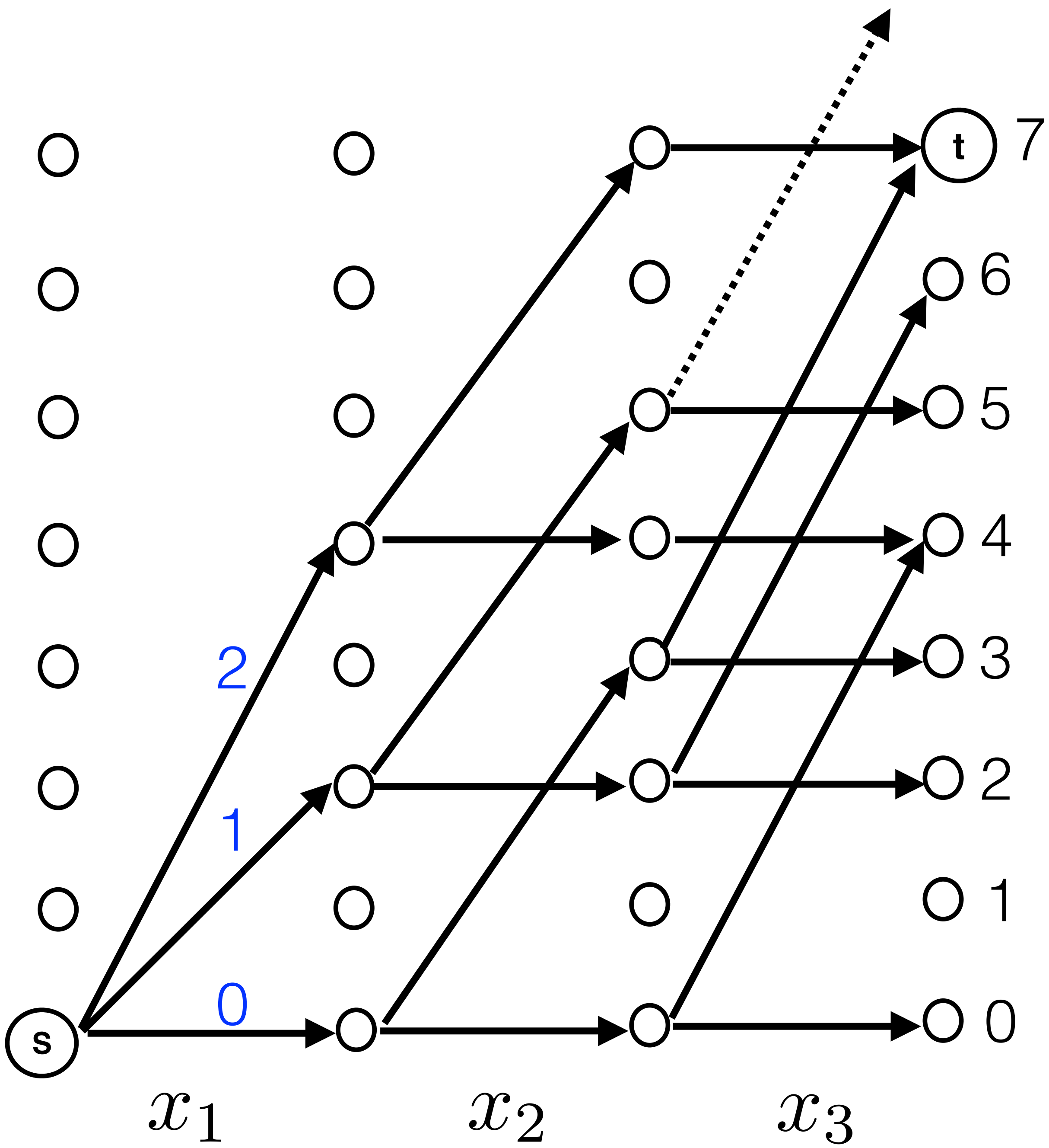
$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

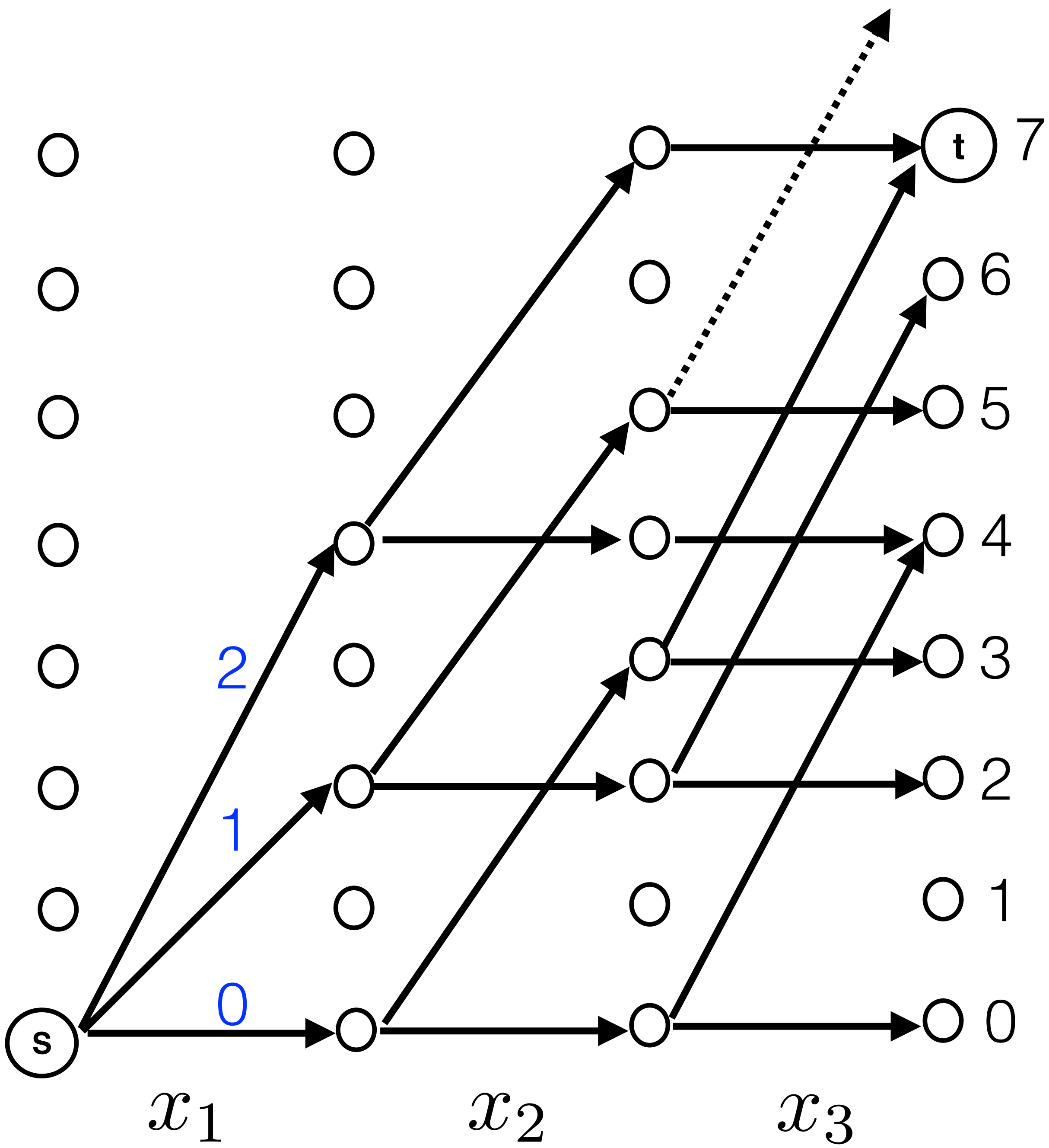
- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



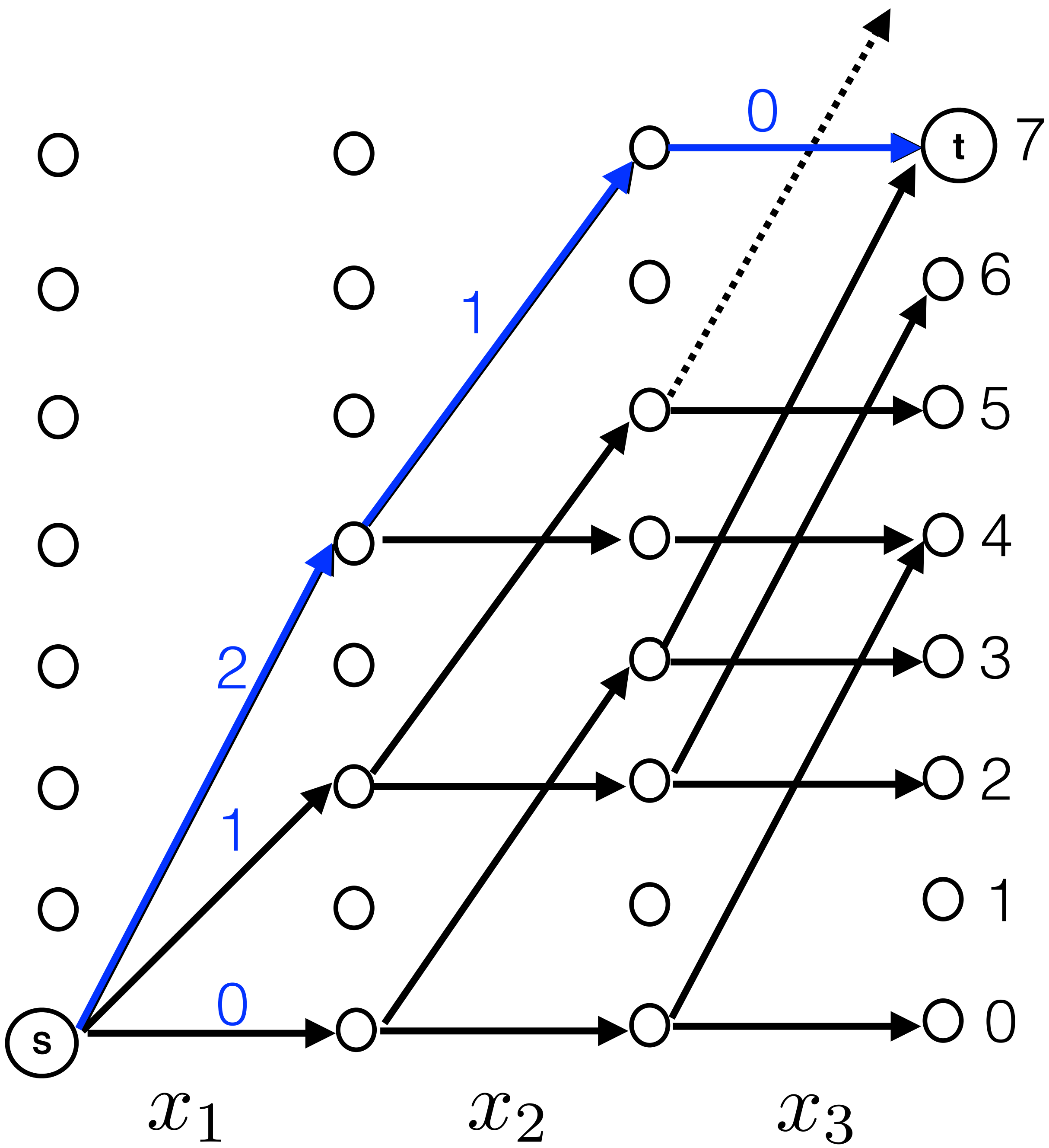
a support = a path from s to t

ex: (2,1,0)

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



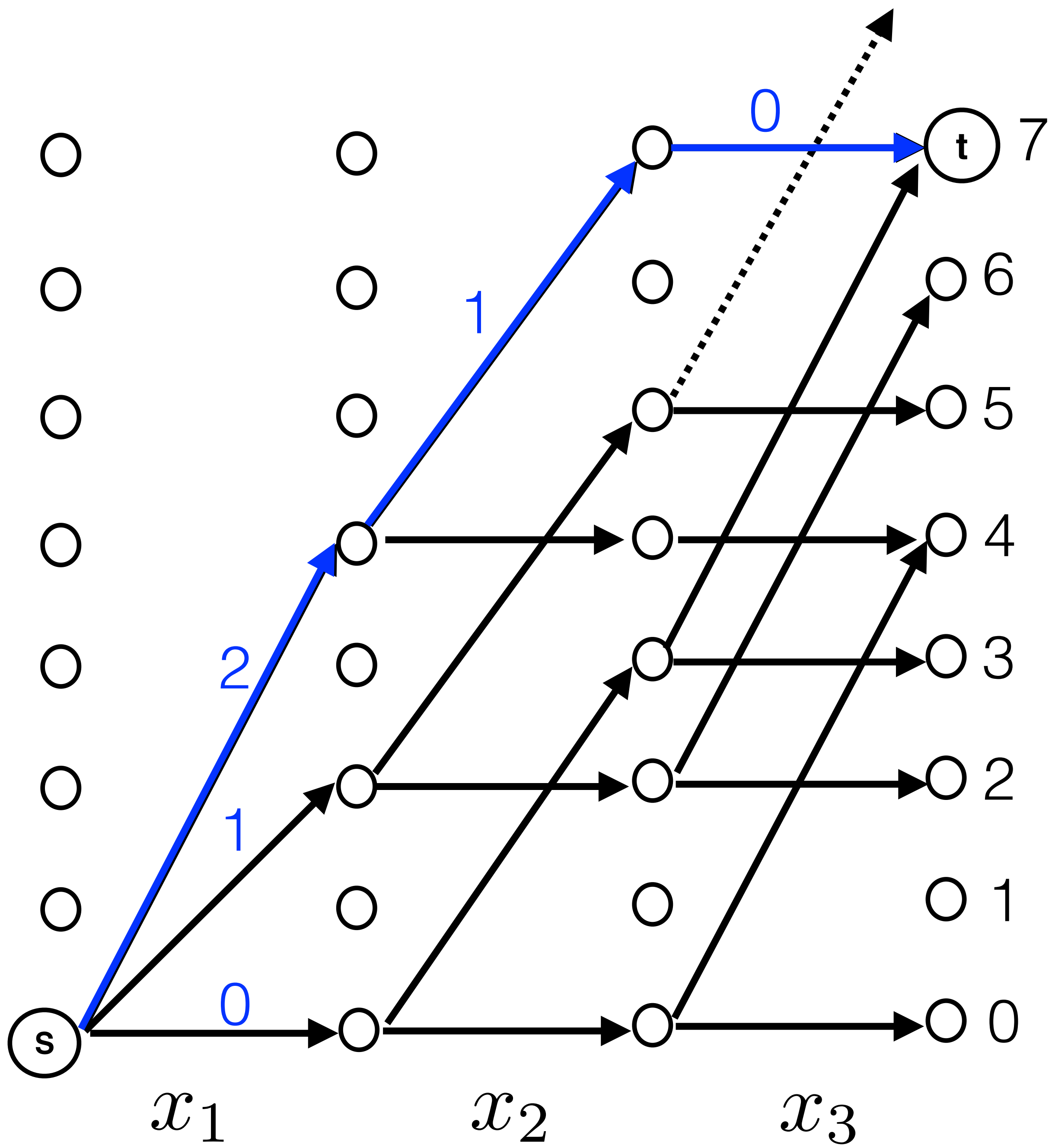
a support = a path from s to t

ex: (2,1,0)

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



a support = a path from s to t

ex: (2,1,0)

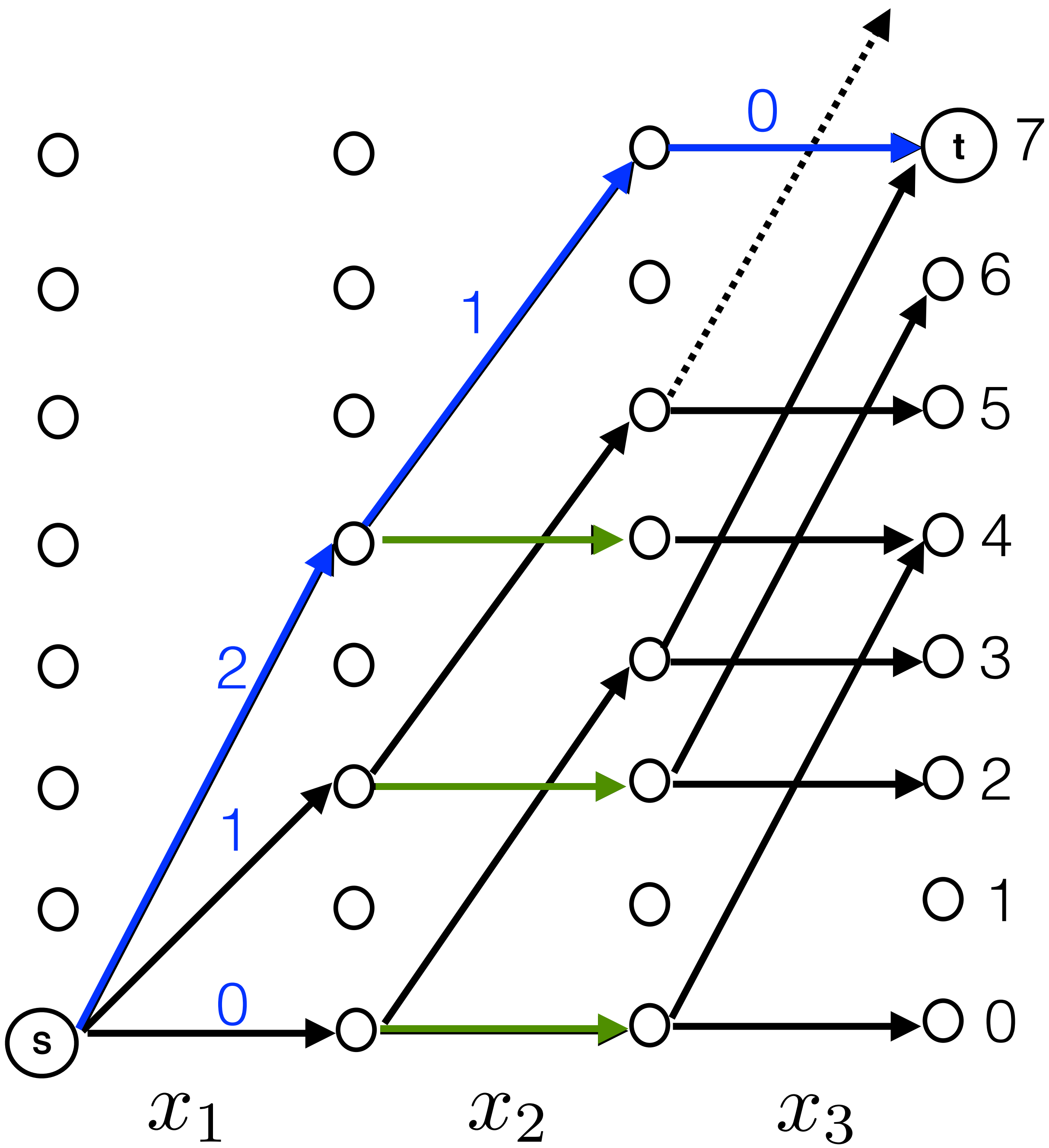
a value of a domain = a set of arcs in the graph

ex: Value 0 of x_2

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



a support = a path from s to t

ex: (2,1,0)

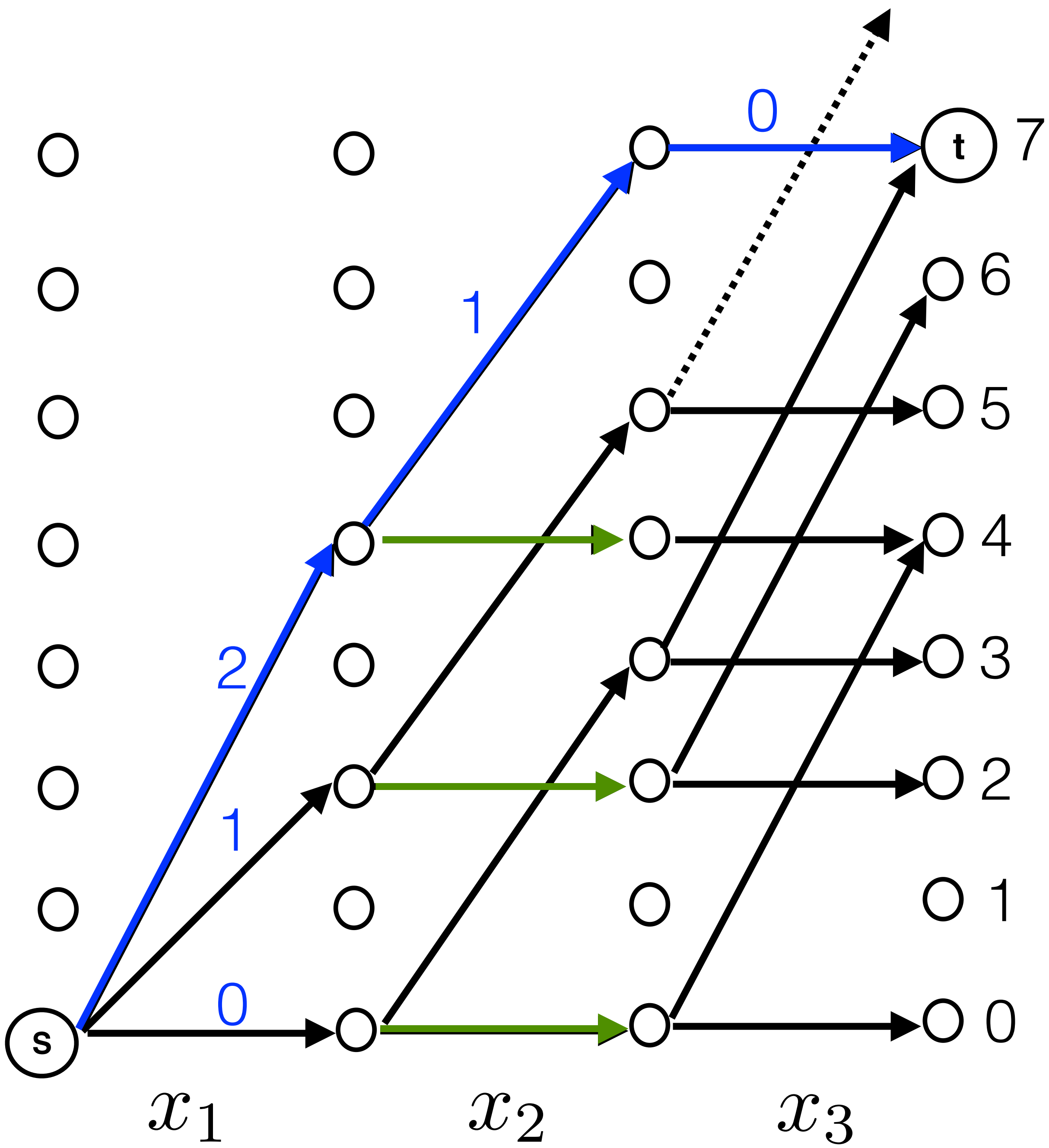
a value of a domain = a set of arcs in the graph

ex: Value 0 of x_2

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



a support = a path from s to t

ex: (2,1,0)

a value of a domain = a set of arcs in the graph

ex: Value 0 of x_2

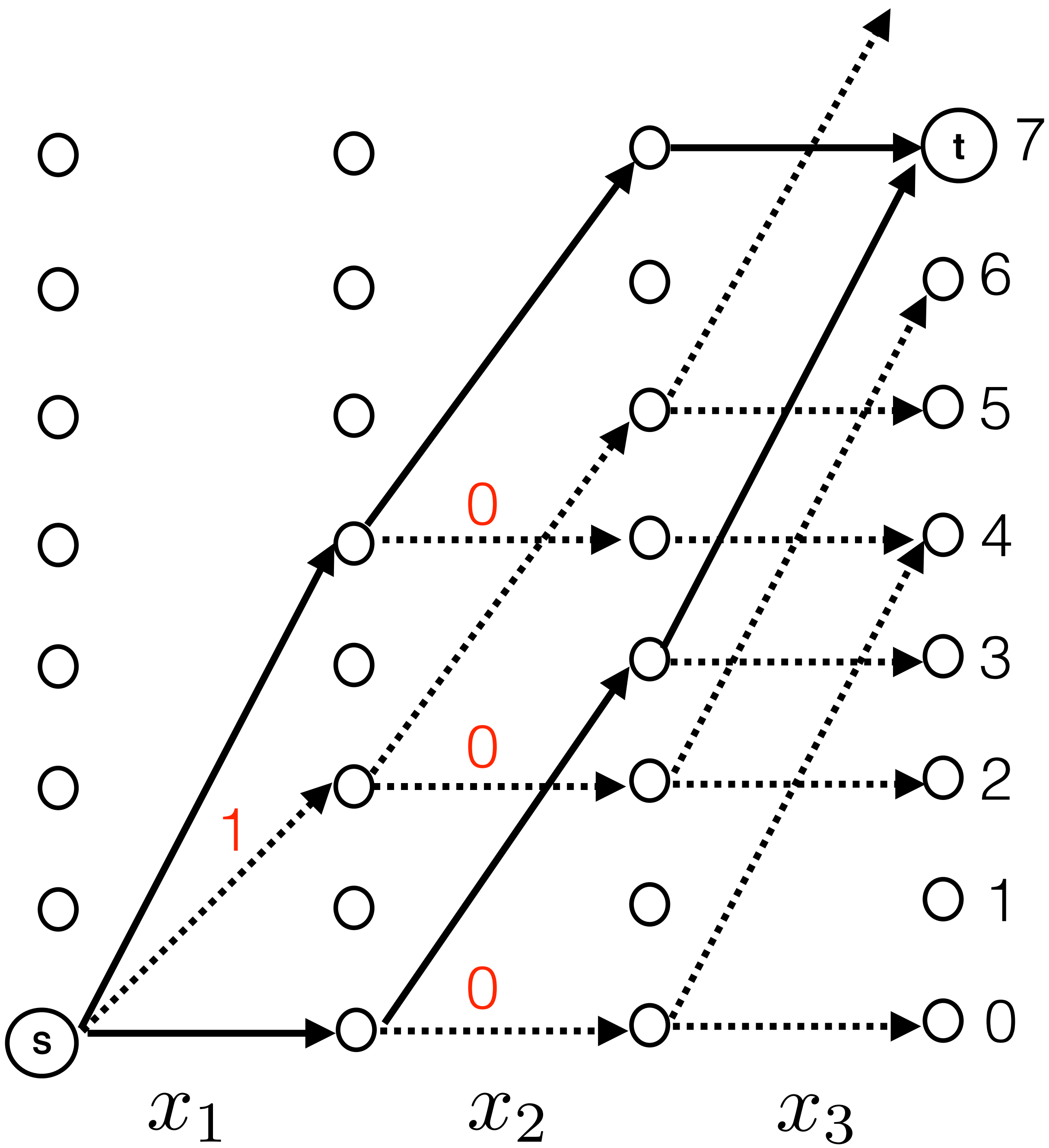
Filtering:

- remove all arcs that do not belong to a path-support
- remove values when they loose all their supporting arcs

Linear equation

$$2x_1 + 3x_2 + 4x_3 = 7 \quad D(x_1) = \{0, \cancel{1}, 2\} \quad D(x_2) = \{\cancel{0}, 1\} \quad D(x_3) = \{0, 1\}$$

- The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



Filtering:

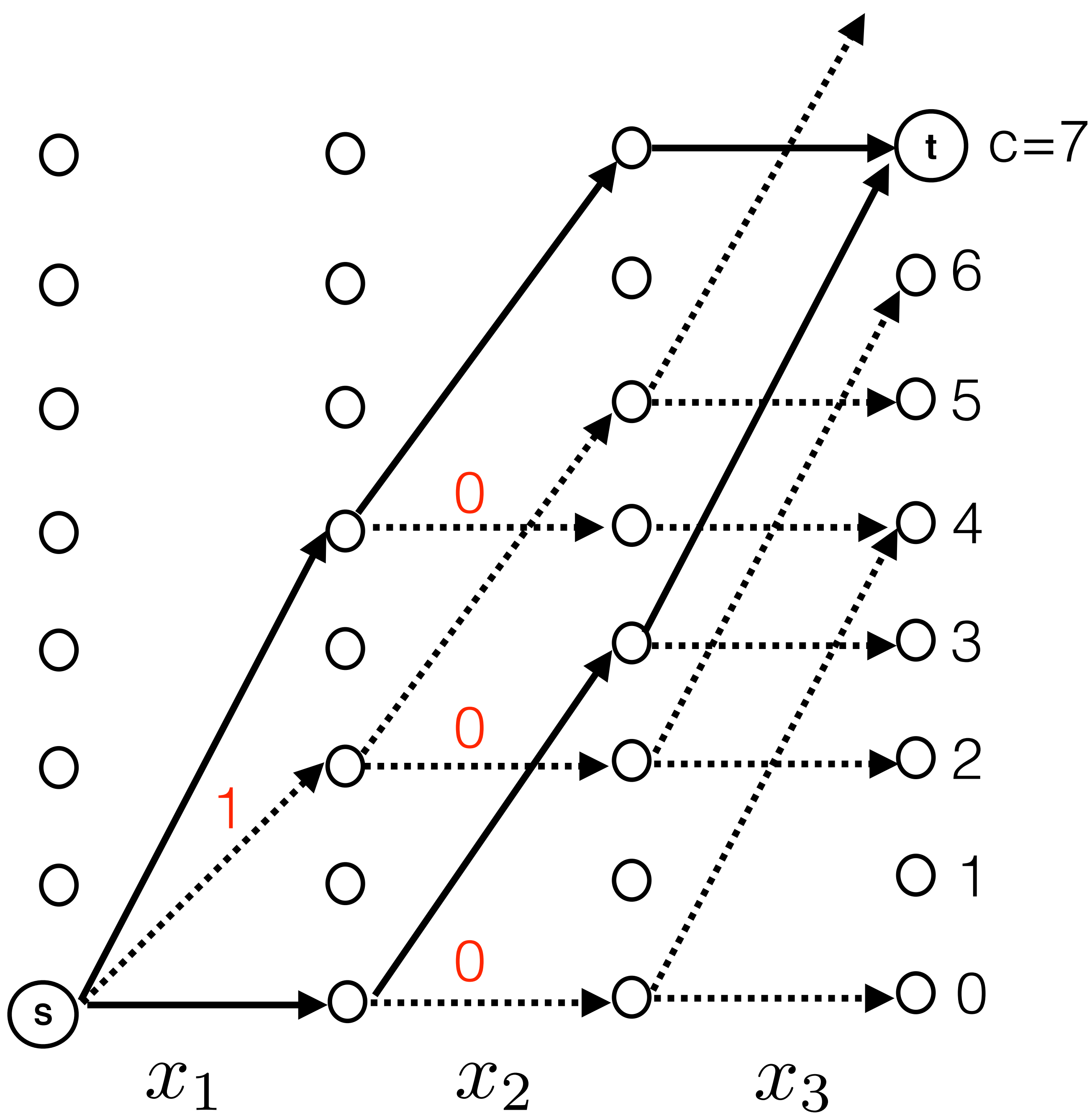
- remove all arcs that do not belong to a path-support
- remove values when they loose all their supporting arcs

Algorithm:

1. forward pass: mark arcs in a breath-first search from s to t
2. backward pass: mark arcs in a breath-first search from t to s
3. remove all non-marked arcs

Linear equation

The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…



Filtering:

- remove all arcs that do not belong to a path-support
- remove values when they loose all their supporting arcs

Algorithm:

1. forward pass: mark arcs in a breath-first search from s to t
2. backward pass: mark arcs in a breath-first search from t to s
3. remove all non-marked arcs

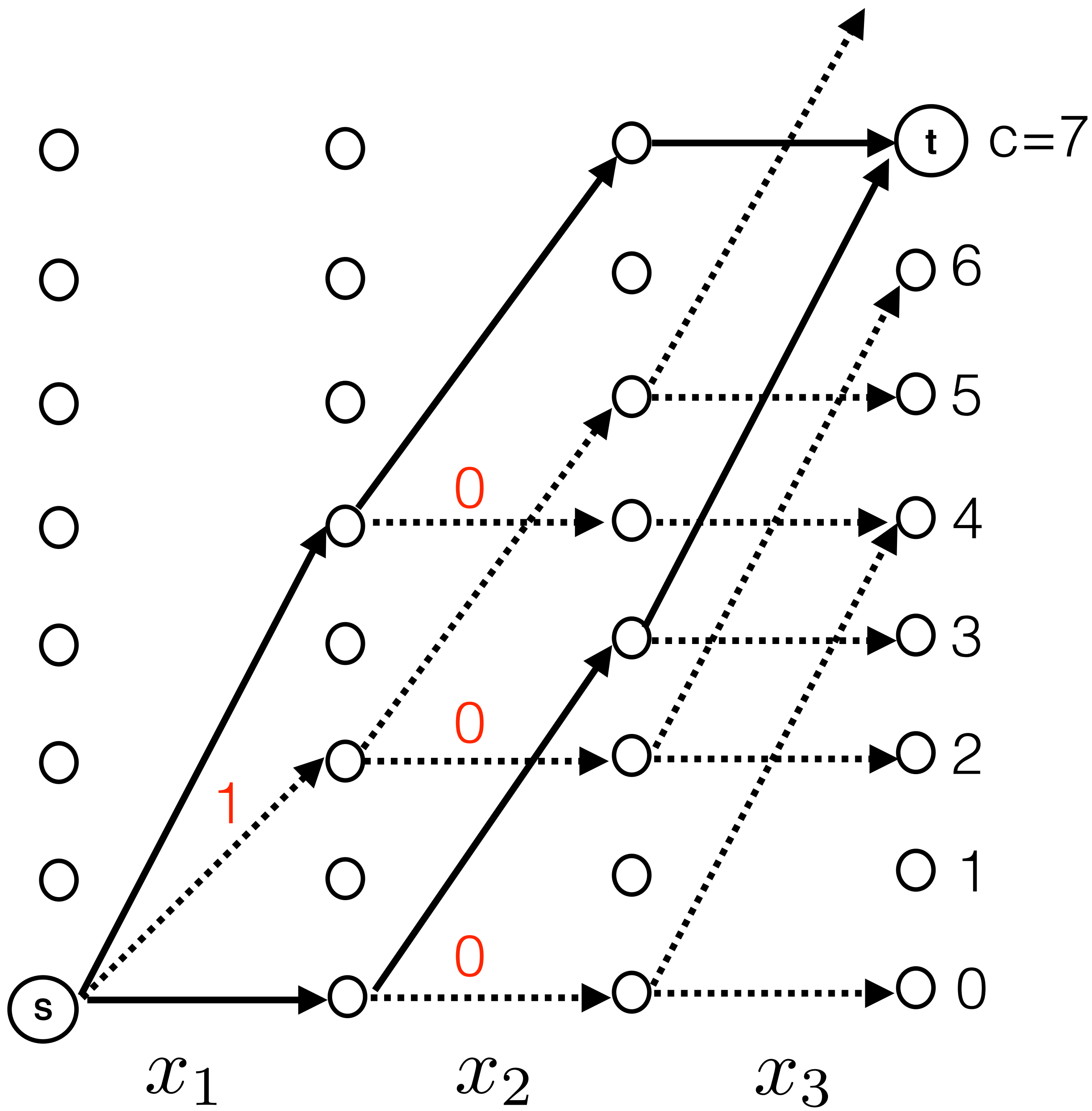
Complexity: $O(nmc)$

(positive domains and coefficients)

Linear equation

The dynamic programming approach: formulate it **a path problem** in a graph with a **pseudo-polynomial size**…

[Trick. 2003]



Filtering:

- remove all arcs that do not belong to a path-support
- remove values when they loose all their supporting arcs

Algorithm:

1. forward pass: mark arcs in a breath-first search from s to t
2. backward pass: mark arcs in a breath-first search from t to s
3. remove all non-marked arcs

Complexity: $O(nmc)$

(positive domains and coefficients)

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

$f(i, K)$ **= true** if sum **K** can be reached with $x_1, \ldots, x_i$

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

$f(i, K)$ **= true** if sum **K** can be reached with $x_1, \ldots, x_i$

$$f(i, K) = \vee_{v_k \in D(x_i)} f(i-1, K - a_i v_k)$$

We are looking for $f(n, c)$

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

$f(i, K)$ **= true** if sum **K** can be reached with $x_1, \ldots, x_i$

$$f(i, K) = \bigvee_{\boxed{v_k \in D(x_i)}} f(i-1, K - a_i v_k)$$

We are looking for $f(n, c)$

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^* \text{ and } D(x_i) \subset \mathbb{N}$

$f(i, K)$ **= true** if sum **K** can be reached with $x_1, \ldots, x_i$

$$f(i, K) = \bigvee_{\boxed{v_k \in D(x_i)}} f(i-1, K - a_i v_k)$$

$$f(0, K) = \text{false} \quad K \neq 0$$

$$f(0, 0) = \text{true}$$

We are looking for $f(n, c)$

Linear equation

$$\sum_{i=1}^{n} a_i x_i = c$$

Suppose for sake of simplicity: $\forall i \quad a_i \in \mathbb{N}^*$ and $D(x_i) \subset \mathbb{N}$

$f(i, K)$ **= true** if sum **K** can be reached with $x_1, \ldots, x_i$

$$f(i, K) = \bigvee_{\boxed{v_k \in D(x_i)}} f(i-1, K - a_i v_k)$$

$$f(0, K) = \text{false} \quad K \neq 0$$

$$f(0, 0) = \text{true}$$

We are looking for $f(n, c)$

Complexity: $O(nmc)$

Dynamic programming for global constraints

- Linear equation
- General principles
- Regular and variants
- WeightedCircuit
- Reformulation of global constraints and MDD domains ?

General principles

1. Formulate the problem of **existence of a support** as **a path problem** in a graph of **pseudo-polynomial size**

2. Define properly the graph model:
   - **support** = a **path, shortest path, longest path**, …
   - **values** of domains = **arcs, nodes**

3. Apply a forward-backward pass to mark edges-nodes with
   - the value of the **best** path **supporting them**

4. Remove all values not supported in the graph

Dynamic programming for global constraints

- Linear equation
- General principles
- Regular and variants
- WeightedCircuit
- Table constraint and MDD domains ?

Regular and variants

Regular and variants

Automaton

- Regular : $\textsc{Regular}([X_1, \ldots, X_n], A)$ [Pesant, 2004]
  - Propagation based on breath-first-search in the unfolded automaton

Regular and variants

Automaton

- Regular : $\textsc{Regular}([X_1, \ldots, X_n], A)$ [Pesant, 2004]
  - Propagation based on breath-first-search in the unfolded automaton

- Cost regular : $\textsc{Regular}([X_1, \ldots, X_n], A) \wedge \sum_{i=1}^{n} c_{iX_i} = Z$
  - Propagation based on shortest/longest path in the unfolded automaton [Demassey, Pesant, Rousseau, 2004]

Regular and variants

Automaton

- Regular : $\textsc{Regular}([X_1, \ldots, X_n], A)$ [Pesant, 2004]
  - Propagation based on breath-first-search in the unfolded automaton

- Cost regular : $\textsc{Regular}([X_1, \ldots, X_n], A) \;\wedge\; \sum_{i=1}^{n} c_{iX_i} = Z$
  - Propagation based on shortest/longest path in the unfolded automaton [Demassey, Pesant, Rousseau, 2004]

- Multi-cost regular : $\textsc{Multi-cost Regular}([X_1, \ldots, X_n], [Z^1, \ldots, Z^R], A)$
  $\textsc{Regular}([X_1, \ldots, X_n], A) \;\wedge\; (\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r = 0, \ldots, R)$
  - Propagation based on **resource constrained shortest/longest path**
  - Sequencing and counting at the same time [Menana, Demassey, 2009]
    - Personnel scheduling
    - Routing
  - Example: combine Regular and GCC

Regular and variants

Regular and variants

- Multi-cost regular :

  $$\textsc{Regular}([X_1,\ldots,X_n],A) \;\wedge\; (\textstyle\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r = 0,\ldots,R)$$

- Example:
  - Schedule 7 shifts of type: **night (N), day (D), rest (R)**
  - (1) “A **Rest must follow a Night** shift”
  - (2) “**Exactly 3 day shifts** and **1 night shift** must take place in the week”

X_1	X_2	X_3	X_4	X_5	X_6	X_7
D	R	N	R	D	D	R

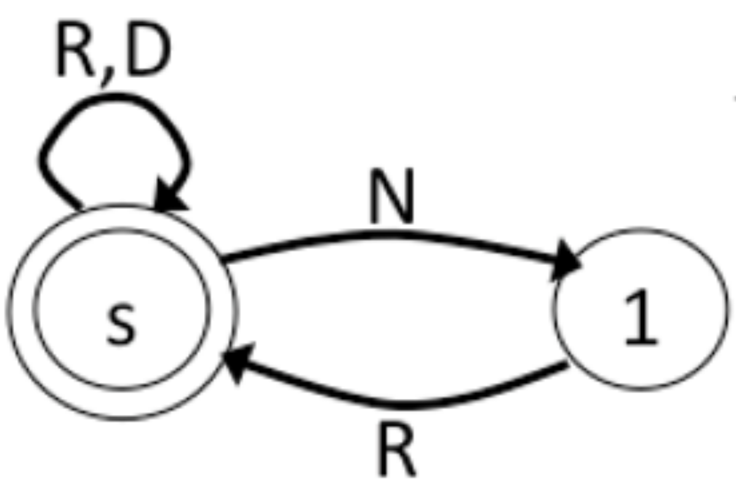
Regular and variants

- Multi-cost regular :

$$\textsc{Regular}([X_1,\dots,X_n],A) \wedge (\textstyle\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r = 0,\dots,R)$$

- Example:
  - Schedule 7 shifts of type: **night (N), day (D), rest (R)**
  - (1) “A **Rest must follow a Night** shift”
  - (2) “**Exactly 3 day shifts** and **1 night shift** must take place in the week”

X_1	X_2	X_3	X_4	X_5	X_6	X_7
D	**R**	**N**	**R**	**D**	**D**	**R**



$R = 2$

$$\begin{array}{c} \qquad\qquad\qquad\qquad\quad \text{D}\ \ \text{R}\ \ \text{N} \\ \mathrm{GCC}([X_1,\dots,X_7],[3,0,1],[3,7,1]) \end{array}$$

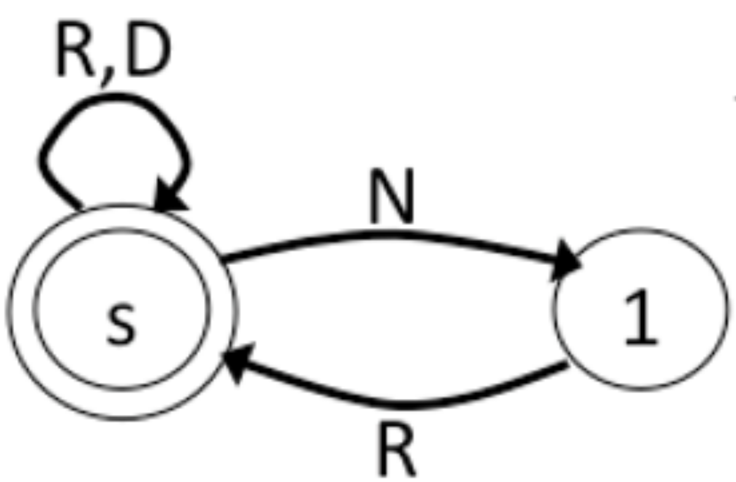
Regular and variants

- Multi-cost regular :

  $$\textsc{Regular}([X_1, \ldots, X_n], A) \wedge (\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r = 0, \ldots, R)$$

- Example:
  - Schedule 7 shifts of type: **night (N), day (D), rest (R)**
  - (1) "A **Rest must follow a Night** shift"
  - (2) "**Exactly 3 day shifts** and **1 night shift** must take place in the week"

X_1	X_2	X_3	X_4	X_5	X_6	X_7
D	R	N	R	D	D	R



$R = 2$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^1_D	1	1	1	1	1	1	1
c^1_N	0	0	0	0	0	0	0
c^1_R	0	0	0	0	0	0	0

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^2_D	0	0	0	0	0	0	0
c^2_N	1	1	1	1	1	1	1
c^2_R	0	0	0	0	0	0	0

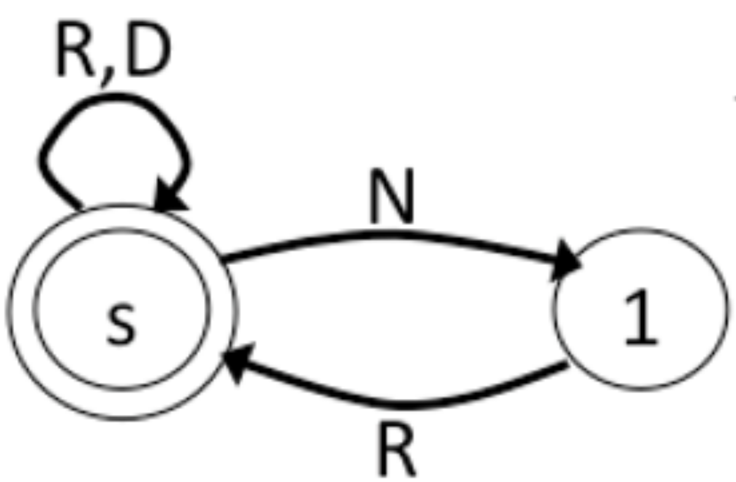
Regular and variants

- Multi-cost regular :

$$\text{REGULAR}([X_1,\dots,X_n],A)\ \wedge\ (\textstyle\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r = 0,\dots,R)$$

- Example:
  - Schedule 7 shifts of type: **night (N), day (D), rest (R)**
  - (1) "A **Rest must follow a Night** shift"
  - (2) "**Exactly 3 day shifts** and **1 night shift** must take place in the week"

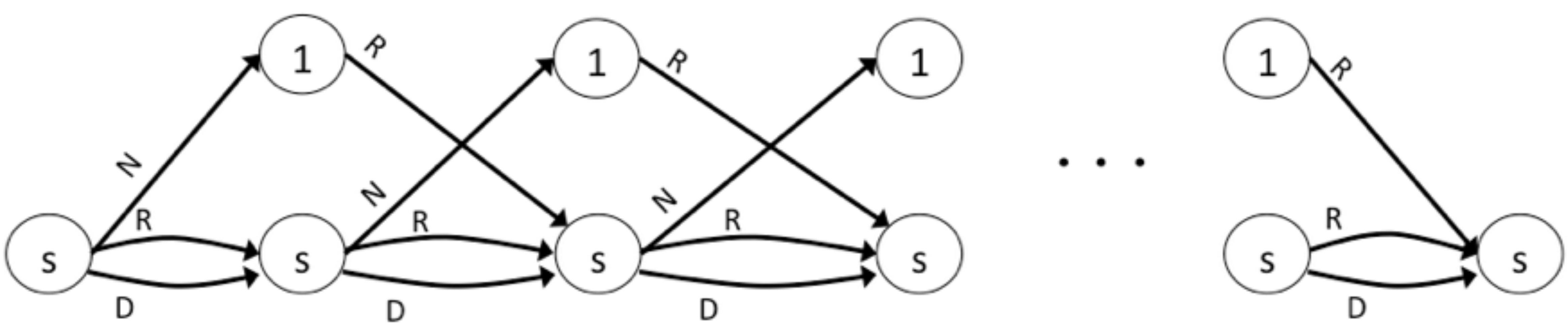
X_1	X_2	X_3	X_4	X_5	X_6	X_7
D	R	N	R	D	D	R



$R = 2$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^1_D	1	1	1	1	1	1	1
c^1_N	0	0	0	0	0	0	0
c^1_R	0	0	0	0	0	0	0

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^2_D	0	0	0	0	0	0	0
c^2_N	1	1	1	1	1	1	1
c^2_R	0	0	0	0	0	0	0



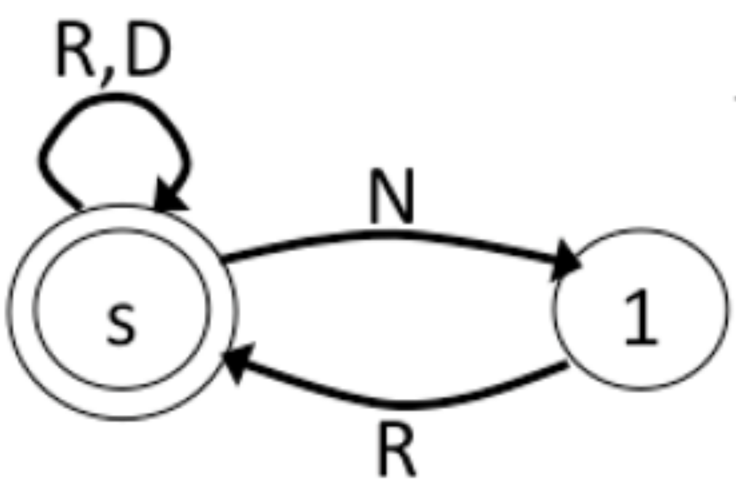
Regular and variants

- Multi-cost regular :

$$\textsc{Regular}([X_1,\ldots,X_n],A)\ \wedge\ (\textstyle\sum_{i=1}^{n} c^r_{iX_i} = Z^r, \forall\, r=0,\ldots,R)$$

- Example:
  - Schedule 7 shifts of type: **night (N), day (D), rest (R)**
  - (1) “A **Rest must follow a Night** shift”
  - (2) “**Exactly 3 day shifts** and **1 night shift** must take place in the week”

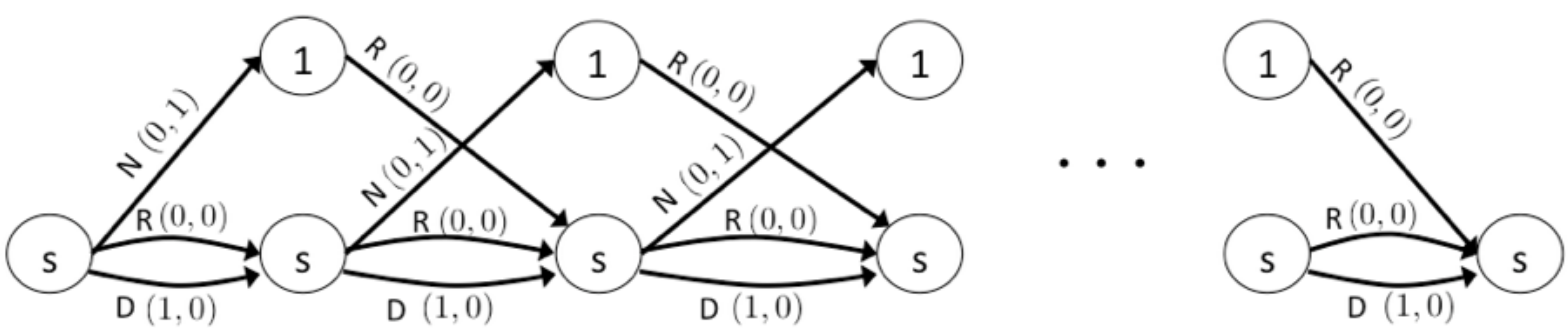
X_1	X_2	X_3	X_4	X_5	X_6	X_7
D	R	N	R	D	D	R



$R = 2$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^1_D	1	1	1	1	1	1	1
c^1_N	0	0	0	0	0	0	0
c^1_R	0	0	0	0	0	0	0

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
c^2_D	0	0	0	0	0	0	0
c^2_N	1	1	1	1	1	1	1
c^2_R	0	0	0	0	0	0	0



Dynamic programming for global constraints

- Linear equation
- General principles
- Regular and variants
- WeightedCircuit
- Reformulation of global constraints and MDD domains ?

Weighted Circuit

$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], z)$$

$next_i$: immediate successor of **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)

next variables must form a tour and $\sum_{i=1}^{n} d_{i,next_i} = z$

Weighted Circuit

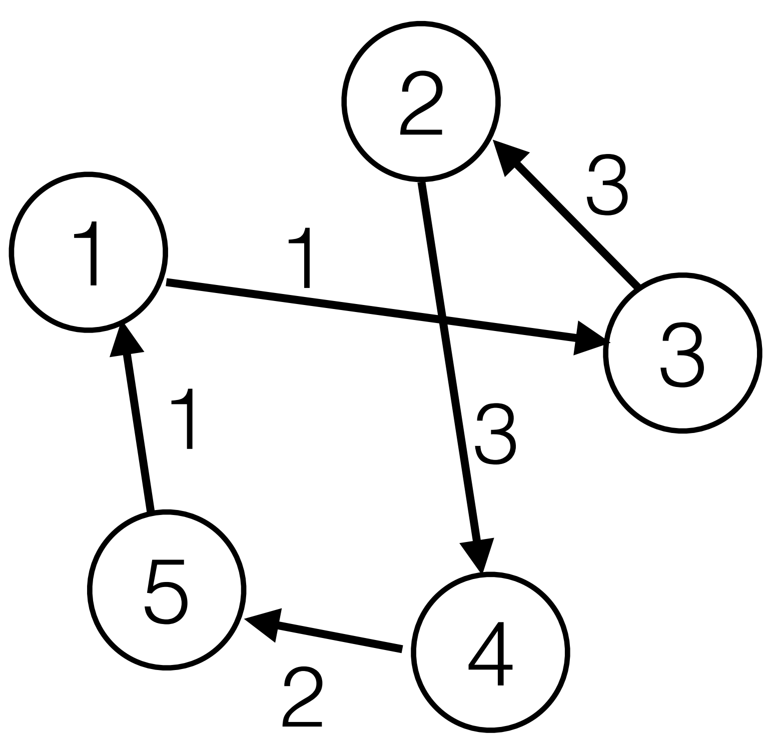
$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], z)$$

$next_i$: immediate successor of **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)

next variables must form a tour and $\sum_{i=1}^{n} d_{i,next_i} = z$



$$z = (1 + 3 + 3 + 2 + 1) = 10$$

$$next_1 = 3$$

$$next_3 = 2$$

$$\ldots$$

$$next_5 = 1$$

Weighted Circuit

$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], z)$$

$next_i$: immediate successor of **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)

next variables must form a tour and $\sum_{i=1}^{n} d_{i,next_i} = z$

- Filter the lower bound of z by solving a relaxation of the TSP
- Detect mandatory/forbidden arcs regarding the upper bound of z
- Applications in routing

Weighted Circuit

$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], z)$$

$next_i$: immediate successor of **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)

next variables must form a tour and $\sum_{i=1}^{n} d_{i,next_i} = z$

- Many problems involve side-constraints such as precedences, time-windows, vehicle capacity, … constraining the position of a city/client in the tour or relative positions of clients

- A useful variable for reasoning:

  pos_i : position of city **i** in the tour

Weighted Circuit

$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], [pos_1, \ldots, pos_n], z)$$

$next_i$: immediate successor of **i** in the tour

pos_i : position of city **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)

Weighted Circuit

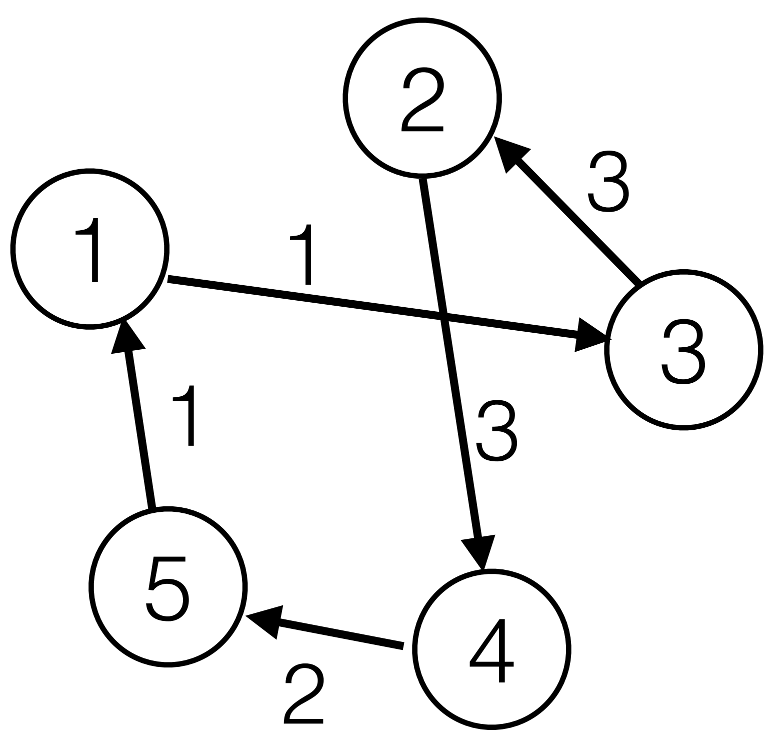
$$\textsc{WeightedCircuit}([next_1, \ldots, next_n], [pos_1, \ldots, pos_n], z)$$

$next_i$: immediate successor of **i** in the tour

pos_i : position of city **i** in the tour

z : distance of the tour

d : matrix of distances. d_{ij} is the distance of arc (i,j)



$$z = (1 + 3 + 3 + 2 + 1) = 10$$

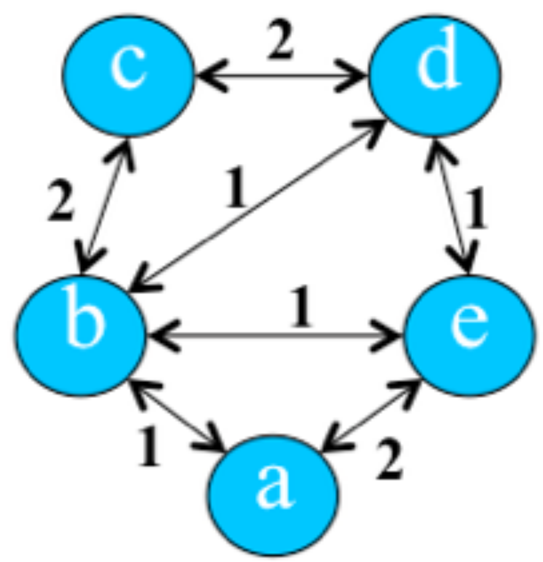
$$next_1 = 3$$
$$next_3 = 2$$
$$\ldots$$
$$next_5 = 1$$

$$pos_1 = 1$$
$$pos_2 = 3$$
$$pos_3 = 2$$
$$pos_4 = 4$$
$$pos_5 = 5$$

Relaxation of TSP to filter $\underline{z}$?

Weighted Circuit - TSP relaxations

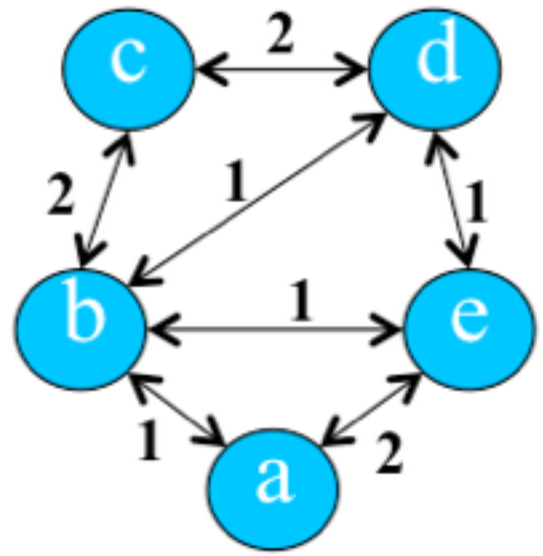
Weighted Circuit - TSP relaxations



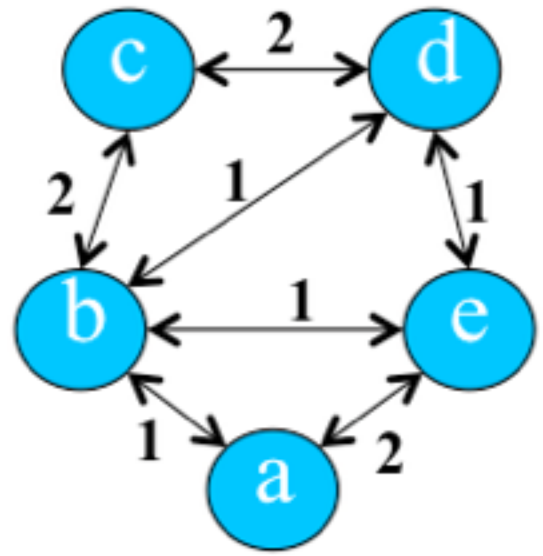
Weighted Circuit - TSP relaxations

Definition 1

- Connectivity
- Degree 2



Weighted Circuit - TSP relaxations

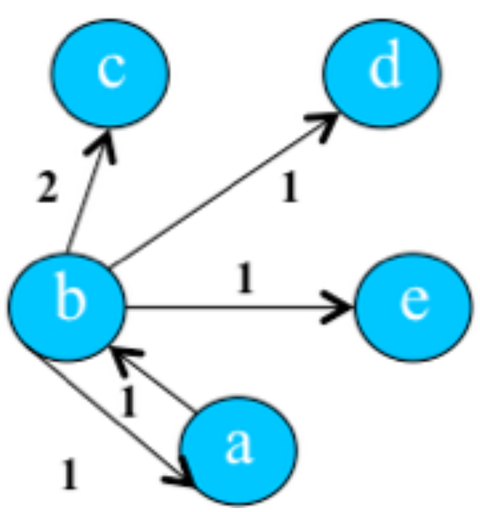


Definition 1

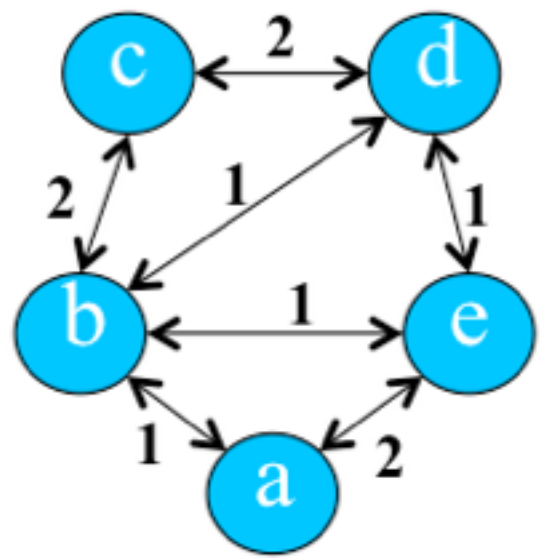
- Connectivity
- Degree 2

- Connectivity
- ~~Degree 2~~

One-Tree



Weighted Circuit - TSP relaxations

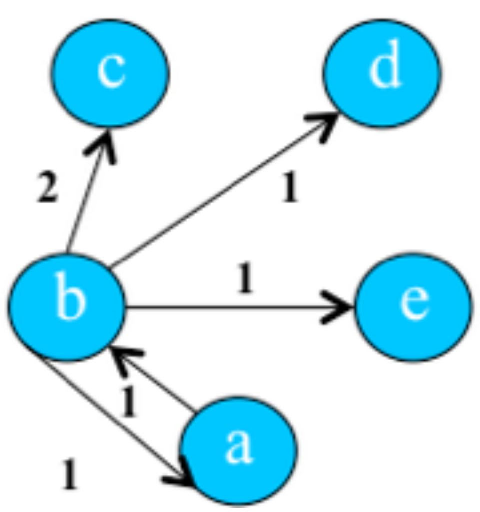


Definition 1

- Connectivity
- Degree 2

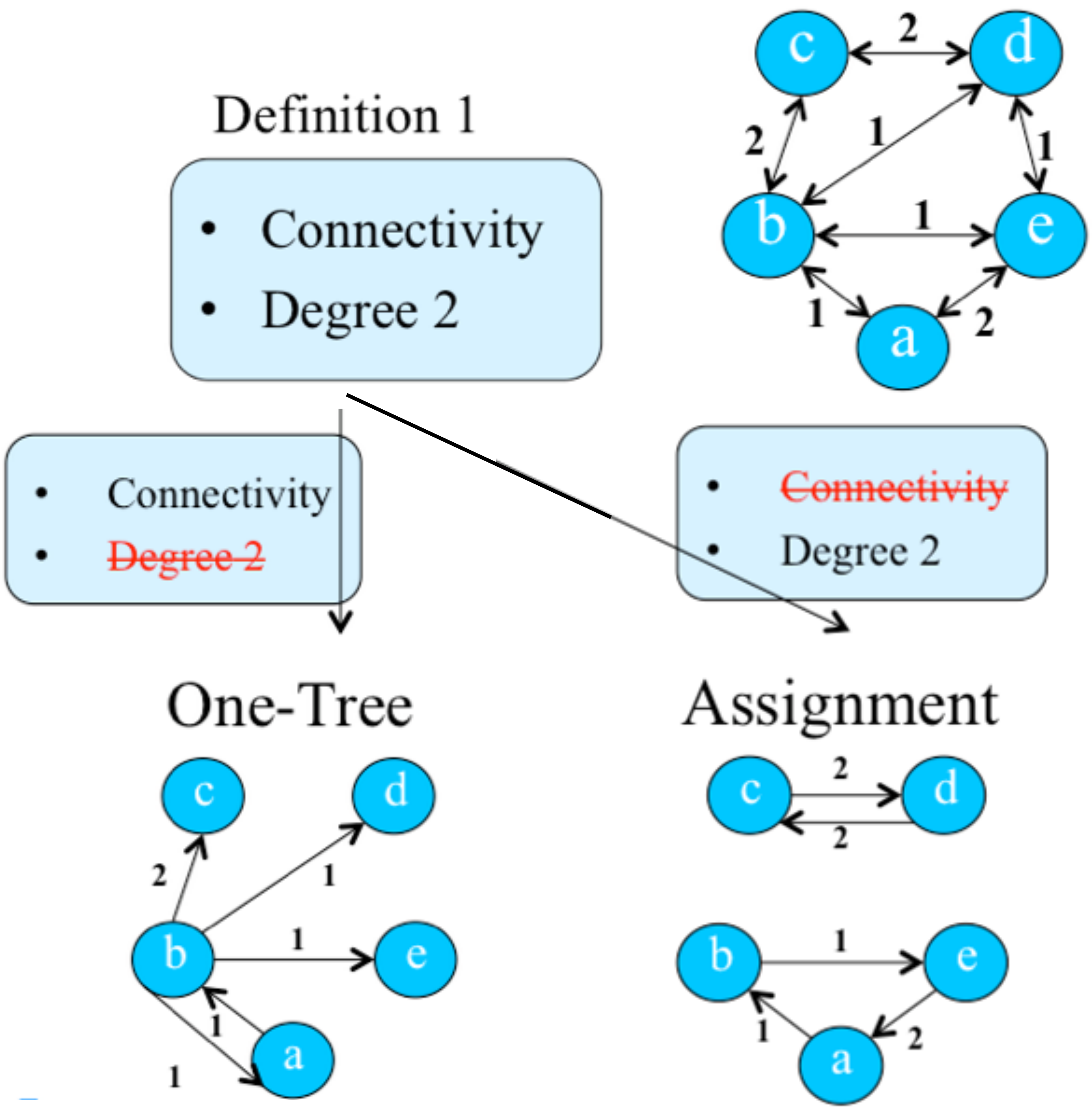
- Connectivity
- ~~Degree 2~~

One-Tree



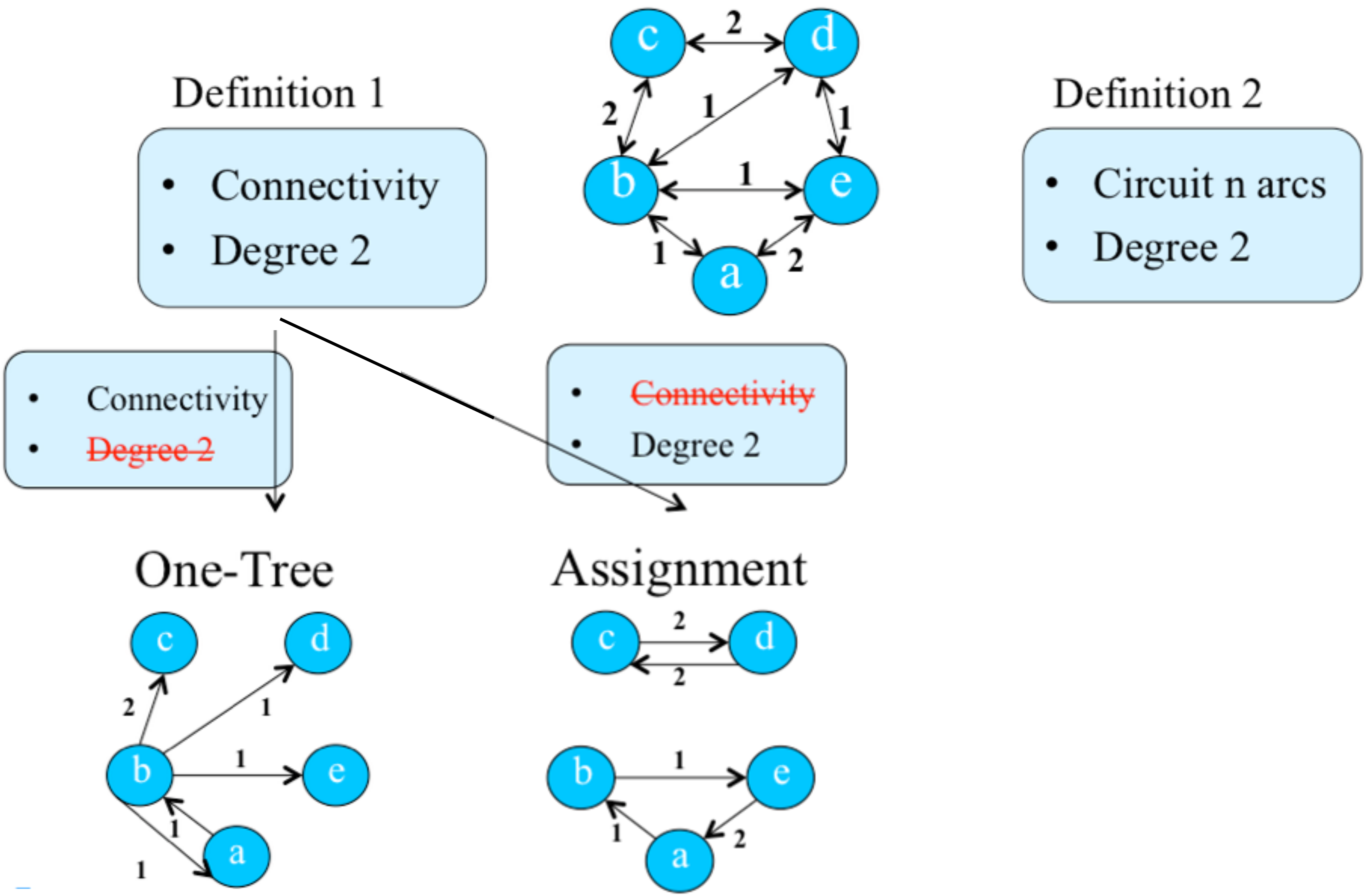
[Held and Karp. 1970]

Weighted Circuit - TSP relaxations



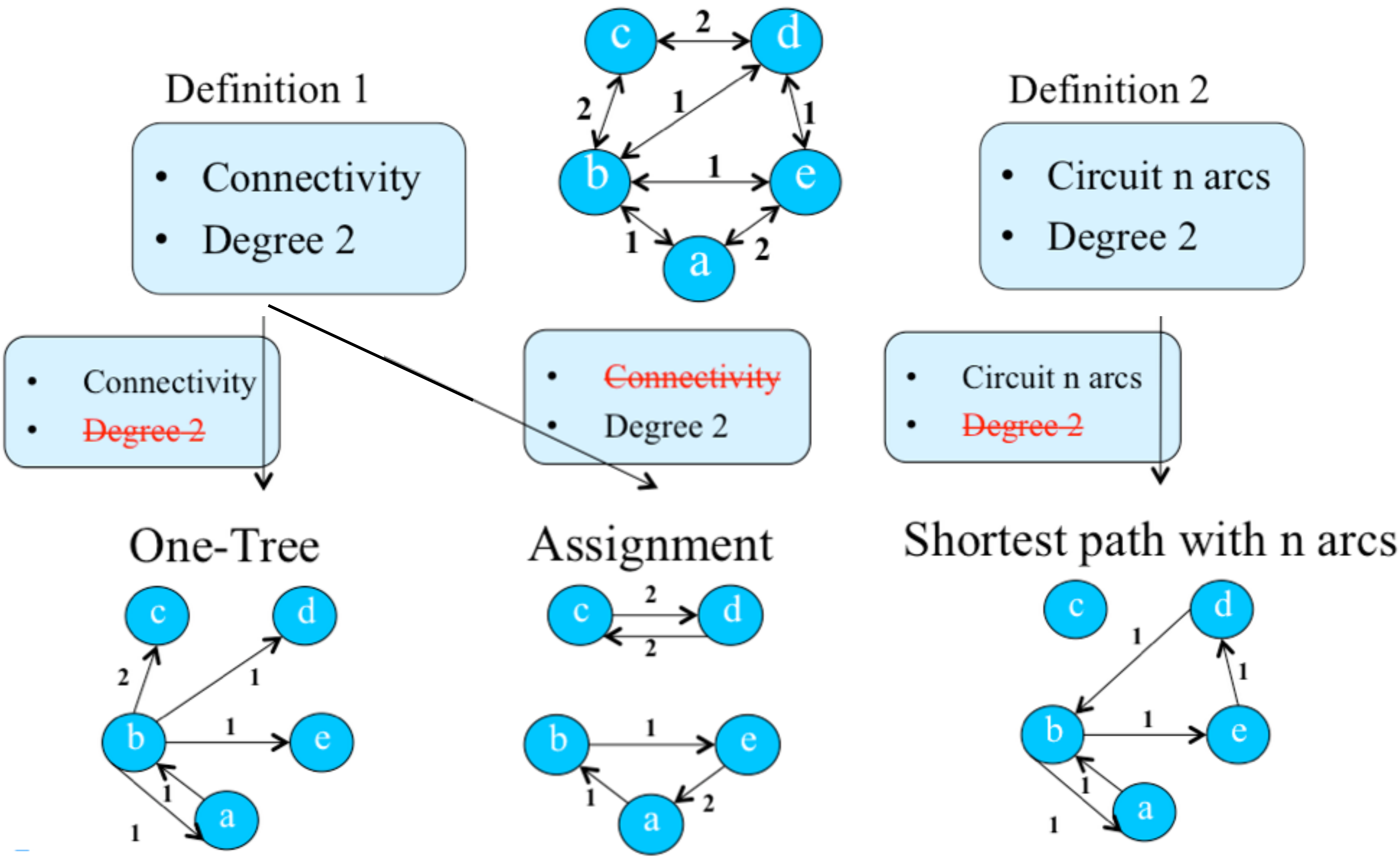
[Held and Karp. 1970]

Weighted Circuit - TSP relaxations



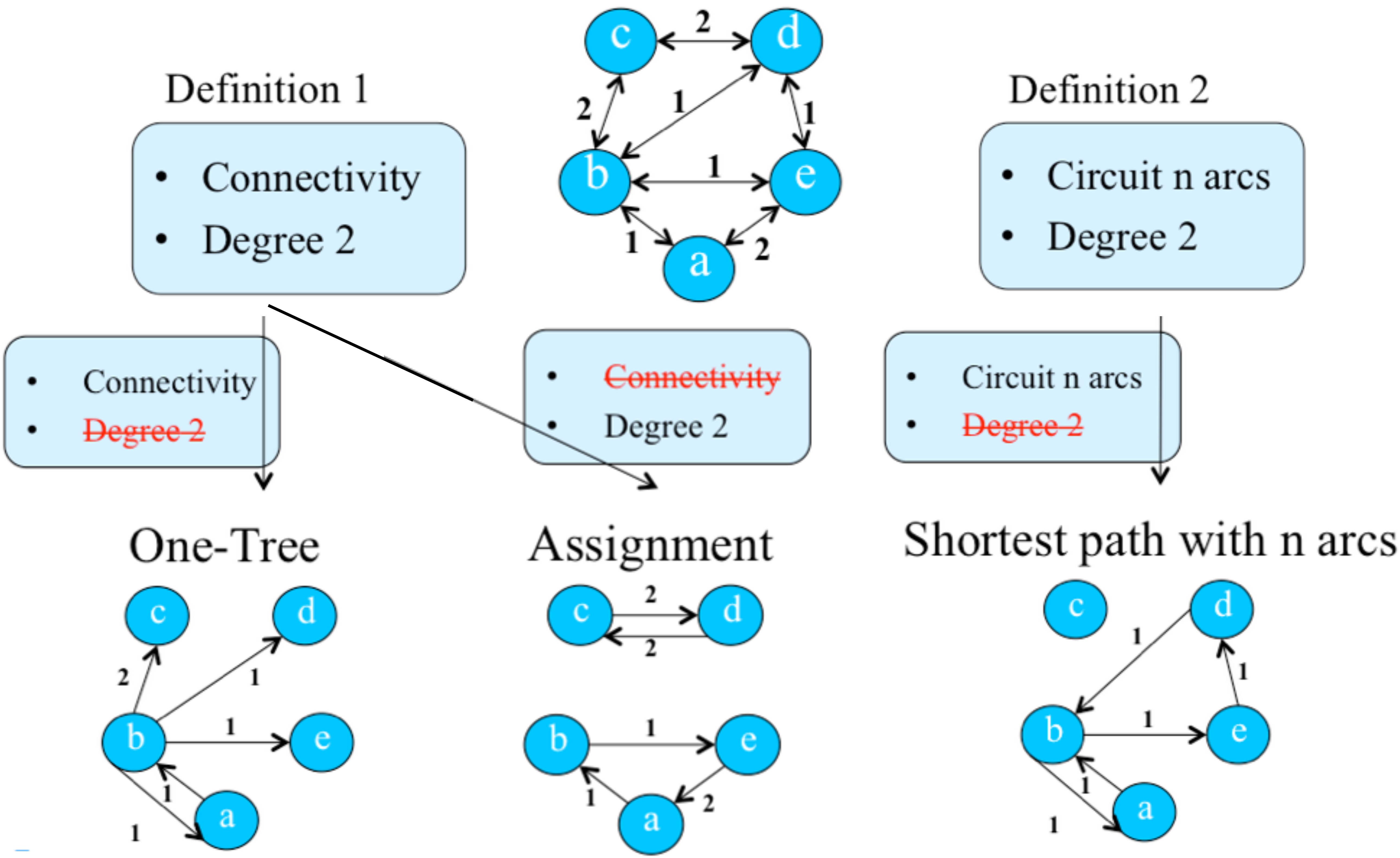
[Held and Karp. 1970]

Weighted Circuit - TSP relaxations



[Held and Karp. 1970]

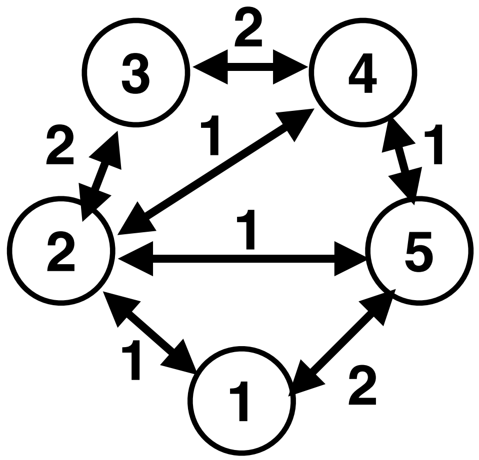
Weighted Circuit - TSP relaxations

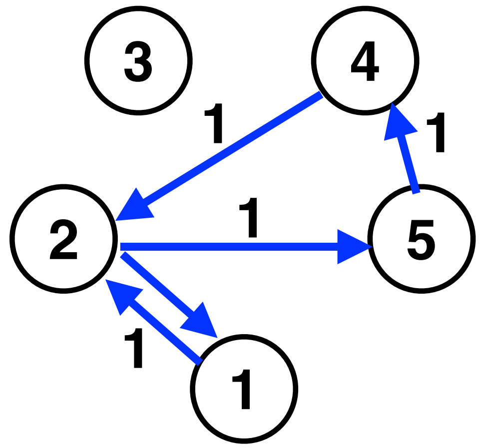


[Held and Karp. 1970]

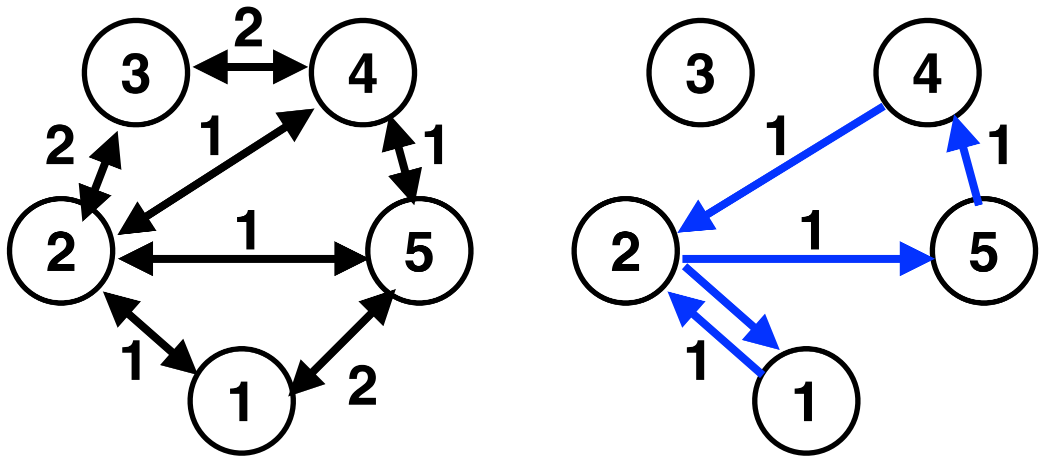
[Christophides et al. 1981]

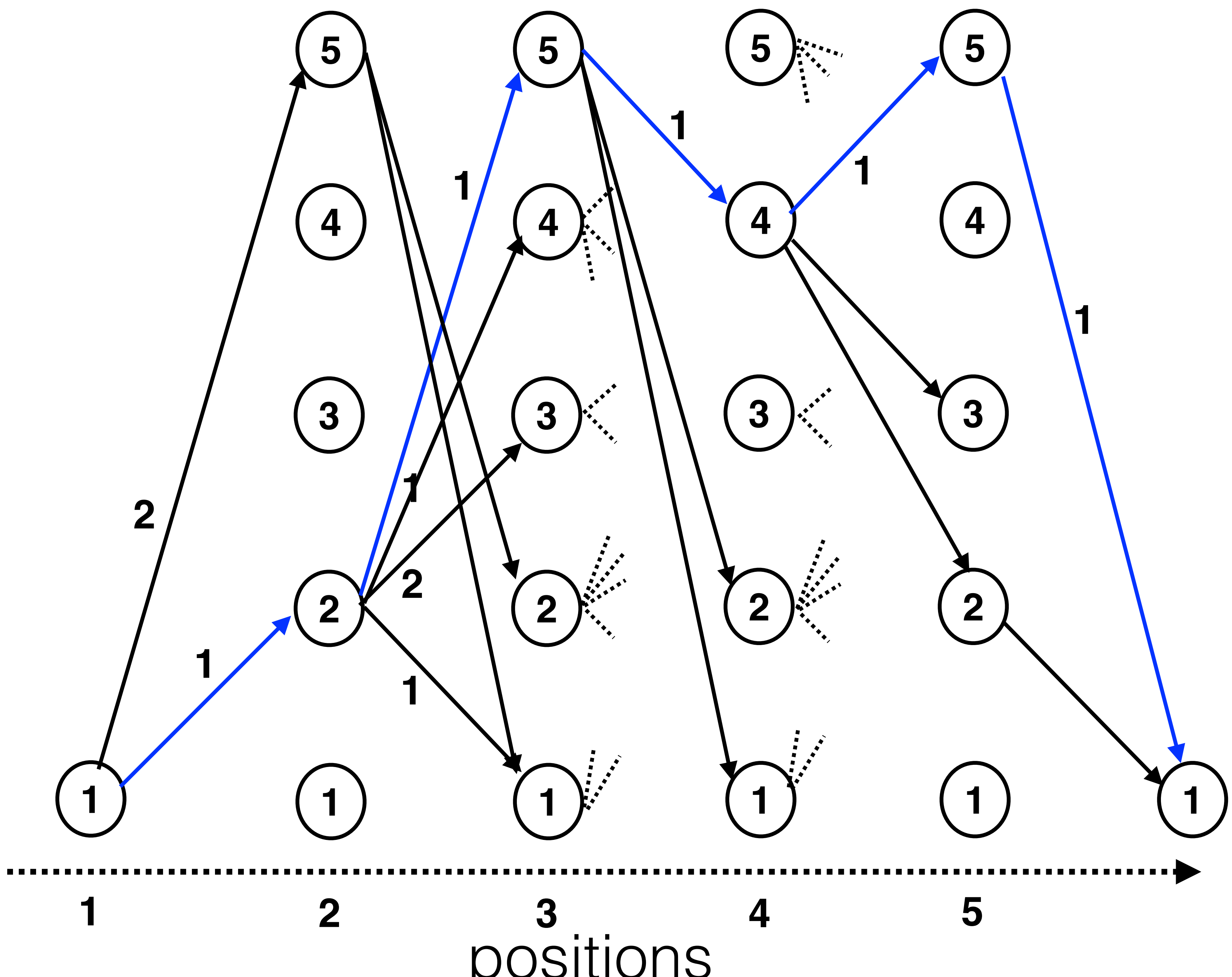
n-path relaxation



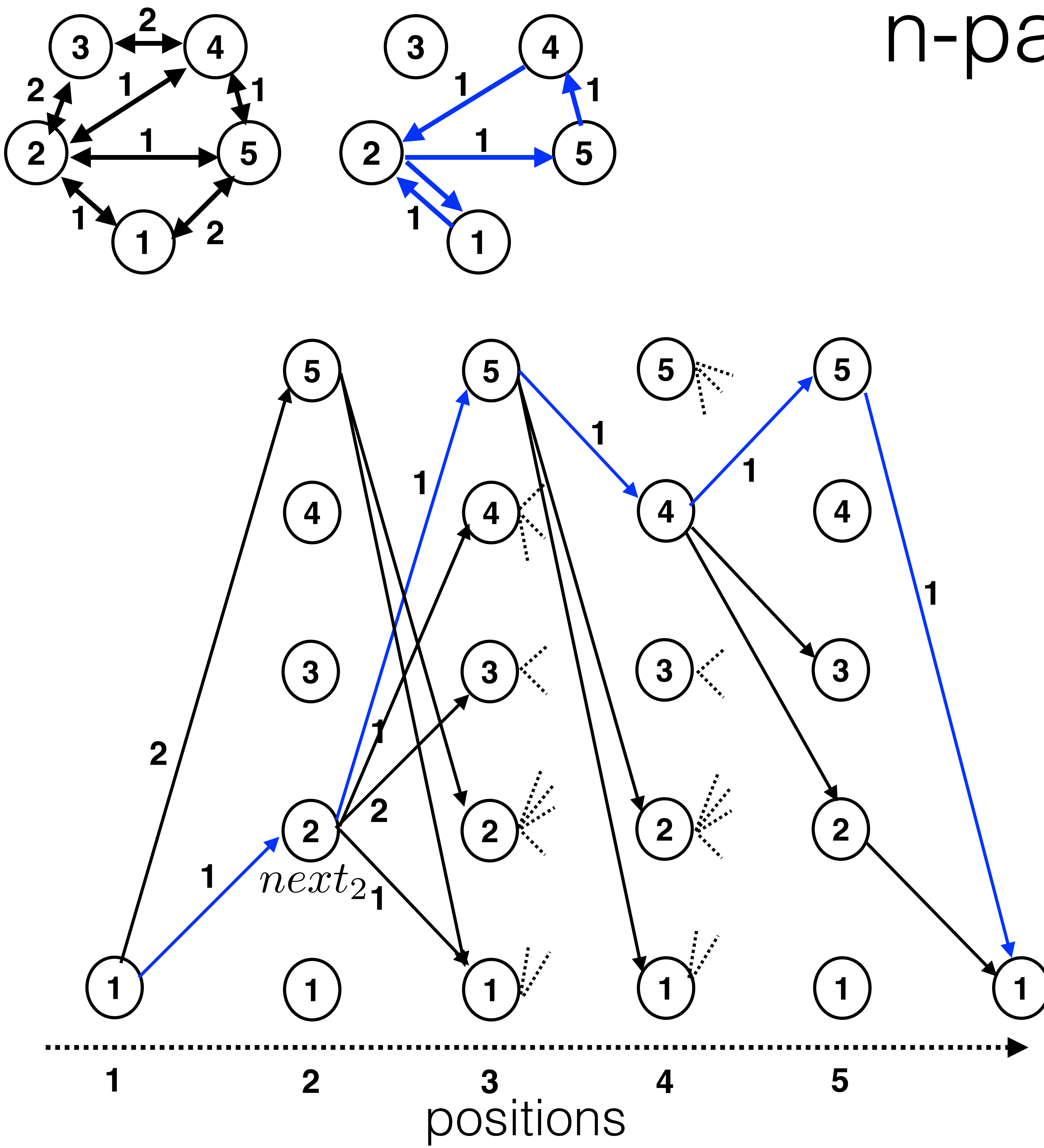


n-path relaxation

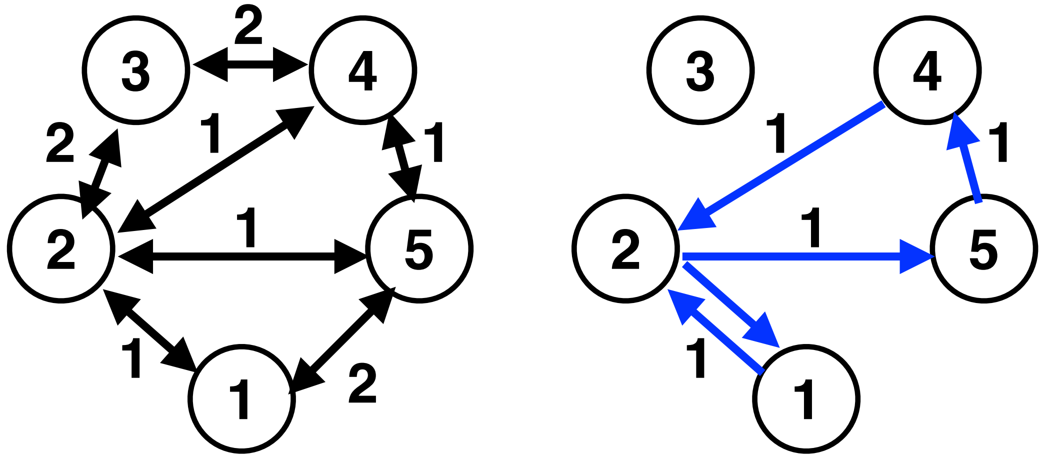




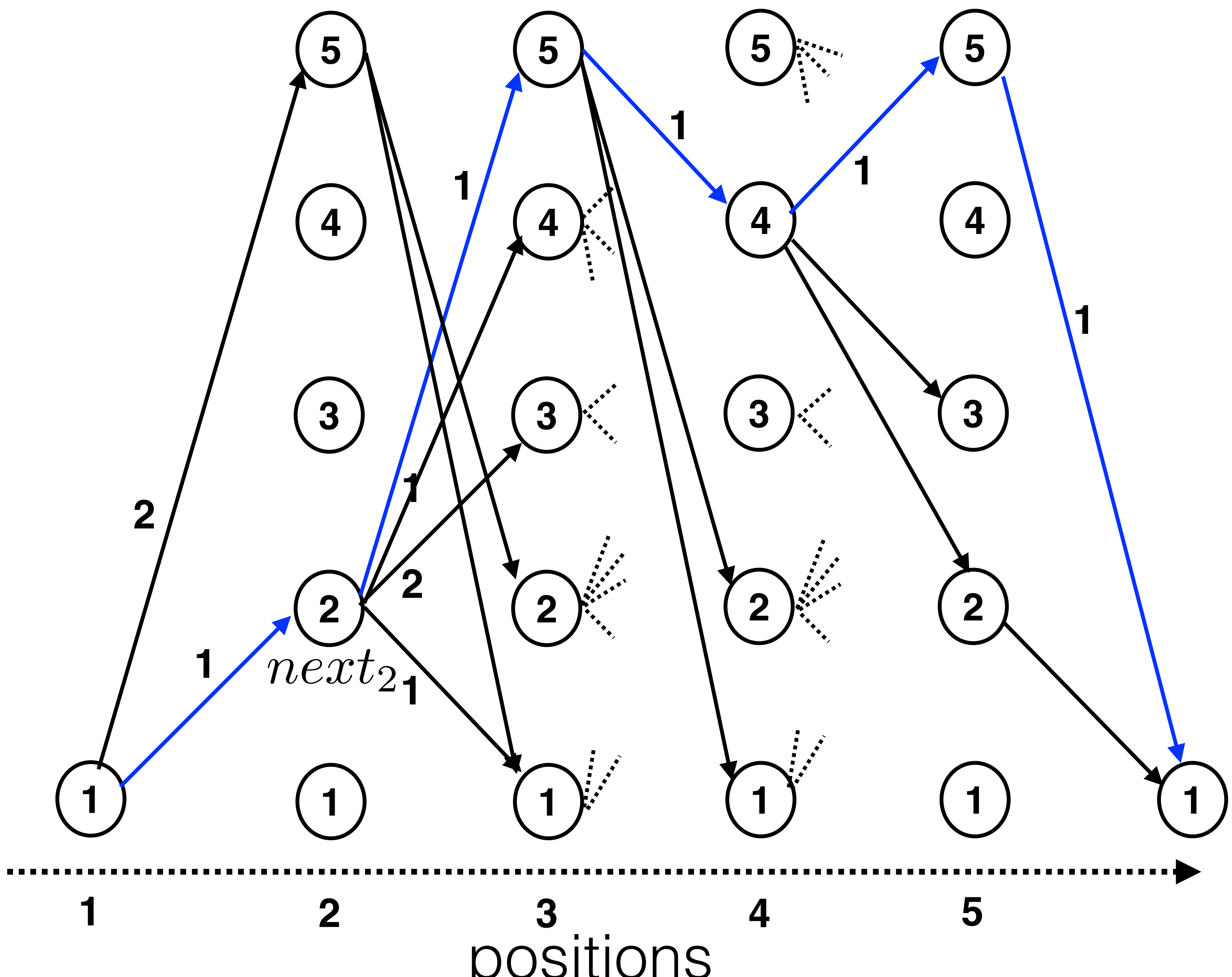
n-path relaxation



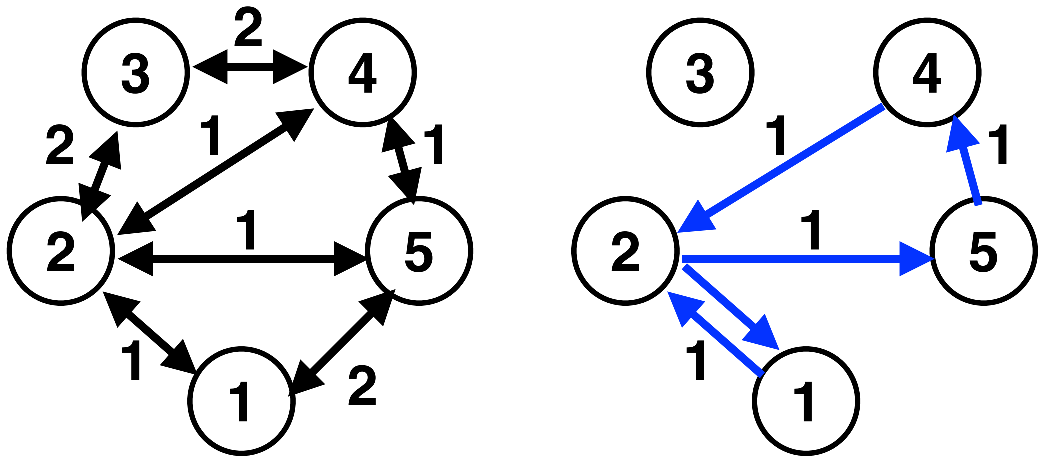
n-path relaxation



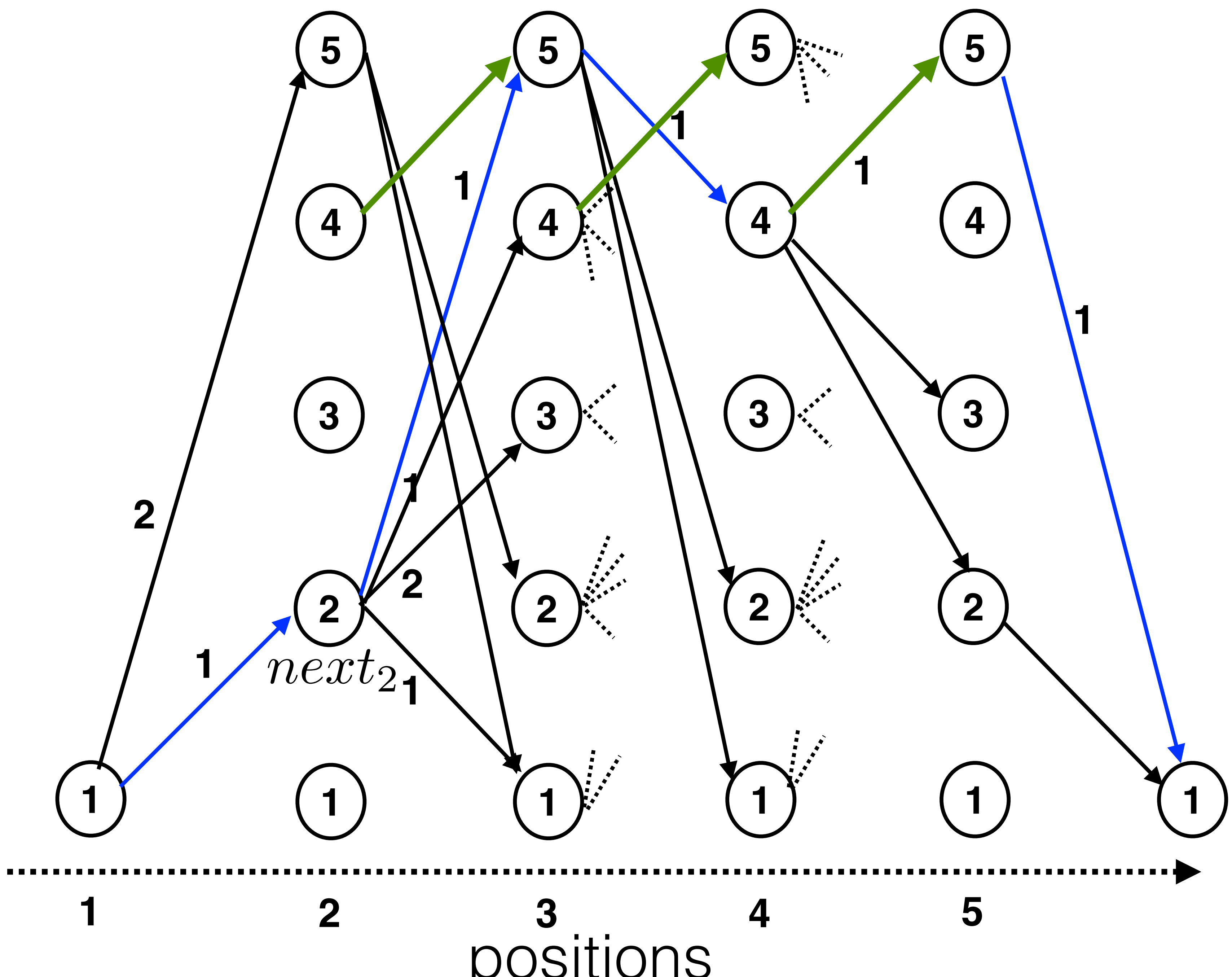
support of $\underline{z}$ = a shortest path



n-path relaxation

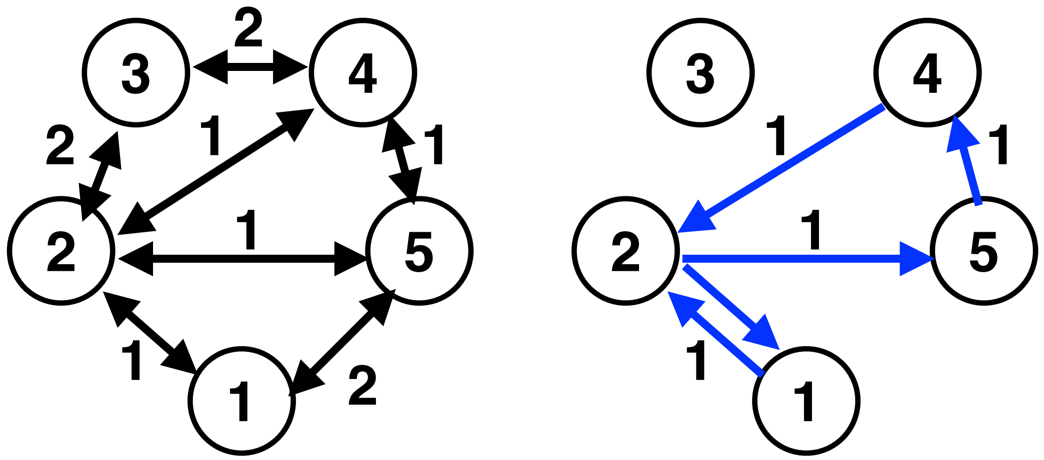


support of $\underline{z}$ = a shortest path



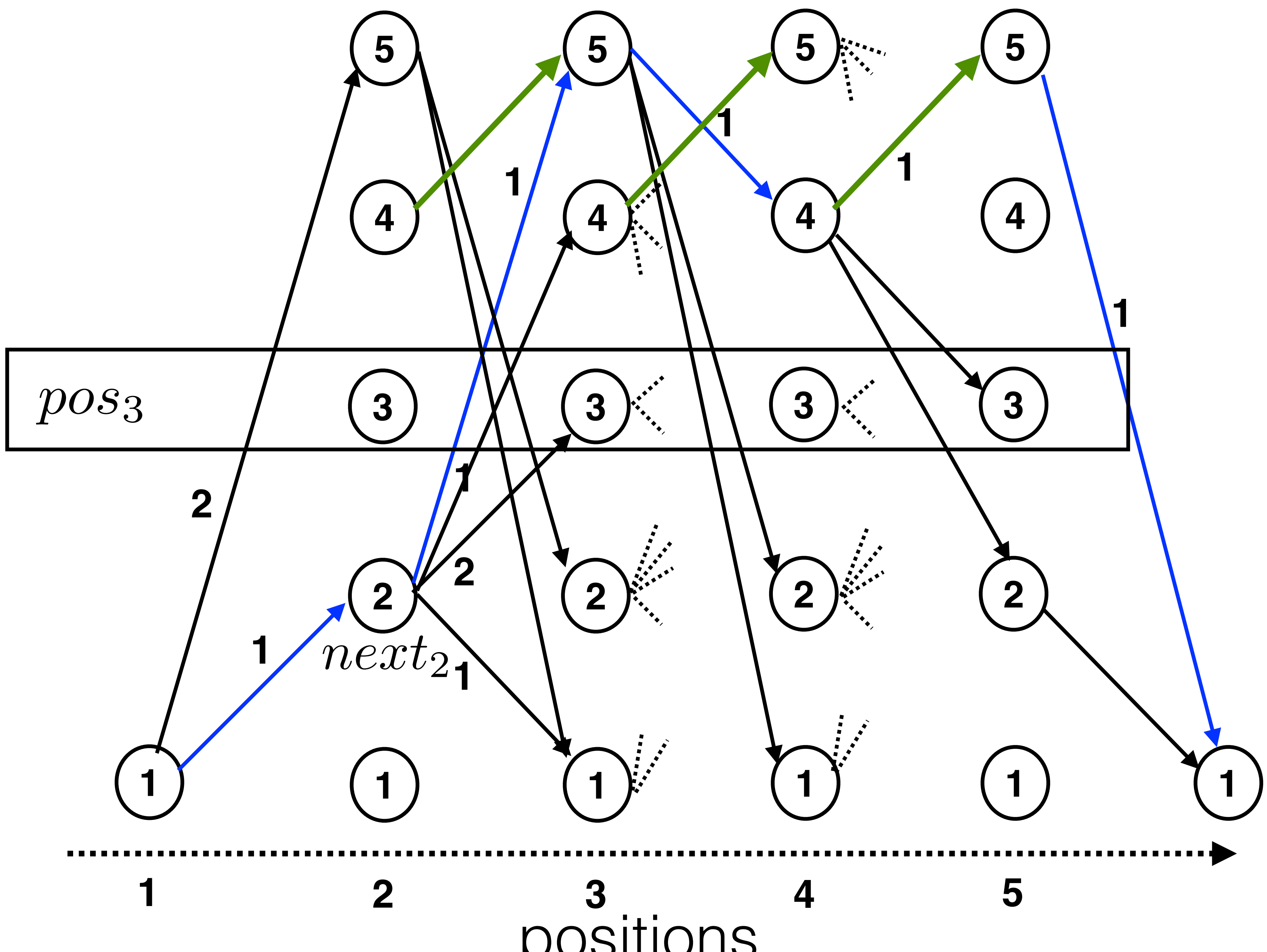
value 5 of $next_4$

n-path relaxation

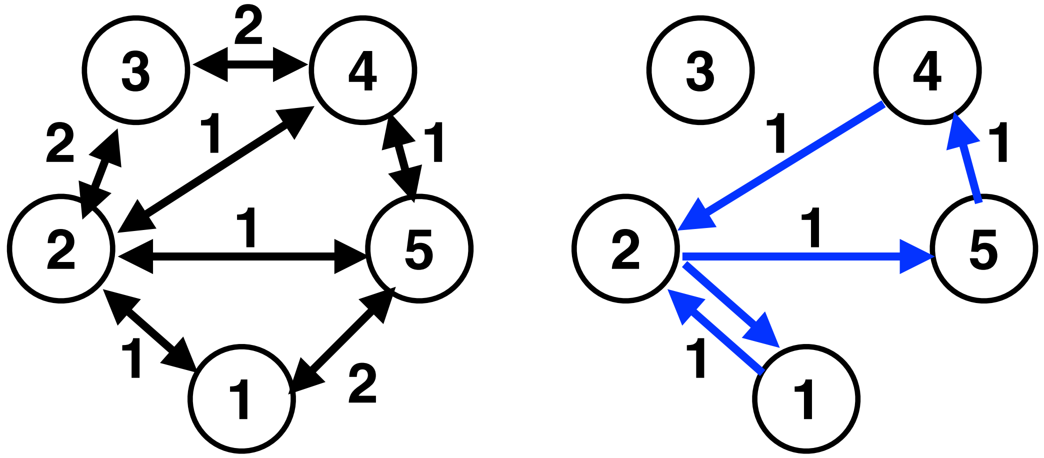


support of $\underline{z}$ = a shortest path

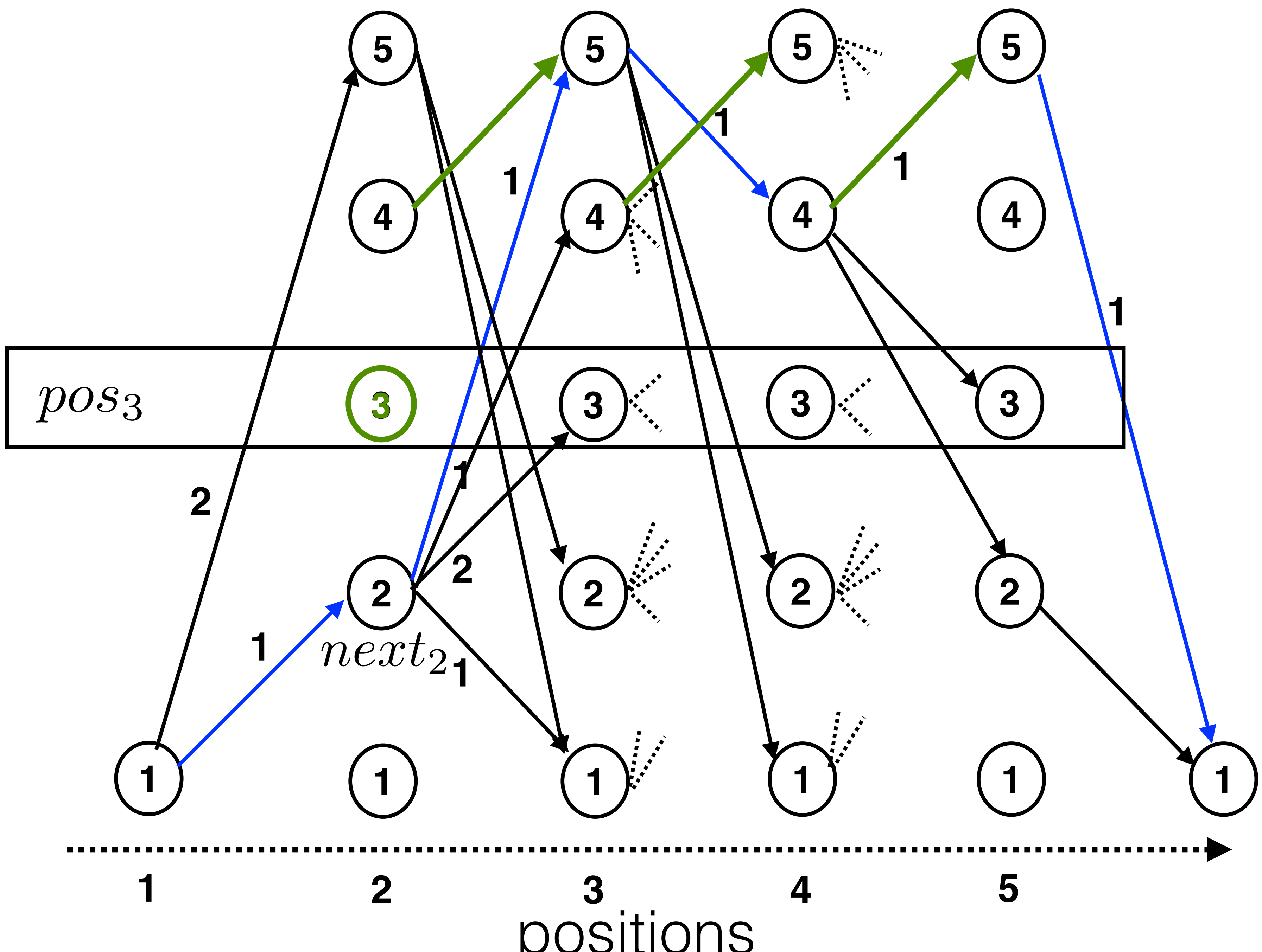
value 5 of $next_4$



n-path relaxation



support of $\underline{z}$ = a shortest path



value 5 of $next_4$

value 2 of pos_3

n-path relaxation

n-path relaxation: a circuit of n-arcs

$f^*(k, i)$: length of an optimal path starting from **1** and reaching **i** in **exactly k** arcs.

We are looking for $f^*(n, 1)$

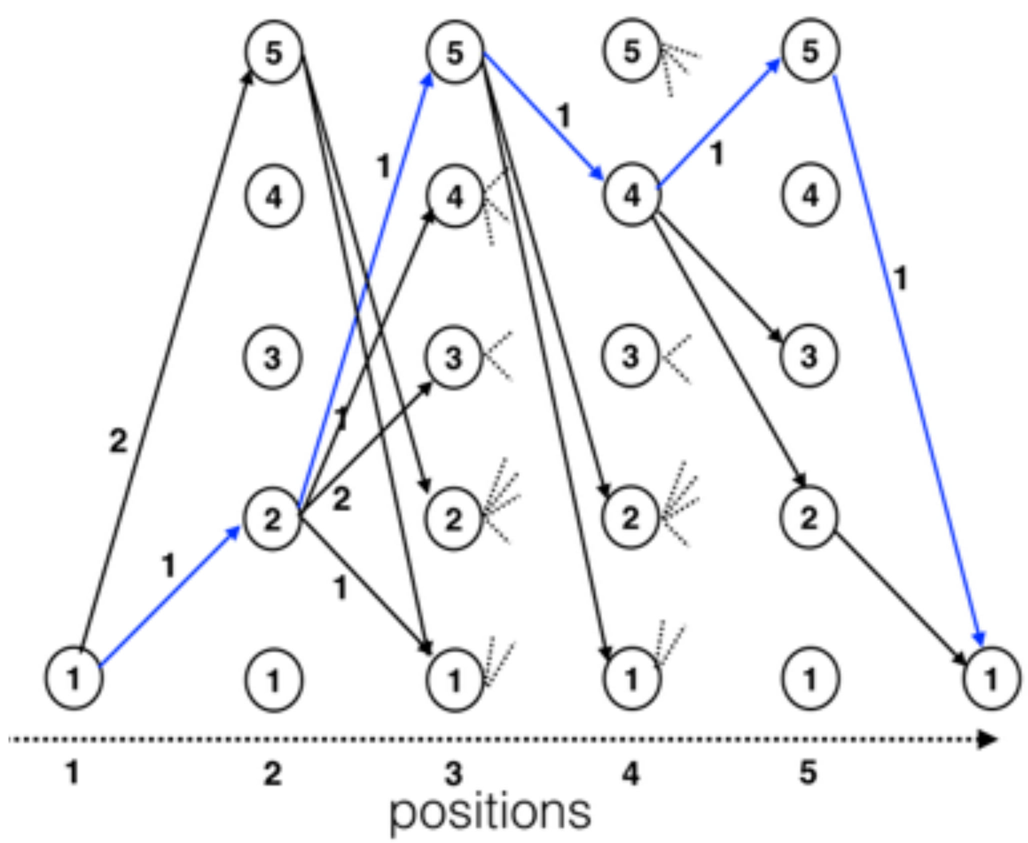
n-path relaxation

n-path relaxation: a circuit of n-arcs

$f^*(k,i)$: length of an optimal path starting from **1** and reaching **i** in **exactly k** arcs.

We are looking for $f^*(n,1)$

$$f^*(k,i) = \min_{j \in \boxed{D(pred_i)}} (f^*(k-1,j) + d_{ji}) \quad \forall k, \forall i \text{ s.t } k \in \boxed{D(pos_i)}$$



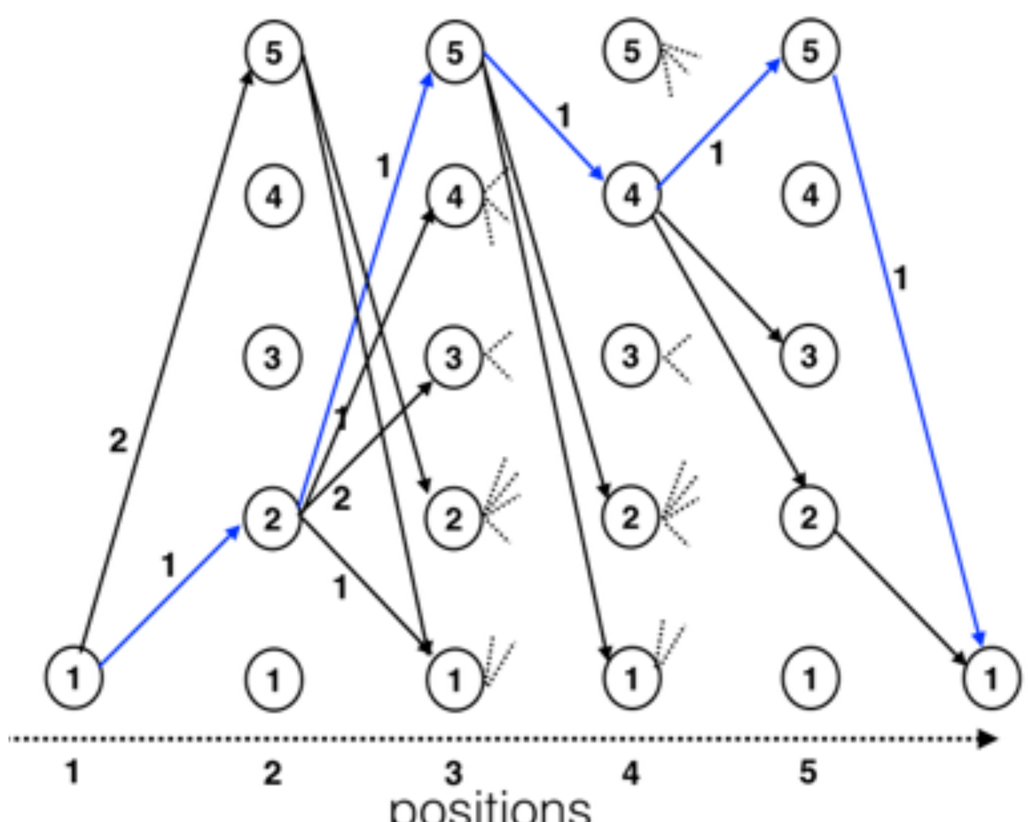
n-path relaxation

n-path relaxation: a circuit of n-arcs

$f^*(k,i)$: length of an optimal path starting from **1** and reaching **i** in **exactly k** arcs.

We are looking for $f^*(n,1)$

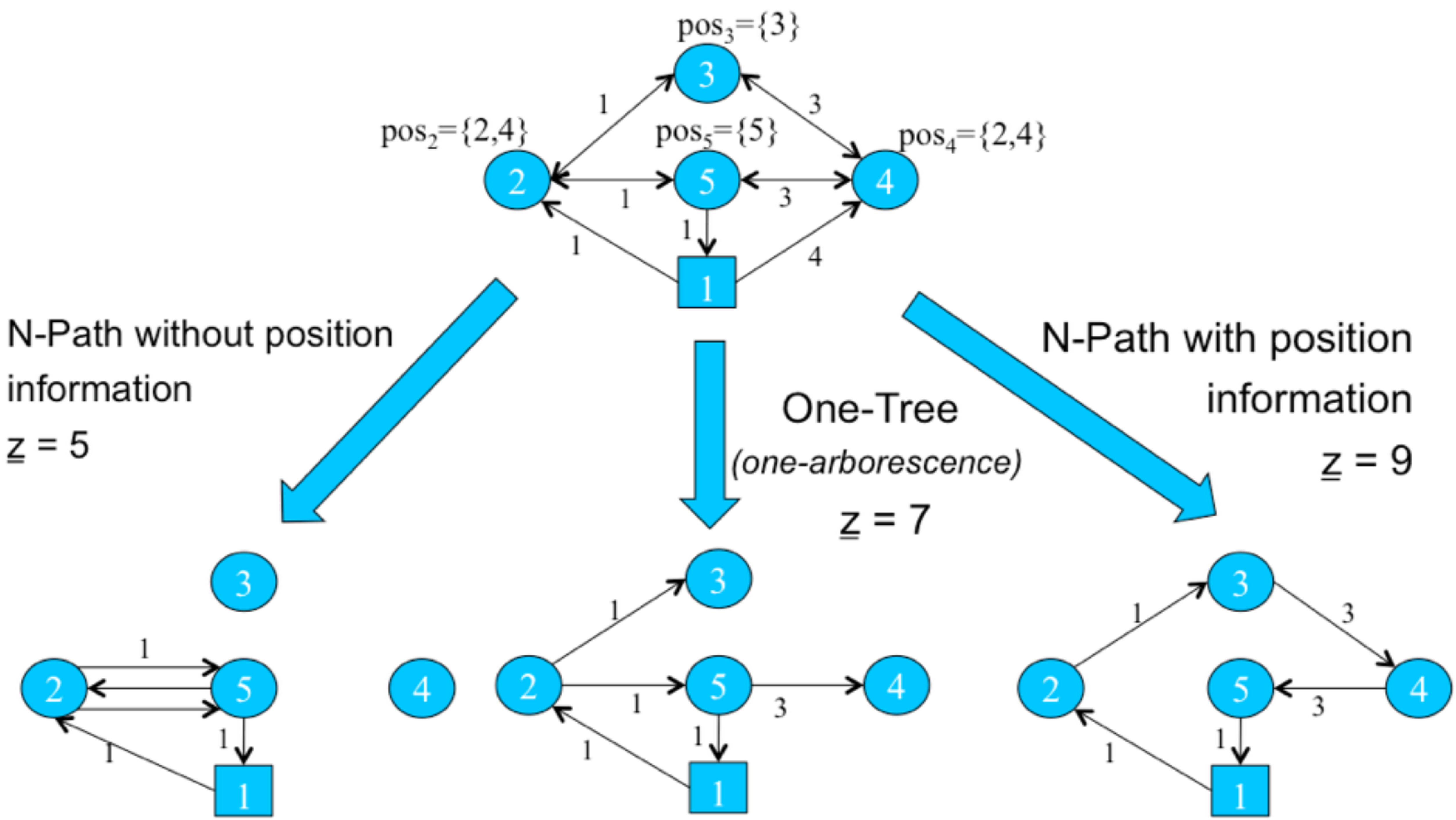
$$f^*(k,i) = \min_{j \in \boxed{D(pred_i)}} (f^*(k-1,j) + d_{ji}) \quad \forall k, \forall i \text{ s.t } k \in \boxed{D(pos_i)}$$



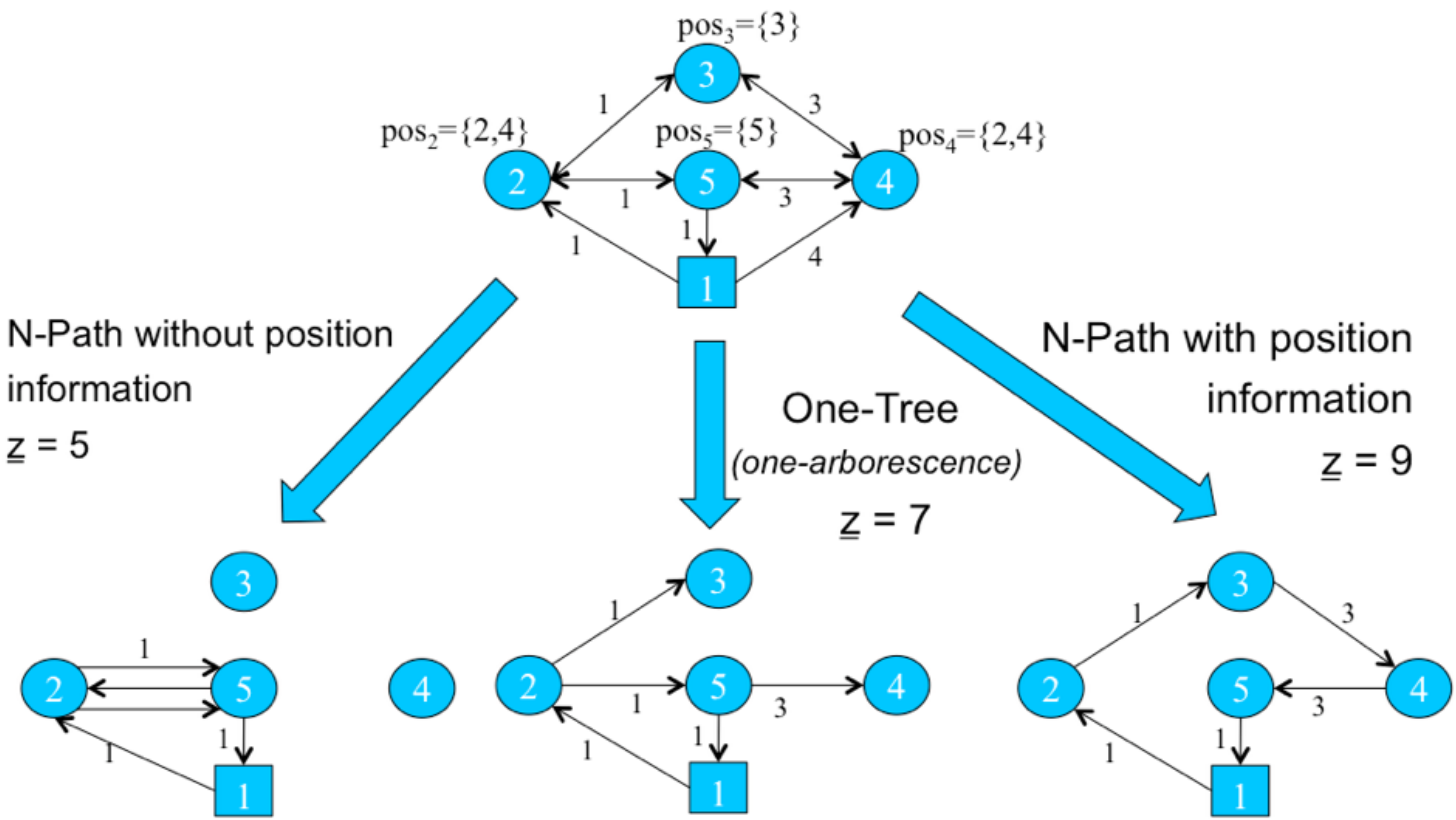
Filtering of both successors and positions

Complexity in $O(n^3)$

one-tree versus n-path



one-tree versus n-path



[Ducomman et al. 2016]

Dynamic programming for global constraints

- Linear equation
- General principles
- Regular and variants
- WeightedCircuit
- Reformulation of global constraints and MDD domains ?

Reformulations of global constraints

- Reformulating global constraints with small arity constraints to simulate the DP algorithm with AC on the corresponding constraint network:
  - ★ Regular [Quimper and Walsh, 2007]
  - ★ Bound AllDifferent
  - ★ Bound GCC
    } [Bessiere et al. 2009]
  - ★ Slides [Bessiere et al. 2008]

Reformulations of global constraints

- Reformulating global constraints with small arity constraints to simulate the DP algorithm with AC on the corresponding constraint network:
  - ★ Regular [Quimper and Walsh, 2007]
  - ★ Bound AllDifferent
  - ★ Bound GCC

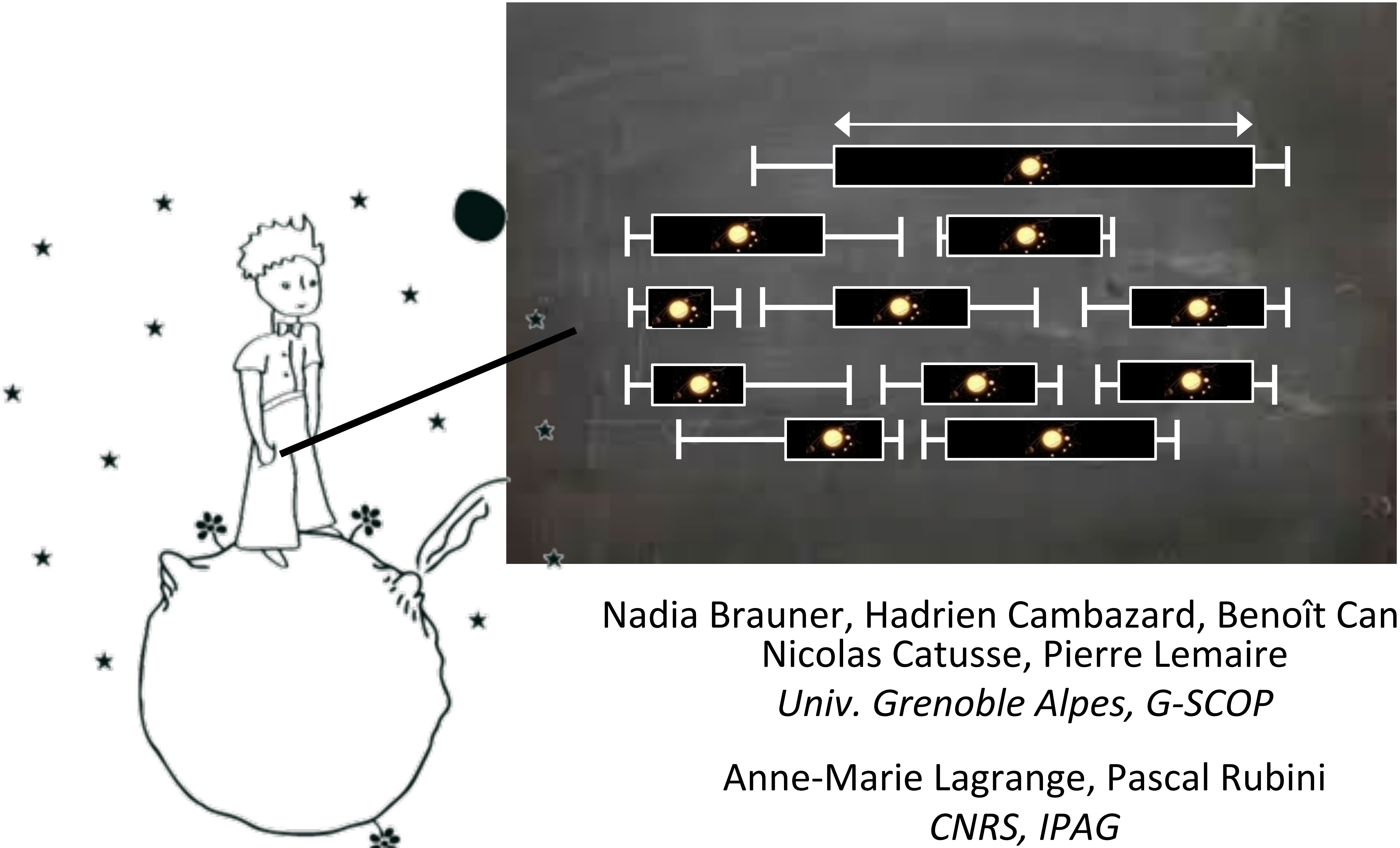
  } [Bessiere et al. 2009]
  - ★ Slides [Bessiere et al. 2008]

- MDD domains, a form of Dynamic programming ?
  - Multi-valued Decision Diagram MDD consistency
  - Explicit representation of more refined potential solution space [Hooker et al. 2007]
  - Limited width defines relaxation MDD
  - Overcome the current limit that : **« constraints are communicating through domains »**

Outline

1. **Reduced-costs based filtering**
    - Linear Programming duality
    - First example: *AtMostNValue*
        - Filtering the upper bound of a 0/1 variable
        - Filtering the lower bound of a 0/1 variable
    - General principles
    - Second example
    - Assignment, Cumulative, Bin-packing, …
2. **Dynamic programming filtering algorithms**
    - Linear equation, WeightedCircuit
    - General principles
    - Other relationships of DP and CP
3. **Illustration with a real-life application**

Star Scheduler



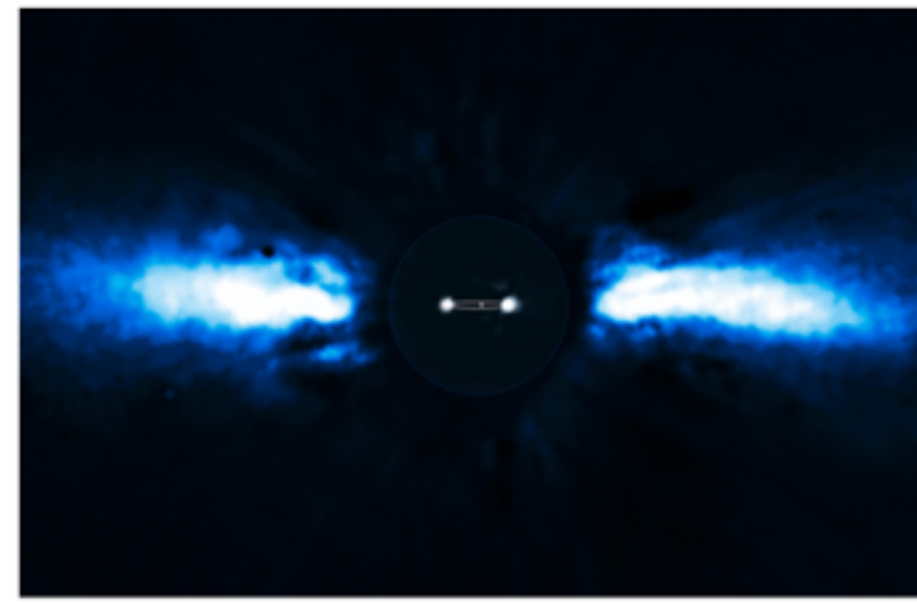
Nadia Brauner, Hadrien Cambazard, Benoît Cance,
Nicolas Catusse, Pierre Lemaire
Univ. Grenoble Alpes, G-SCOP

Anne-Marie Lagrange, Pascal Rubini
CNRS, IPAG

Star Scheduler

Planet that orbits a star $\neq$ sun

- Earth twin ?



$\approx$ **2000 planets discovered**

- A few dozens with direct imaging
- Some light years distance from earth
- million times less brilliant than their stars

New Observation tools:

VLT SPHERE

- Anne-Marie Lagrange
- Beta pictoris b (2008)

Star Scheduler

Extrasolar planet observation

From earth: the VLT (Chili)

The Astrophysicists

- Survey potential stars
- Book a fixed set of nights within the budget

About 100.000 euros a night

- Decide the observation schedule for each night to **maximize scientific interest**

Star Scheduler

Extrasolar planet observation

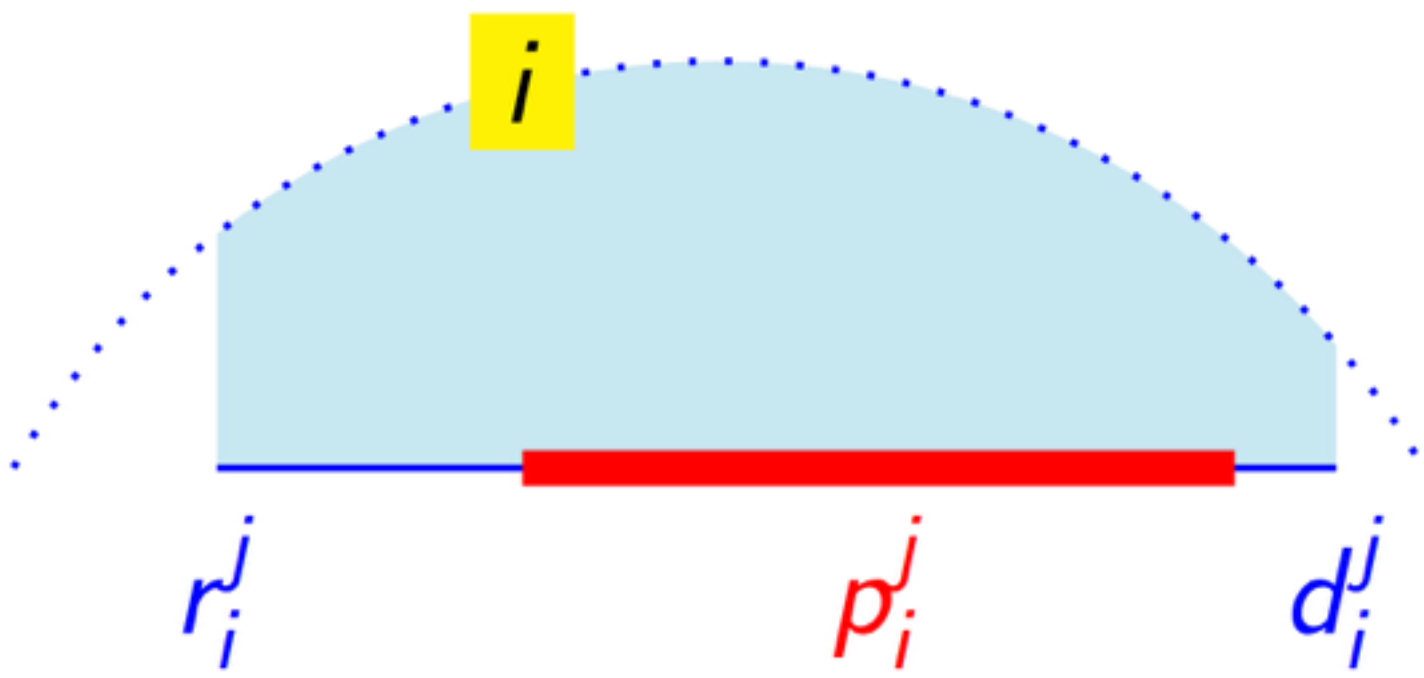
From earth: the VLT (Chili)

Main constraints

- Visibility period of the stars
- Position in the sky influence
  - Quality of the observation
  - Length of the observation
- Some stars are scientifically more important than others
- Calibration (runs, earthquake)

Star Scheduler

Observation *i* in night *j*

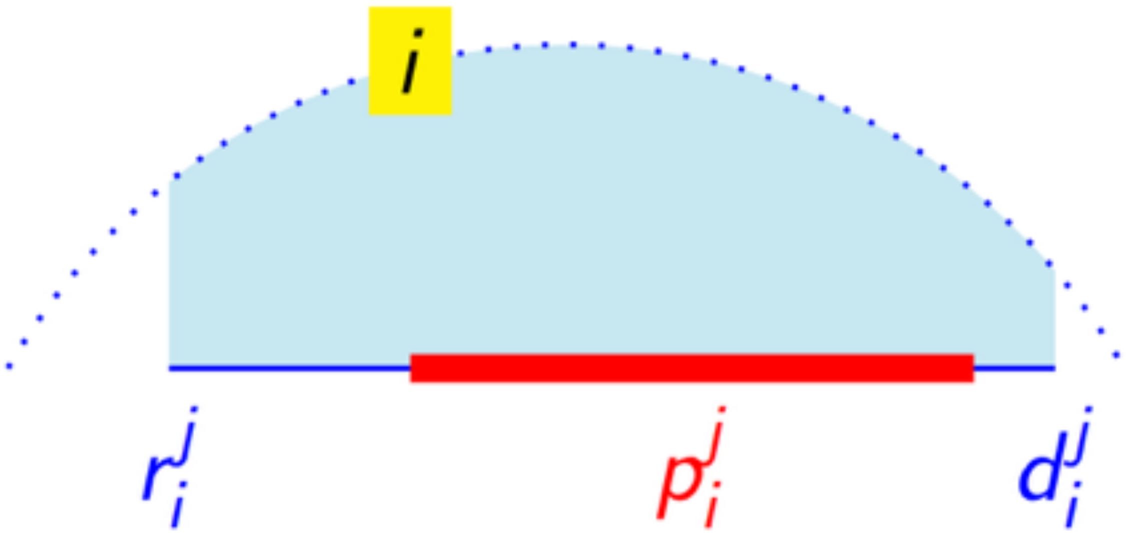


$[r_i^j, d_i^j[$: visibility interval

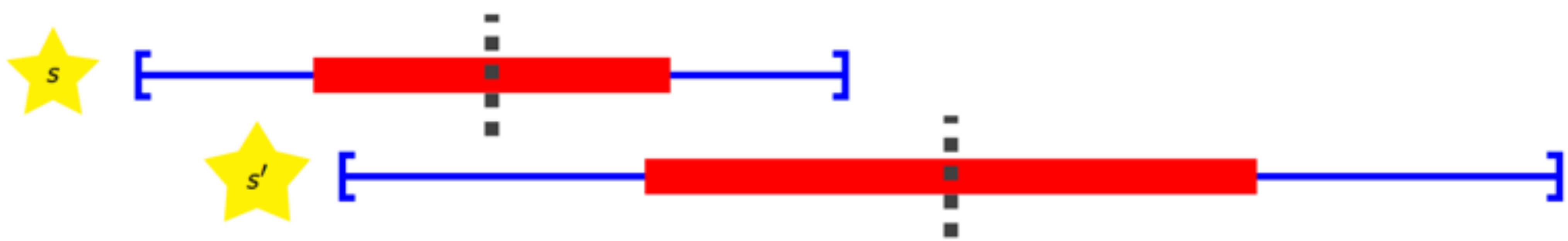
p_i^j : duration of the observation

w_i : scientific interest

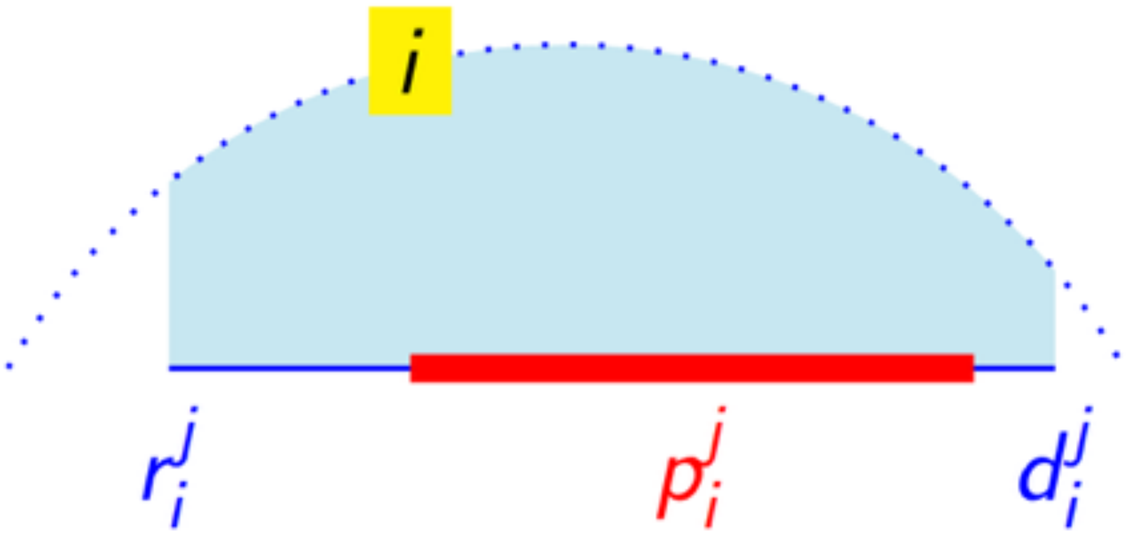
Star Scheduler



The meridian instant $m_i = \frac{d_i^j - r_i^j}{2}$ is a mandatory instant of observation, that is for every star **i:** $p_i^j \geq \frac{d_i^j - r_i^j}{2}$



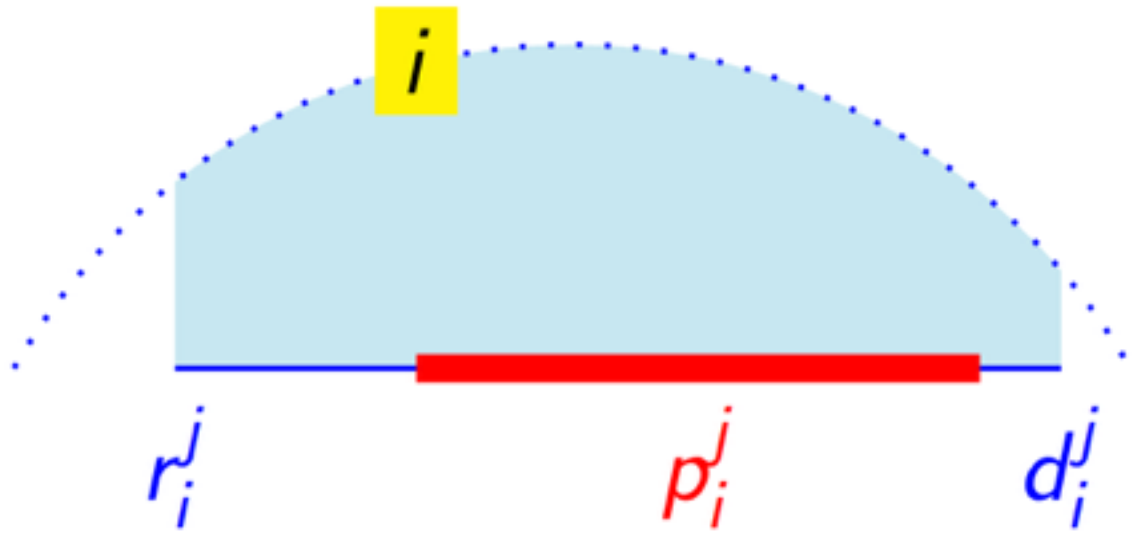
➡ The observations must be scheduled by non-decreasing meridian time



Star Scheduler

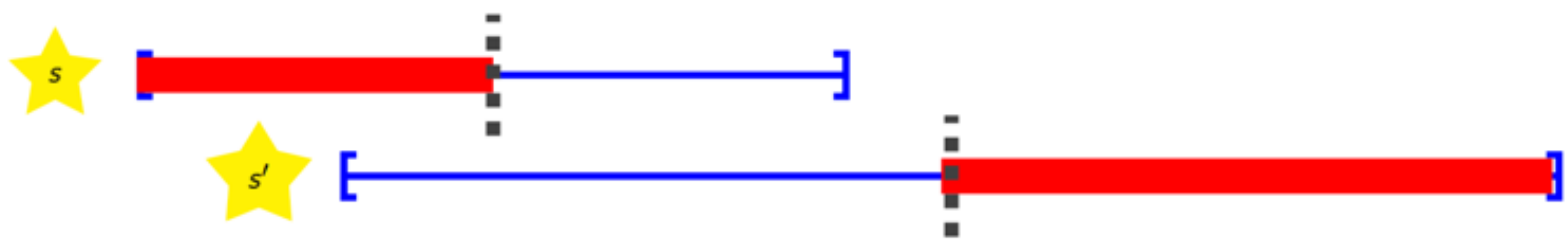
The meridian instant $m_i = \frac{d_i^j - r_i^j}{2}$ is a mandatory instant of observation, that is for every star **i:** $p_i^j \geq \frac{d_i^j - r_i^j}{2}$

The observations must be scheduled by non-decreasing meridian time



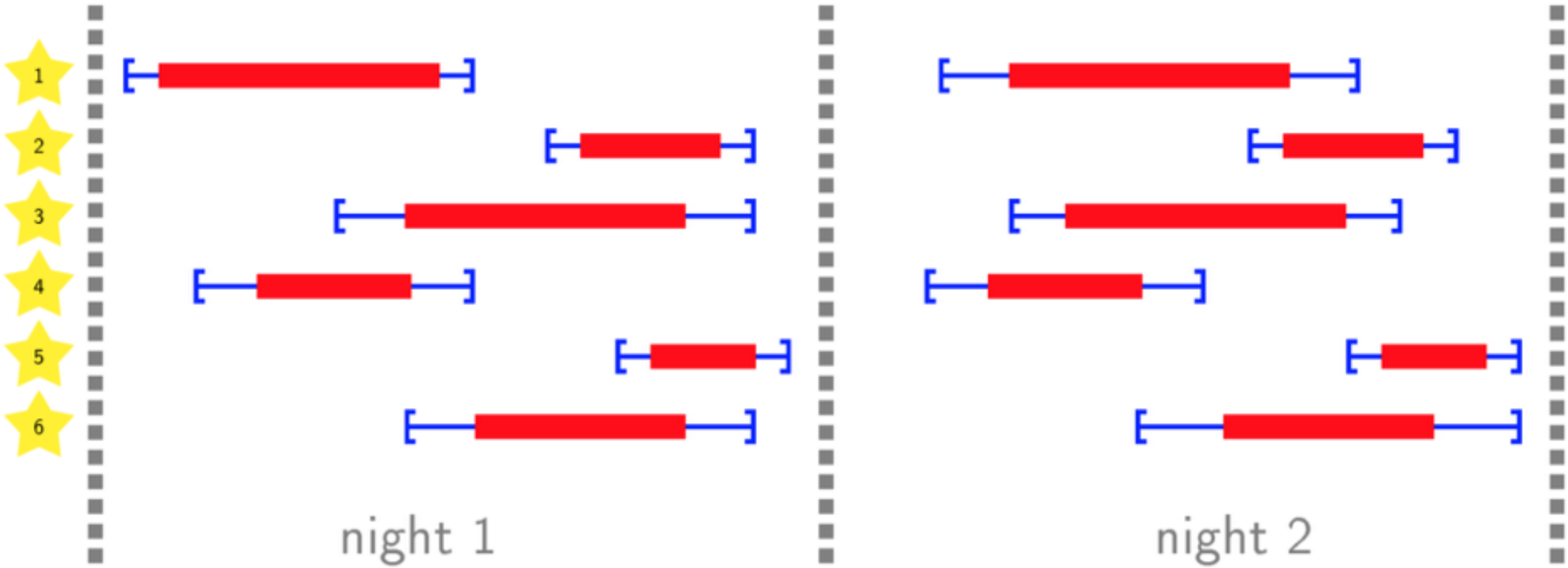
Star Scheduler

The meridian instant $m_i = \frac{d_i^j - r_i^j}{2}$ is a mandatory instant of observation, that is for every star **i:** $p_i^j \geq \frac{d_i^j - r_i^j}{2}$

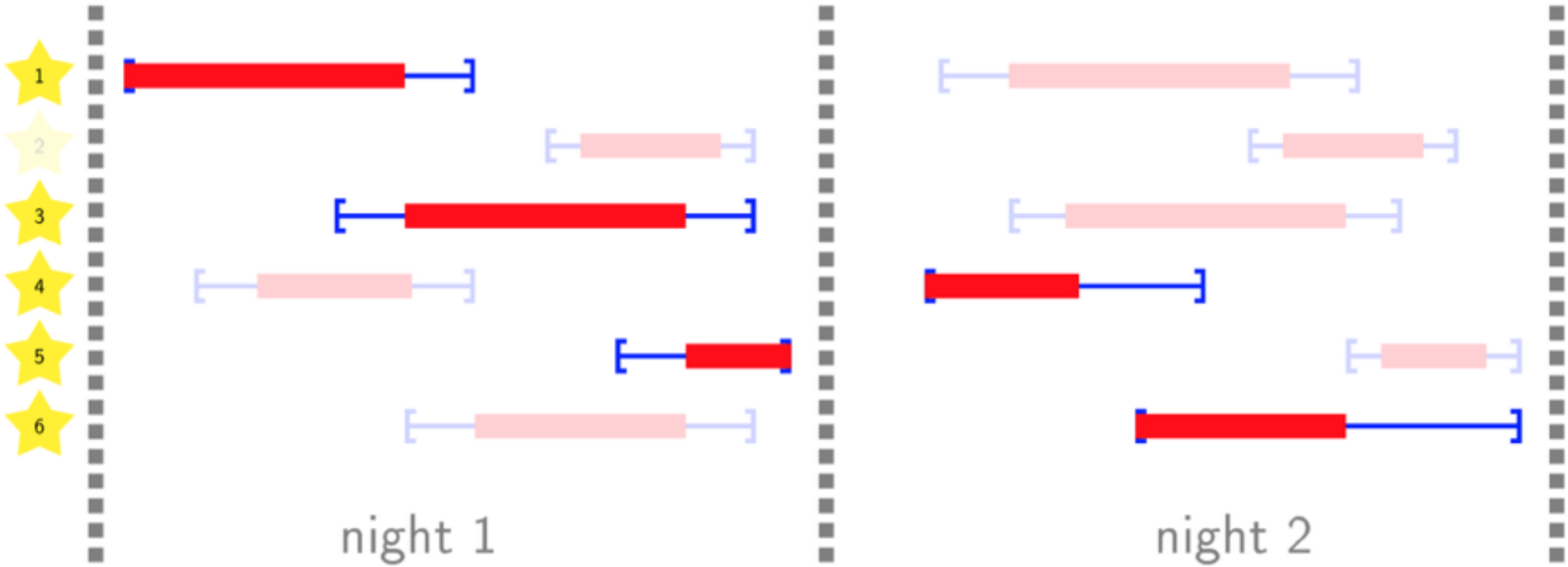


The observations must be scheduled by non-decreasing meridian time

Star Scheduler



Star Scheduler



A solution

Star Scheduler
A MIP model

$$\sum_j \boxed{z_i^j} = z_i$$

= 1 iff **i** is observed in night **j**

visibility interval of night j

$$\begin{cases} r_i^j z_i^j \le \boxed{t_i} \\ t_i + p_i^j z_i^j \le d_i^j z_i^j + M(1 - z_i^j) \end{cases}$$

= starting time of **i**

i1 < i2 if on the same night

$$\begin{cases} z_{i_1}^j + z_{i_2}^j \le \boxed{y_{i_1,i_2}} + 1 \\ t_{i_1} + p_{i_1}^j \le t_{i_2} + M(1 - y_{i_1,i_2}) \end{cases}$$

= 1 iff **i1** and **i2** are observed the same night

Star Scheduler
A MIP model

i: observations
j: nights

$$\max \sum_i w_i \boxed{z_i}$$

$= 1$ iff **i** is observed

$$\sum_j \boxed{z_i^j} = z_i$$

$= 1$ iff **i** is observed in night **j**

visibility interval of night j

$$\begin{cases} r_i^j z_i^j \le \boxed{t_i} \\ t_i + p_i^j z_i^j \le d_i^j z_i^j + M(1 - z_i^j) \end{cases}$$

$=$ starting time of **i**

i1 < i2 if on the same night

$$\begin{cases} z_{i_1}^j + z_{i_2}^j \le \boxed{y_{i_1,i_2}} + 1 \\ t_{i_1} + p_{i_1}^j \le t_{i_2} + M(1 - y_{i_1,i_2}) \end{cases}$$

$= 1$ iff **i1** and **i2** are observed the same night

Star Scheduler
A MIP model

i: observations
j: nights

$$\max \sum_i w_i \boxed{z_i}$$

= 1 iff **i** is observed

$$\sum_j \boxed{z_i^j} = z_i$$

= 1 iff **i** is observed in night **j**

visibility interval of night j

$$\begin{cases} r_i^j z_i^j \leq \boxed{t_i} \\ t_i + p_i^j z_i^j \leq d_i^j z_i^j + M(1 - z_i^j) \end{cases}$$

= starting time of **i**

i1 < i2 if on the same night

$$\begin{cases} z_{i_1}^j + z_{i_2}^j \leq \boxed{y_{i_1,i_2}} + 1 \\ t_{i_1} + p_{i_1}^j \leq t_{i_2} + M(1 - y_{i_1,i_2}) \end{cases}$$

= 1 iff **i1** and **i2** are observed the same night

Very poor linear relaxation, does not scale in memory $O(n^2 m)$

Star Scheduler - A CP model

A CP model:

- Use optional tasks of CPO and NoOverlap for each night

Star Scheduler - A CP model

A CP model:

- Use optional tasks of CPO and NoOverlap for each night

$$
\begin{aligned}
&\max\ z = \textstyle\sum_i w_i z_i \\
&\quad \textstyle\sum_j z_i^j = z_i && \forall\ i \\
&\quad z_i^j = 1 \Leftrightarrow \boxed{task_i^j} \text{ is present} && \forall\ i\ \forall\ j \\
&\quad \textsc{NoOverlap}([task_1^j, \ldots, task_n^j]) && \forall\ j
\end{aligned}
$$

Star Scheduler - A CP model

A CP model:

- Use optional tasks of CPO and NoOverlap for each night

$$
\begin{aligned}
\max\ z = \textstyle\sum_i w_i z_i & \\
\textstyle\sum_j z_i^j = z_i & \qquad \forall\, i \\
z_i^j = 1 \Leftrightarrow \boxed{task_i^j} \text{ is present} & \qquad \forall\, i\ \forall\, j \\
\textsc{NoOverlap}([task_1^j, \ldots, task_n^j]) & \qquad \forall\, j
\end{aligned}
$$

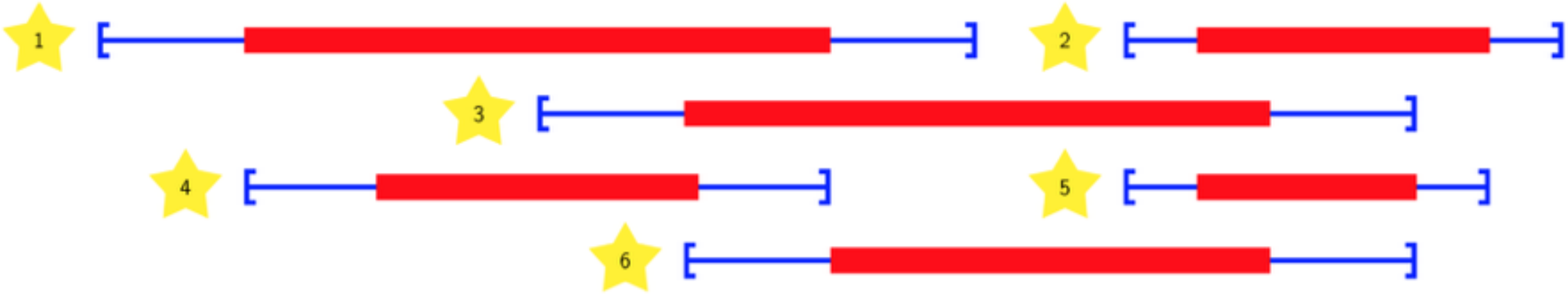
+ precedences when on the same night

+ clique of known incompatible observations

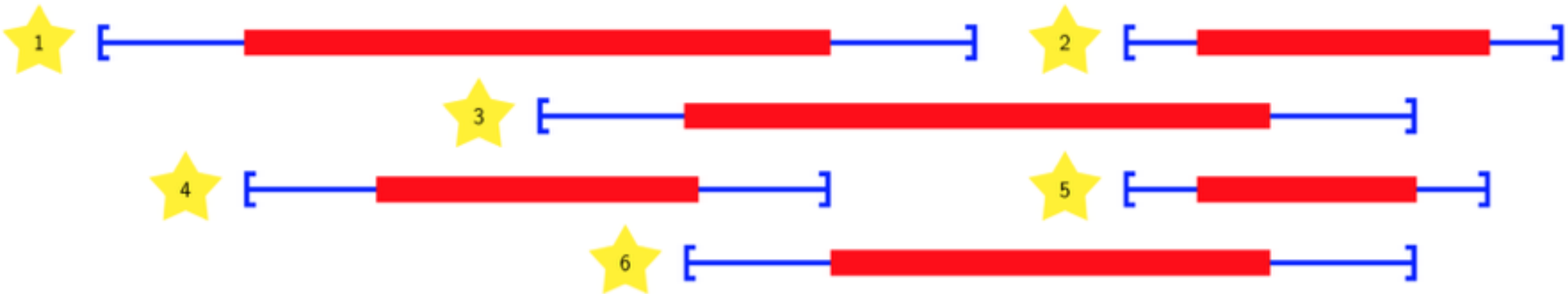
- Best results (LNS) with a blackbox model but remains unable to handle the real-life dataset (800 observations, 142 nights)
- No effective filtering and no interesting global upper bound

Star Scheduler - The single night problem

Star Scheduler - The single night problem



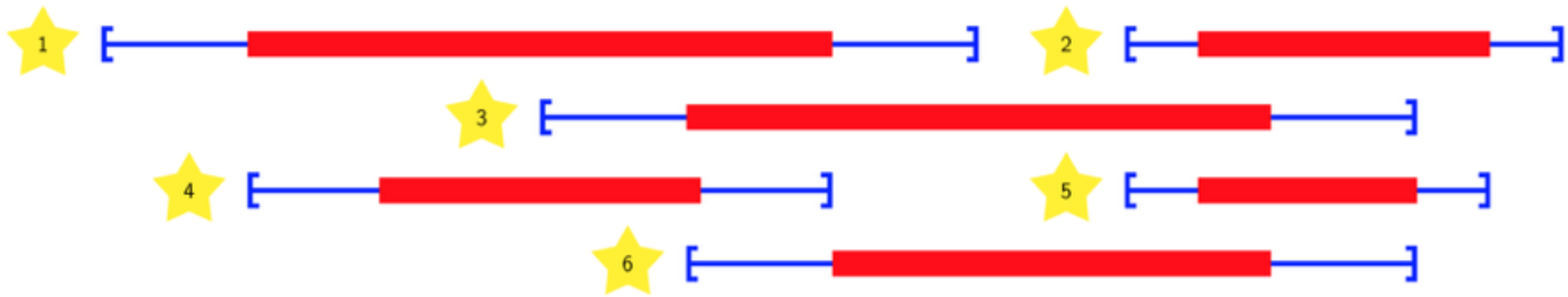
Star Scheduler - The single night problem



Find and schedule a subset S of observations s.t

$\sum_i w_i$ is maximized

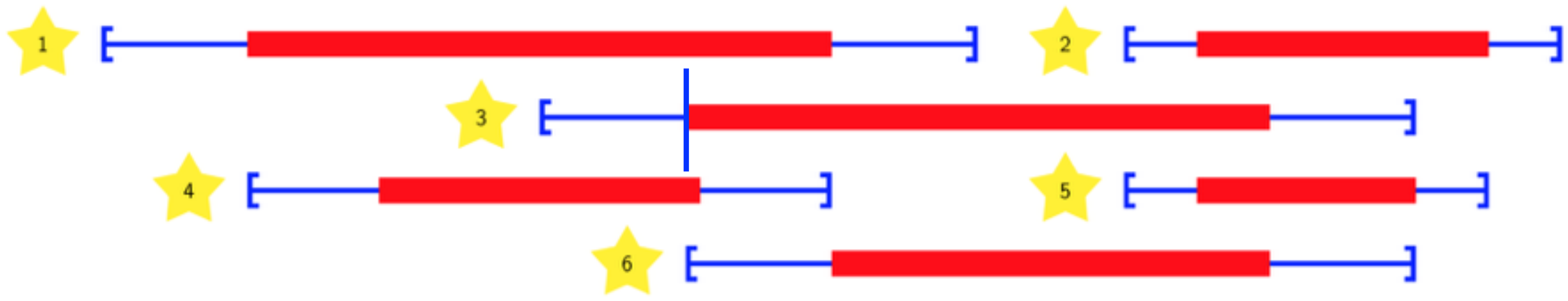
Star Scheduler - The single night problem



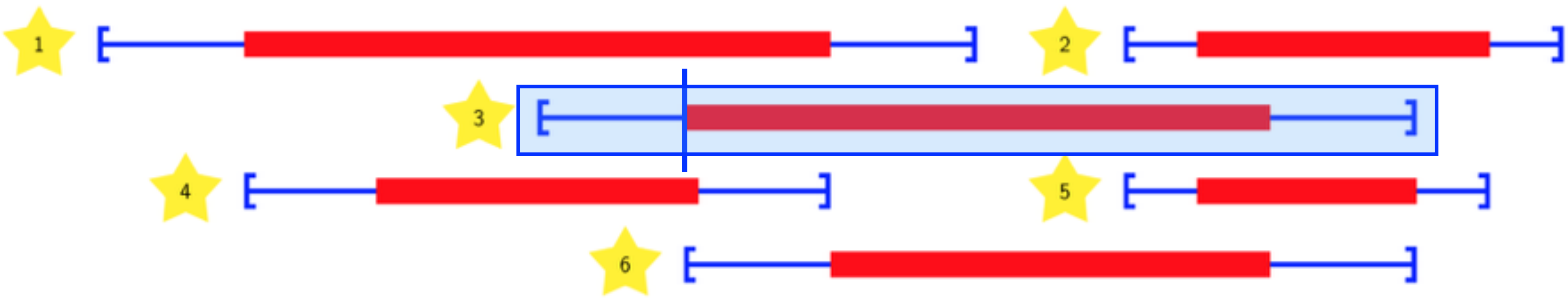
Find and schedule a subset S of observations s.t

$\sum_i w_i$ is maximized

Star Scheduler - The single night problem

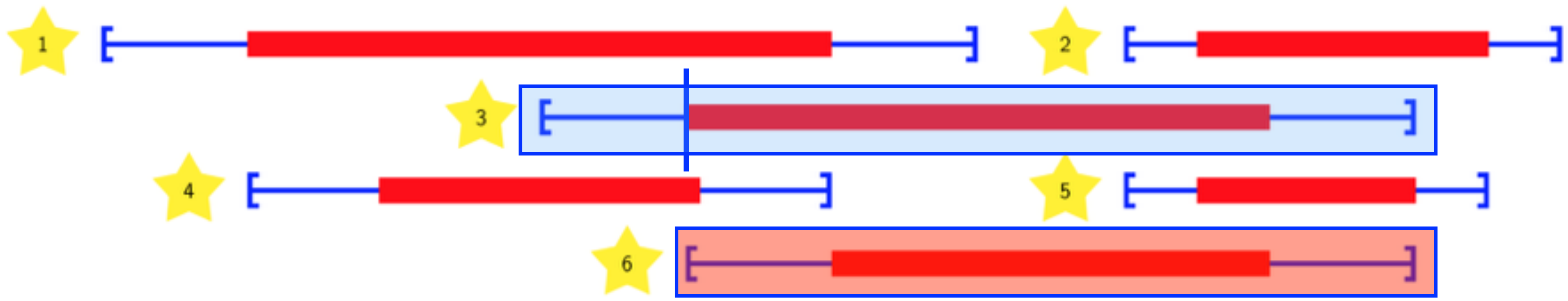


Star Scheduler - The single night problem



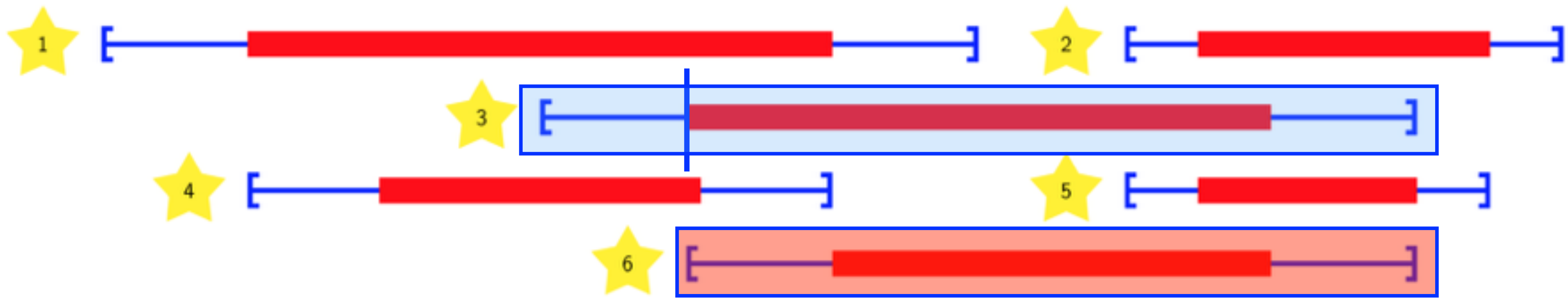
- Suppose observation 3 is scheduled

Star Scheduler - The single night problem

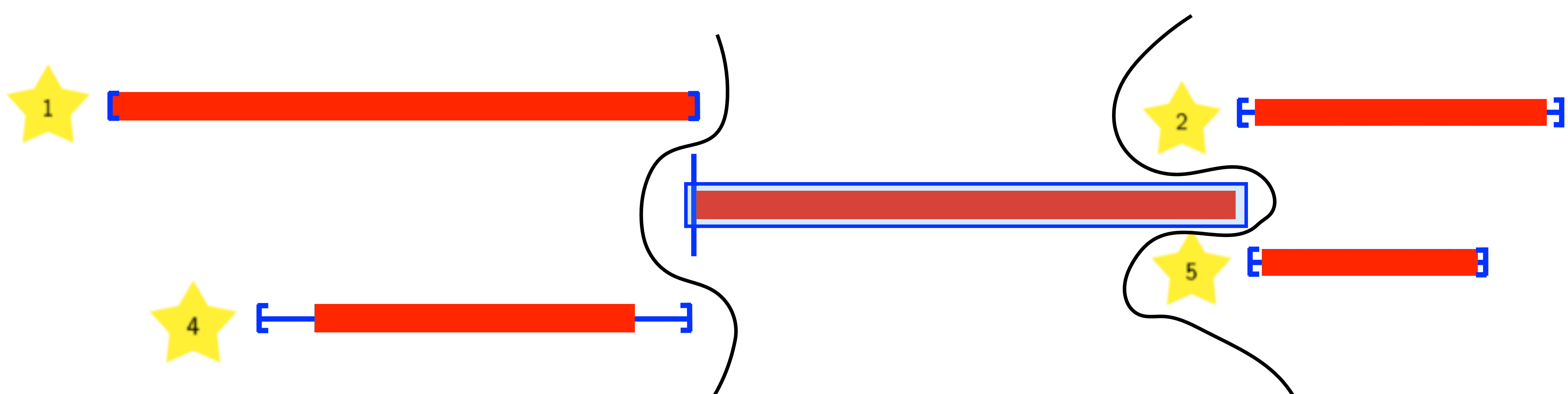


- Suppose observation 3 is scheduled
- 6 is incompatible

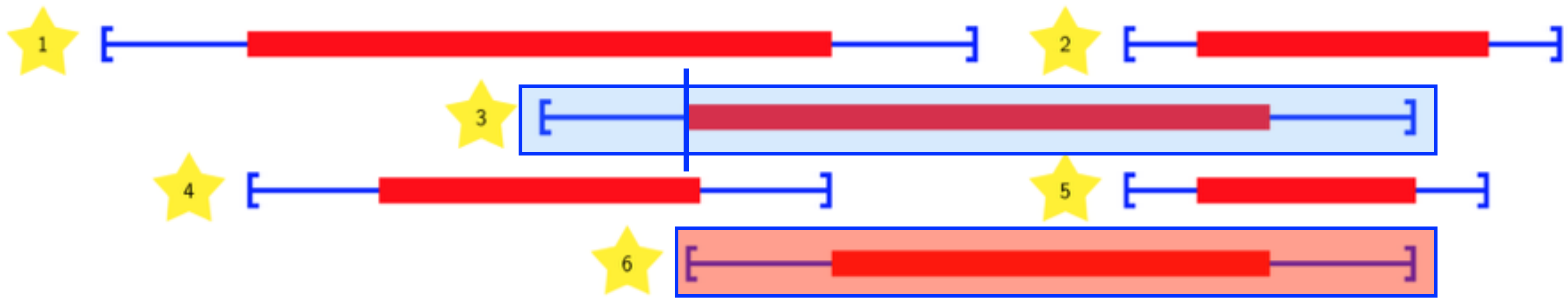
Star Scheduler - The single night problem



- Suppose observation 3 is scheduled
- 6 is incompatible
- Left and right subproblems are independent (observations are scheduled in non-decreasing time of their meridians)



Star Scheduler - The single night problem



$f(i,t)$: maximum interest with observations 1 to i (schedule order) and such that i ends before time t

$$f(i,t) = \begin{cases} max(f(i-1,t), f(i-1,t-p_i)+w_i) & i \in [1,n], t \in [r_i+p_i, d_i] \\ f(i-1,t) & i \in [1,n], t \in [0, r_i+p_i[\\ f(i,d_i) & i \in [1,n], t \in]d_i, T] \\ 0 & i=0, t \in [0,T] \end{cases}$$

$f(n,T)$ can be found in $O(nT)$

Star Scheduler - An improved CP model

interest of night **j**

$$\max\ z = \sum_j \boxed{interest_j}$$

$$\sum_j z_i^j \leq 1 \qquad \forall\, i$$

$$\boxed{\textsc{NightNoOverlap}}([z_1^j, \ldots, z_n^j], interest_j) \qquad \forall\, j$$

- Update $\underline{interest_j}$ based on the observations assigned in the night
- Filter observations that can not fit in the night anymore
- Filter $\overline{interest_j}$ using DP
- Force (in the night) observations that are mandatory to reach $\underline{interest_j}$

Star Scheduler - An improved CP model

interest of night **j**

$$\max\ z = \sum_j \boxed{interest_j}$$

$$\sum_j z_i^j \leq 1 \qquad \forall\, i$$

$$\boxed{\textsc{NightNoOverlap}}([z_1^j, \ldots, z_n^j], interest_j) \qquad \forall\, j$$

+ scheduling is excluded from the search space
+ strong filtering for each night

- nights remains filtered independently, no strong lower bound

Star Scheduler - Back to MIP

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

$$\max \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \qquad \forall\, i$$

$$\rho_j^k \in \{0, 1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

$$\max \sum_j \sum_{k \in \Omega_j} w_j^k \boxed{\rho_j^k}$$

= 1 iff **k**-th schedule of night **j** is selected

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \qquad \forall\, i$$

$$\rho_j^k \in \{0, 1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

$$\max \sum_j \sum_{k\in \boxed{\Omega_j}} \boxed{w_j^k} \boxed{\rho_j^k}$$

= 1 iff **k**-th schedule of night **j** is selected

$$\sum_{k\in\Omega_j} \rho_j^k = 1 \qquad \forall j$$

$$\sum_j \sum_{k\in\Omega_j} \boxed{s_{i,j}^k} \rho_j^k \leq 1 \qquad \forall i$$

$$\rho_j^k \in \{0,1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

$$\max \sum_j \sum_{k \in \boxed{\Omega_j}} \boxed{w_j^k} \boxed{\rho_j^k}$$

= 1 iff **k**-th schedule of night **j** is selected

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall j$$

(exactly one schedule for each night)

$$\sum_j \sum_{k \in \Omega_j} \boxed{s_{i,j}^k} \rho_j^k \leq 1 \qquad \forall i$$

$$\rho_j^k \in \{0, 1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

An extended LP formulation:

- One variable (a column) = one night schedule
- Constraints of the LP:
  - Exactly one schedule for each night
  - One observation occurs in at most one schedule
- Objective is the find the combination of schedules with maximum interest

$$\max \sum_j \sum_{k \in \boxed{\Omega_j}} \boxed{w_j^k} \boxed{\rho_j^k}$$

= 1 iff **k**-th schedule of night **j** is selected

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j$$

(exactly one schedule for each night)

$$\sum_j \sum_{k \in \Omega_j} \boxed{s_{i,j}^k} \rho_j^k \leq 1 \qquad \forall\, i$$

(observations are assigned to at most one night)

$$\rho_j^k \in \{0,1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

An extended LP formulation

Ω_j : the set all possible schedules of night j

$s_{i,j}^k = 1$ iff observation **i** belongs to the **k**-th schedule of night **j**

$(s_{1,j}^k, \ldots, s_{n,j}^k)$: 0/1 description of the **k**-th schedule of night **j**

$w_j^k = \sum_i w_i s_{i,j}^k$: **interest** of the **k**-th schedule of night **j**

$$\max \sum_j \sum_{k \in \boxed{\Omega_j}} \boxed{w_j^k} \boxed{\rho_j^k}$$

ρ_j^k = 1 iff **k**-th schedule of night **j** is selected

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall j$$

(exactly one schedule for each night)

$$\sum_j \sum_{k \in \Omega_j} \boxed{s_{i,j}^k} \rho_j^k \leq 1 \qquad \forall i$$

(observations are assigned to at most one night)

$$\rho_j^k \in \{0, 1\} \qquad \forall k \in \Omega_j, \forall j$$

Star Scheduler - Back to MIP

$$\max \ \sum_j \sum_{k \in \Omega_j} w_j^k \boxed{\rho_j^k}$$

= 1 iff **k**-th schedule of night **j** is selected $\rho_j^k \in \{0, 1\}$

$$\sum_{k \in \Omega_j} \rho_j^k = 1$$ (exactly one schedule for each night)

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1$$ (observations are assigned to at most one night)

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost
- Only a tiny fraction of the variables are needed

Star Scheduler - Back to MIP

$$\max \; \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \;\; \forall\, i$$

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost

Star Scheduler - Back to MIP

$$\max \ \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j \quad (\alpha_j)$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \quad \forall\, i \quad (\beta_i)$$

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost

Star Scheduler - Back to MIP

$$\max \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j \qquad (\alpha_j)$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \quad \forall\, i \qquad (\beta_i)$$

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

Star Scheduler - Back to MIP

$$\max \ \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j \quad \boxed{(\alpha_j)}$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \quad \forall\, i \quad \boxed{(\beta_i)}$$

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

$$rc(\rho_j^k) = \sum_i (w_i - \beta_i) s_{i,j}^k - \alpha_j$$

Star Scheduler - Back to MIP

$$\max \ \sum_j \sum_{k \in \Omega_j} w_j^k \rho_j^k$$

$$\sum_{k \in \Omega_j} \rho_j^k = 1 \qquad \forall\, j \quad (\alpha_j)$$

$$\sum_j \sum_{k \in \Omega_j} s_{i,j}^k \rho_j^k \leq 1 \quad \forall\, i \quad (\beta_i)$$

The LP relaxation can be solved by **column generation**:

- Iteratively add a variable (schedule) of maximum reduced cost

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

$$rc(\rho_j^k) = \sum_i (w_i - \beta_i) s_{i,j}^k - \alpha_j$$

- Solve the one night problem where w_i is replaced by

$$(w_i - \beta_i)$$

Star Scheduler - An improved CP model

$$\max\ z = \sum_j \boxed{interest_j}$$

$$\sum_j z_i^j \leq 1 \qquad \forall\, i$$

$$\textsc{NightNoOverlap}([z_1^j, \ldots, z_n^j], interest_j) \qquad \forall\, j$$

$$\textsc{Objective}([z_1^1, \ldots, z_n^m], z)$$

Solve the LP relaxation by column generation:

- Filter the upper bound of z
- Reduced-cost filtering to exclude/force observations into nights ?

Branch and price algorithm implemented in a CP framework

Star Scheduler - Back to MIP

- The reduced cost of the **k**-th schedule of night **j**

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

Star Scheduler - Back to MIP

- The reduced cost of the **k**-th schedule of night **j**

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

- How to filter the upper bound of a z_i^j variable, i.e. excluding observation **i** from night **j** ?

Star Scheduler - Back to MIP

- The reduced cost of the **k**-th schedule of night **j**

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

- How to filter the upper bound of a z_i^j variable, i.e. excluding observation **i** from night **j** ?
- What is smallest decrease of the objective over all possible schedules that includes **i** in night **j** ?

Star Scheduler - Back to MIP

- The reduced cost of the **k**-th schedule of night **j**

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

- How to filter the upper bound of a z_i^j variable, i.e. excluding observation **i** from night **j** ?
- What is smallest decrease of the objective over all possible schedules that includes **i** in night **j** ?

$$z_{LP}^* + \boxed{\max_{k \in \Omega_j | s_{i,j}^k = 1} (rc(\rho_j^k))} < \underline{z} \implies z_i^j \neq 1$$

Star Scheduler - Back to MIP

- The reduced cost of the **k**-th schedule of night **j**

$$rc(\rho_j^k) = w_j^k - \alpha_j - \sum_i s_{i,j}^k \beta_i$$

- How to filter the upper bound of a z_i^j variable, i.e. excluding observation **i** from night **j** ?
- What is smallest decrease of the objective over all possible schedules that includes **i** in night **j** ?

$$z_{LP}^* + \boxed{\max_{k \in \Omega_j | s_{i,j}^k = 1} (rc(\rho_j^k))} < \underline{z} \implies z_i^j \neq 1$$

- The two steps backward-forward resolution of the DP provides exactly this information.

Star Scheduler - Results

Branch and price proves to be extremely efficient (benchmark of 21 instances):

- The real-life instance (800 observations, 142 nights) is solved optimally in less than 10 minutes
- 18 instances are solved optimally between 1 to 20 minutes
- 3 instances remains open in 2h time limit but the optimality gap is less than 0.11 %
- All feasible solutions significantly improves the MIP/CP approach

Star Scheduler - Results

[Catusse et al. 2016]

Branch and price proves to be extremely efficient (benchmark of 21 instances):

- The real-life instance (800 observations, 142 nights) is solved optimally in less than 10 minutes
- 18 instances are solved optimally between 1 to 20 minutes
- 3 instances remains open in 2h time limit but the optimality gap is less than 0.11 %
- All feasible solutions significantly improves the MIP/CP approach

Outline

1. **Reduced-costs based filtering**
    - Linear Programming duality
    - First example: *AtMostNValue*
        - Filtering the upper bound of a 0/1 variable
        - Filtering the lower bound of a 0/1 variable
    - General principles
    - Second example
    - Assignment, Cumulative, Bin-packing, …
2. **Dynamic programming filtering algorithms**
    - Linear equation, WeightedCircuit
    - General principles
    - Other relationships of DP and CP
3. **Illustration with a real-life application**

Conclusion

Focus of this talk:

Investigate/understand filtering techniques beyond polynomial sub-problems (beyond local-consistencies)

Help us to grow a better understanding of OR