# Crew Scheduling and Column Generation

Ralf Borndörfer

2015 Workshop on
Combinatorial Optimization with Applications in
Transportation and Logistics

Beijing, 29.07.2015

# Overview

- Crew Scheduling Example
- Path Covering and Column Generation
- Practice
- Extensions

Management Science vol. 3, No. 1, 1956

# A MODEL FOR THE OPTIMAL PROGRAMMING OF RAILWAY FREIGHT TRAIN MOVEMENTS*

A. CHARNES AND M. H. MILLER

*Purdue University and Carnegie Institute of Technology*

The structure shown in Table 1 can be translated into equation form by moving a row of λ's, one for each column, up through the rows and inserting the equal sign to the right of the $P_0$ column. The first two equations, for example, would be:

$$4 = 1\lambda_1 + 1\lambda_4 - 1\lambda_6 + 1\lambda_{12}$$

$$1 = 1\lambda_1 + 1\lambda_5 - 1\lambda_7 + 1\lambda_{13}$$

With the addition of the variables, the problem has been reduced to a standard simplex problem of the form:

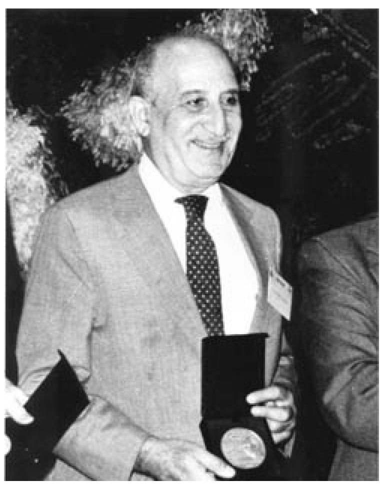
$$\text{Min.} \sum_{i=1}^{n} \lambda_i c_i$$

subject to:

$$\sum_{i=1}^{n} \lambda_i P_i = P_0$$

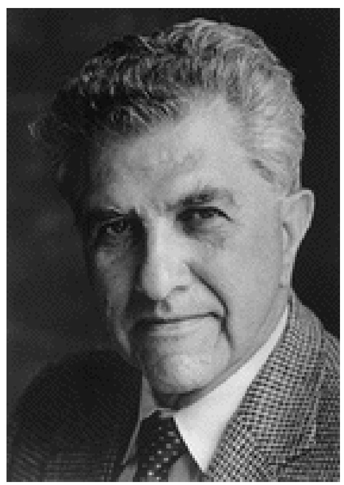
$$\lambda_i \geqq 0$$

and can be solved by the simplex technique.



Abraham Charnes

Finalist for the Nobel Prize in Economics 1975



Merton H. Miller

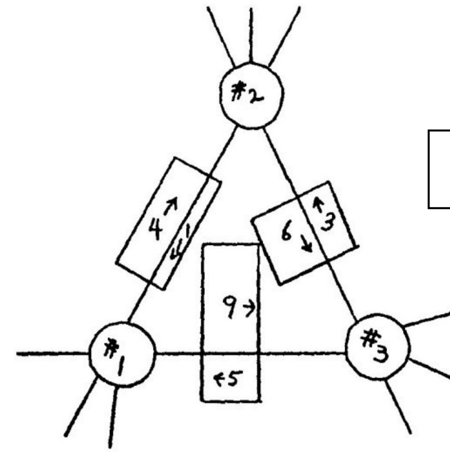
Nobel Prize for Economics 1990 w. Markowitz & Sharpe



TABLE 1

*Structural tableau of train-scheduling model*

| $c_j \rightarrow$ | | | 1.0 | 1.0 | 1.0 | 1.2 | 1.2 | 0 | 0 | 0 | 0 | 0 | 0 | M | M | M | M | M | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Shipment Requirements | Routes | | | | | Surplus Vectors (light moves) | | | | | | Artificial Vectors (legs) | | | | | |
| From | To | | 1,2 | 1,3 | 2,3 | 1,2,3 | 1,3,2 | 1–2 | 2–1 | 1–3 | 3–1 | 2–3 | 3–2 | 1–2 | 2–1 | 1–3 | 3–1 | 2–3 | 3–2 |
| | | $P_0$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ | $P_{15}$ | $P_{16}$ | $P_{17}$ |
| 1 | 2 | 4 | 1 | | | 1 | | −1 | | | | | | 1 | | | | | |
| 2 | 1 | 1 | 1 | | | | 1 | | −1 | | | | | | 1 | | | | |
| 1 | 3 | 9 | | 1 | | | 1 | | | −1 | | | | | | 1 | | | |
| 3 | 1 | 5 | | 1 | | 1 | | | | | −1 | | | | | | 1 | | |
| 2 | 3 | 6 | | | 1 | 1 | | | | | | −1 | | | | | | 1 | |
| 3 | 2 | 3 | | | 1 | | 1 | | | | | | −1 | | | | | | 1 |




+ rules

CHART 1. Simplified map of terminal switching railroad, showing connections with trunklines, major interchange and customer yard areas, and traffic requirements (in trainloads) between major points.

postponed until the description of the model and the computational routine has been completed.

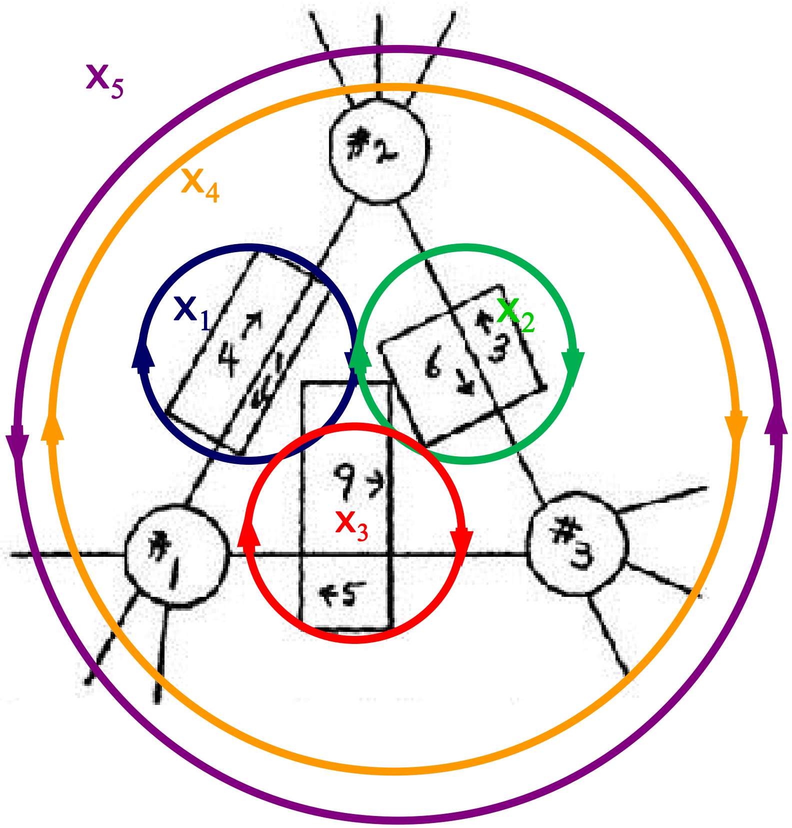
Above the routes, in the row labeled $c_j$, are entered the costs of assigning a single crew and engine package to the route in question. These costs may be stated either as the standard crew and engine expense, or as the expected costs reflecting the fact that on longer runs there is a greater probability of running into overtime. We constructed working models both ways and found, that optimal programs were not particularly sensitive to variations in the cost of crews. In fact, it was usually possible to simplify the calculation by minimizing the number of crews, that is treating the cost of each crew as 1.

$P_6$ to $P_{11}$ in the tableau are overfulfillment slack vectors. In the train scheduling context they correspond to "light moves", or trips by a crew and engine without cars. If, for example, four crews should be assigned to the route $P_1$—which runs

# Integer Programming (IP)



## Graph theoretical model




## Algebraic model (Integer Program)

$$\text{Min } x_1 + x_2 + x_3 + 1.2x_4 + 1.2x_5$$

$$4 \leq x_1 + x_4$$

$$1 \leq x_1 + x_5$$

$$9 \leq x_3 + x_5$$

$$5 \leq x_3 + x_4$$

$$6 \leq x_2 + x_4$$

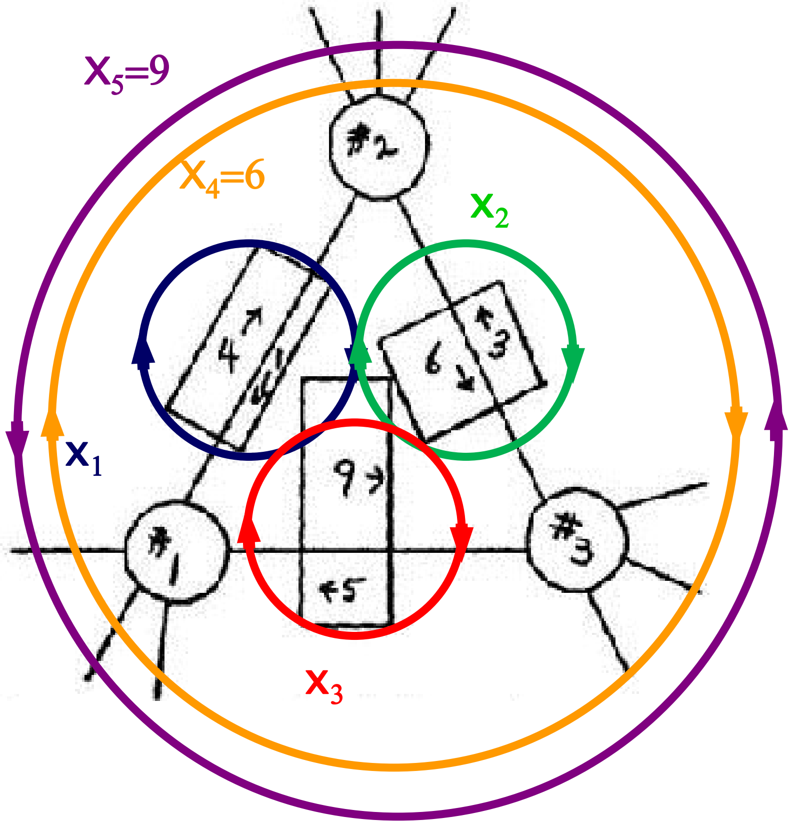
$$3 \leq x_2 + x_5$$

$$x \geq 0$$

$$x \text{ integer}$$

# Ad Hoc Solution

## Graph theoretical model




## Solution

- $x_1 = 0$
- $x_2 = 0$
- $x_3 = 0$
- $x_4 = 6$
- $x_5 = 9$

## Price

- $\pi_{12} = 0.4$
- $\pi_{21} = 0.4$
- $\pi_{13} = 0.4$
- $\pi_{31} = 0.4$
- $\pi_{23} = 0.4$
- $\pi_{32} = 0.4$

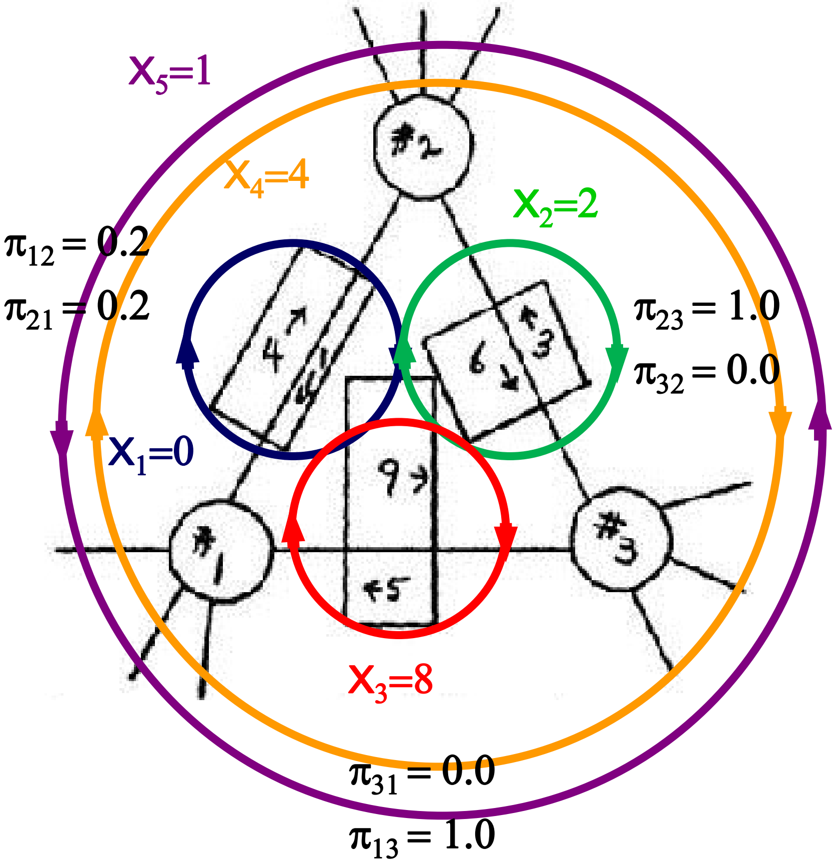
- $c = 7.2 + 10.8 = 18$
- $c = 28 * 0.4 = 11.2$

- Uncertainty = 6.8/18 = 37.78%

# Optimal Solution

Freie Universität Berlin

## Graph theoretical model




**Solution**

- $x_1 = 0$
- $x_2 = 2$
- $x_3 = 8$
- $x_4 = 4$
- $x_5 = 1$

- $c = 16$

**Price**

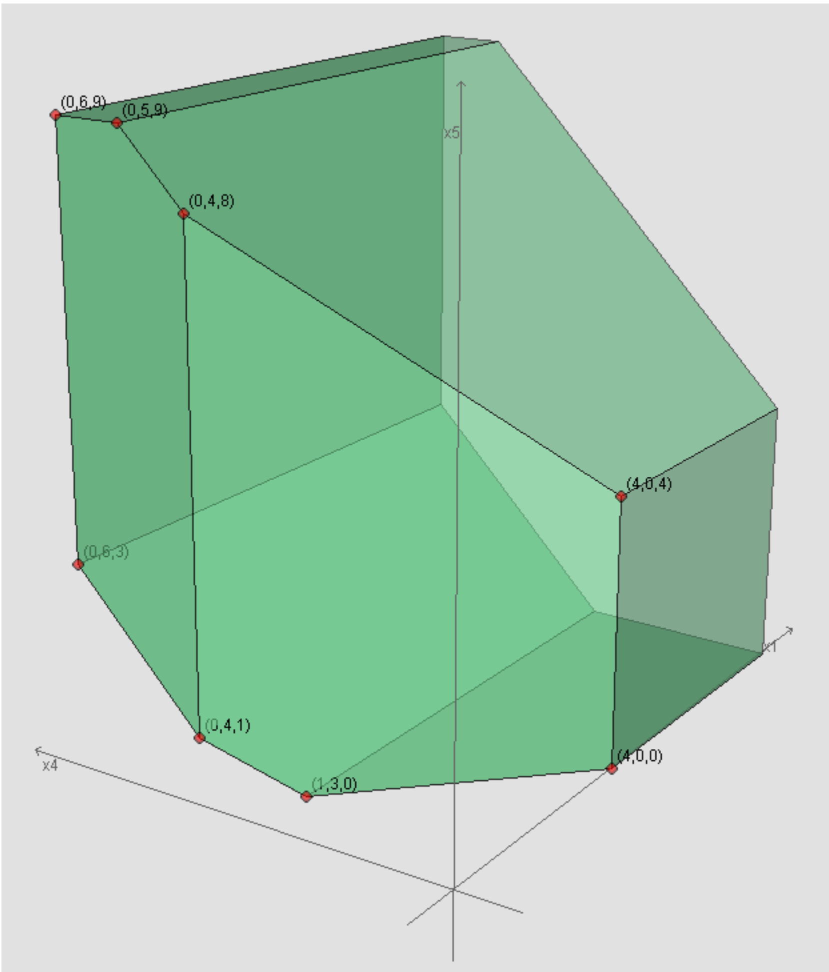
- $\pi_{12} = 0.2$
- $\pi_{21} = 0.2$
- $\pi_{13} = 1.0$
- $\pi_{31} = 0.0$
- $\pi_{23} = 1.0$
- $\pi_{32} = 0.0$

- $c = 16$

- Uncertainty = 0.0/16 = 0.00%

# Integer Programming (IP)

## Graph theoretical model
## Polytope




## Algebraic model (Integer Program)
## Algebraic model (Linear Program)

**Min** $x_1 + x_2 + x_3 + 1.2x_4 + 1.2x_5$

$4 \le x_1 + x_4$

$1 \le x_1 + x_5$

$9 \le x_3 + x_5$

$5 \le x_3 + x_4$

$6 \le x_2 + x_4$

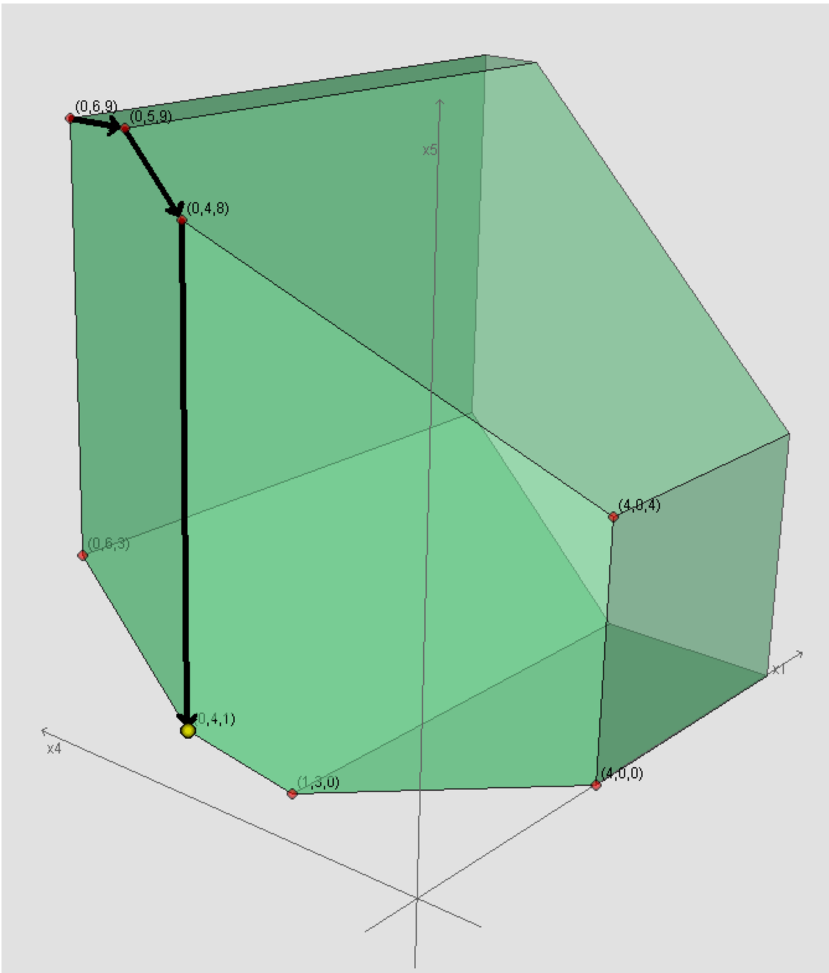
$3 \le x_2 + x_5$

$x \ge 0$

~~**x integer**~~

## Polytope

## Simplex Algorithm




## Algebraic model (Linear Program)

$$\text{Min } x_1 + x_2 + x_3 + 1.2x_4 + 1.2x_5$$

$$4 \leq x_1 + x_4$$

$$1 \leq x_1 + x_5$$

$$9 = x_3 + x_5$$

$$5 \leq x_3 + x_4$$

$$6 = x_2 + x_4$$

$$3 \leq x_2 + x_5$$

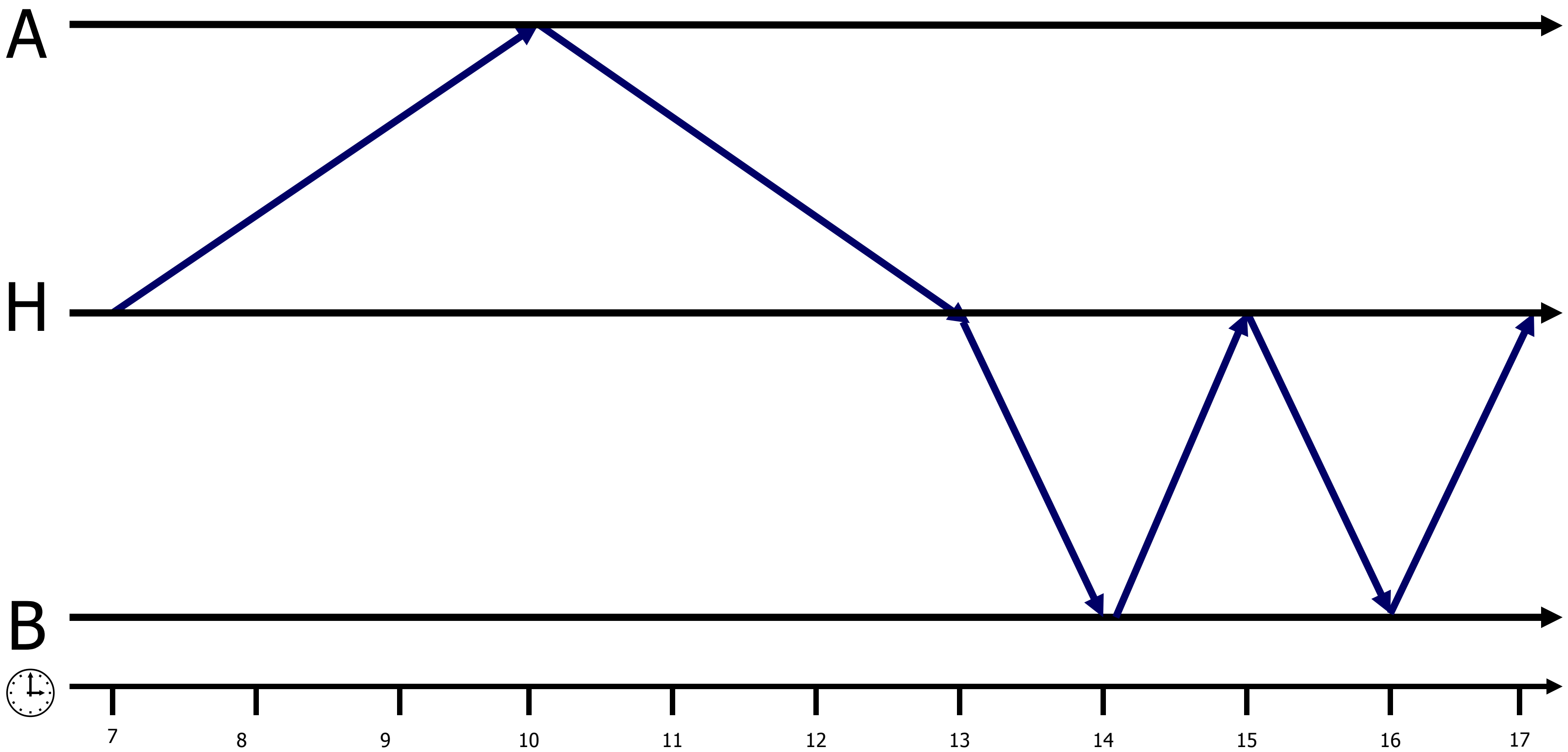
$$x \geq 0$$

# Path Covering 1956-1998

| *Article* | *Constraints* | *Variables* | *Time* |
| --- | --- | --- | --- |
| Charnes & Miller [1956] | 6 | 17 | by hand |
| Hoffman & Padberg [1993] | 145 | 1 053 137 | 5 min |
| Bixby, Gregory, Lustig, Marsten, Shanno [1992] | 837 | 12 753 313 | 249 sec |
| Barnhart, Johnson, Nemhauser, Savelsbergh, Vance [1998] | >10 000 | nobody knows | several days |

# Crew Scheduling Problem

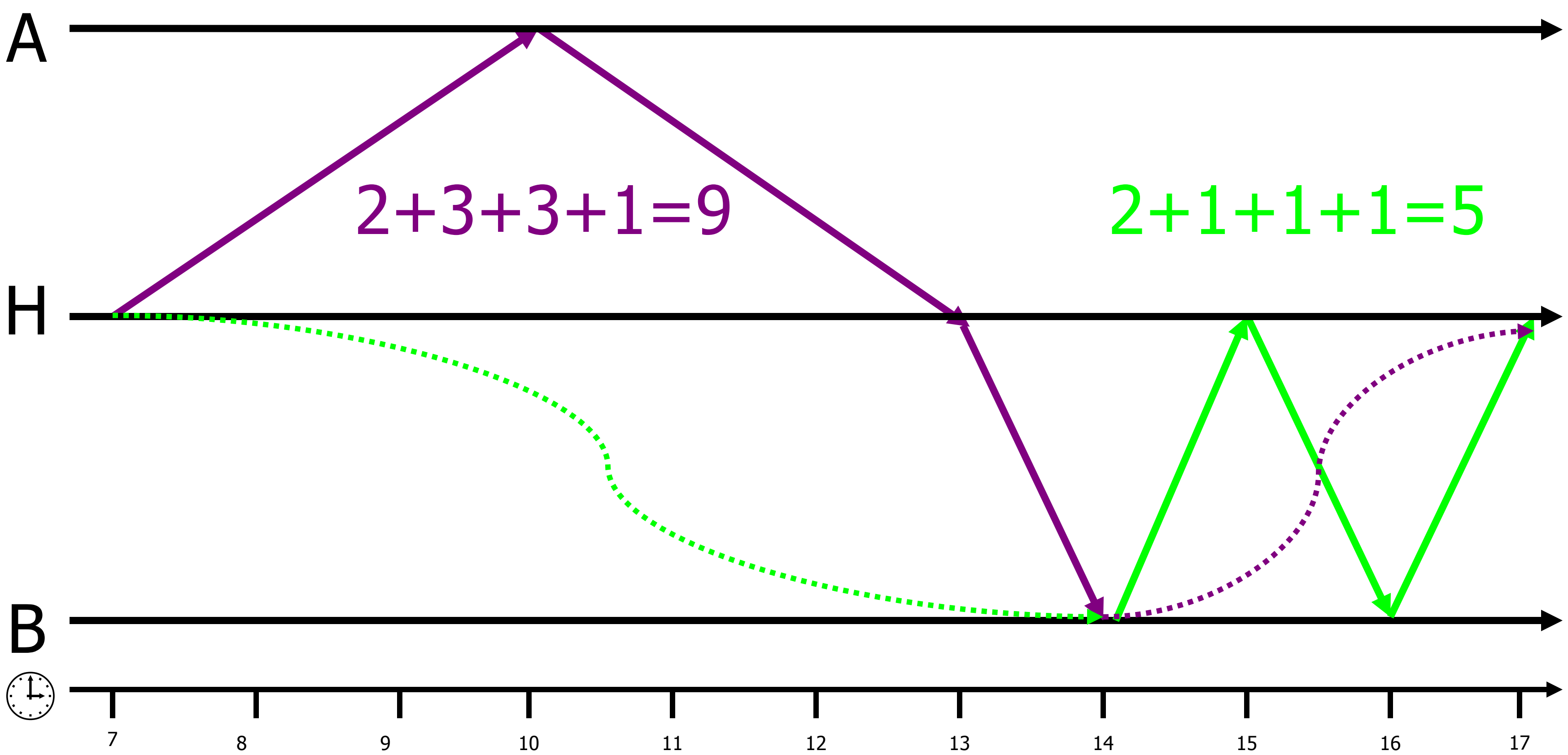





- Rules: Flight time ≤ 7 h, connections ≤ 3 h
- Costs: 2 + working time

# Crew Scheduling Problem

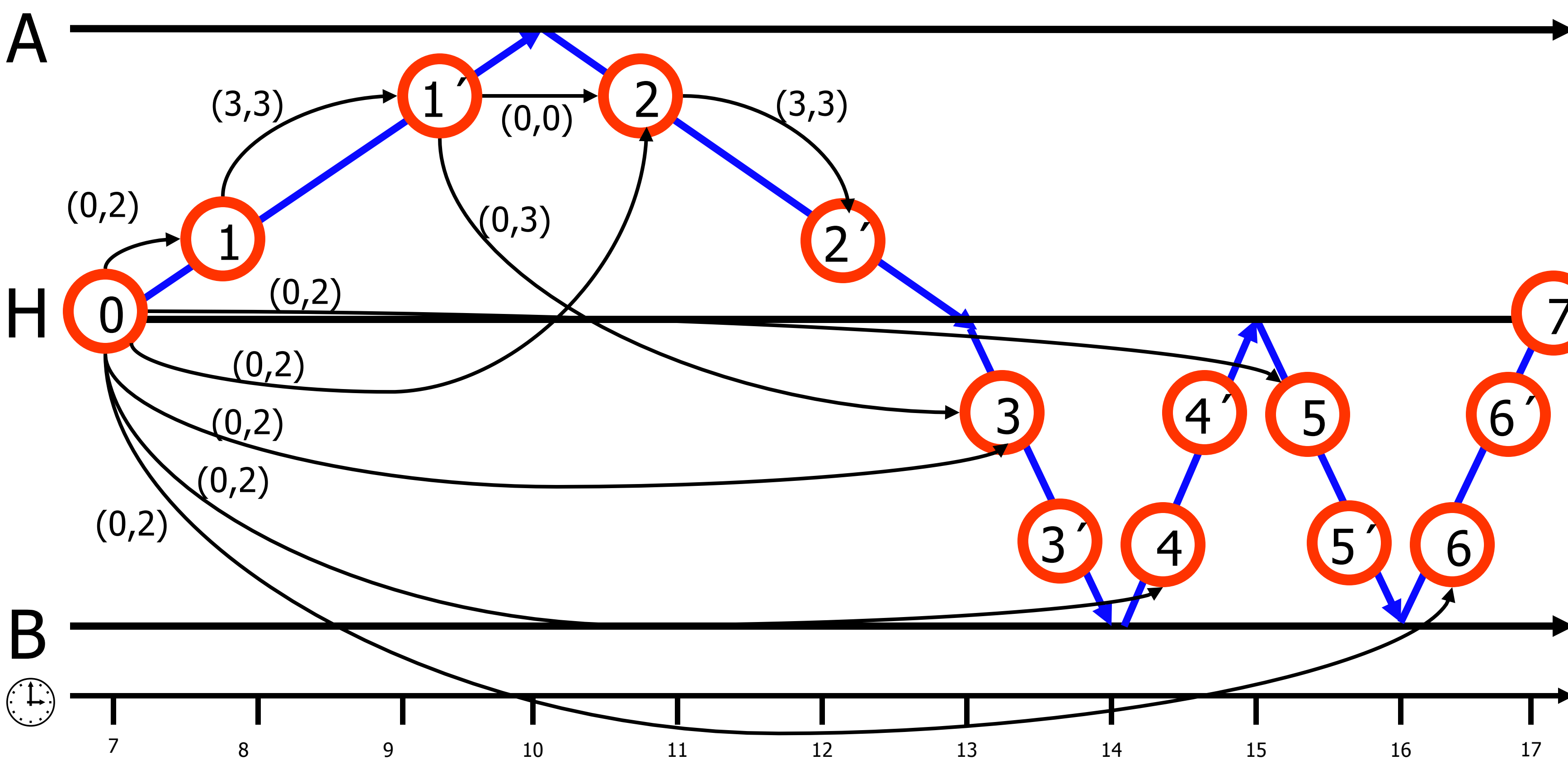





- Rules: Flight time ≤ 7 h, connections ≤ 3 h
- Costs: 2 + working time

# Graph Theoretic Model

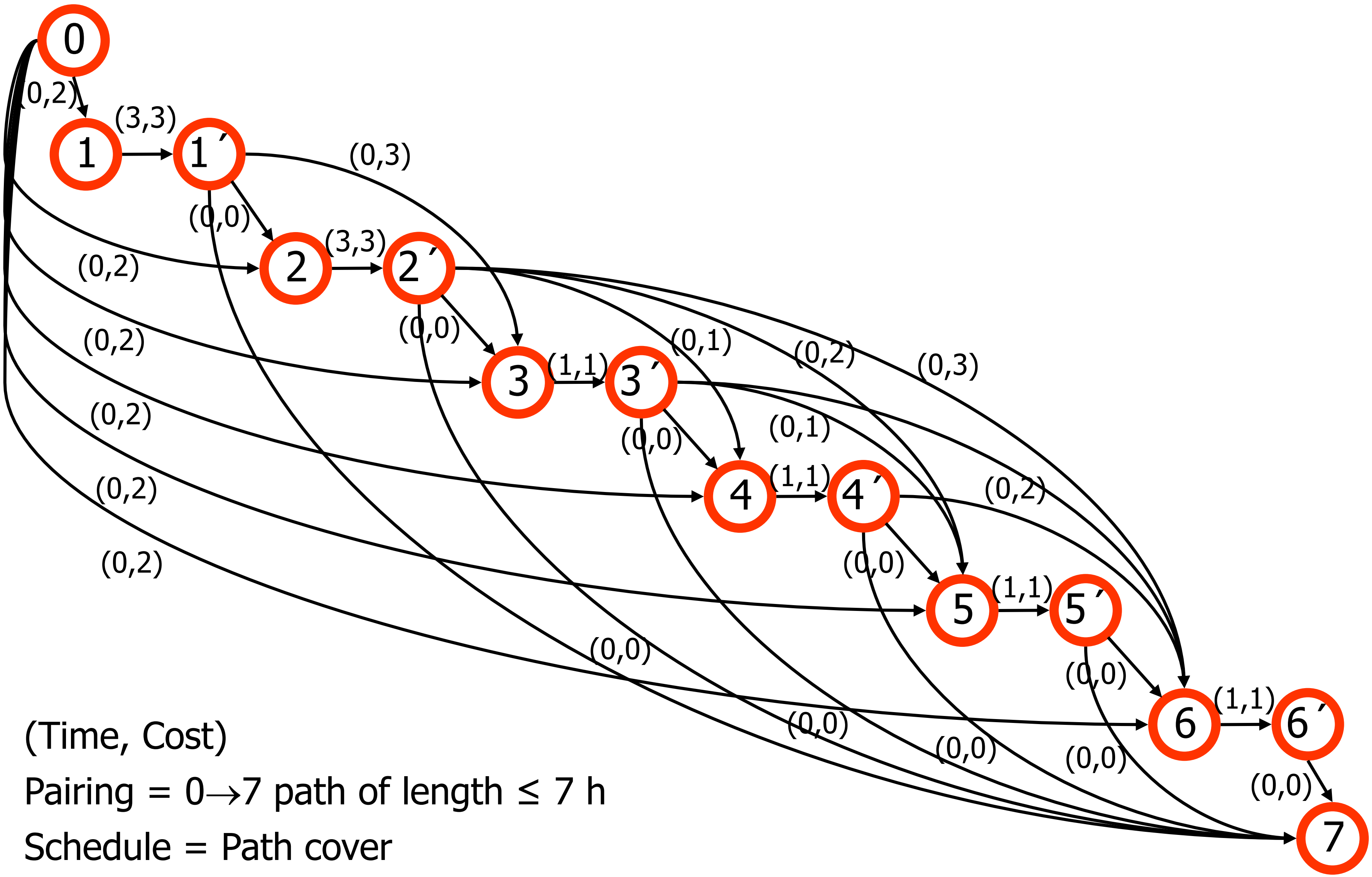





- Rules: Flight time ≤ 7 h, connections ≤ 3 h
- Costs: 2 + working time

# Graph Theoretic Model

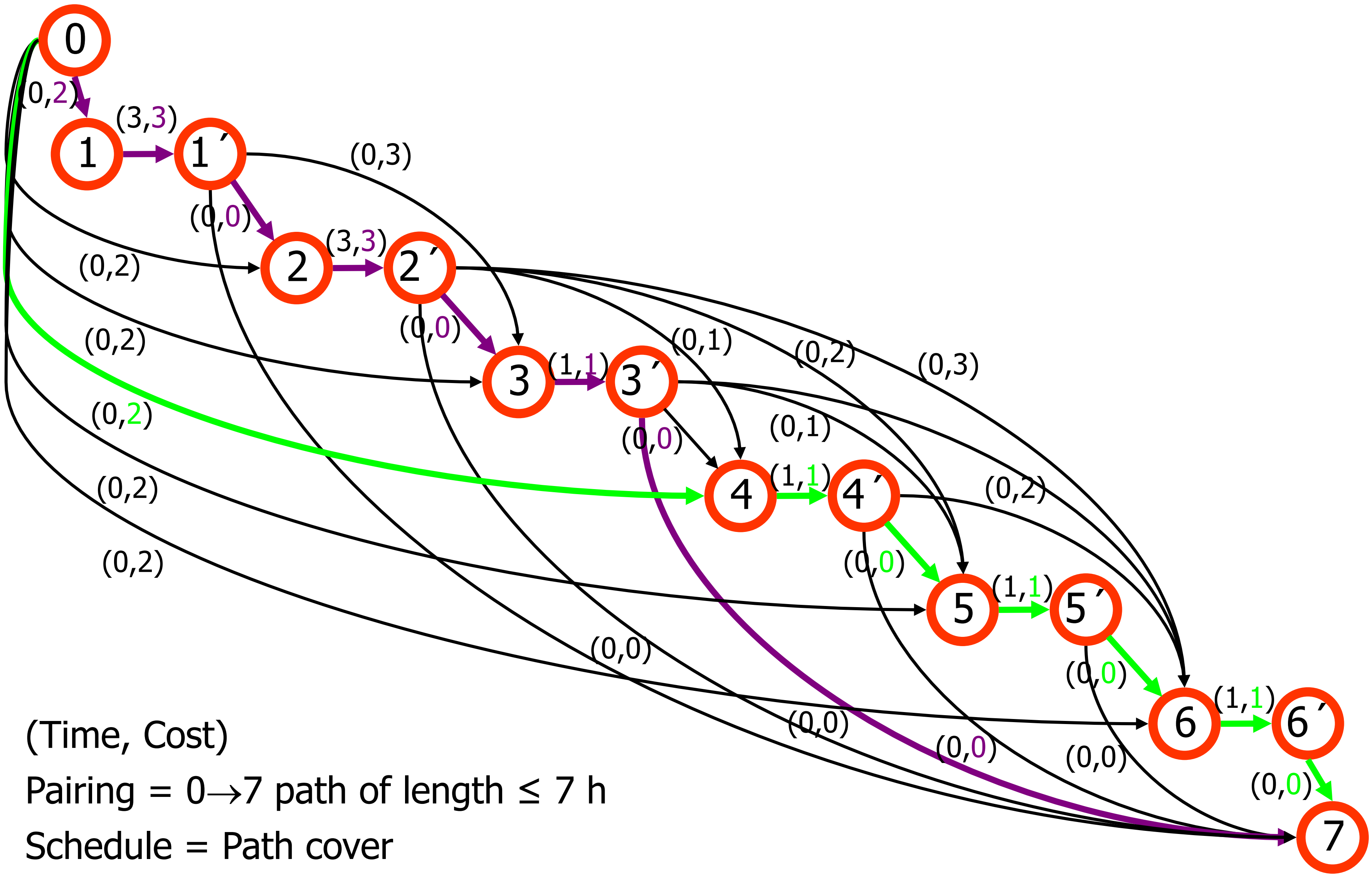





- (Time, Cost)
- Pairing = 0→7 path of length ≤ 7 h
- Schedule = Path cover

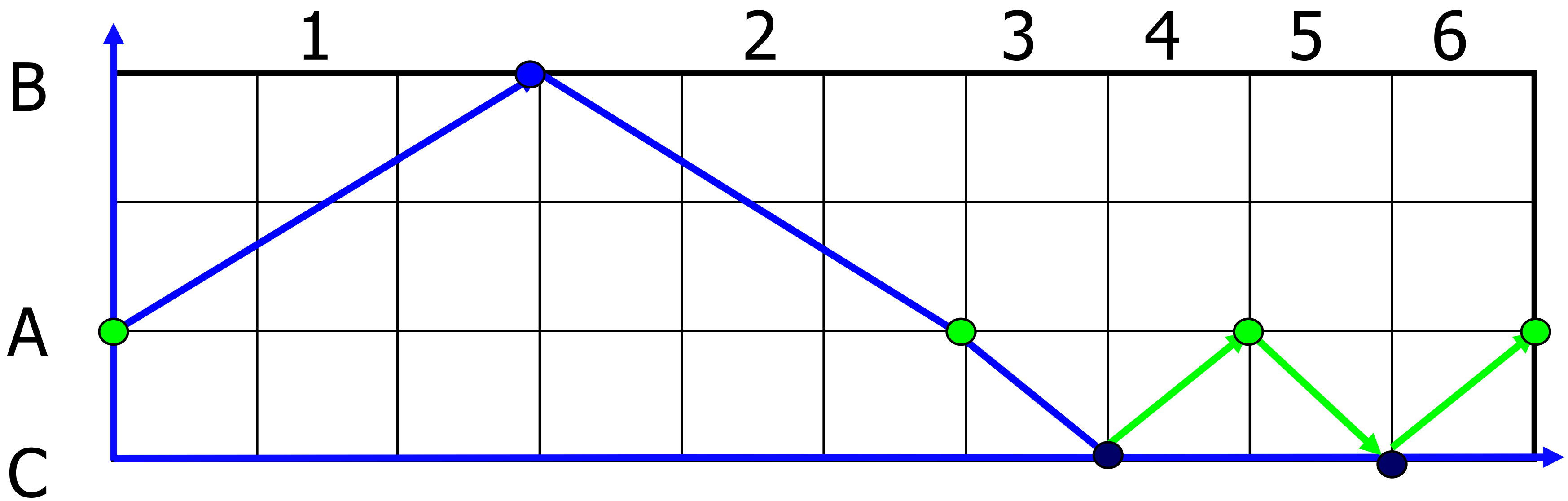
# Graph Theoretic Model






- (Time, Cost)
- Pairing = 0→7 path of length ≤ 7 h
- Schedule = Path cover

# Set Partitioning Model




| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 |

# Set Partitioning Model

| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 1 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 1 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 |
| | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | x10 | x11 | x12 | x13 | x14 | x15 | x16 | x17 | x18 | x19 | x20 | x21 | x22 | x23 | x24 | x25 | x26 | x27 | x28 | x29 | x30 | x31 | x32 | x33 | x34 | x35 | x36 | x37 | |

min $5x_1+5x_2+\ldots+12x_{36}+9x_{37}$

s.t. $x_1+x_7+x_8+x_{19}+x_{20}+x_{21}+x_{22}+x_{29}+x_{30}+x_{31}+x_{36}=1$

$x_2+x_7+x_9+x_{10}+x_{11}+x_{12}+x_{19}+x_{20}+x_{21}+x_{22}+x_{23}+x_{24}+x_{25}+x_{32}+x_{33}+x_{34}+x_{37}=1$

$x_3+x_8+x_9+x_{13}+x_{14}+x_{15}+x_{19}+x_{23}+x_{24}+x_{25}+x_{26}+x_{27}+x_{29}+x_{30}+x_{32}+x_{33}+x_{35}+x_{36}+x_{37}=1$

$x_4+x_{10}+x_{13}+x_{16}+x_{17}+x_{20}+x_{23}+x_{26}+x_{27}+x_{28}+x_{29}+x_{30}+x_{31}+x_{32}+x_{33}+x_{34}+x_{35}+x_{36}+x_{37}=1$

$x_5+x_{11}+x_{14}+x_{16}+x_{18}+x_{21}+x_{24}+x_{26}+x_{28}+x_{29}+x_{31}+x_{32}+x_{34}+x_{35}+x_{36}+x_{37}=1$

$x_6+x_{12}+x_{15}+x_{17}+x_{18}+x_{22}+x_{25}+x_{27}+x_{28}+x_{30}+x_{31}+x_{33}+x_{34}+x_{35}+x_{36}+x_{37}=1$

$x_1,\ldots,x_{37} \geq 0$

$x_1,\ldots,x_{37}$ integer

# Set Partitioning Model

| no | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | ! |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 1 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 1 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 |

min $5x_1+5x_2+\ldots+12x_{36}+9x_{37}$

$x_1+x_7+x_8+x_{19}+x_{20}+x_{21}+x_{22}+x_{29}+x_{30}+x_{31}+x_{36}=1$

$x_2+x_7+x_9+x_{10}+x_{11}+x_{12}+x_{19}+x_{20}+x_{21}+x_{22}+x_{23}+x_{24}+x_{25}+x_{32}+x_{33}+x_{34}+x_{37}=1$

$x_3+x_8+x_9+x_{13}+x_{14}+x_{15}+x_{19}+x_{23}+x_{24}+x_{25}+x_{26}+x_{27}+x_{29}+x_{30}+x_{32}+x_{33}+x_{35}+x_{36}+x_{37}=1$

$x_4+x_{10}+x_{13}+x_{16}+x_{17}+x_{20}+x_{23}+x_{26}+x_{27}+x_{28}+x_{29}+x_{30}+x_{31}+x_{32}+x_{33}+x_{34}+x_{35}+x_{36}+x_{37}=1$

$x_5+x_{11}+x_{14}+x_{16}+x_{18}+x_{21}+x_{24}+x_{26}+x_{28}+x_{29}+x_{31}+x_{32}+x_{34}+x_{35}+x_{36}+x_{37}=1$

$x_6+x_{12}+x_{15}+x_{17}+x_{18}+x_{22}+x_{25}+x_{27}+x_{28}+x_{30}+x_{31}+x_{33}+x_{34}+x_{35}+x_{36}+x_{37}=1$

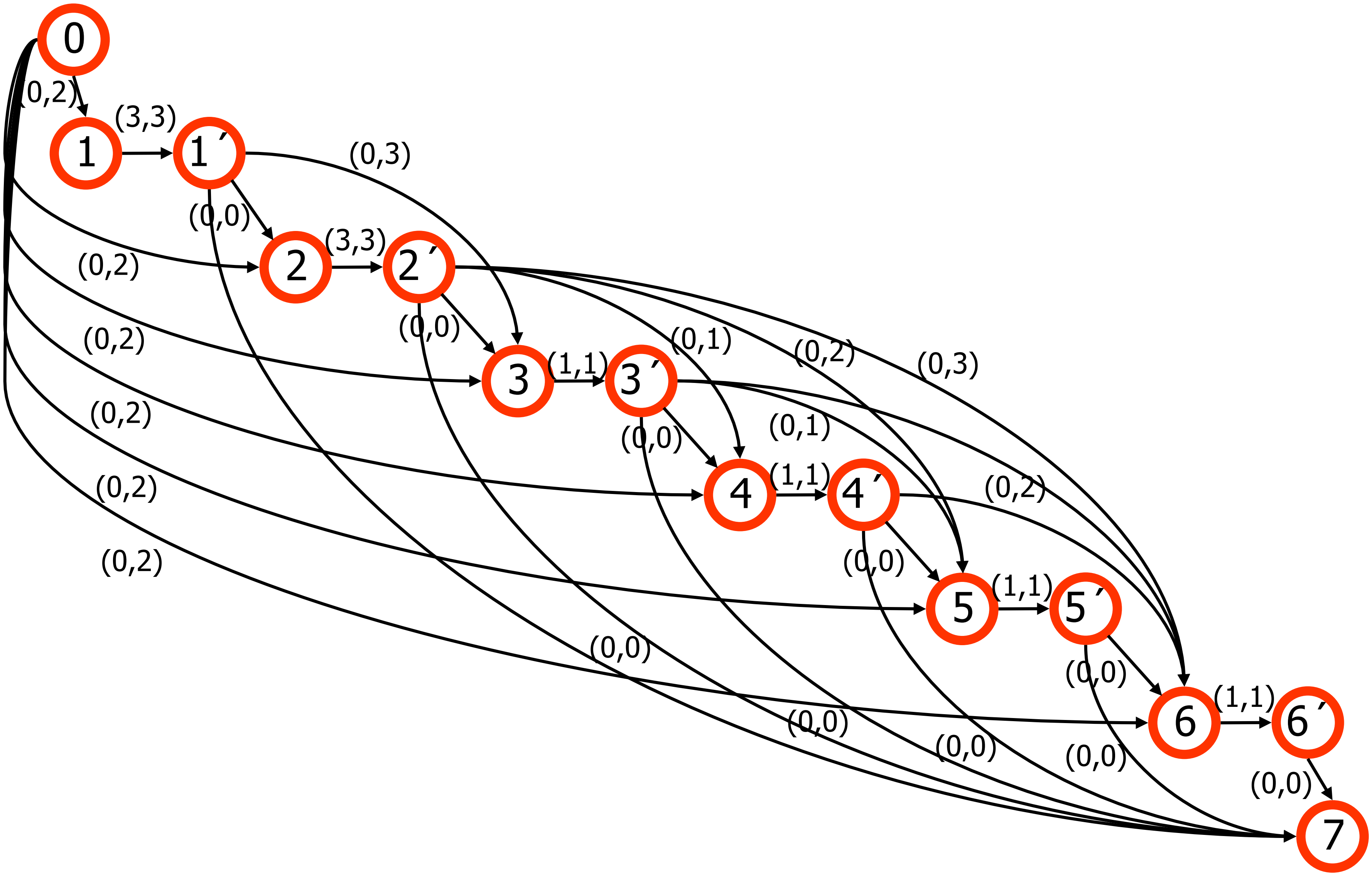
$0 \le x_1,\ldots,x_{37} \le 1$

$x_1,\ldots,x_{37}$ integer

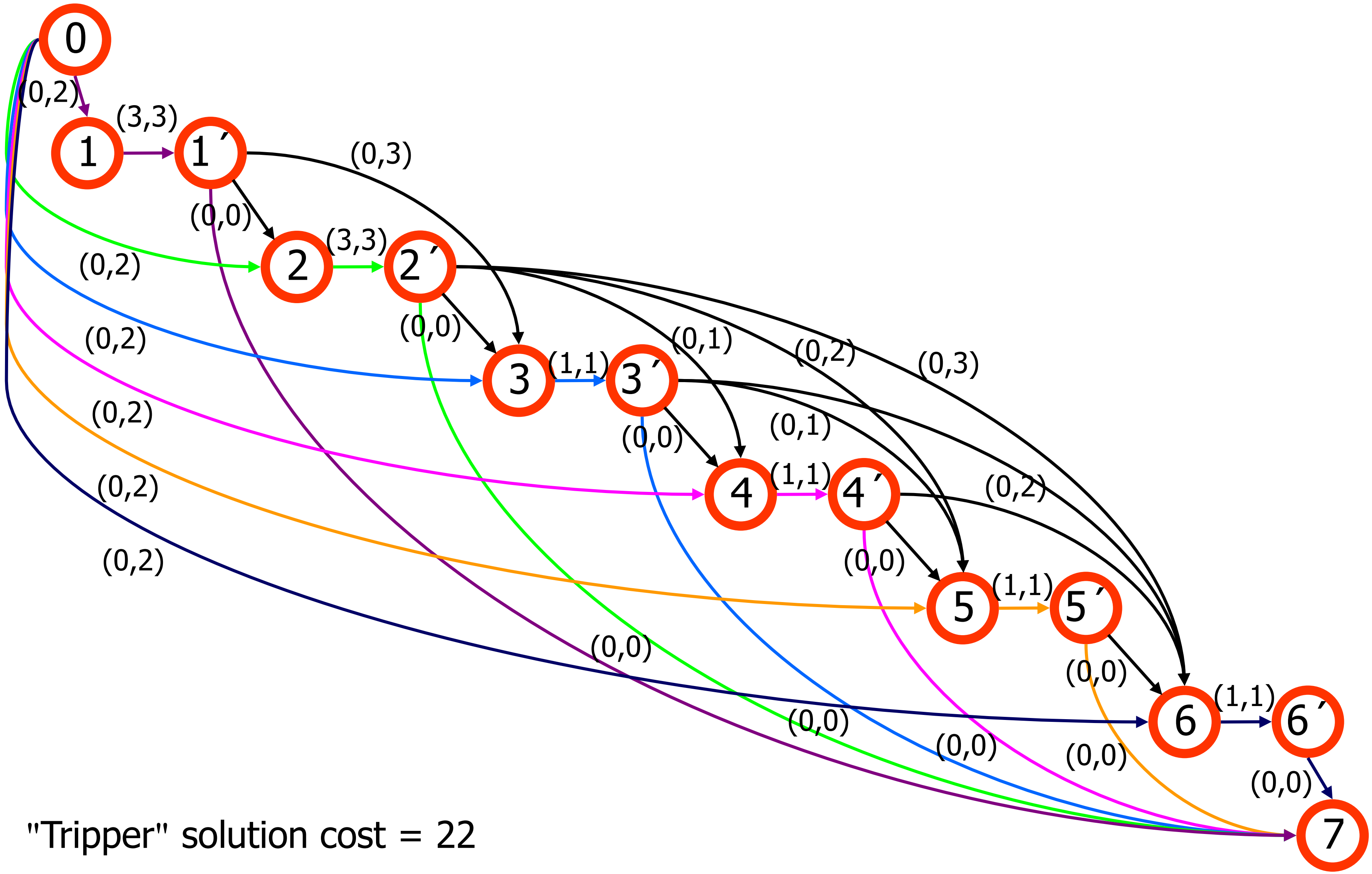
$\Leftrightarrow$ min $\sum c_j x_j,\quad \sum_{j=1,\ldots,n} a_{ij} x_j = 1,\quad i = 1,\ldots,m,\quad 0 \le x_j \le 1,\quad x_j \in \mathbb{Z}$

$\Leftrightarrow$ min $c^T x,\quad Ax = 1,\quad 0 \le x \le 1,\quad x \in \mathbb{Z}^n$

# Crew Scheduling Algorithm

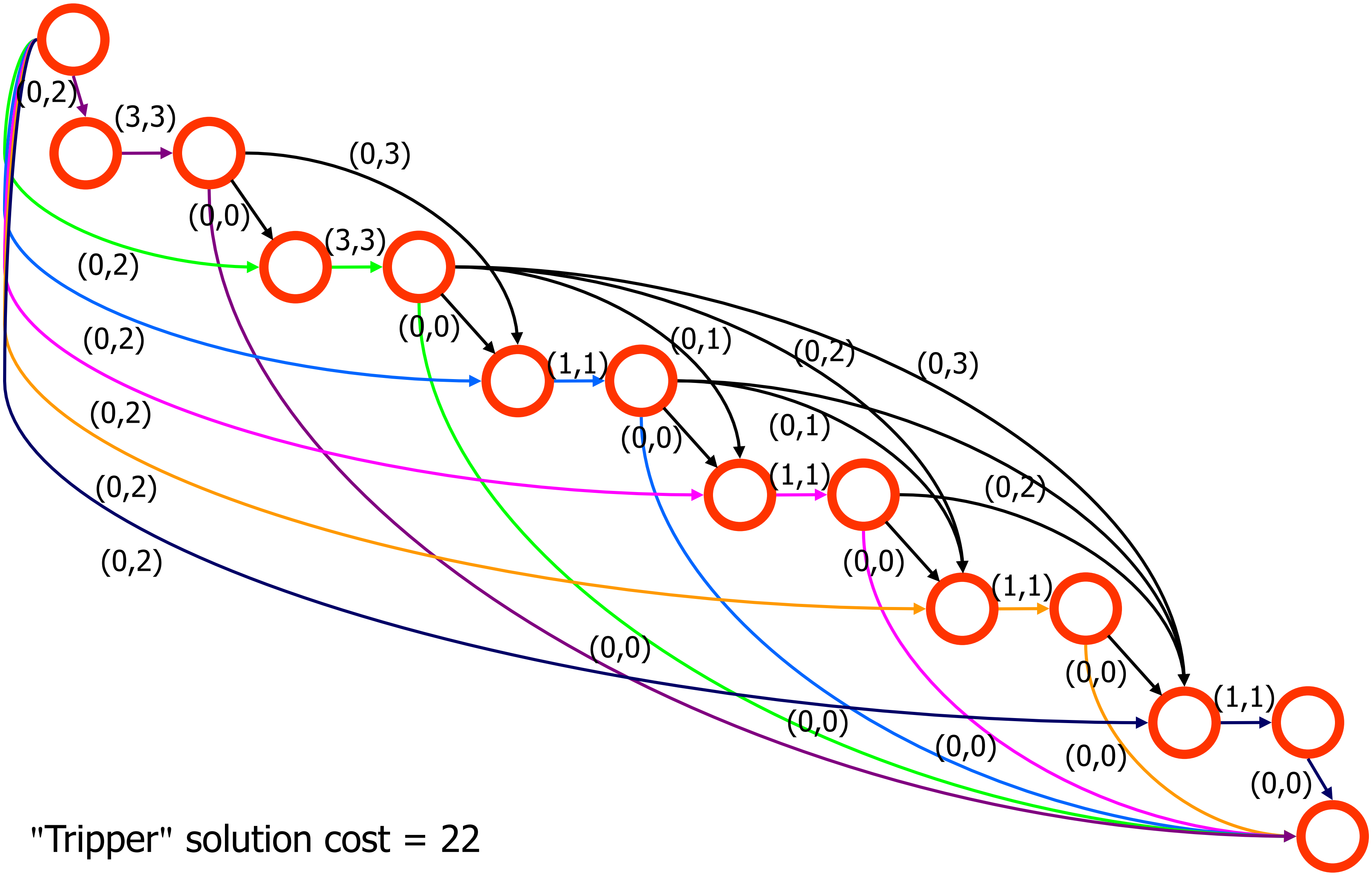


# Crew Scheduling Algorithm




- "Tripper" solution cost = 22

# Crew Scheduling Algorithm




- "Tripper" solution cost = 22

# Crew Scheduling Algorithm

| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | y |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 5 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 5 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 3 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 3 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 3 |
| x | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

$x_1=x_2=x_3=x_4=x_5=x_6=1$, cost $2*5+4*3=22$

$y_1=5$

$y_2=5$

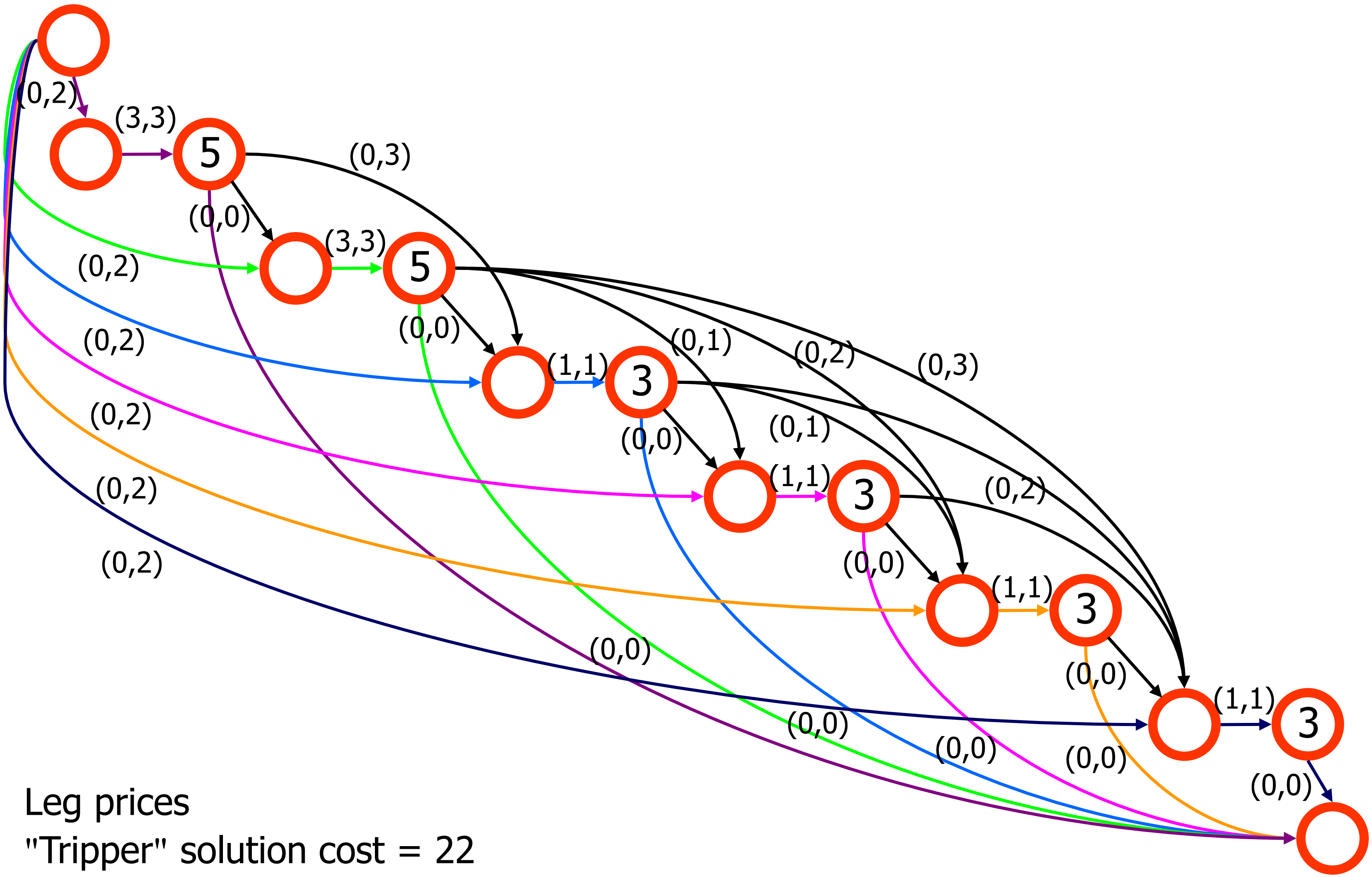
$y_3=3$

$y_4=3$

$y_5=3$

$y_6=3$

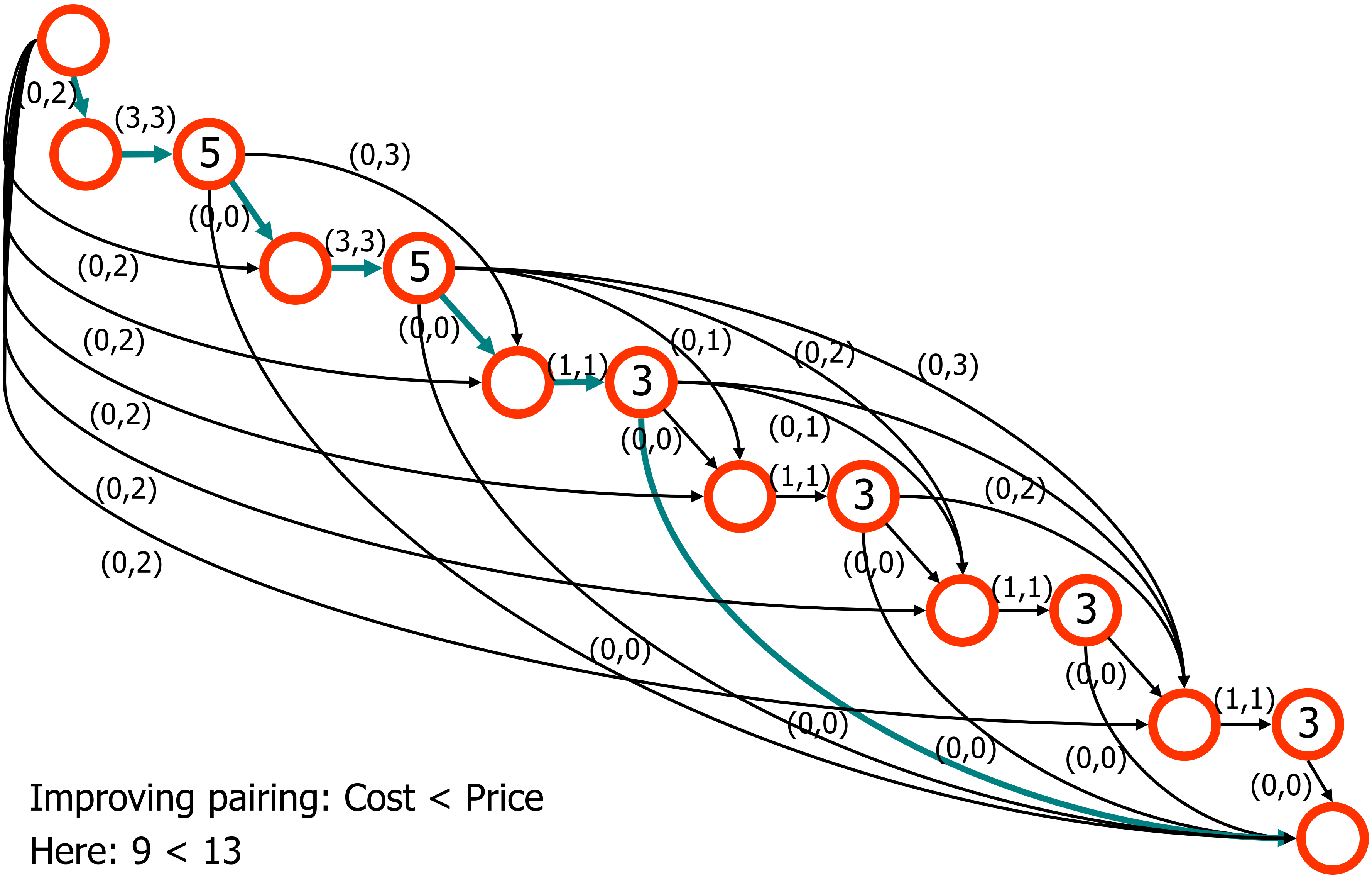
# Crew Scheduling Algorithm




- Leg prices
- "Tripper" solution cost = 22

# Crew Scheduling Algorithm






- Improving pairing: Cost < Price
- Here: 9 < 13

# Constrained Shortest Path Problem



**Definition:** *Length-constrained acyclic shortest path problem* (CSP)

- Input: Acyclic Digraph D=(V,A) with source s, sink t, integer lengths $z_{ij}$ and costs $c_{ij}$ for all arcs ij and an integer bound L.
- Output: An (s,t)-path with length ≤ L with minimal, possibly negative, costs.

**Observation:** The problem to construct an improving pairing is a length-constrained acyclic shortest path problem.

**Theorem (Desrochers):** The CSP is NP-hard.

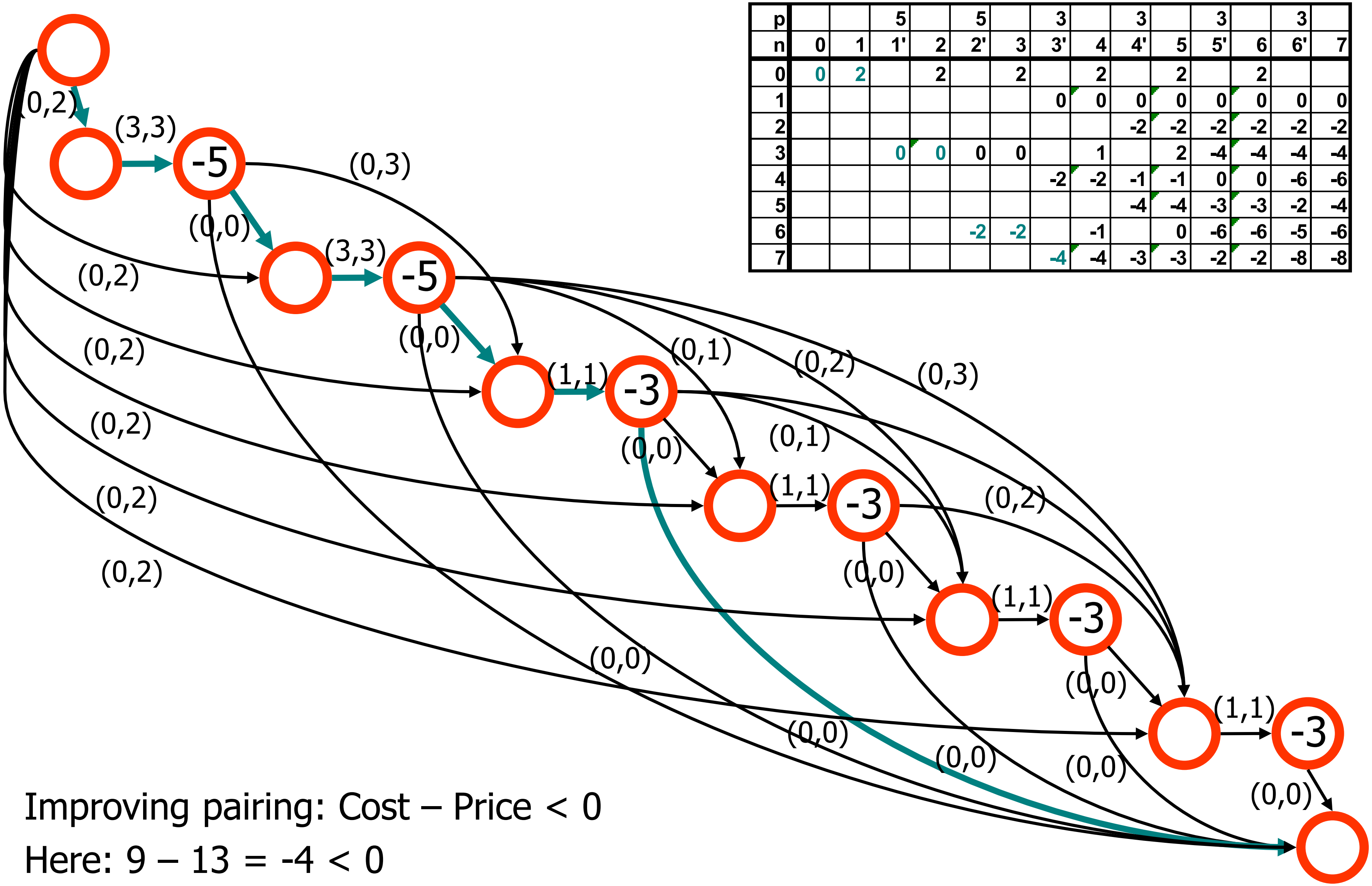
**Theorem:** The CSP can be solved in pseudo-polynomial time by dynamic programming.

**Theorem (Warburton, Hassin):** There are FPAS for the CSP.

# Crew Scheduling Algorithm



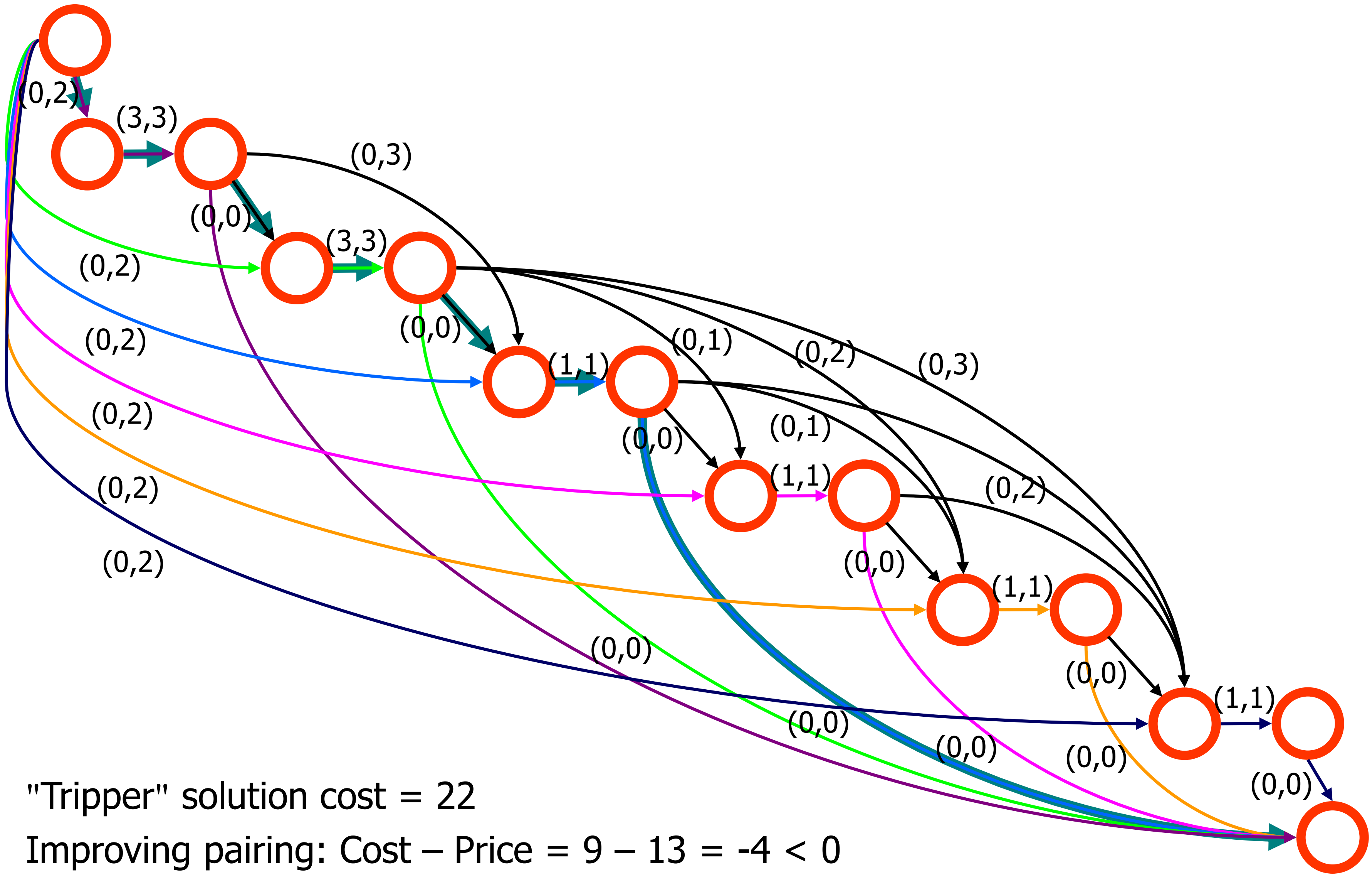
| p | | | 5 | | 5 | | 3 | | 3 | | 3 | | 3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | **0** | **1** | **1'** | **2** | **2'** | **3** | **3'** | **4** | **4'** | **5** | **5'** | **6** | **6'** | **7** |
| 0 | 0 | 2 | | 2 | | 2 | | 2 | | 2 | | 2 | | |
| 1 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | | | | | | | | | -2 | -2 | -2 | -2 | -2 | -2 |
| 3 | | | 0 | 0 | 0 | 0 | | 1 | | 2 | -4 | -4 | -4 | -4 |
| 4 | | | | | | | -2 | -2 | -1 | -1 | 0 | 0 | -6 | -6 |
| 5 | | | | | | | | | -4 | -4 | -3 | -3 | -2 | -4 |
| 6 | | | | | -2 | -2 | | -1 | | 0 | -6 | -6 | -5 | -6 |
| 7 | | | | | | | -4 | -4 | -3 | -3 | -2 | -2 | -8 | -8 |




- Improving pairing: Cost – Price < 0
- Here: 9 – 13 = -4 < 0

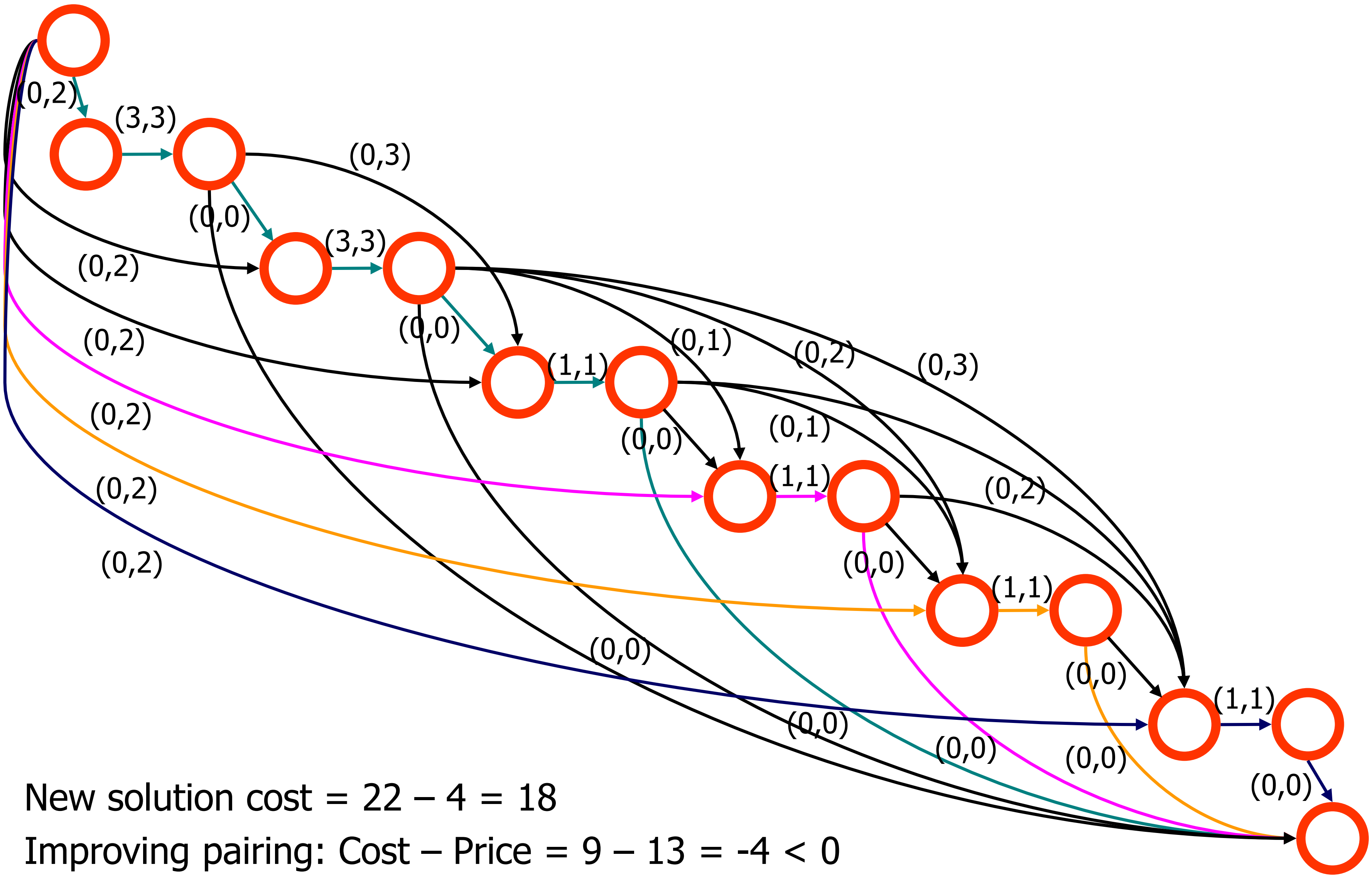
# Crew Scheduling Algorithm






- "Tripper" solution cost = 22
- Improving pairing: Cost – Price = 9 – 13 = -4 < 0

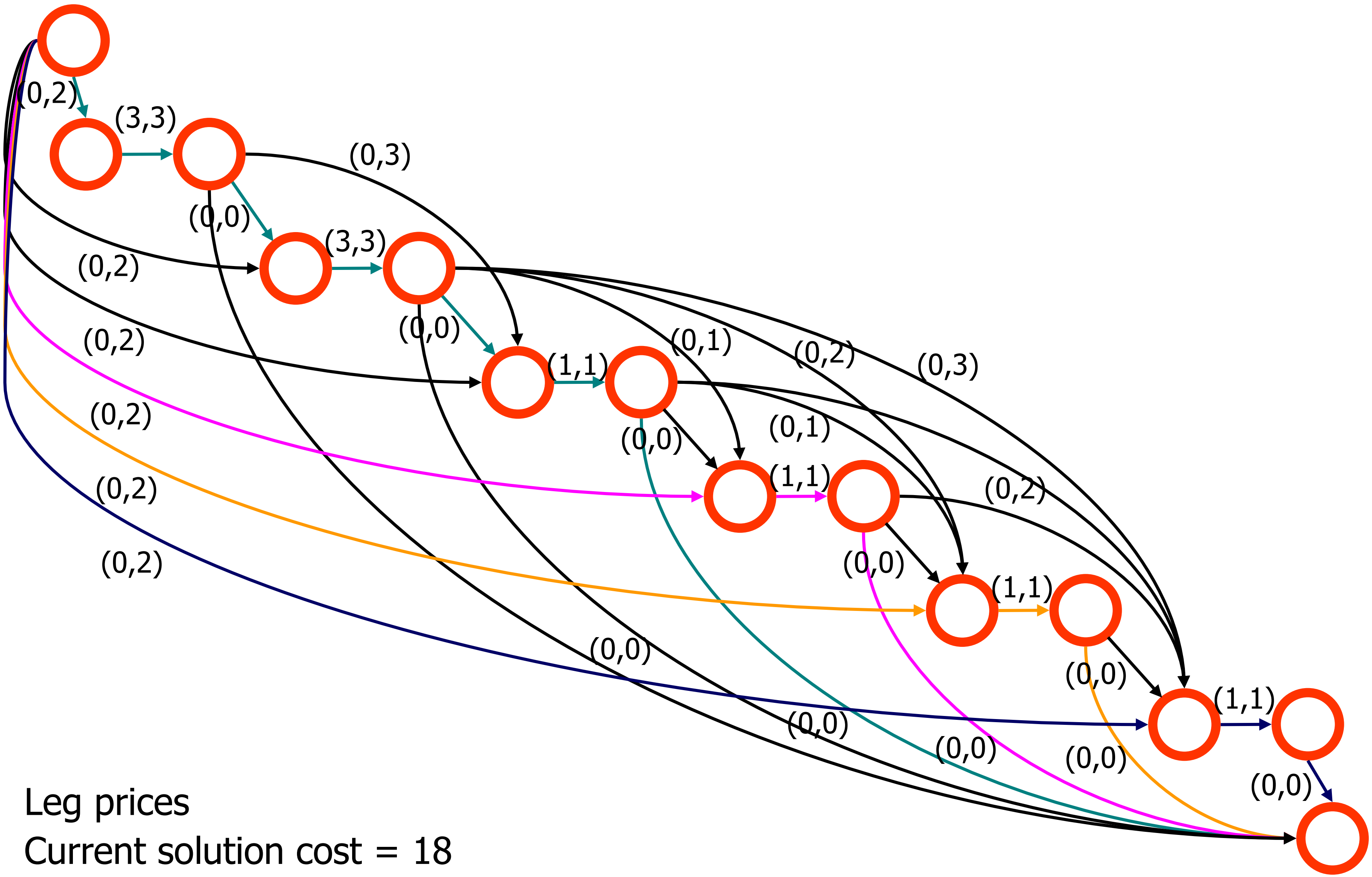
# Crew Scheduling Algorithm




- New solution cost = 22 – 4 = 18
- Improving pairing: Cost – Price = 9 – 13 = -4 < 0

# Crew Scheduling Algorithm






- Leg prices
- Current solution cost = 18

# Column Generation

| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | y |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 1 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 5 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 3 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 3 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 3 |
| x | 1 | 1 | 1 | | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | | | | | | | | |

$x_1=x_2=x_3=0$, $x_4=x_5=x_6= x_{19}=1$, cost 9+3*3=18 [22]

$y_4=3$, $y_5=3$, $y_6=3$, $y_1+y_2+y_3=9$

$y_1=1$, $y_2=5$, $y_3 =3$, $y_4=3$, $y_5=3$, $y_6=3$

Other solutions possible

# Column Generation: Primal Master LP

$$
\begin{array}{lllllllll}
\min 5x_1 + 5x_2 + 3x_3 + 3x_4 + 3x_5 + 3x_6 + 9x_{19} & & & & & & & & \\
x_1 & & & & & & + & x_{19} & = 1 \\
& x_2 & & & & & + & x_{19} & = 1 \\
& & x_3 & & & & + & x_{19} & = 1 \\
& & & x_4 & & & & & = 1 \\
& & & & x_5 & & & & = 1 \\
& & & & & x_6 & & & = 1 \\
x_1, & x_2, & x_3, & x_4, & x_4, & x_6, & x_{19} & & \geq 0
\end{array}
$$

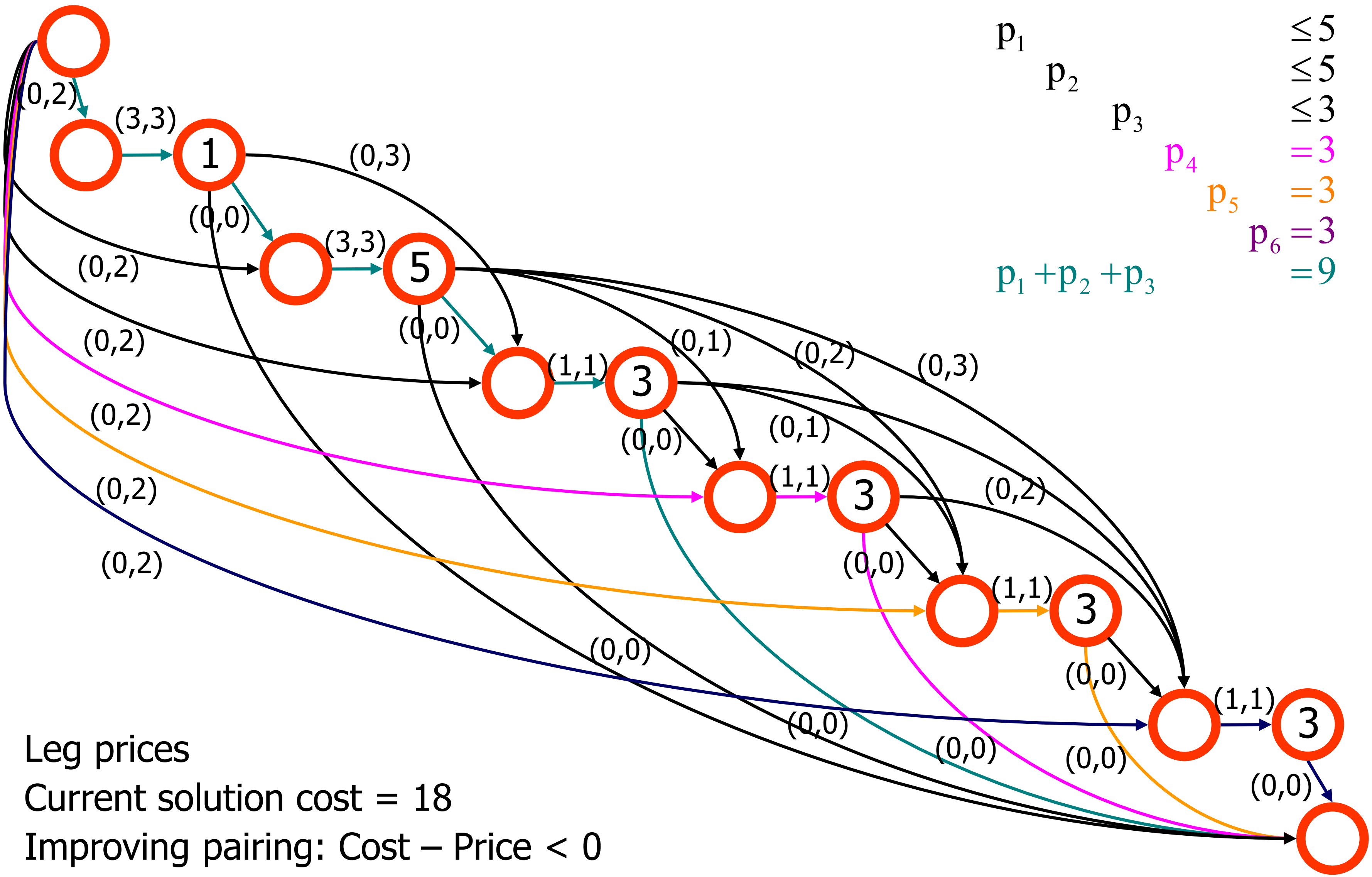
# Column Generation: Dual Master LP



$$
\begin{aligned}
\max\ & y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \\
& y_1 \le 5 \\
& y_2 \le 5 \\
& y_3 \le 3 \\
& y_4 \le 3 \\
& y_5 \le 3 \\
& y_6 \le 3 \\
& y_1 + y_2 + y_3 \le 9
\end{aligned}
$$

# Crew Scheduling Algorithm



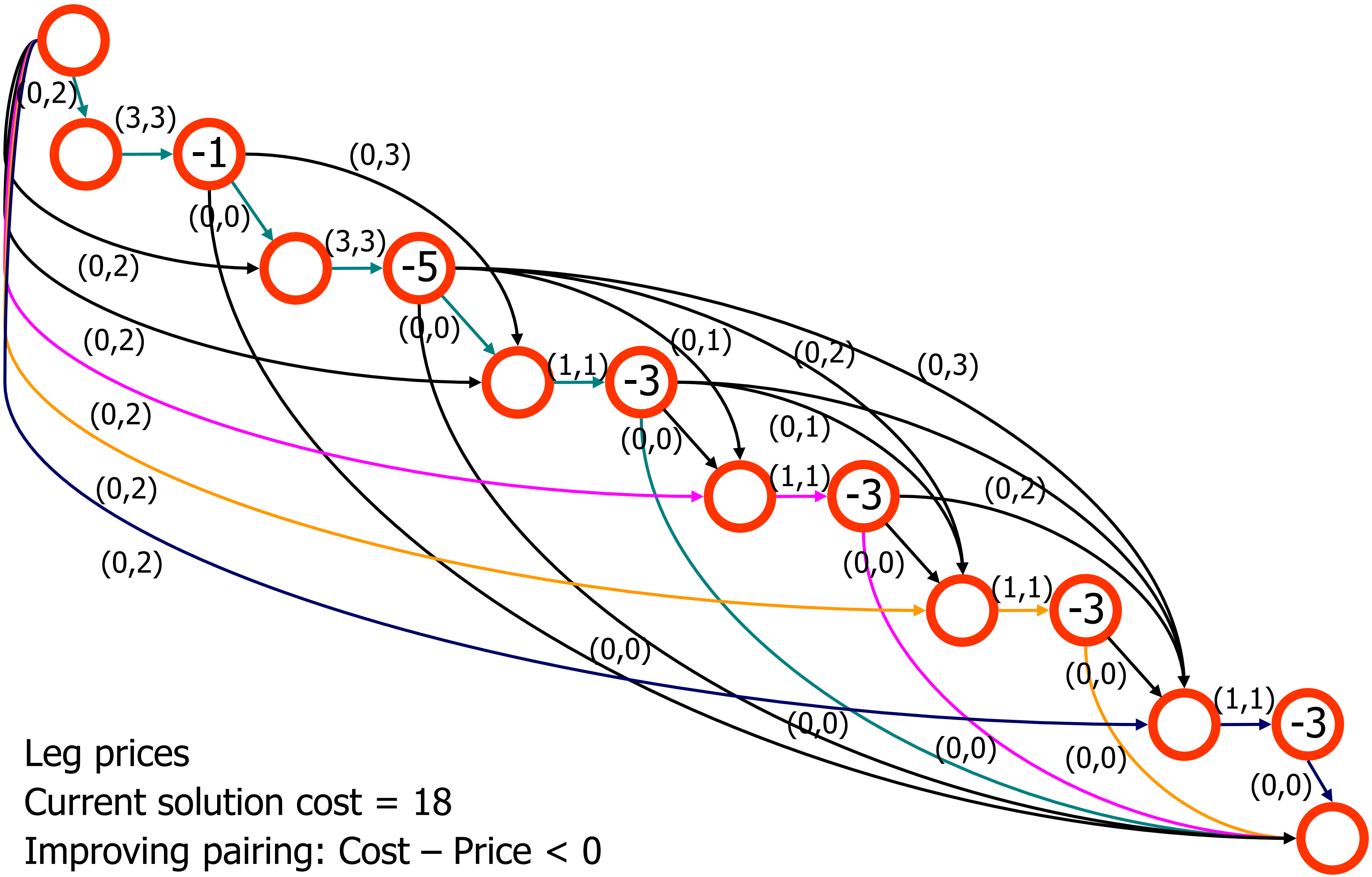



$$
\begin{aligned}
p_1 &\leq 5 \\
p_2 &\leq 5 \\
p_3 &\leq 3 \\
p_4 &= 3 \\
p_5 &= 3 \\
p_6 &= 3 \\
p_1 + p_2 + p_3 &= 9
\end{aligned}
$$

- Leg prices
- Current solution cost = 18
- Improving pairing: Cost – Price < 0

# Crew Scheduling Algorithm



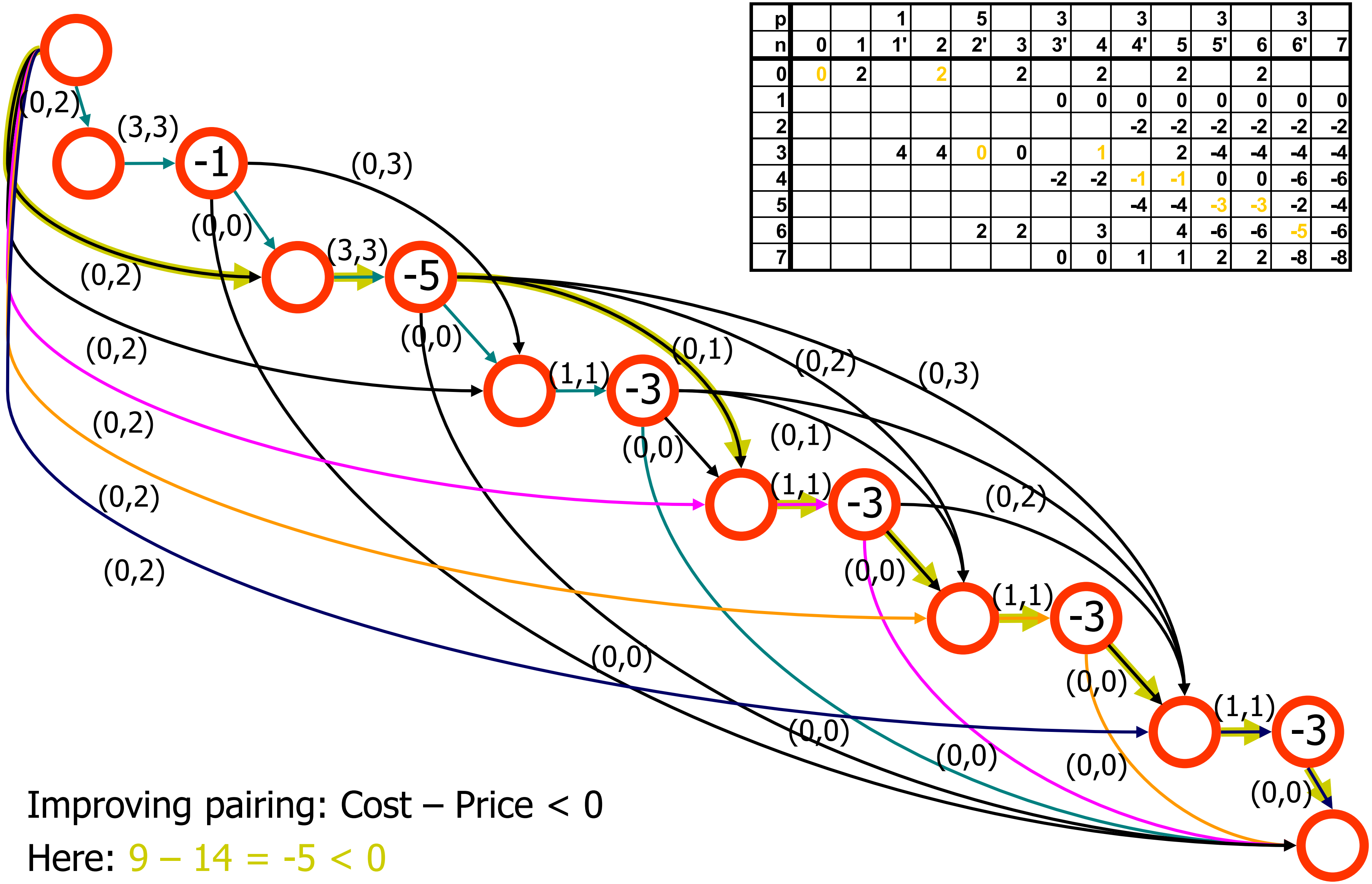



- Leg prices
- Current solution cost = 18
- Improving pairing: Cost – Price < 0

# Crew Scheduling Algorithm



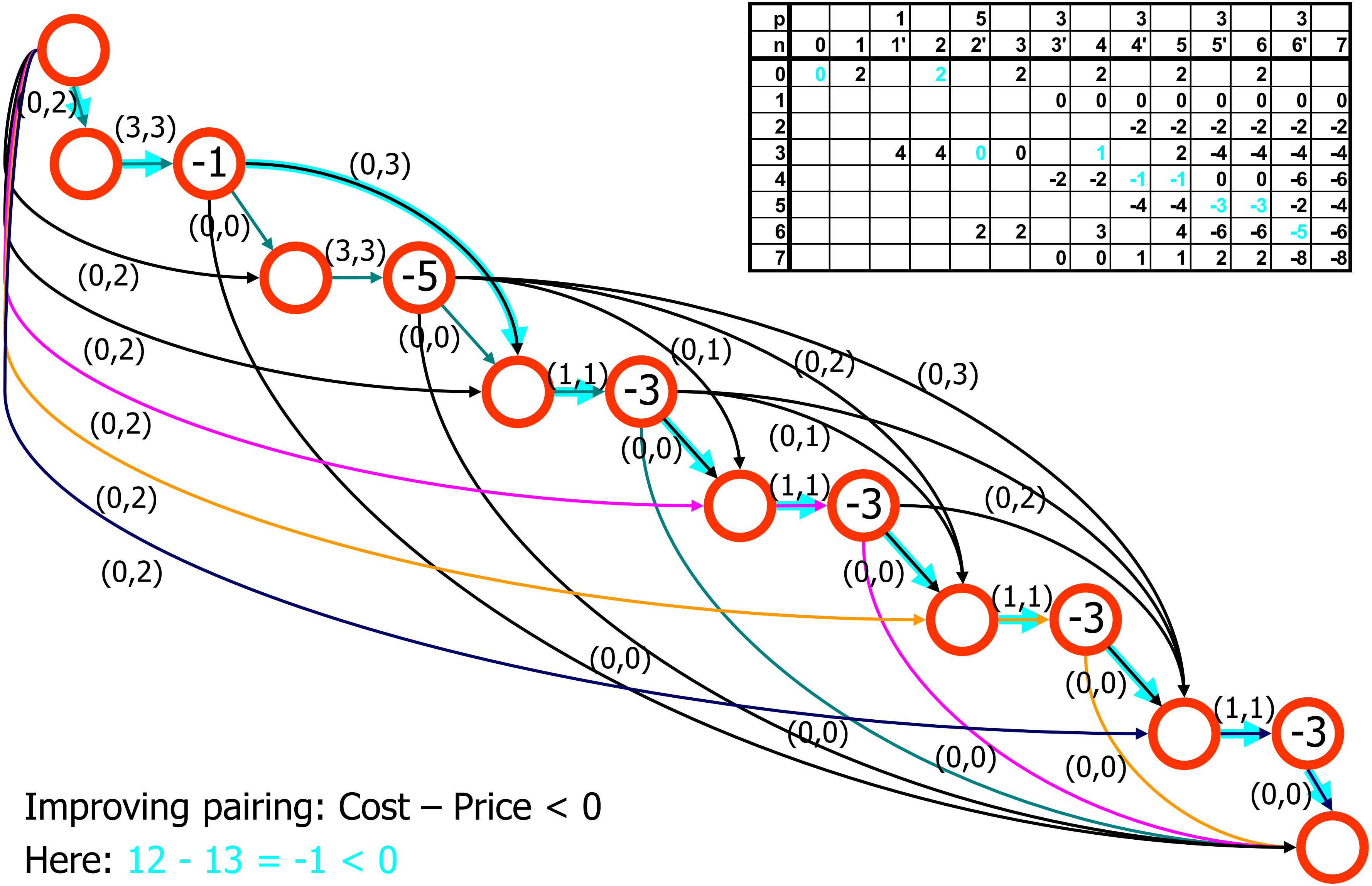
| p | | | 1 | | 5 | | 3 | | 3 | | 3 | | 3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | 0 | 1 | 1' | 2 | 2' | 3 | 3' | 4 | 4' | 5 | 5' | 6 | 6' | 7 |
| 0 | 0 | 2 | | 2 | | 2 | | 2 | | 2 | | 2 | | |
| 1 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | | | | | | | | | -2 | -2 | -2 | -2 | -2 | -2 |
| 3 | | | 4 | 4 | 0 | 0 | | 1 | | 2 | -4 | -4 | -4 | -4 |
| 4 | | | | | | | -2 | -2 | -1 | -1 | 0 | 0 | -6 | -6 |
| 5 | | | | | | | | | -4 | -4 | -3 | -3 | -2 | -4 |
| 6 | | | | | 2 | 2 | | 3 | | 4 | -6 | -6 | -5 | -6 |
| 7 | | | | | | | 0 | 0 | 1 | 1 | 2 | 2 | -8 | -8 |




- Improving pairing: Cost – Price < 0
- Here: 9 – 14 = -5 < 0

# Crew Scheduling Algorithm

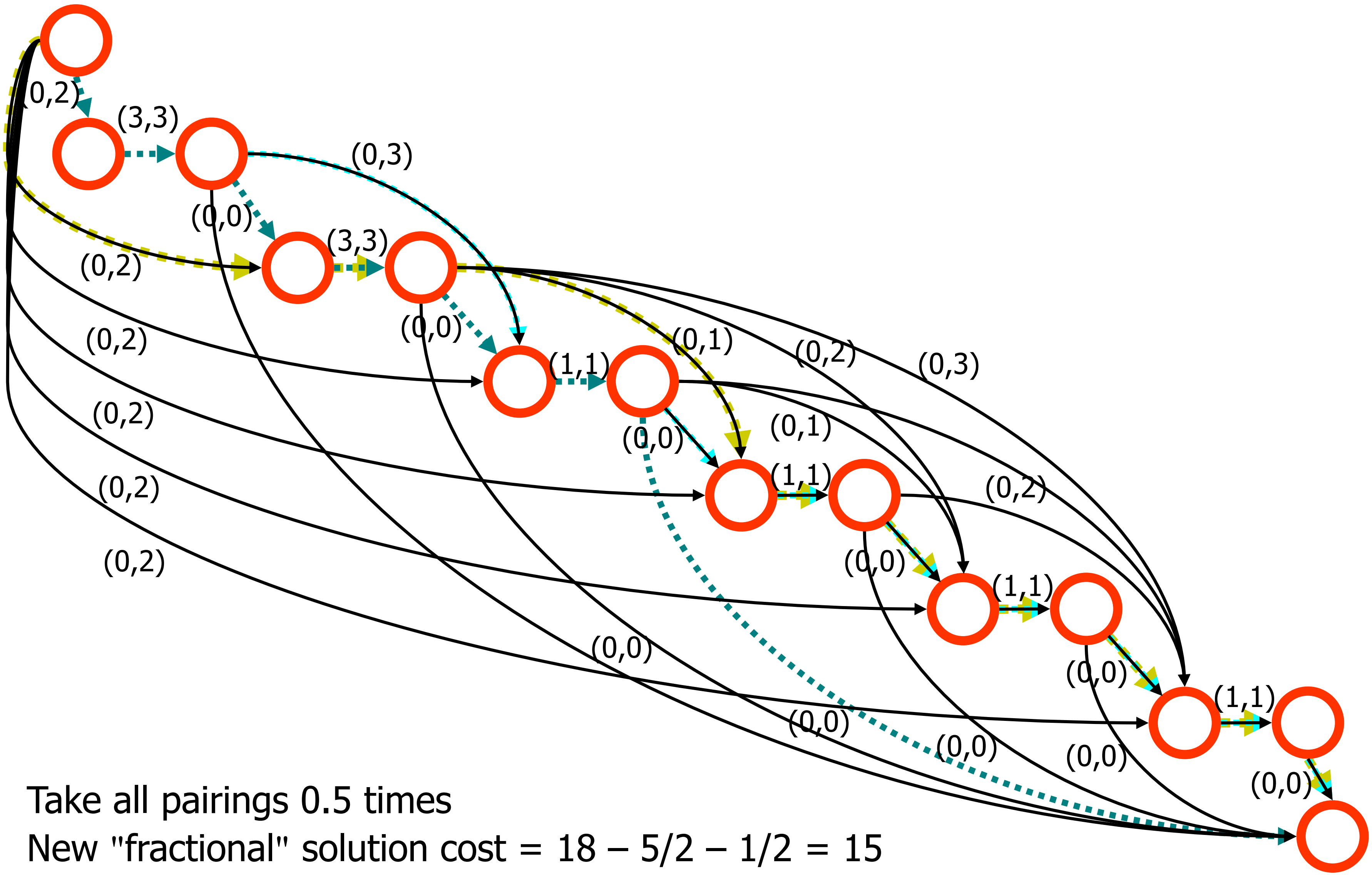
| p | | | 1 | | 5 | | 3 | | 3 | | 3 | | 3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | 0 | 1 | 1' | 2 | 2' | 3 | 3' | 4 | 4' | 5 | 5' | 6 | 6' | 7 |
| 0 | 0 | 2 | | 2 | | 2 | | 2 | | 2 | | 2 | | |
| 1 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | | | | | | | | | -2 | -2 | -2 | -2 | -2 | -2 |
| 3 | | | 4 | 4 | 0 | 0 | | 1 | | 2 | -4 | -4 | -4 | -4 |
| 4 | | | | | | | -2 | -2 | -1 | -1 | 0 | 0 | -6 | -6 |
| 5 | | | | | | | | | -4 | -4 | -3 | -3 | -2 | -4 |
| 6 | | | | | 2 | 2 | | 3 | | 4 | -6 | -6 | -5 | -6 |
| 7 | | | | | | | 0 | 0 | 1 | 1 | 2 | 2 | -8 | -8 |




- Improving pairing: Cost – Price < 0
- Here: 12 - 13 = -1 < 0

# Crew Scheduling Algorithm






- Take all pairings 0.5 times
- New "fractional" solution cost = 18 – 5/2 – 1/2 = 15

| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | y |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 5 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 3 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 3 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 0 |
| x | | | | | | | | | | | | | | | | | | | ½ | | | | | | | | | | | | | | | ½ | | ½ | | |

$x_{34}= x_{34}= x_{19}= 1/2$, cost $(9+9+12)/2=15$ [18]

$y_1+y_2+y_3=9$

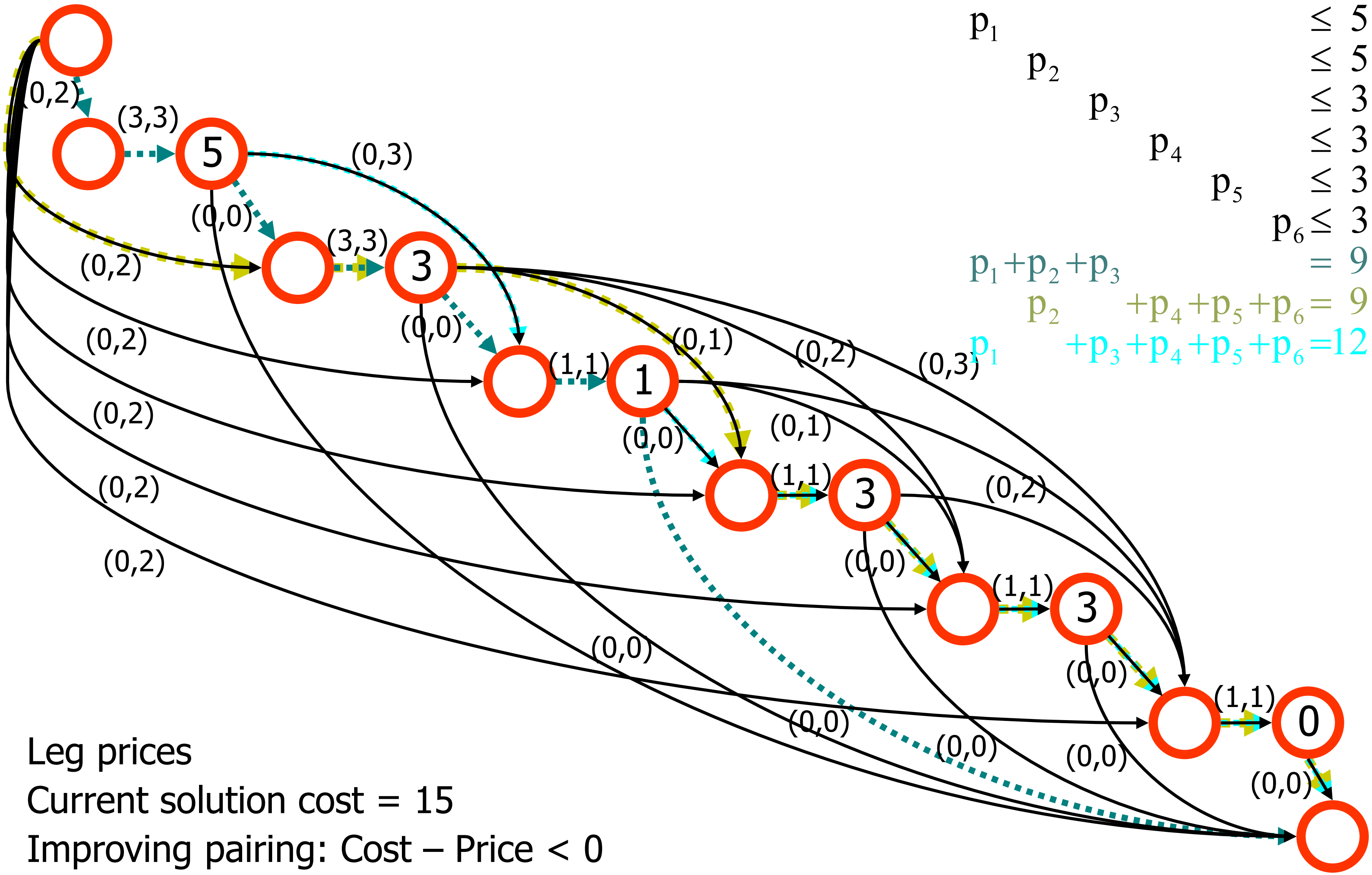
$y_2+y_4+y_5+y_6 =9$

$y_1+y_3+y_4+y_5+y_5=12$

$\Rightarrow\ y_1=5,\ y_2 = y_4 = y_5 = 3,\ y_3= 1,\ y_6= 0$

# Crew Scheduling Algorithm





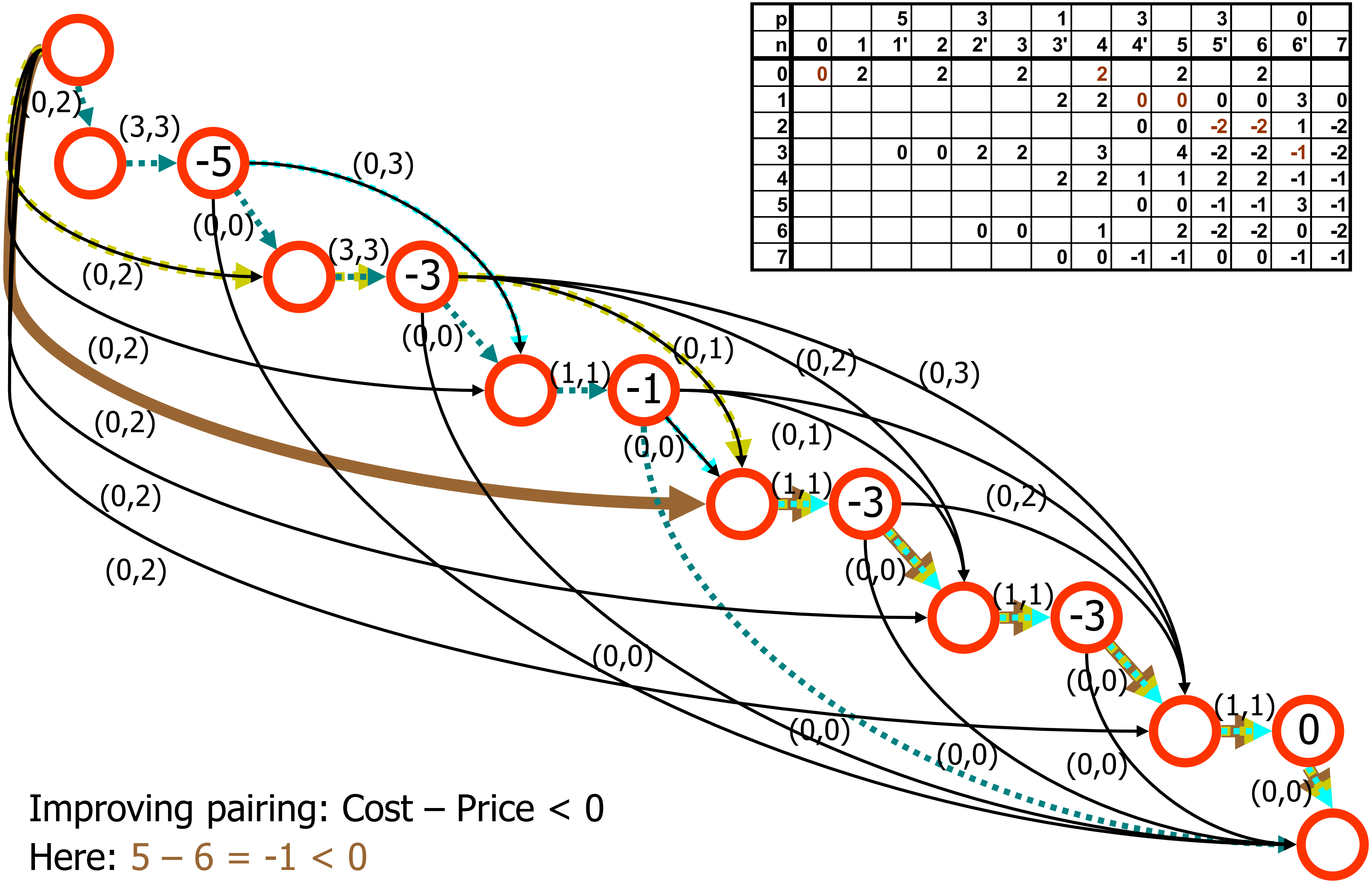

$$
\begin{aligned}
p_1 &\le 5\\
p_2 &\le 5\\
p_3 &\le 3\\
p_4 &\le 3\\
p_5 &\le 3\\
p_6 &\le 3\\
p_1+p_2+p_3 &= 9\\
p_2+p_4+p_5+p_6 &= 9\\
p_1+p_3+p_4+p_5+p_6 &= 12
\end{aligned}
$$

- Leg prices
- Current solution cost = 15
- Improving pairing: Cost – Price < 0

# Crew Scheduling Algorithm



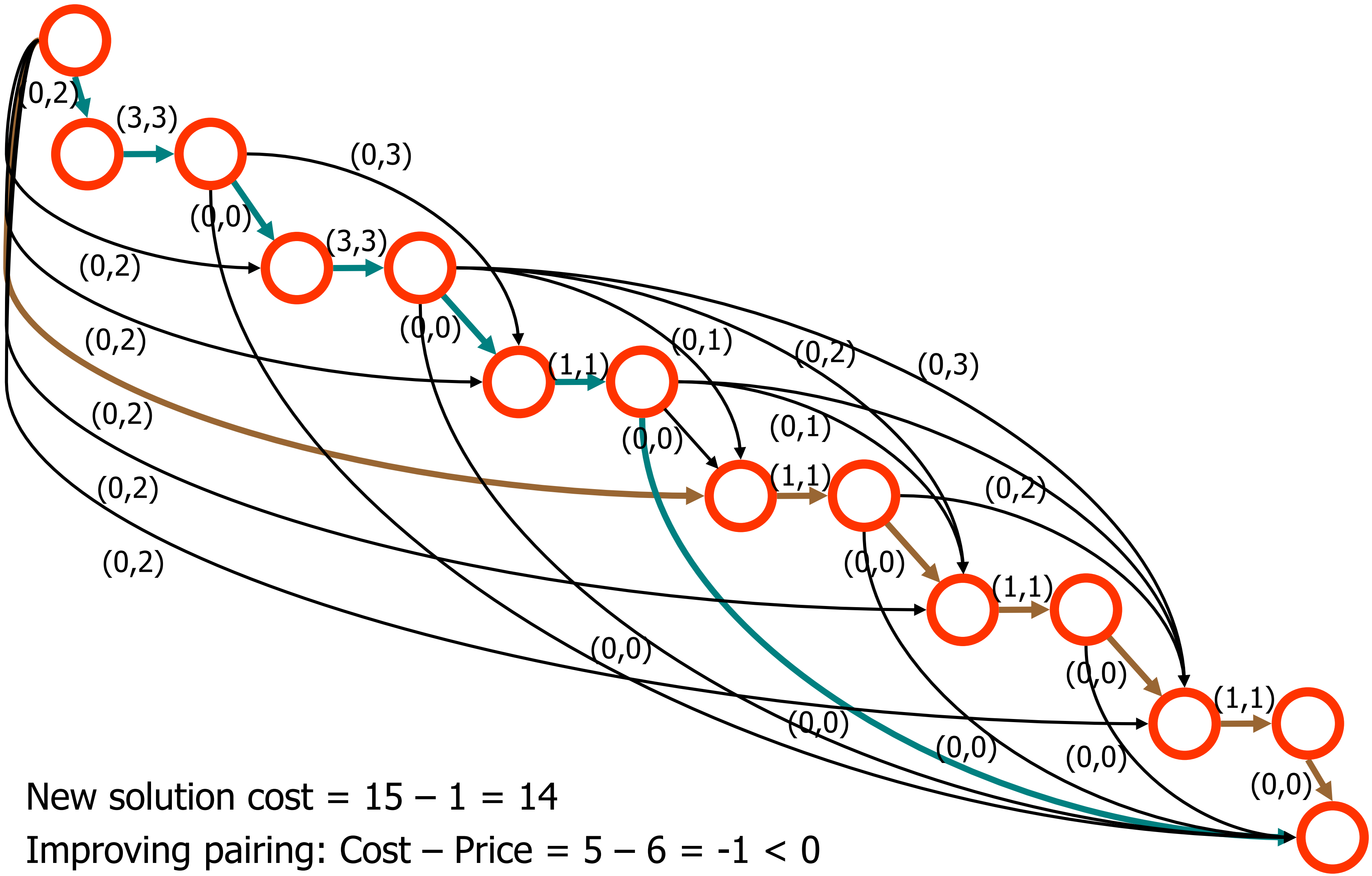
| p | | | 5 | | 3 | | 1 | | 3 | | 3 | | 0 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | 0 | 1 | 1' | 2 | 2' | 3 | 3' | 4 | 4' | 5 | 5' | 6 | 6' | 7 |
| 0 | 0 | 2 | | 2 | | 2 | | 2 | | 2 | | 2 | | |
| 1 | | | | | | | 2 | 2 | 0 | 0 | 0 | 0 | 3 | 0 |
| 2 | | | | | | | | | 0 | 0 | -2 | -2 | 1 | -2 |
| 3 | | | 0 | 0 | 2 | 2 | | 3 | | 4 | -2 | -2 | -1 | -2 |
| 4 | | | | | | | 2 | 2 | 1 | 1 | 2 | 2 | -1 | -1 |
| 5 | | | | | | | | | 0 | 0 | -1 | -1 | 3 | -1 |
| 6 | | | | | 0 | 0 | | 1 | | 2 | -2 | -2 | 0 | -2 |
| 7 | | | | | | | 0 | 0 | -1 | -1 | 0 | 0 | -1 | -1 |




- Improving pairing: Cost – Price < 0
- Here: 5 – 6 = -1 < 0

# Crew Scheduling Algorithm






- New solution cost = 15 – 1 = 14
- Improving pairing: Cost – Price = 5 – 6 = -1 < 0

# Column Generation



| no | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | 5 | 5 | 3 | 3 | 3 | 3 | 8 | 9 | 6 | 7 | 8 | 9 | 4 | 5 | 6 | 4 | 5 | 4 | 9 | 10 | 11 | 12 | 7 | 8 | 9 | 5 | 6 | 5 | 11 | 12 | 12 | 8 | 9 | 9 | 6 | 12 | 9 | y |
| 1 | 1 | | | | | | 1 | 1 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | | | 1 | | 5 |
| 2 | | 1 | | | | | 1 | | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 | 3 |
| 3 | | | 1 | | | | | 1 | 1 | | | | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | 1 | 1 | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 |
| 4 | | | | 1 | | | | | | 1 | | | 1 | | | 1 | 1 | | | 1 | | | 1 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 5 | | | | | 1 | | | | | | 1 | | | 1 | | 1 | | 1 | | | 1 | | | 1 | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 2 |
| 6 | | | | | | 1 | | | | | | 1 | | | 1 | | 1 | 1 | | | | 1 | | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 |
| x | | | | | | | | | | | | | | | | | | | 1 | | | | | | | | | 1 | | | | | | | | | | |

x19= x28= 1, cost 9+5=14 [15]

y1+y2+y3=9

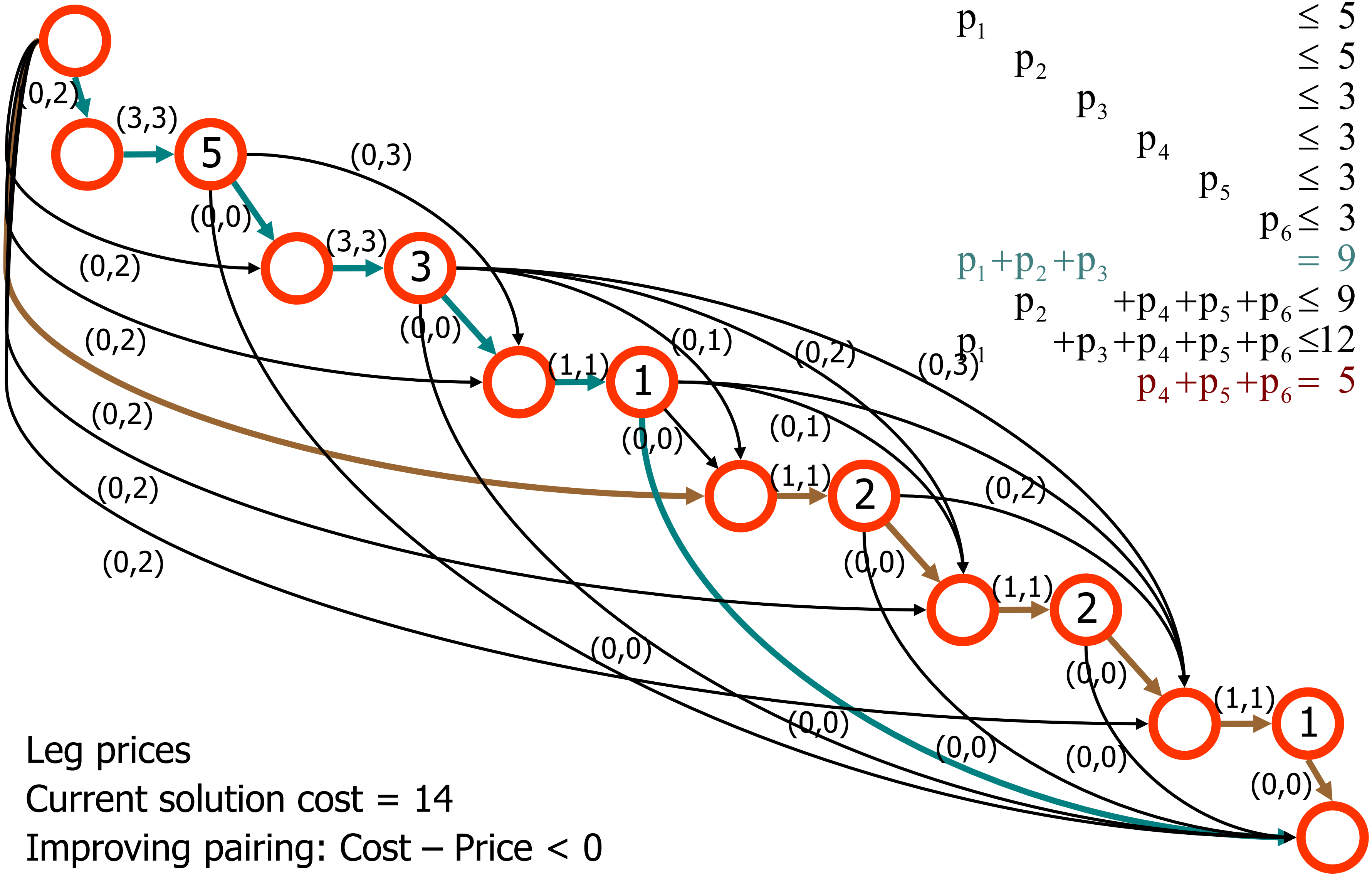
y4+y5+y6 =5 ⇒ y1= 5, y2= 3, y3= y6= 1, y4= y5 = 2

Prices buy all duties

Duality theorem (or simplex criterion) ⇒ x* and y* are optimal

# Crew Scheduling Algorithm





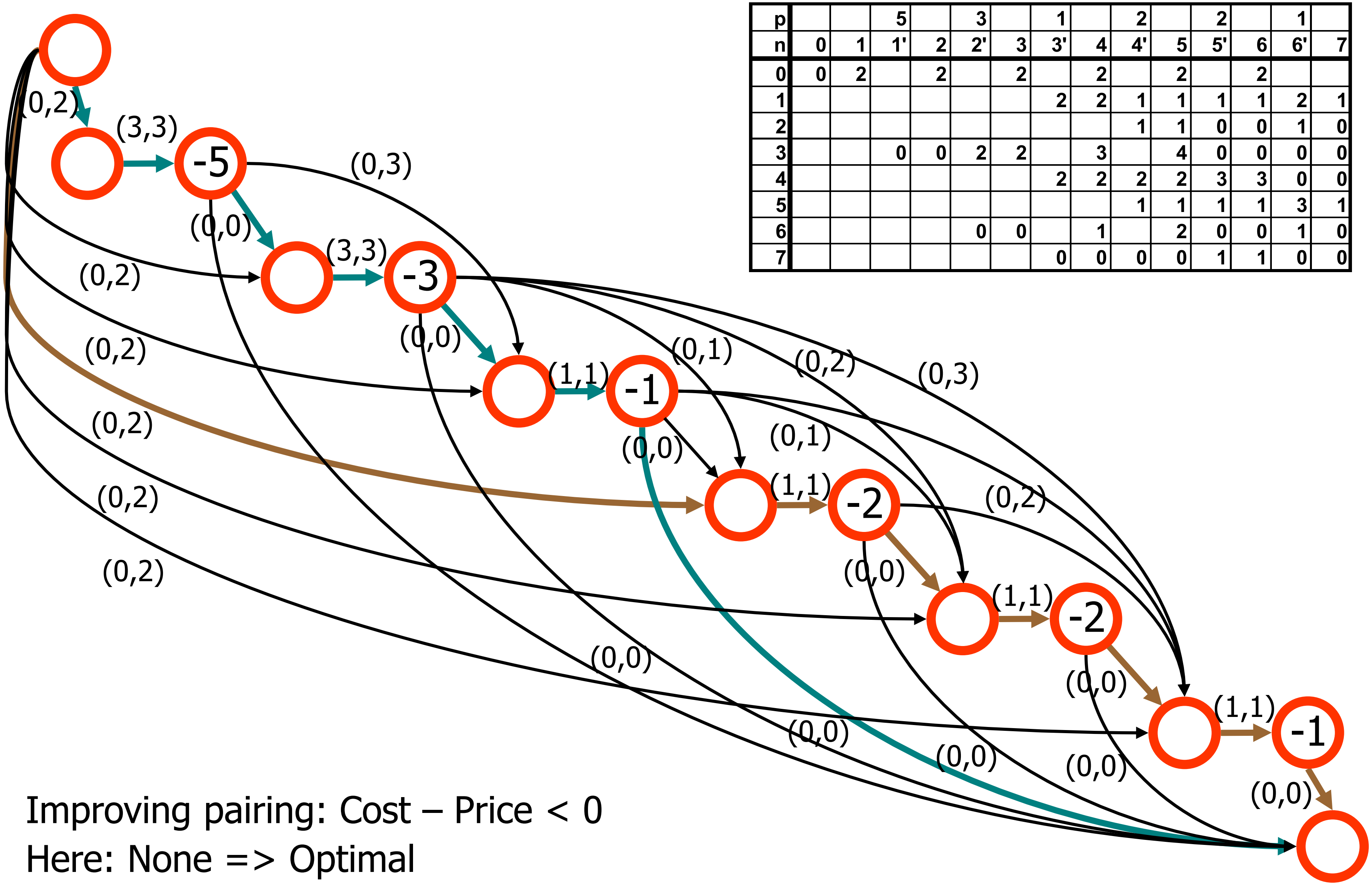

$$
\begin{array}{lcl}
p_1 & \leq & 5 \\
p_2 & \leq & 5 \\
p_3 & \leq & 3 \\
p_4 & \leq & 3 \\
p_5 & \leq & 3 \\
p_6 & \leq & 3 \\
p_1 + p_2 + p_3 & = & 9 \\
p_2 + p_4 + p_5 + p_6 & \leq & 9 \\
p_1 + p_3 + p_4 + p_5 + p_6 & \leq & 12 \\
p_4 + p_5 + p_6 & = & 5
\end{array}
$$

- Leg prices
- Current solution cost = 14
- Improving pairing: Cost – Price < 0

# Crew Scheduling Algorithm

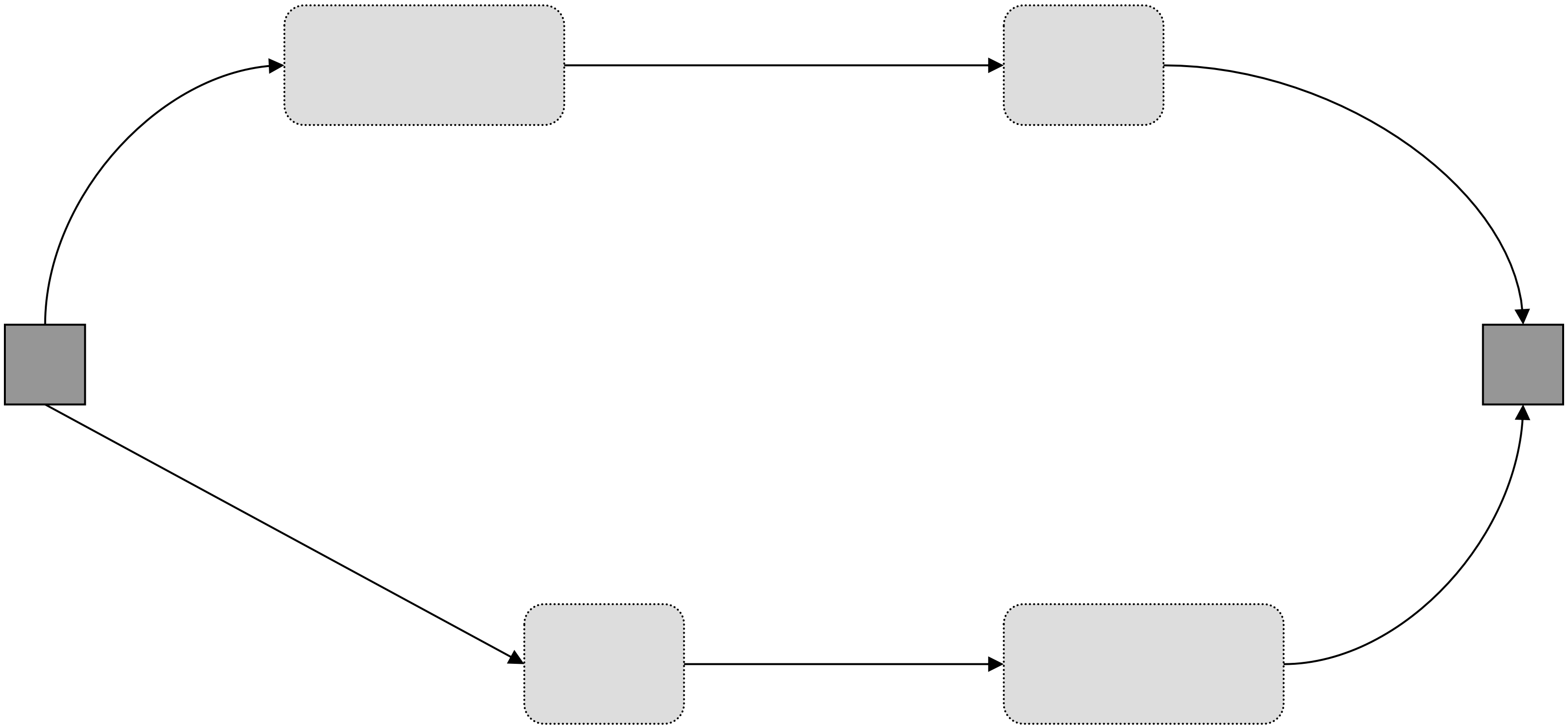


| p | | | 5 | | 3 | | 1 | | 2 | | 2 | | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **n** | **0** | **1** | **1'** | **2** | **2'** | **3** | **3'** | **4** | **4'** | **5** | **5'** | **6** | **6'** | **7** |
| **0** | 0 | 2 | | 2 | | 2 | | 2 | | 2 | | 2 | | |
| **1** | | | | | | | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1 |
| **2** | | | | | | | | | 1 | 1 | 0 | 0 | 1 | 0 |
| **3** | | | 0 | 0 | 2 | 2 | | 3 | | 4 | 0 | 0 | 0 | 0 |
| **4** | | | | | | | 2 | 2 | 2 | 2 | 3 | 3 | 0 | 0 |
| **5** | | | | | | | | | 1 | 1 | 1 | 1 | 3 | 1 |
| **6** | | | | | 0 | 0 | | 1 | | 2 | 0 | 0 | 1 | 0 |
| **7** | | | | | | | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |




- Improving pairing: Cost – Price < 0
- Here: None => Optimal

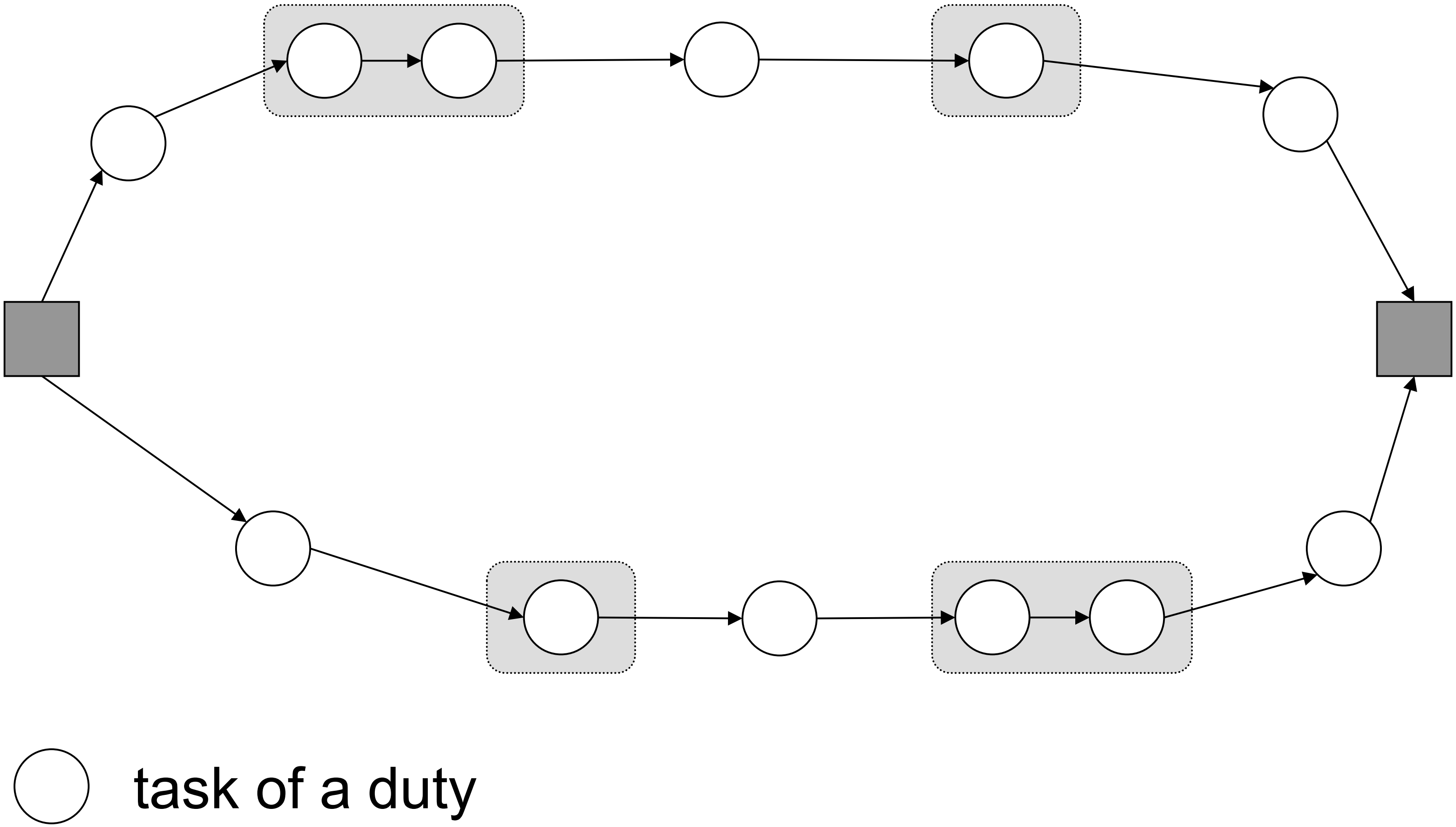
# Graph Theoretic Model



 timetabled trip → deadhead trip

# Graph Theoretic Model

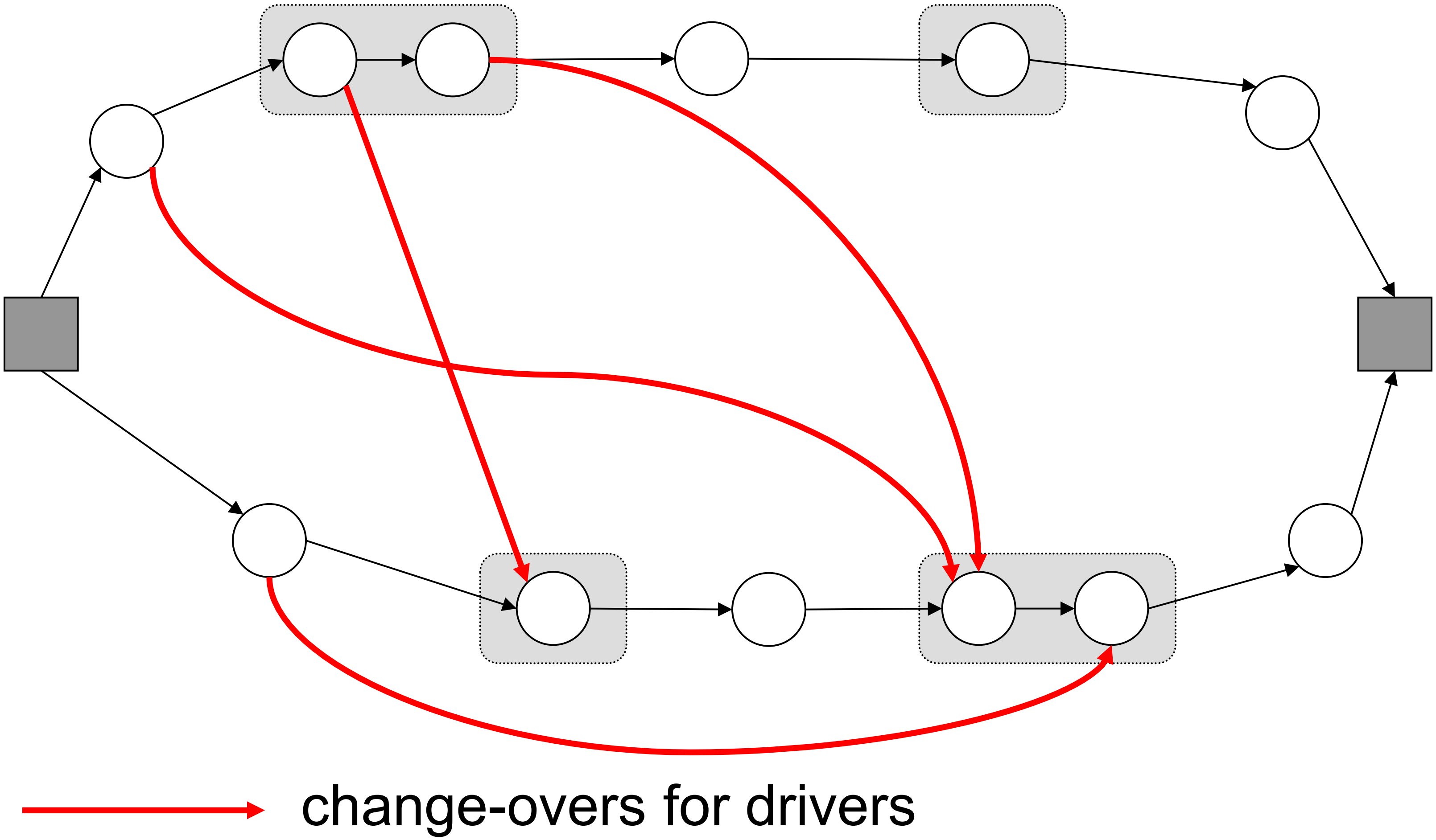




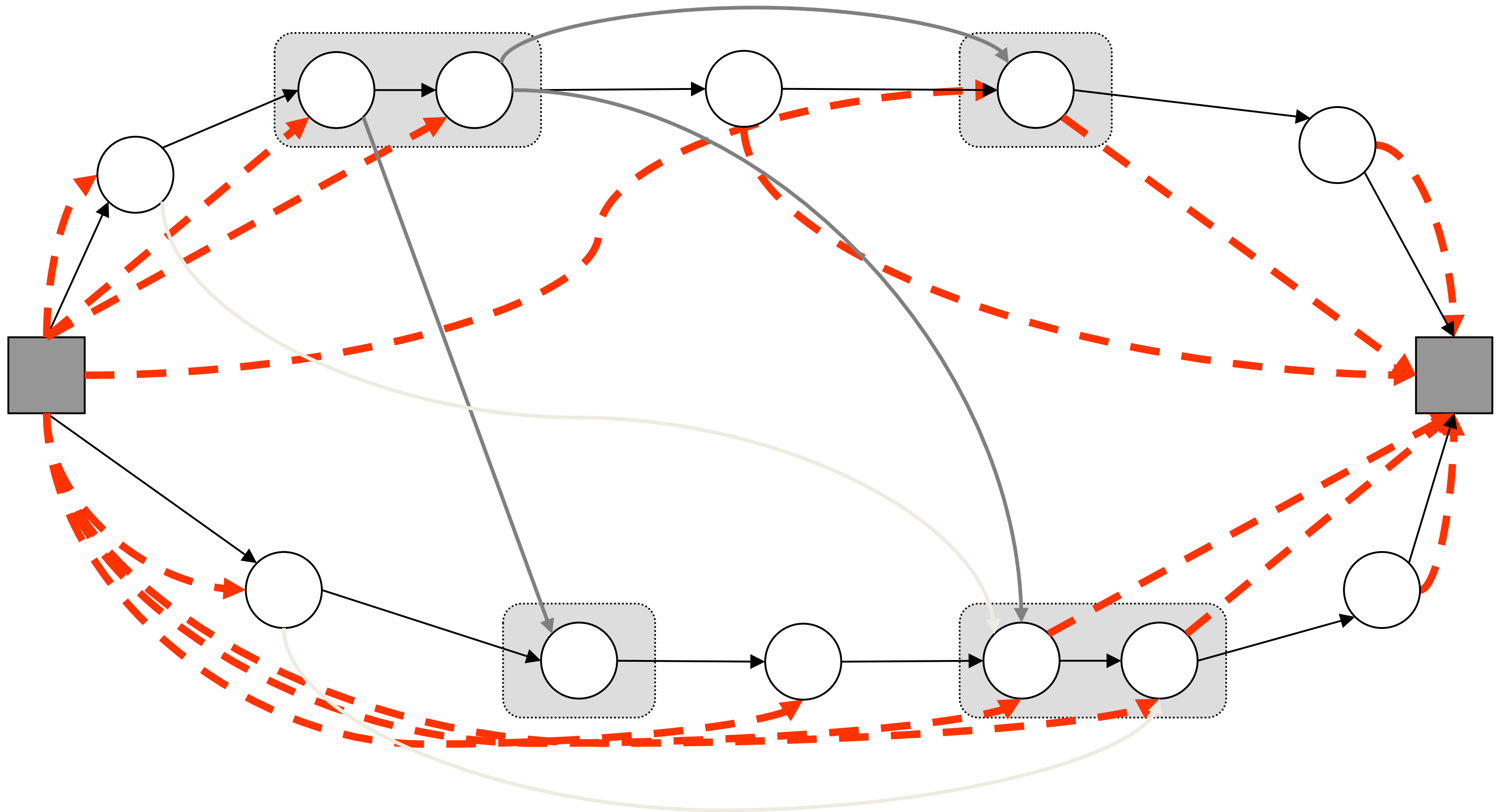
task of a duty

# Graph Theoretic Model

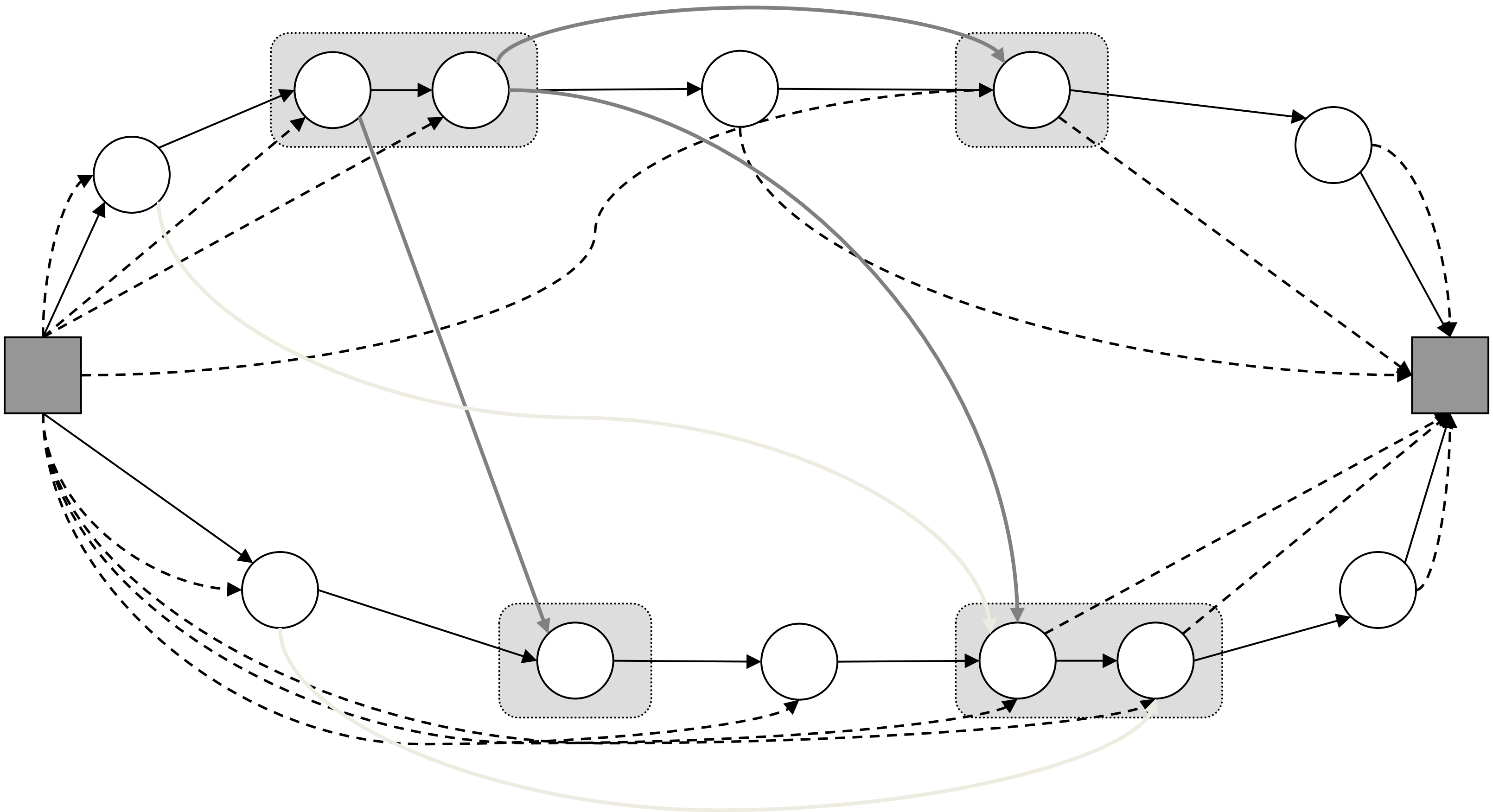




# Graph Theoretic Model

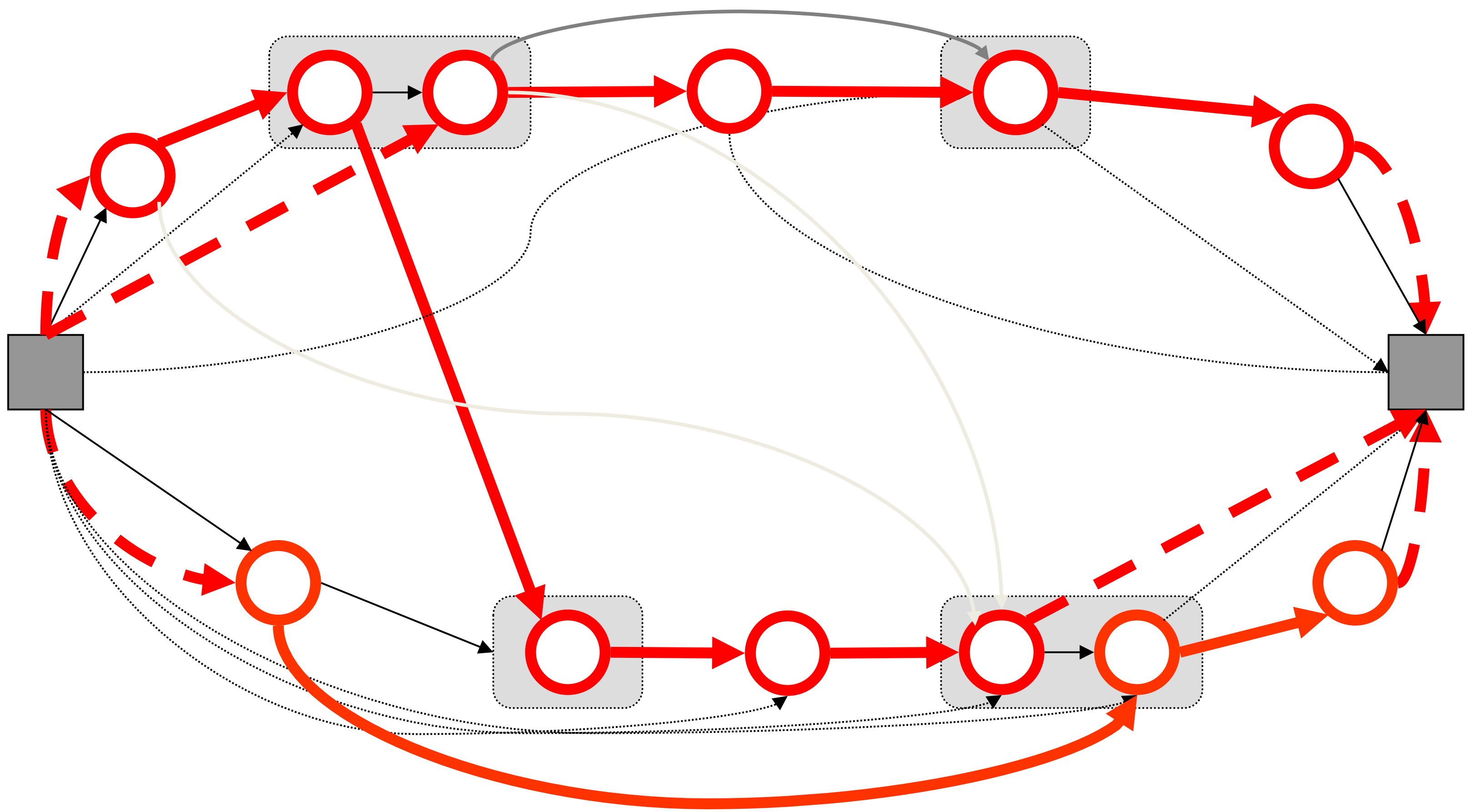


start or end of a duty



# Graph Theoretic Model





# Path Covering Problems

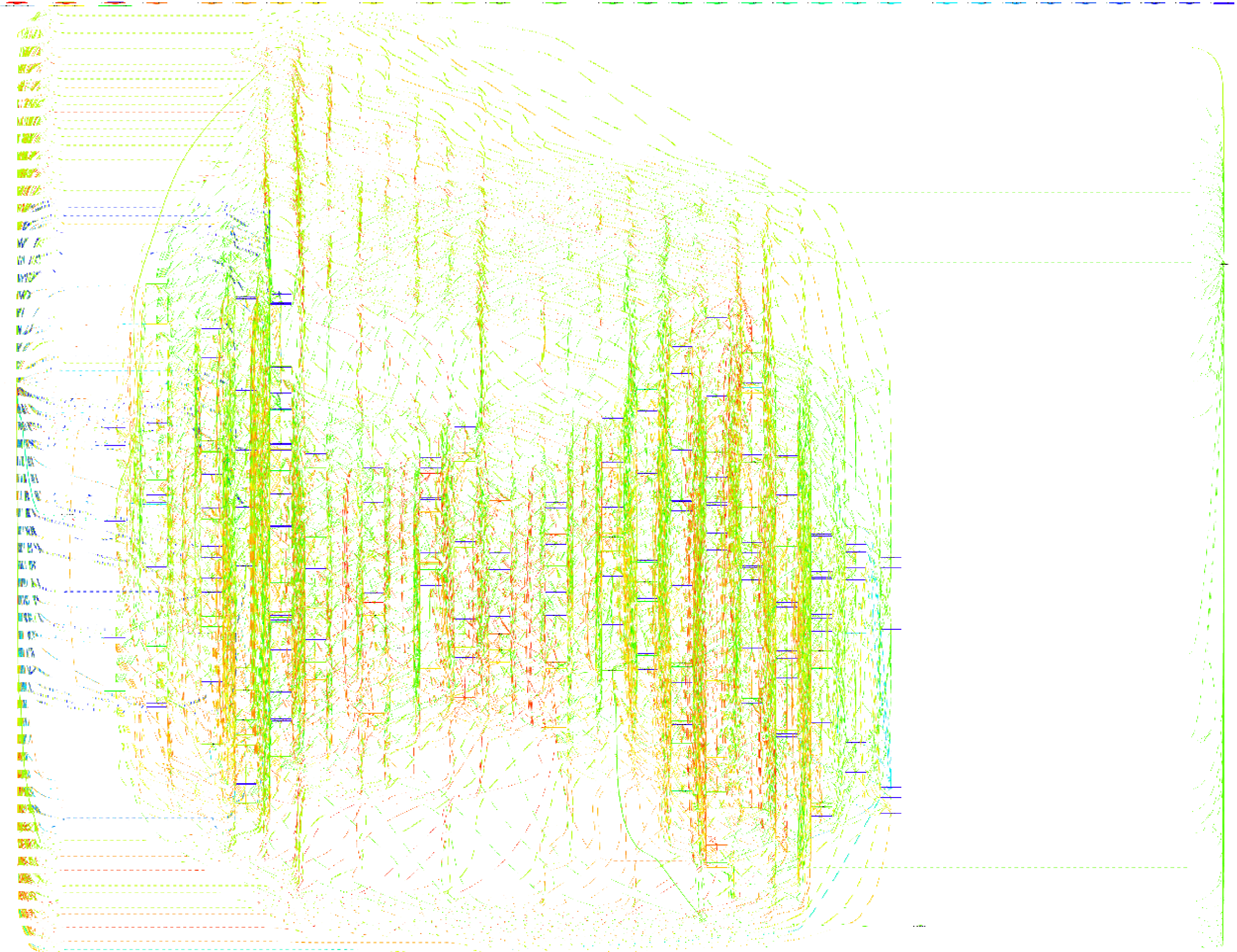


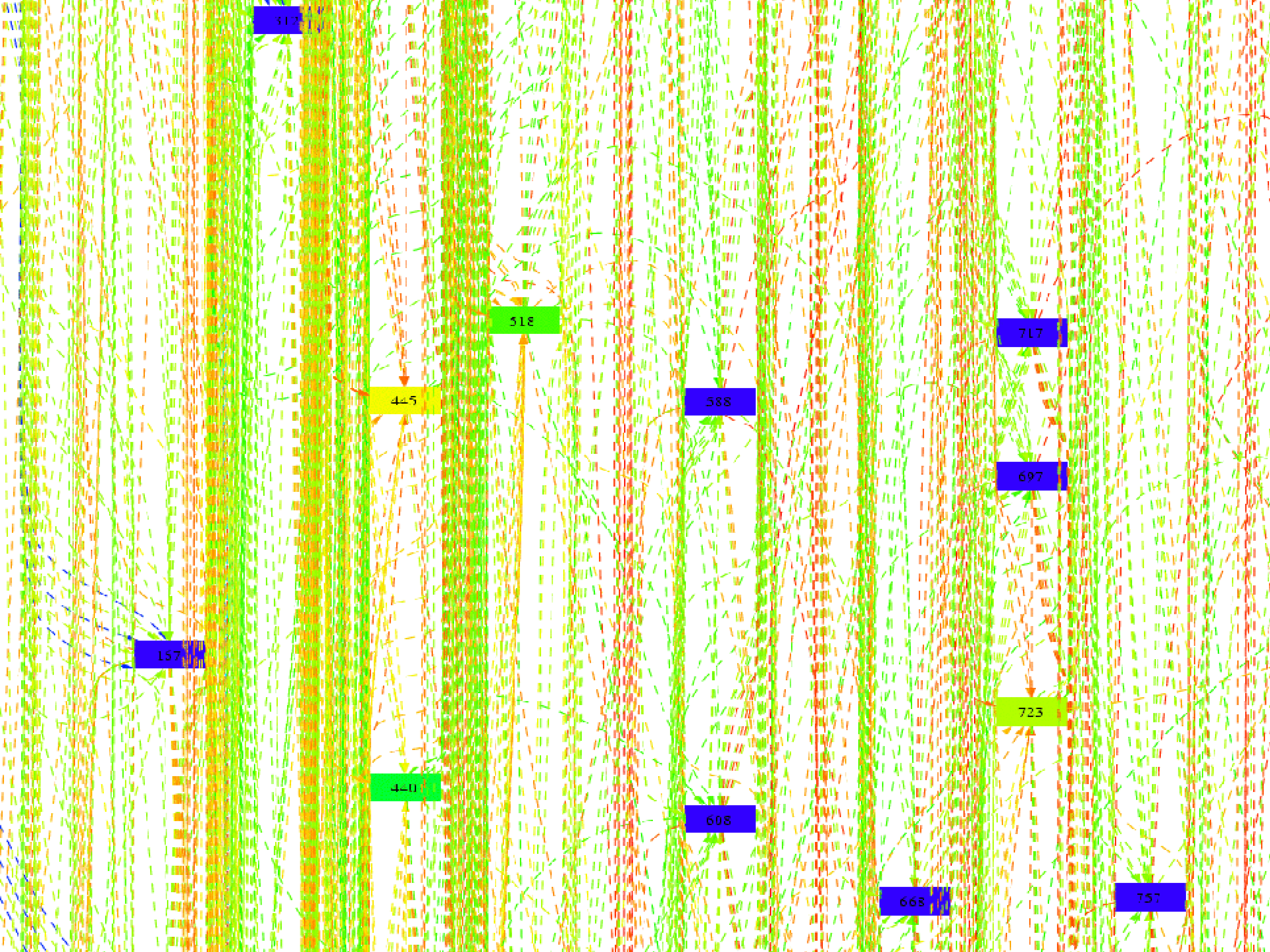
**Definition:** *Path Covering Problem* (PCP)

- Input: Acyclic Digraph D=(V,A) with source s, sink t, integer lengths $z_{ij}$ and costs $c_{ij}$ for all arcs ij and an integer bound L.
- Output: A set $\{P_1,\ldots,P_k\}$ of (s,t)-paths of length ≤ L, such that every node except s and t is contained in exactly one path and the such that total cost of the paths is minimal.

**Observation:** Crew scheduling problems are path covering problems (with additional constraints for individual paths and for the path mix).

**Proposition:** Path covering problems are NP-hard (already for one length restricted path).

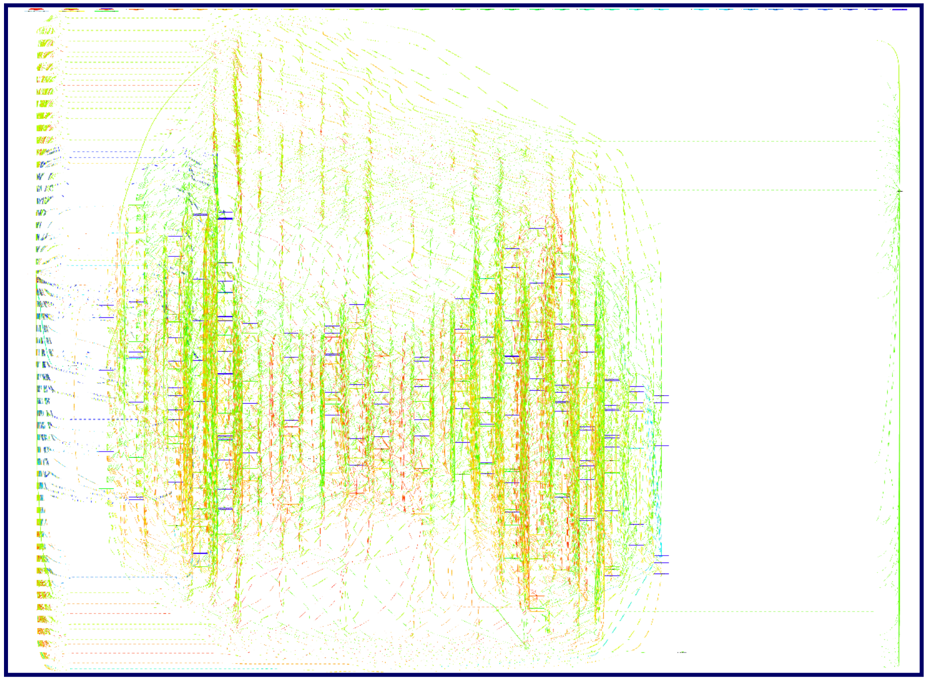




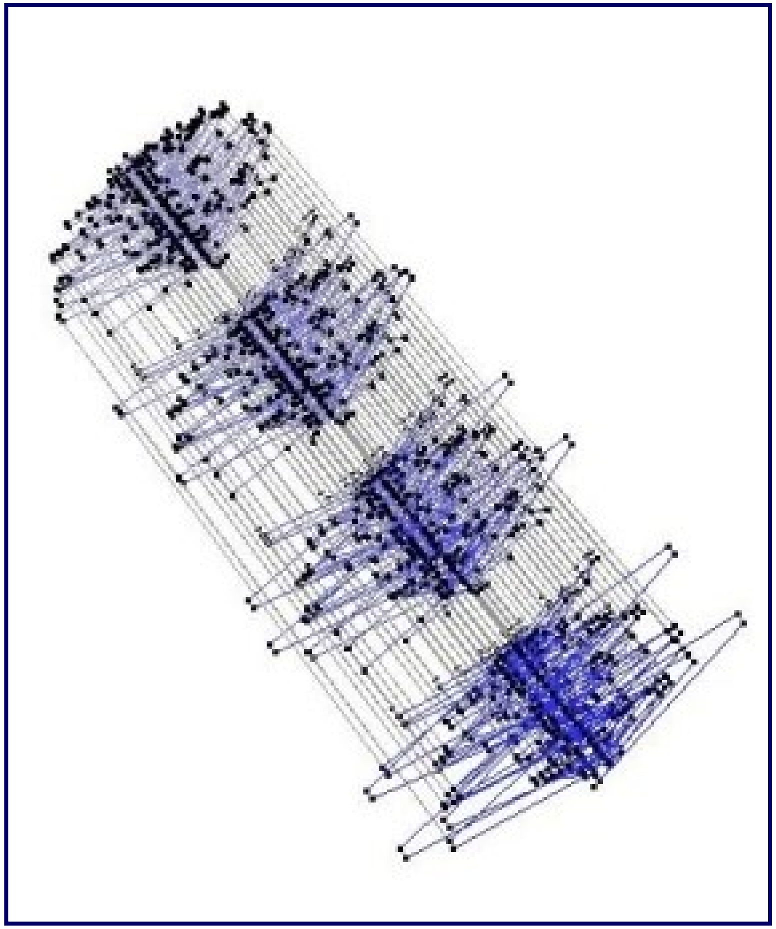
518
717
445
588
697
157
723
440
608
668
757

# Scheduling Graphs





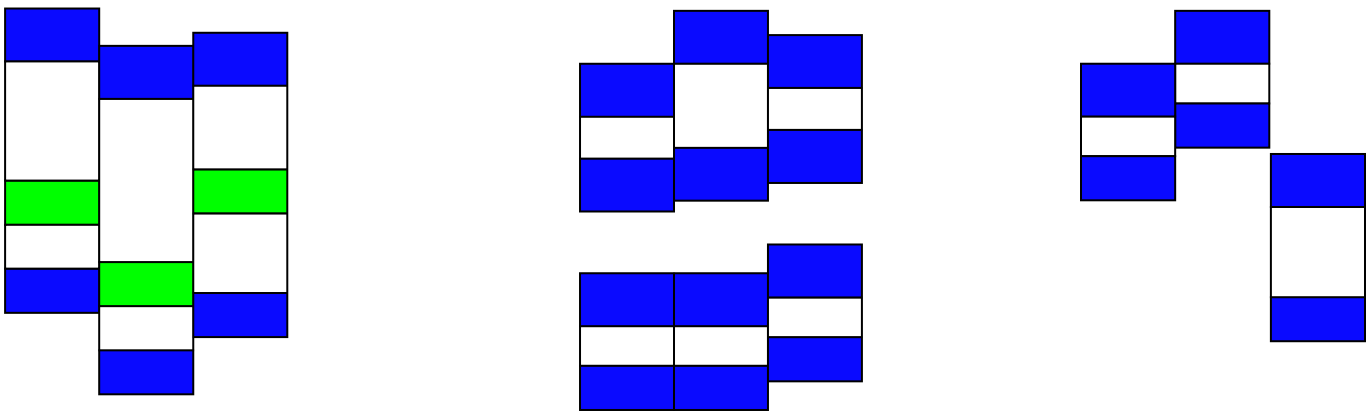
Duty Scheduling in Public Transit

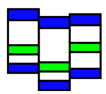
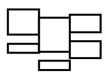
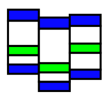
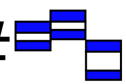


Airline Crew Scheduling

# Duty Mix Rules



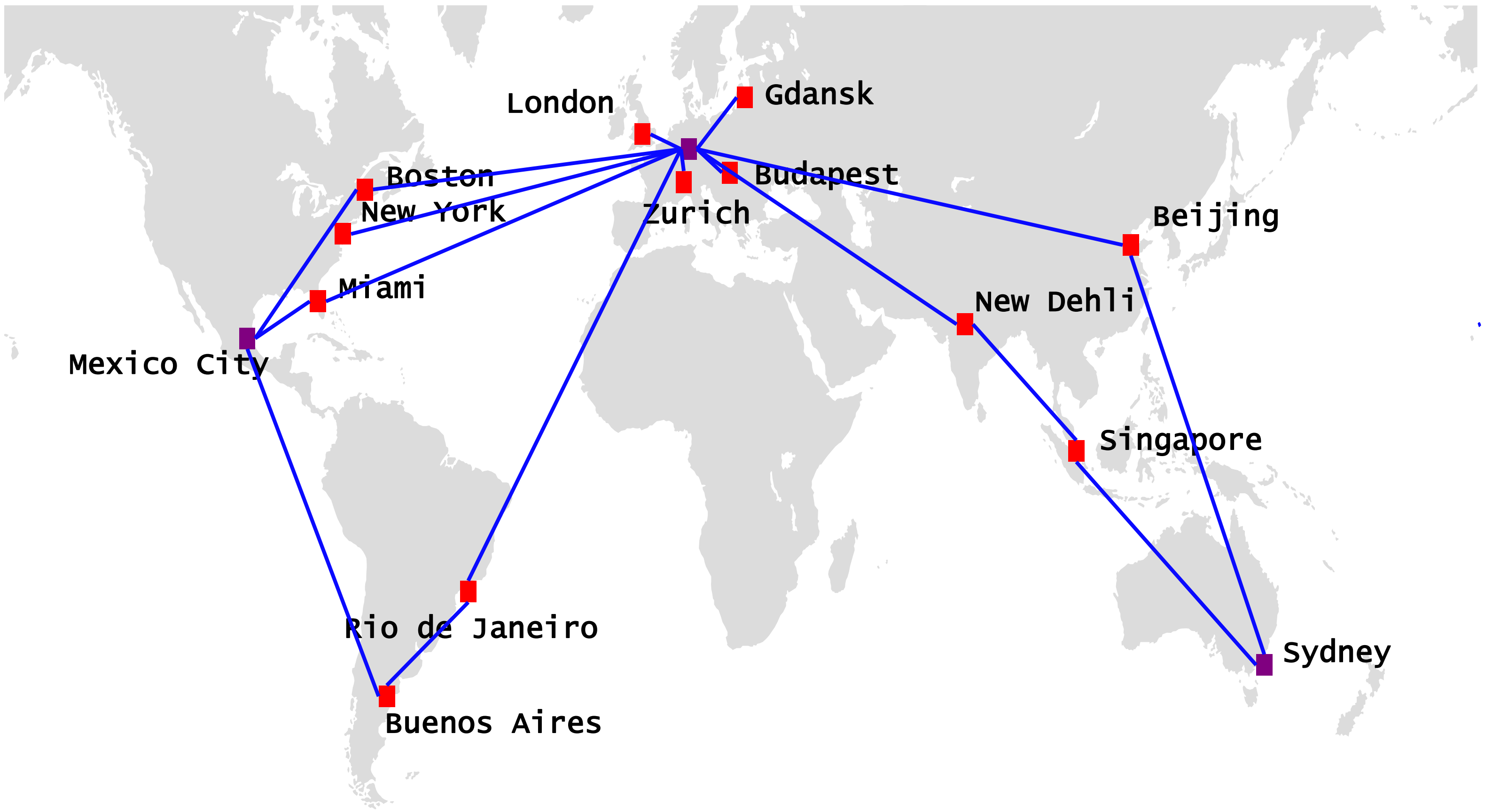


| | | | |
|---|---|---|---|
| Min | #  | $\geq 3$ | (Ex.: min. no.) |
| Opt | Σ  | /3 = 5:00 | (Ex.: average) |
| Max | #  | $\leq$ #  | (Ex.: duty mix) |

- Per type, depot, geo area, time period, ...
- Legality and cost (per type)
  - Linear rules
  - Capacities, averages, duty mix
  - Automatic penalty calculation

# Base Constraints






- Total no. of pairings starting from a base
- Total no. of pairings starting from a base at each day
- Etc.

# Integer Programming Model

(Set Partitioning Problem with Base Constraints)



$$\min \sum_{d} c_d x_d$$

$$\sum_{t \in d} x_d = 1 \quad \forall\, t \quad \text{Tasks}$$

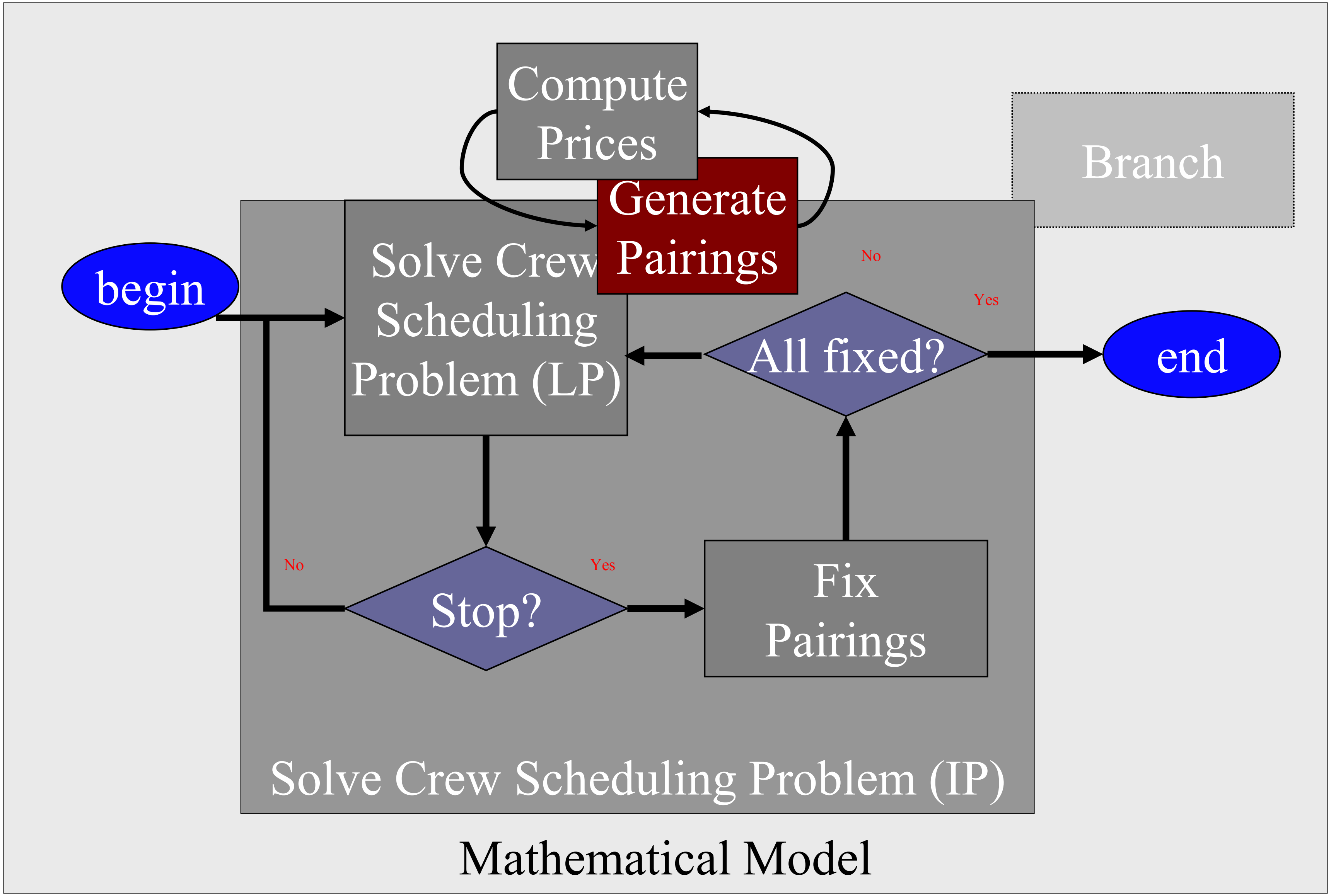
$$\sum_{d \in m} x_d \leq \kappa_m \quad \forall\, m \quad \text{Mix}$$

$$x_d \in \{0,1\} \quad \forall\, d \quad \text{Integrality}$$

# Column Generation Method

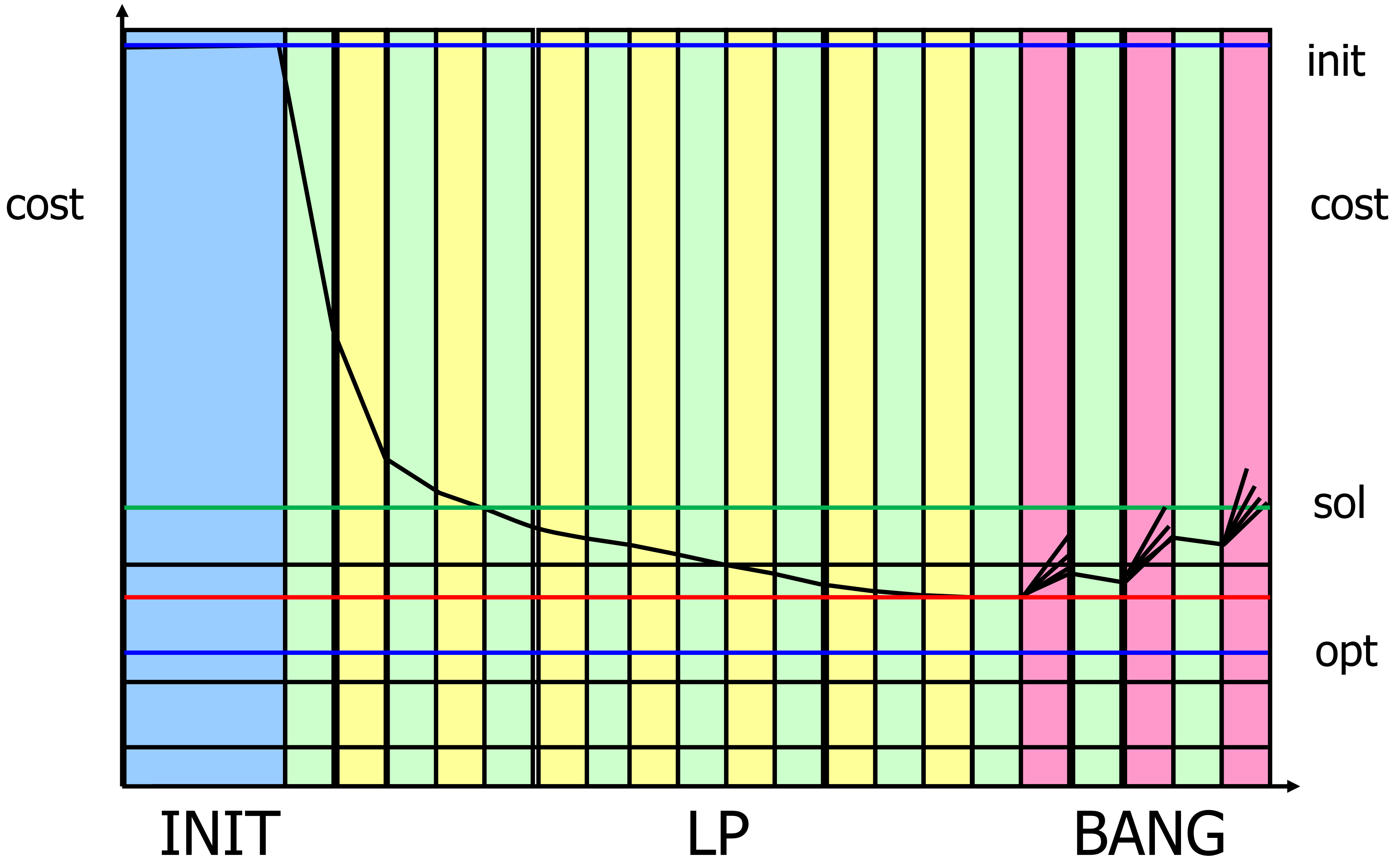
(Branch-and-Generate, Marsten 1994)





# Branch-&-Generate Algorithmus
**(Marsten [1994])**

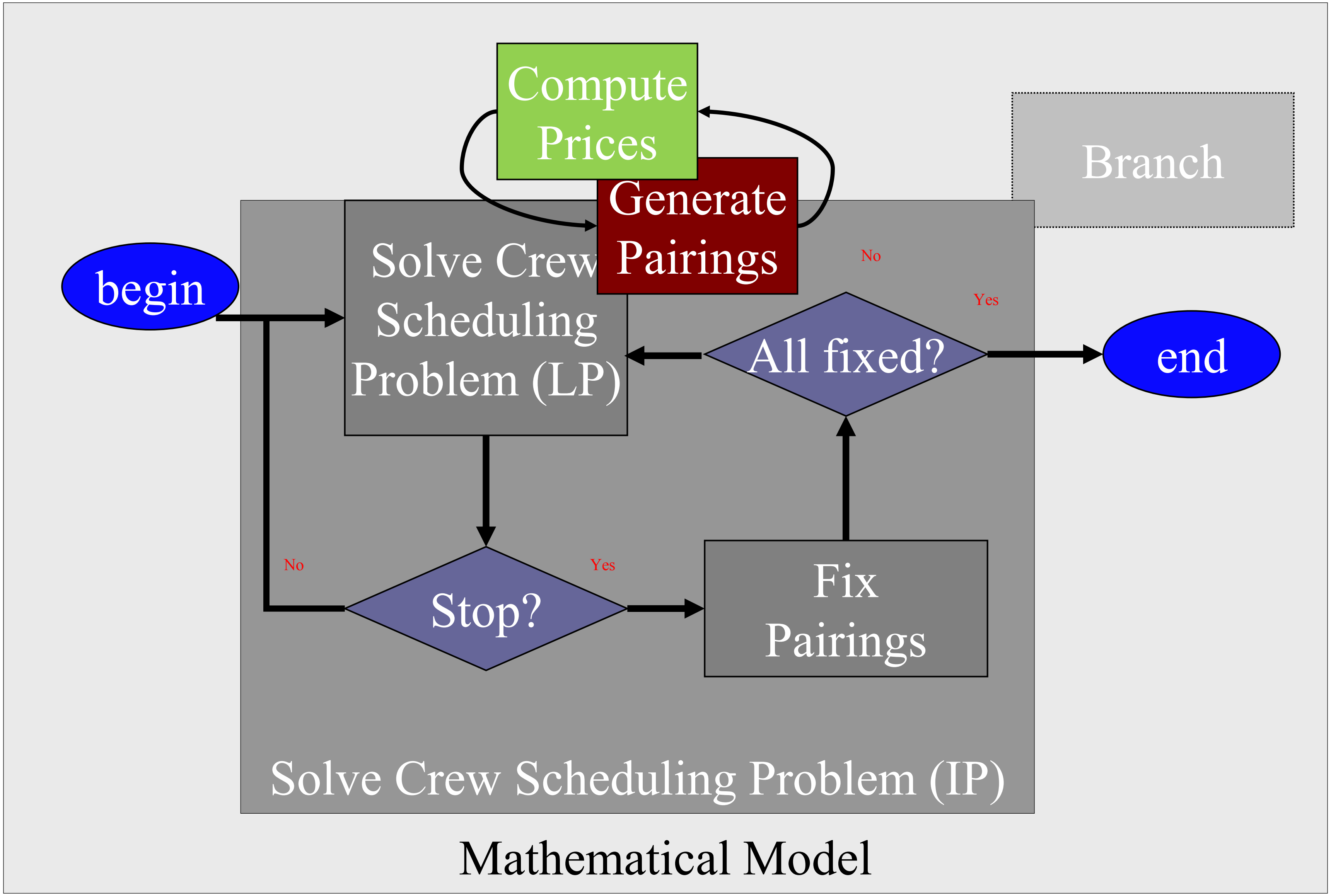




# Column Generation Method

(Branch-and-Generate, Marsten 1994)





# Master LP Lagrange Relaxation



$$\min \sum_{d} c_d x_d, \sum_{d \ni t} x_d = 1 \ \forall t, \sum_{d \in m} x_d \leq \kappa_m \forall\, m, x \geq 0$$

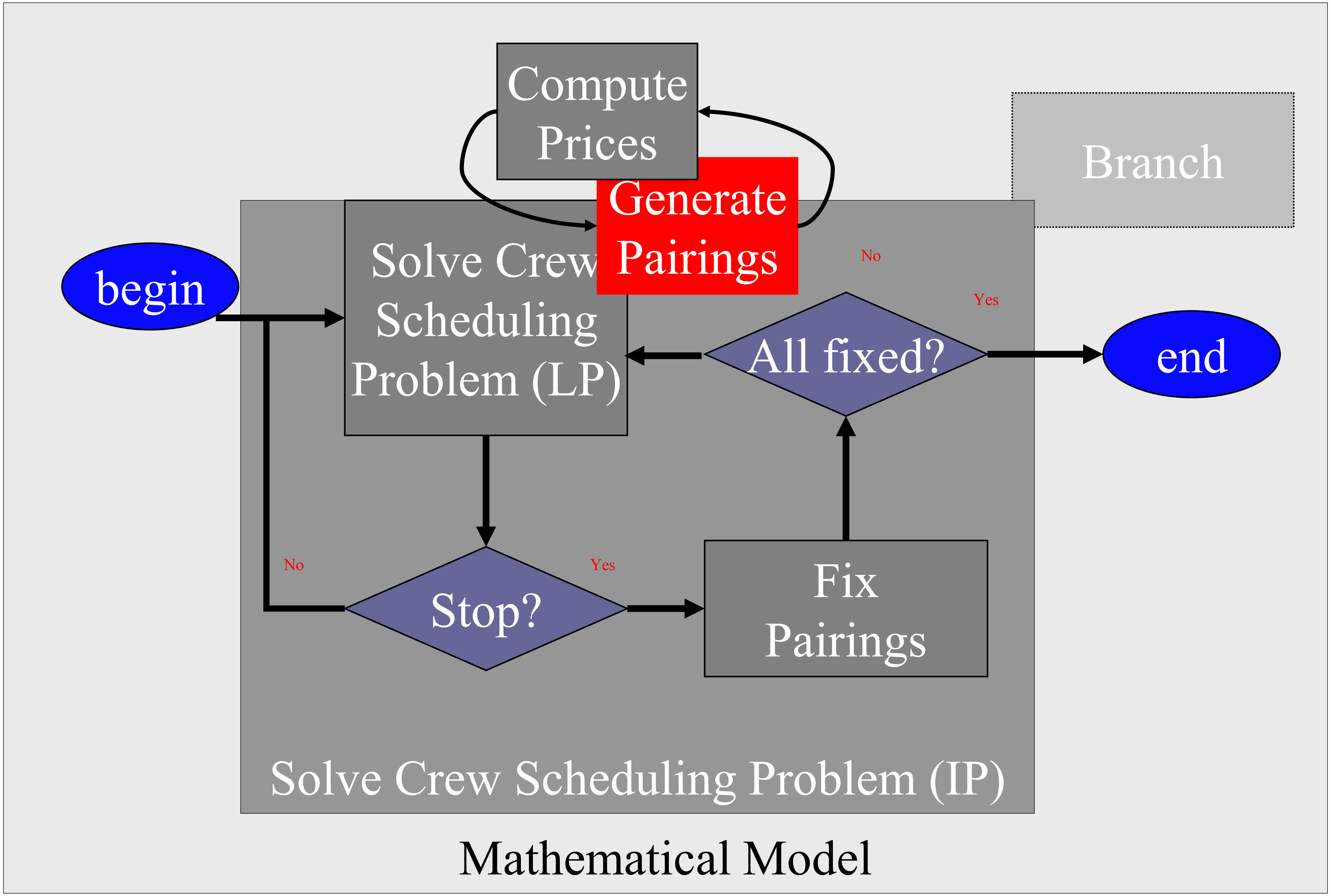
$$\Leftrightarrow \min \sum_{d} c_d x_d, \sum_{d \ni t} x_d = 1 \ \forall t, \sum_{d \in m} x_d \leq \kappa_m \forall\, m, 0 \leq x \leq 1$$

$$\Leftrightarrow \max_{\lambda,\mu \geq 0} \min \sum_{d} c_d x_d - \lambda_t \left( \sum_{d \ni t} x_d - 1 \right) + \mu_m \left( \sum_{d \in m} x_d - \kappa_m \right), 0 \leq x \leq 1$$

# Column Generation Method

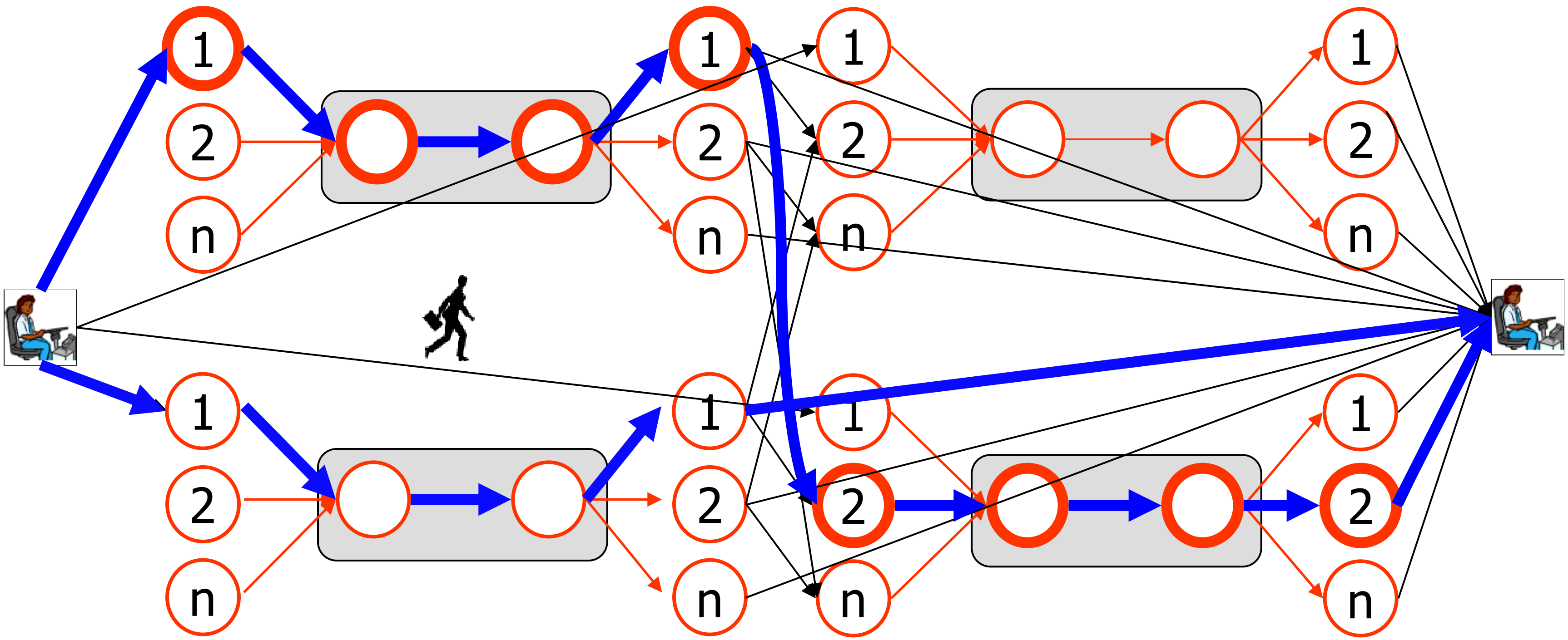
(Branch-and-Generate, Marsten 1994)





# Graph Theoretic Model

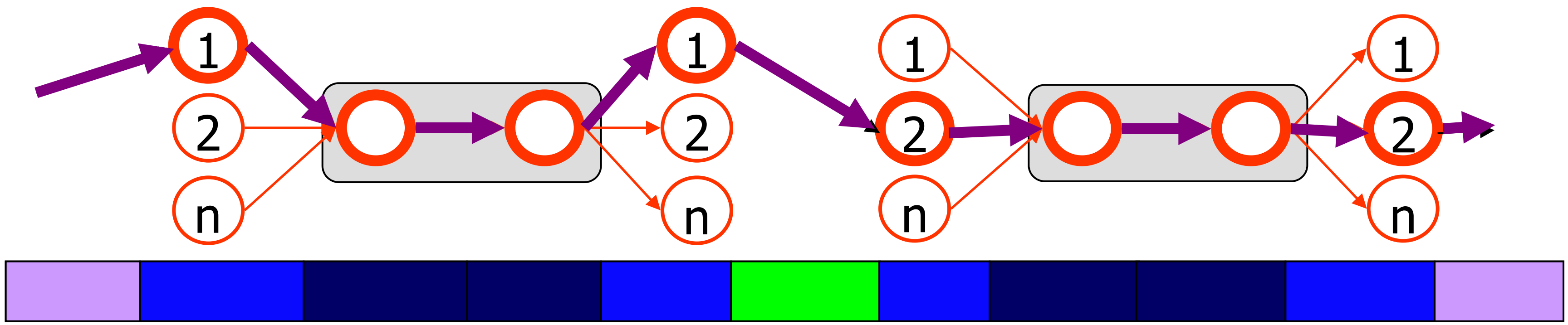


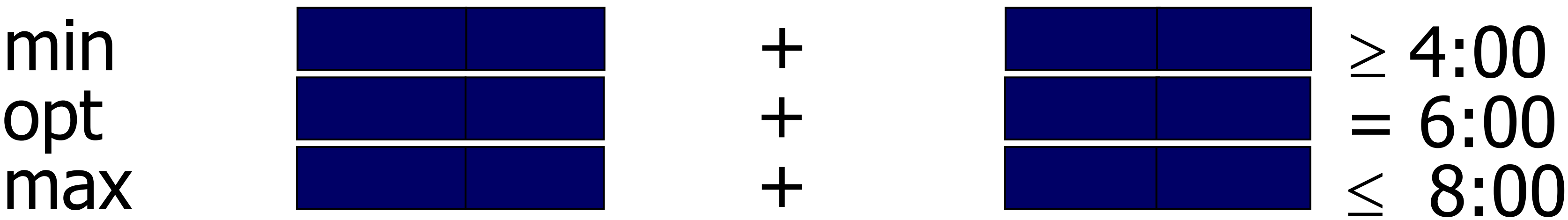



- Extension elements (check in/out, position, ...)
- Links (deadheads, transports, overnights, ...)

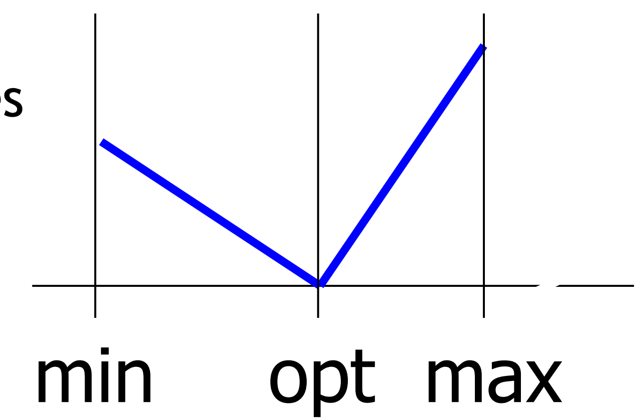
# Duty Construction Rules (Lengths)






- Feasibility + Costs
  - Linear Rules & Penalties





- Feasibility + Costs
  - Linear Rules & Penalties

# Duty Construction Rules



**REGULATION (EC) No 561/2006 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) No 3821/85 and (EC) No 2135/98 and repealing Council Regulation (EEC) No 3820/85**

**Article 6**

1. The daily driving time shall not exceed nine hours. However, the daily driving time may be extended to at most 10 hours not more than twice during the week.
2. The weekly driving time shall not exceed 56 hours and shall not result in the maximum weekly working time laid down in Directive 2002/15/EC being exceeded.
3. The total accumulated driving time during any two consecutive weeks shall not exceed 90 hours.
4. Daily and weekly driving times shall include all driving time on the territory of the Community or of a third country.
5. A driver shall record as other work any time spent as described in Article 4(e) as well as any time spent driving a vehicle used for commercial operations not falling within the scope of this Regulation, and shall record any periods of availability, as defined in Article 15(3)(c) of Regulation (EEC) No 3821/85, since his last daily or weekly rest period. This record shall be entered either manually on a record sheet, a printout or by use of manual input facilities on recording equipment.

**Article 7**

After a driving period of four and a half hours a driver shall take an uninterrupted break of not less than 45 minutes, unless he takes a rest period.

This break may be replaced by a break of at least 15 minutes followed by a break of at least 30 minutes each distributed over the period in such a way as to comply with the provisions of the first paragraph.

**Second Execution Decree for the Regulation of Aircraft Operations (2. DV LuftBO)**

§ 8 Flight duty times of crew members

(1) The unrestricted flight duty time of any crew member between two rest periods is 10 hours. It is feasible to extend the flight duty times according to sentence 1 up to four hours and up to four times between 7 consecutive days, if the sum of the extensions does not exceed 8 hours within 7 consecutive days. The time period of 7 consecutive days starts at any one time at 00:00 Standard Greenwich Time (GMT) of the first day and ends at 24:00 GMT of the seventh day. A crew member, who requires full time or part time assistance by another crew member in his function as a pilot during the flight duty time according to sentence 1, is not subject to paragraphs 2 and 3.

(2) The maximum flight duty time extension of fours hours according to paragraph 1 is diminished 1. by 1 hour, if the flight duty time is longer than 2, but less than 4 hours, 2. by 2 hours, if the flight duty time contains 4 hours or more between 01:00 and 07:00 local time of the departure airport (winter time).

(3) A flight duty time that is diminished according to paragraph 2 is to be further diminished 1. by 1 more hour for more than 3, but less than 6 landings, 2. by 2 more hours for more than 5 landings.

(4) If the minimum crew size is extended and if suitable sleeping accommodation is provided in a space separate from the cockpit and the cabin, the regulation authority can extend the flight duty time according to paragraph 1, sentence 1 up to two times by 8 hours within 7 consecutive days upon written request. The time that any crew member spends piloting and operating the aircraft must never exceed 12 hours. The cabin crew must be provided with suitable breaks during the flight. To this purpose, rest seats must be provided. In any way, § 12 applies accordingly.

(5) The flight duty times must not exceed 210 hours within 30 consecutive days, and 1800 hours within one calendar year.

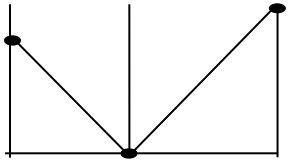
# Linear Resource Constraints

- Max constraints: $w_r^T x \le b_r$
  duty time, working time, driving time

- Min constraints: $w_r^T x \ge b_r \quad (w_r^T x \ge q w_r^T x)$
  dito, quotient break rule relaxations

- Opt constraints: $w_r^T x + s_r^+ - s_r^- = b_r \quad (\min d_r^T s_r)$
  duty time, working time, driving time

- Path Elimination constraints: $\chi_P^T x \le \chi_P^T 1 - 1$
  all other rules



# Constrained Shortest Path Problem

**Definition:** *Length-constrained acyclic shortest path problem* (CSP)

- Input: Acyclic Digraph D=(V,A) with source s, sink t, integer lengths $z_{ij}$ and costs $c_{ij}$ for all arcs ij and an integer bound L.
- Output: An (s,t)-path with length ≤ L with minimal, possibly negative, costs.

**Observation:** The problem to construct an improving pairing is a length-constrained acyclic shortest path problem.

**Theorem (Desrochers):** The CSP is NP-hard.

**Theorem:** The CSP can be solved in pseudo-polynomial time by dynamic programming.

**Theorem (Warburton, Hassin):** There are FPAS for the CSP.

# Duty Generation by Acyclic Constrained Shortest Paths

$$
\begin{aligned}
\text{(ACSP)}\ \min\ \overline{c}^{T} x & \\
Nx & = e_0 - e_{n+1} \\
Wx + I^{\pm} s & = b \\
0 \le s & \le u \\
x & \in \{0,1\}^{A}
\end{aligned}
$$

- NP hard for |R|=1 (Meggido [1977])
- FPAS for |R| fixed (Warburton [1987] etc.)
- LP in O(log(nL)) SP computations for |R|=1 (Mehlhorn, Ziegelmann [2000])

# Lagrangean Shortest Path Relaxation



$$\max_{y} \min L(x,y) = \bar{c}^T x - y^T W x + d^T s - y^T I^{\pm} s + y^T b$$

$$\begin{aligned} Nx &= e_0 - e_{n+1} \\ s &\leq u \\ x, s &\geq 0 \end{aligned}$$

L(x,y)

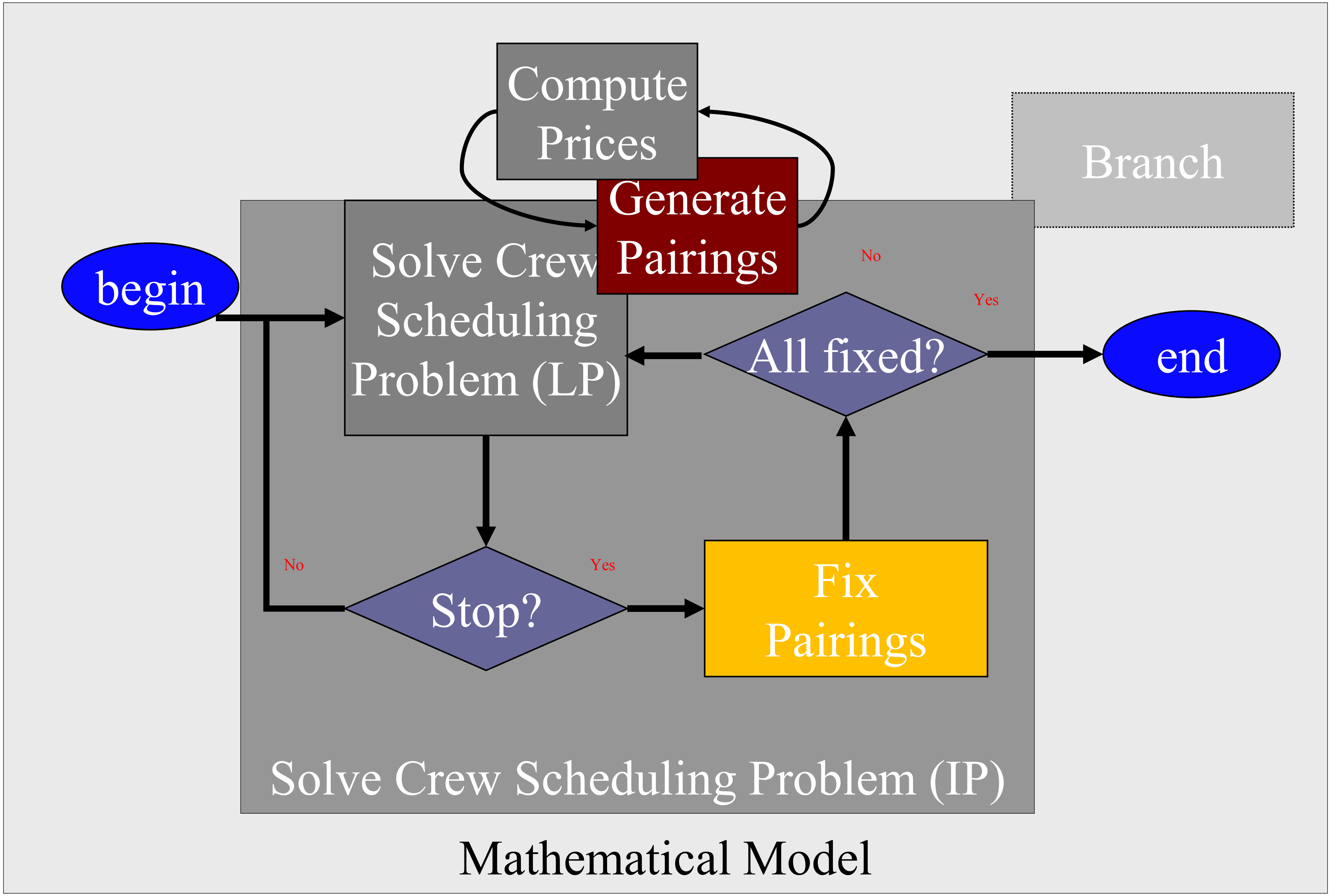
- Rexation of Linear Constraints
- Lower-Bound-Tree for any multipliers y
- Computation by Bundle Method

Lower Bound in Dynamic Program/Multilabel Shortest Path Algorithm

# Column Generation Method

(Branch-and-Generate, Marsten 1994)





# Perturbation Branching



$\min c^T x \quad Ax \geq b,\ 0 \leq x \leq 1,\ x \in \{0,1\}^n$

Sequence i = 1,2,... of IPs with perturbed objectives

$c_j^{i+1} := c_j^i - \alpha(x_j^i)^2 \qquad$ for all j

Set of fixing candidates in iteration i

$B^i := \{ j : x_j^i \geq 1 - \varepsilon \}$

Potential function in iteration i

$v^i := c^T x^i - w|B^i|$

Set of fixed variables (many)

$B^* := B^{\operatorname{argmin} v^i}$

Go on while not integer and potential decreases, else

- Perturb for $k_{max}$ additional iterations, if still not successful
  - Fix a single variable and reset objective every $k_s$ iterations

# Rapid Branching

$B^* := \{j_1, \dots, j_m\}$, $\bar{c}_{j_1} \leq \dots \leq \bar{c}_{j_m}$

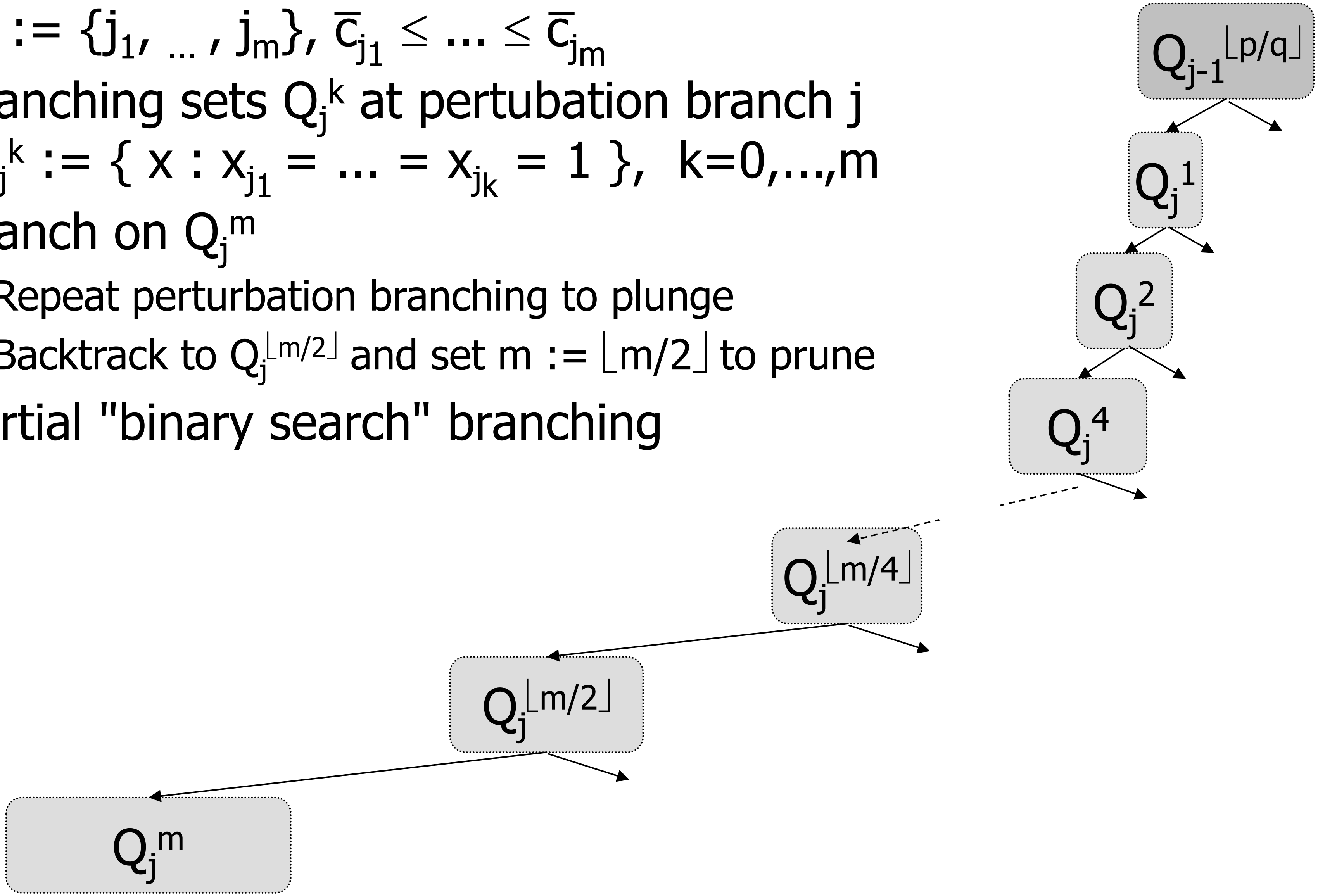
Branching sets $Q_j^k$ at pertubation branch j

$Q_j^k := \{ x : x_{j_1} = \dots = x_{j_k} = 1 \}$, $k=0,\dots,m$

Branch on $Q_j^m$

- Repeat perturbation branching to plunge
- Backtrack to $Q_j^{\lfloor m/2 \rfloor}$ and set $m := \lfloor m/2 \rfloor$ to prune

Partial "binary search" branching

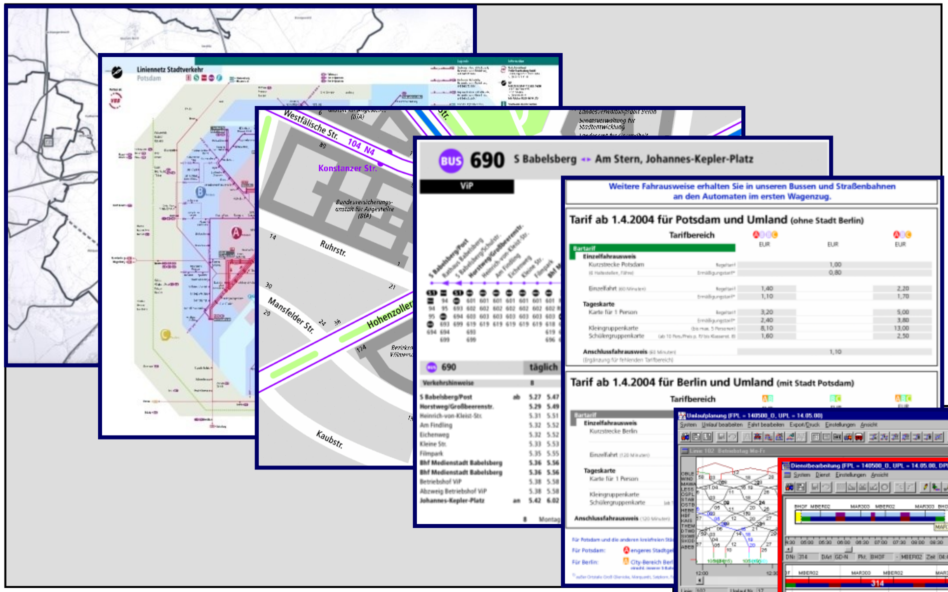


# Rapid Branching & Set Partitioning

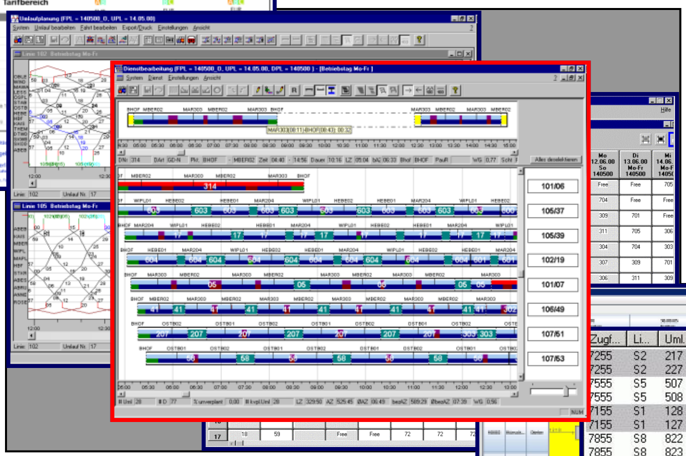
| problem | | *ivu53b* | *ivu60* | *aa01* | *aa04* | *us01* |
|---|---|---|---|---|---|---|
| cols | | 560,233 | 1,148,050 | 8,904 | 7,195 | 1,053,137 |
| rows | | 2,421 | 1,979 | 823 | 426 | 145 |
| bcs | | 6 | 2 | – | – | – |
| nzs | | 12,840,061 | 21,384,769 | 72,965 | 52,121 | 13,636,541 |
| nzs/col | | 22.92 | 18.63 | 8.19 | 7.24 | 12.95 |
| LP | | 261.33 | 159.99 | 55,535.43 | 25,877.61 | 9,962.64 |
| DS | obj | 263.90 | 162.34 | – | – | – |
| | time | 9,098 | 29,982 | – | – | – |
| STATIC | obj | 265.31 | 172.22 | 57,832 | 27,202 | 10,098 |
| | time | 609 | 1,353 | 56 | 42 | 176 |
| CPLEX | obj | 263.57 | – | 56,354 | 26,433 | 10,090 |
| | time | 13,238.34 | > 4d | 128 | 34 | 335 |

# Planning Problems in Public Transit

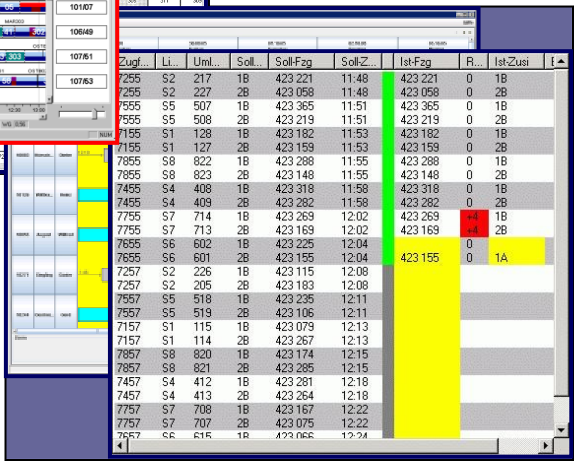





Service Design

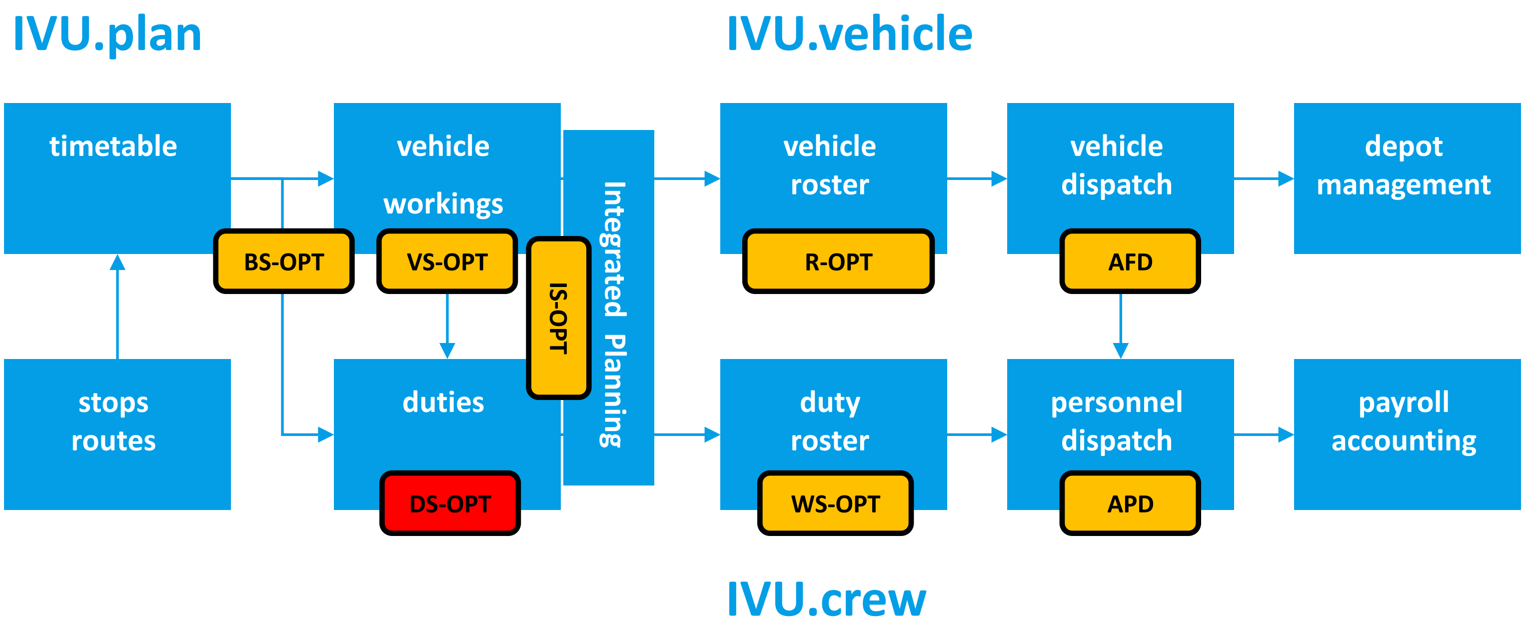


Operational Planning



Operations Control

# IVU.plan / IVU.crew / IVU.vehicle
# System Overview with Optimization Modules



# Thank you for your attention

Ralf Borndörfer

Freie Universität Berlin
Zuse-Institute Berlin
Takustr. 7
14195 Berlin-Dahlem

Fon (+49 30) 84185-243
Fax (+49 30) 84185-269

borndoerfer@zib.de

www.zib.de/borndoerfer