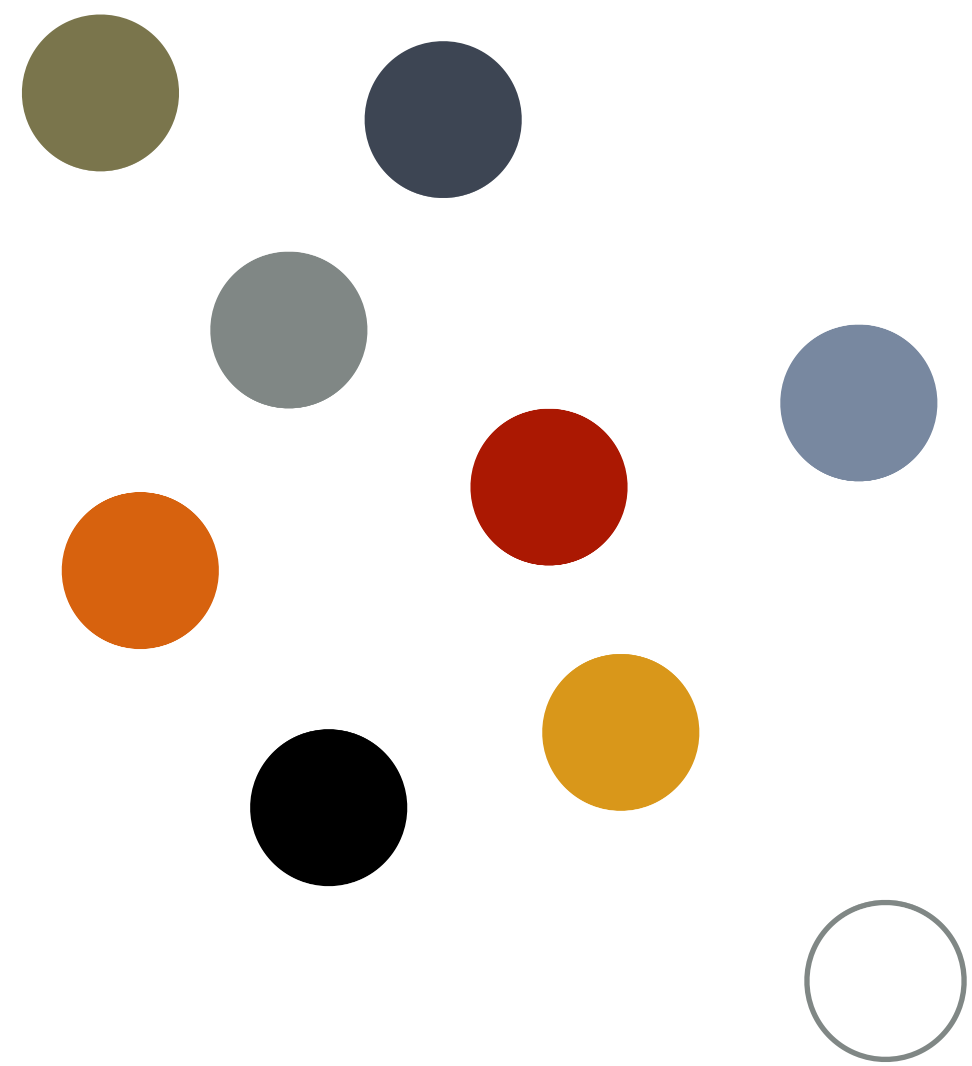
SESSION TWO

Modeling

SELECT THE BALLS

- Suppose we have n balls, if you select any of them the cost associated with that ball should be paid.

- Formulate the total payment:



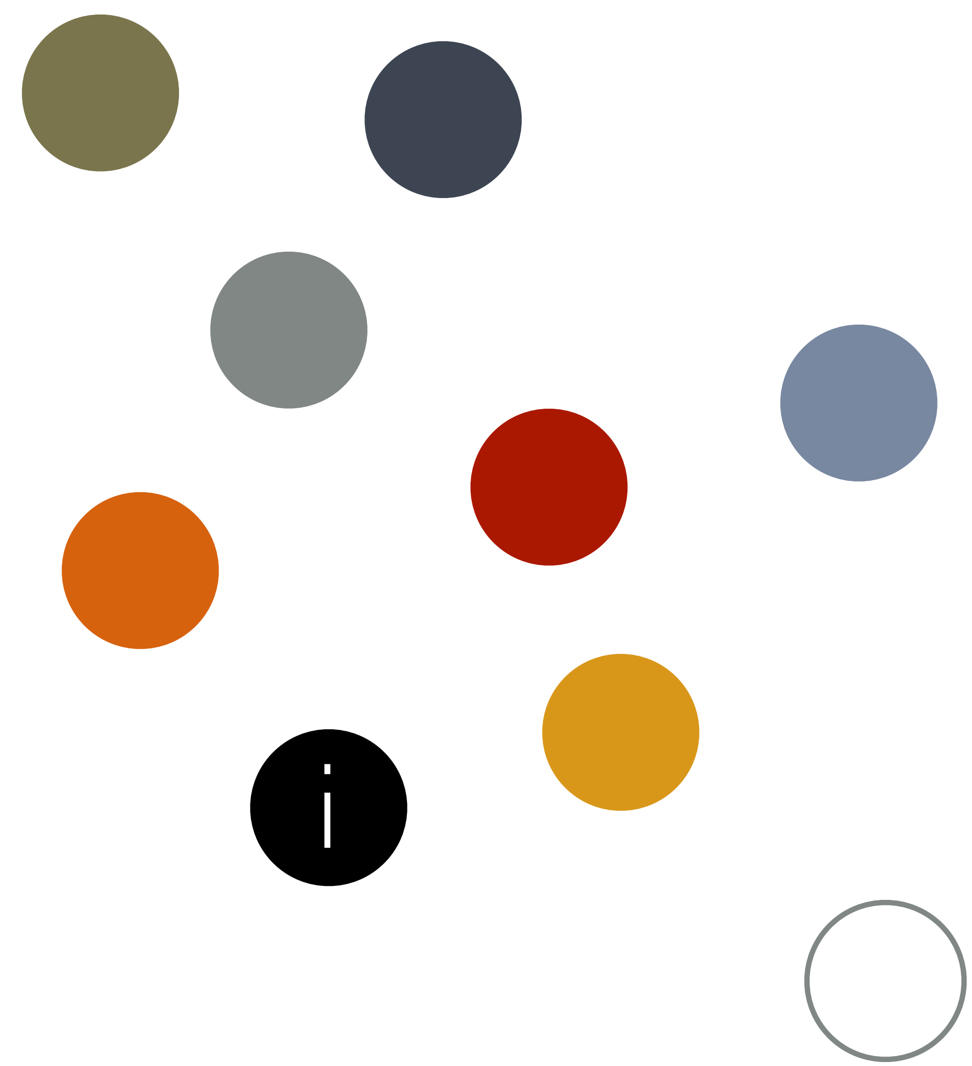
-

SELECT THE BALLS

- Suppose we have n balls, if you select any of them the cost associated with that ball should be paid.

- Formulate the total payment:

- $$Cost = \sum_{i \in N} x_i c_i$$



SELECT THE BALLS

- Select exactly two of them:

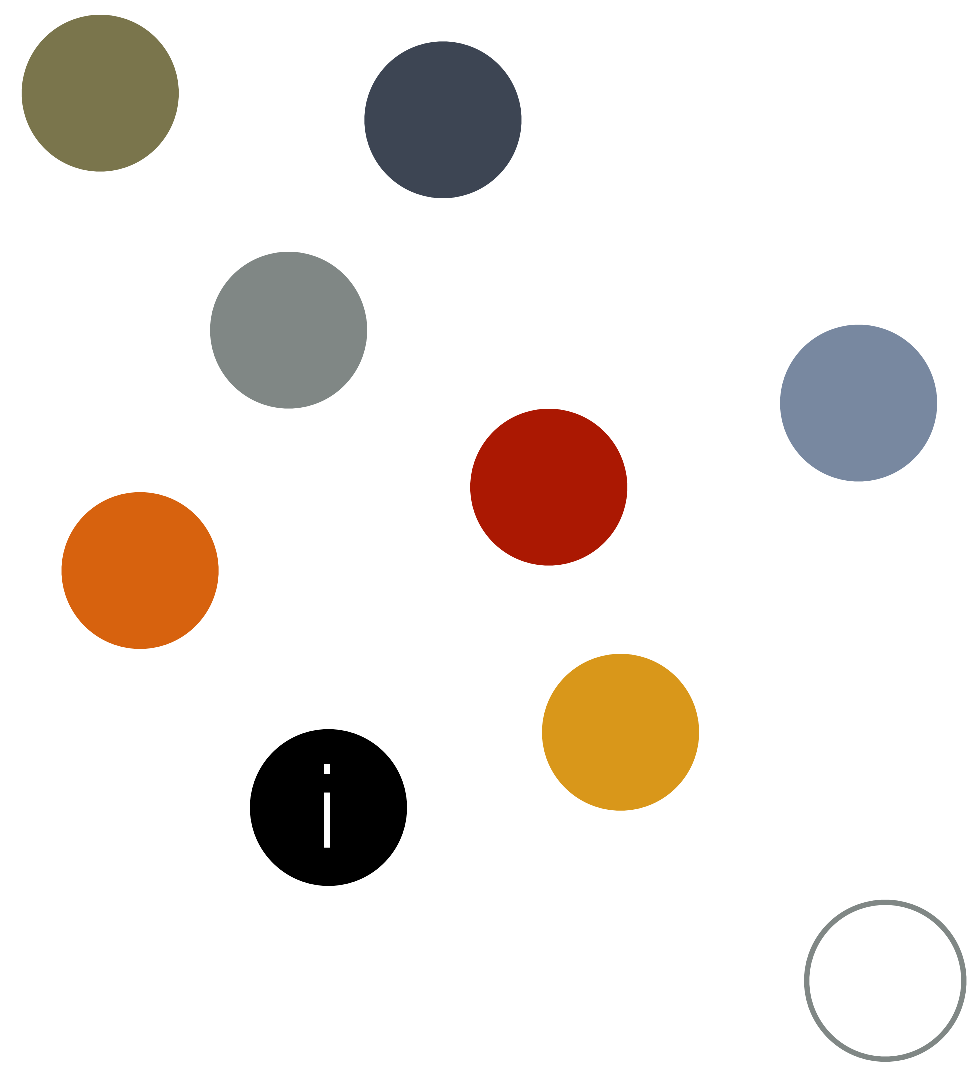
$$\sum_{i \in N} x_i = 2$$

- Select at least two of them:

$$\sum_{i \in N} x_i \geq 2$$

- Select at most two of them:

- $$\sum_{i \in N} x_i \leq 2$$



SELECT THE BALLS

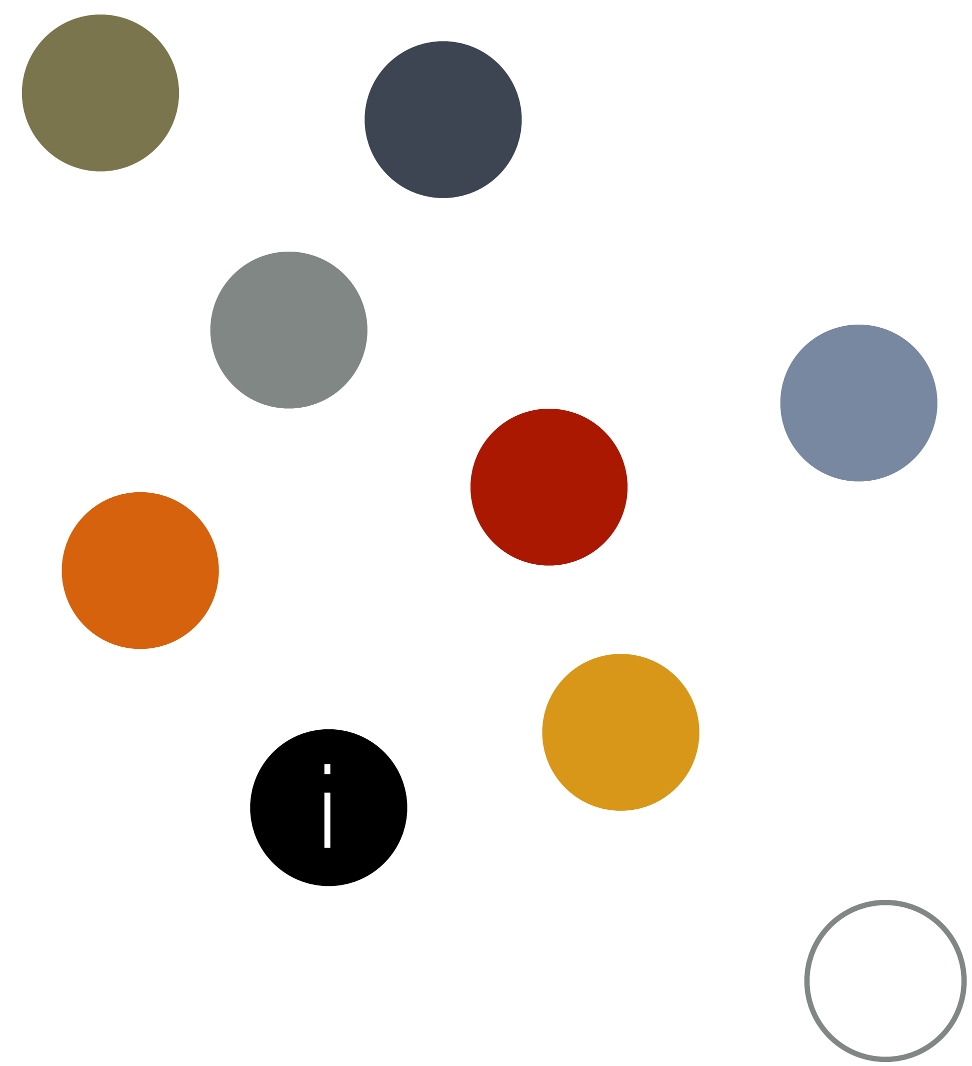
- If ball #3 is selected then ball 4 and ball 5 can not be selected :

$$x_4 \leq 1 - x_3$$
$$x_5 \leq 1 - x_3$$

- If ball #3 is selected then ball 4 and ball 5 should be selected :

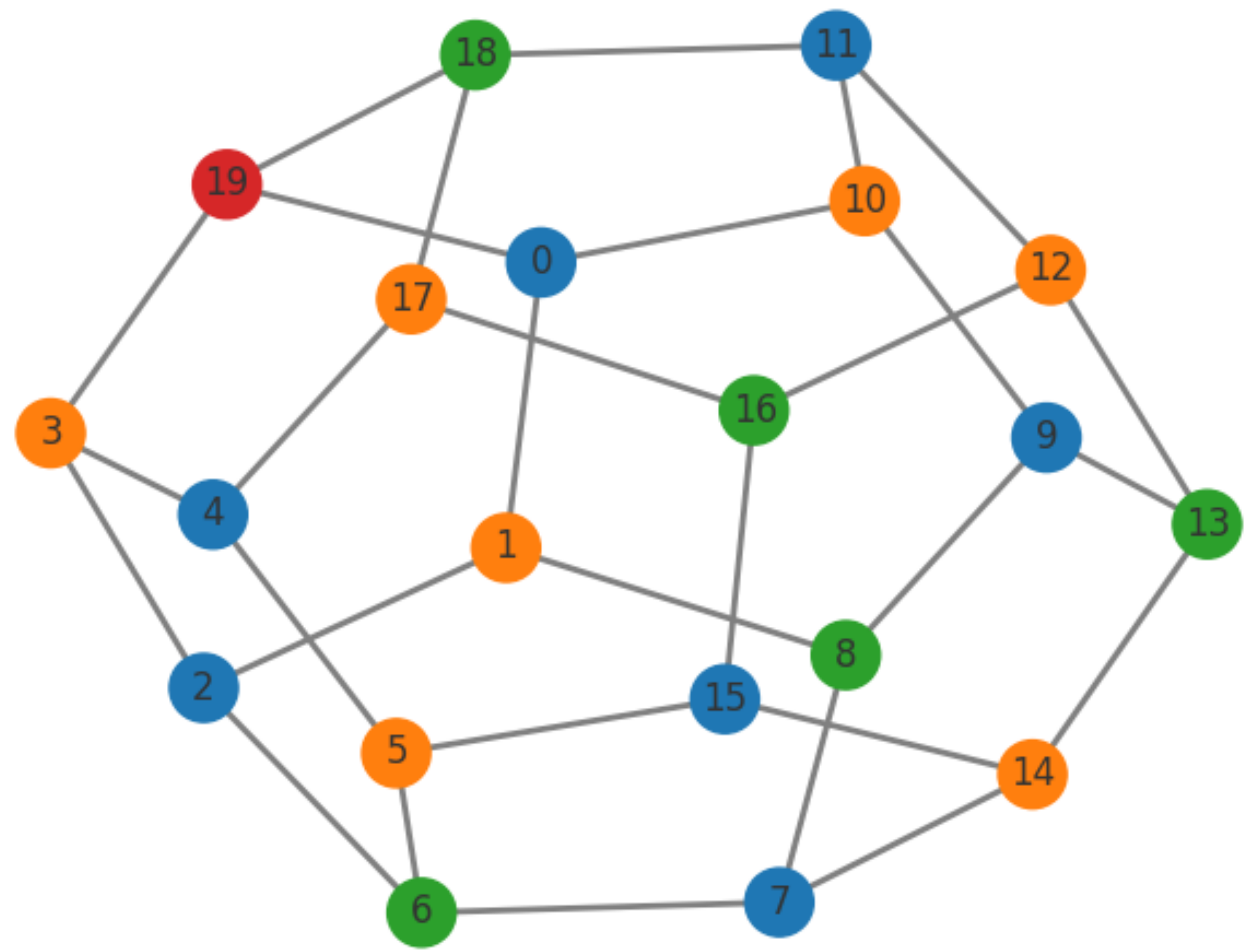
$$x_4 \geq x_3$$
$$x_5 \geq x_3$$

-



SHORTEST PATH

- How to start ?
- What is the input data?
- Set?
- Decision variables?
- Constraints?
- Objective function ?



SHORTEST PATH

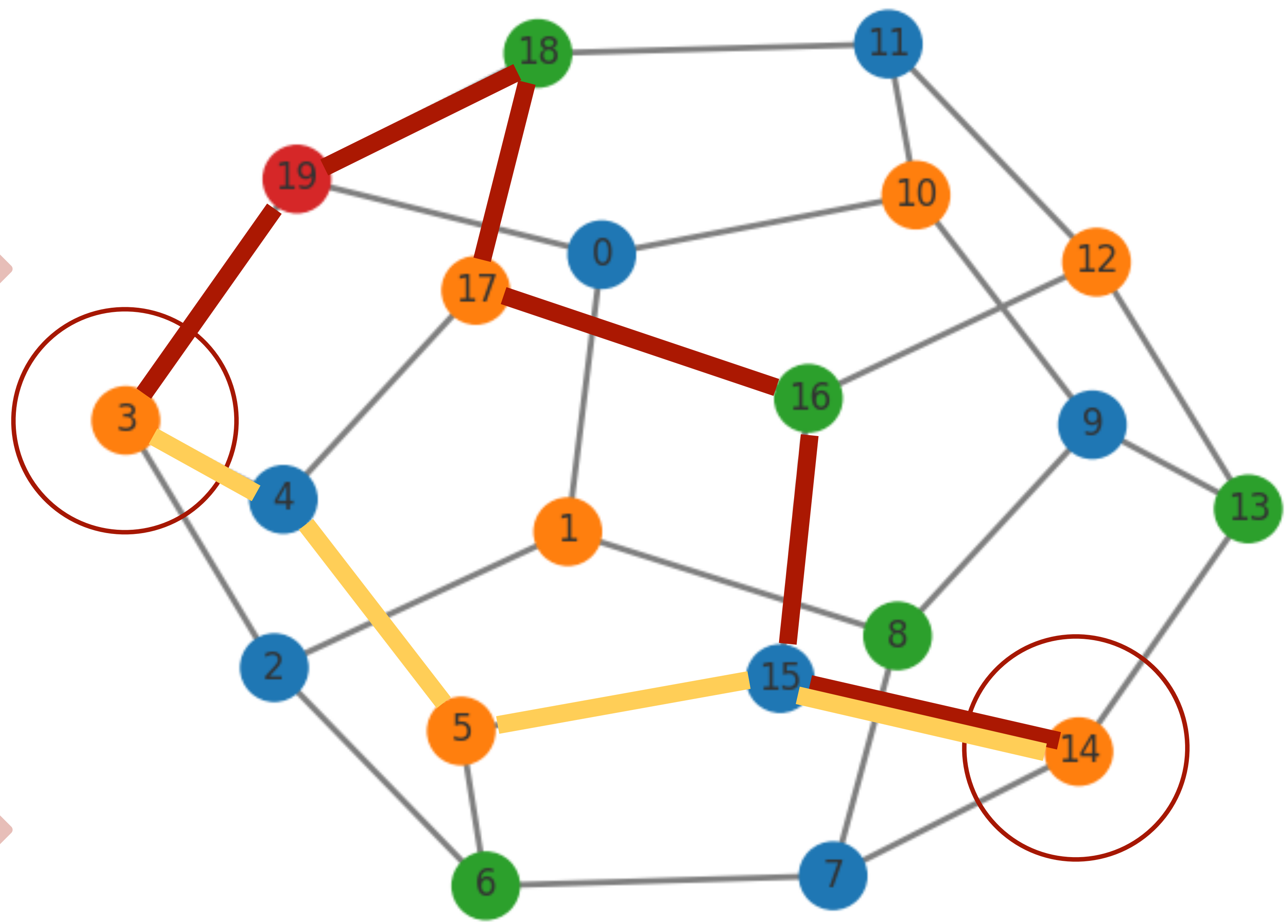
Assume that we have 35 edges

$U_{i,j}$

$$OF = \sum_{i,j} D_{i,j} U_{i,j}$$

$$\forall_i \quad \sum_j U_{i,j} = 1$$

$$\forall_i \quad \sum_j U_{j,i} = 1$$



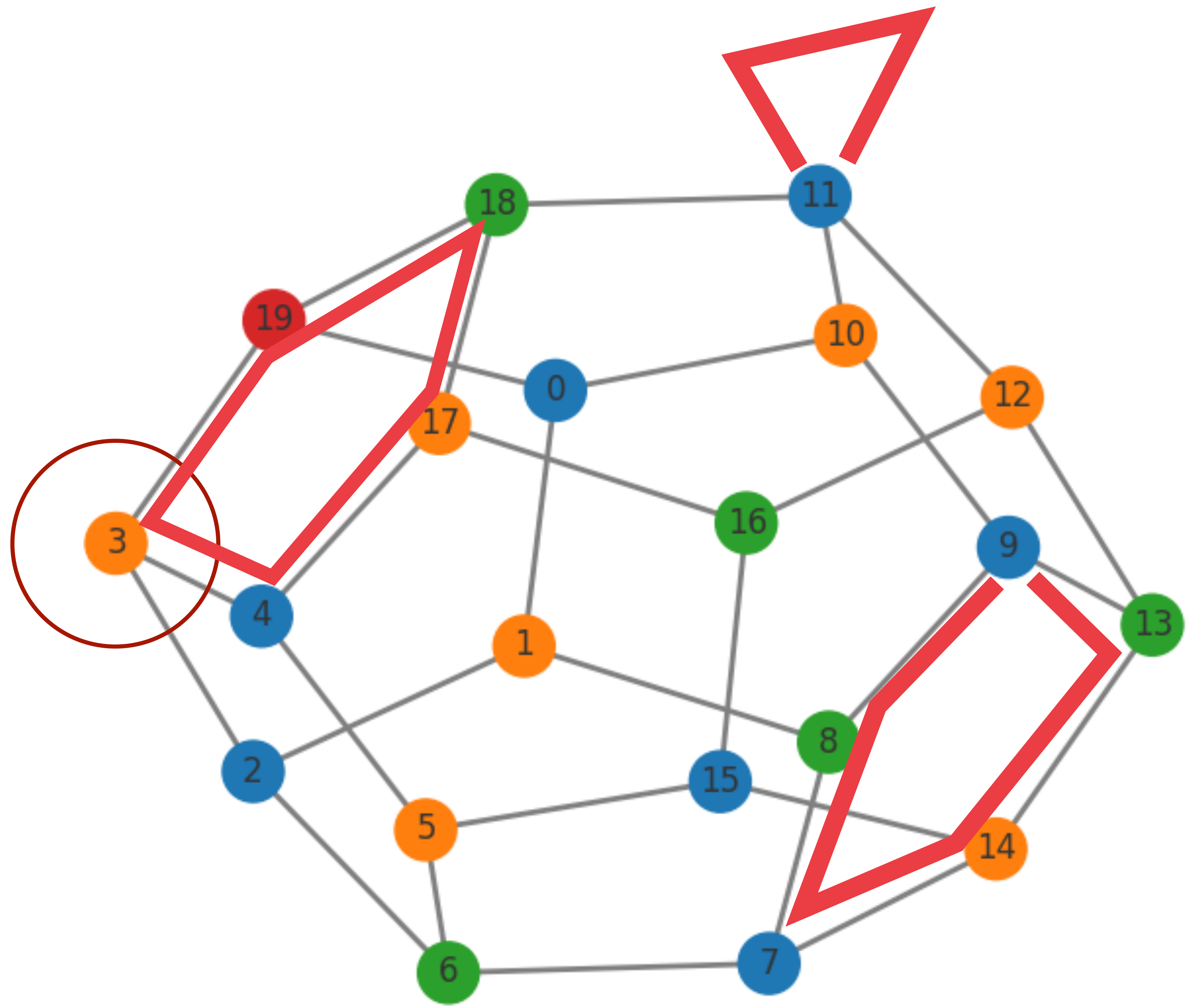
SHORTEST PATH

$$U_{i,j}$$

$$OF = \sum_{i,j} D_{i,j} U_{i,j}$$

$$\forall_i \quad \sum_j U_{i,j} = 1$$

$$\forall_i \quad \sum_j U_{j,i} = 1$$



SHORTEST PATH

$$OF = \sum_{i,j} D_{i,j} U_{i,j}$$

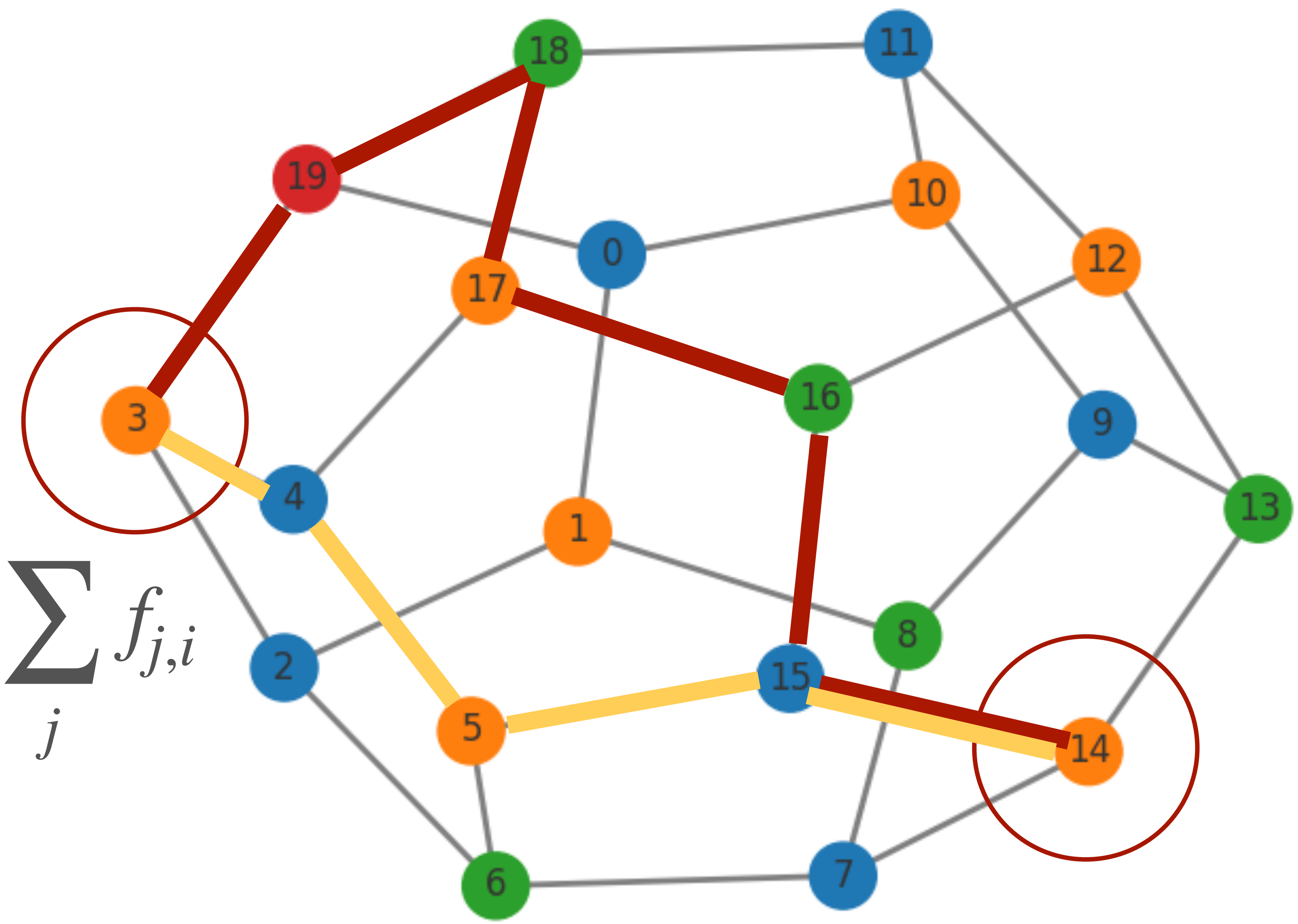
$$\forall_i \quad \sum_j U_{i,j} = 1$$

$$\forall_i \quad \sum_j U_{j,i} = 1$$

Parametr

$$\forall_i \quad S_i - D_i = \sum_j f_{i,j} - \sum_j f_{j,i}$$

$$\forall_{i,j} \quad f_{i,j} \le M \times U_{i,j}$$



SHORTEST PATH

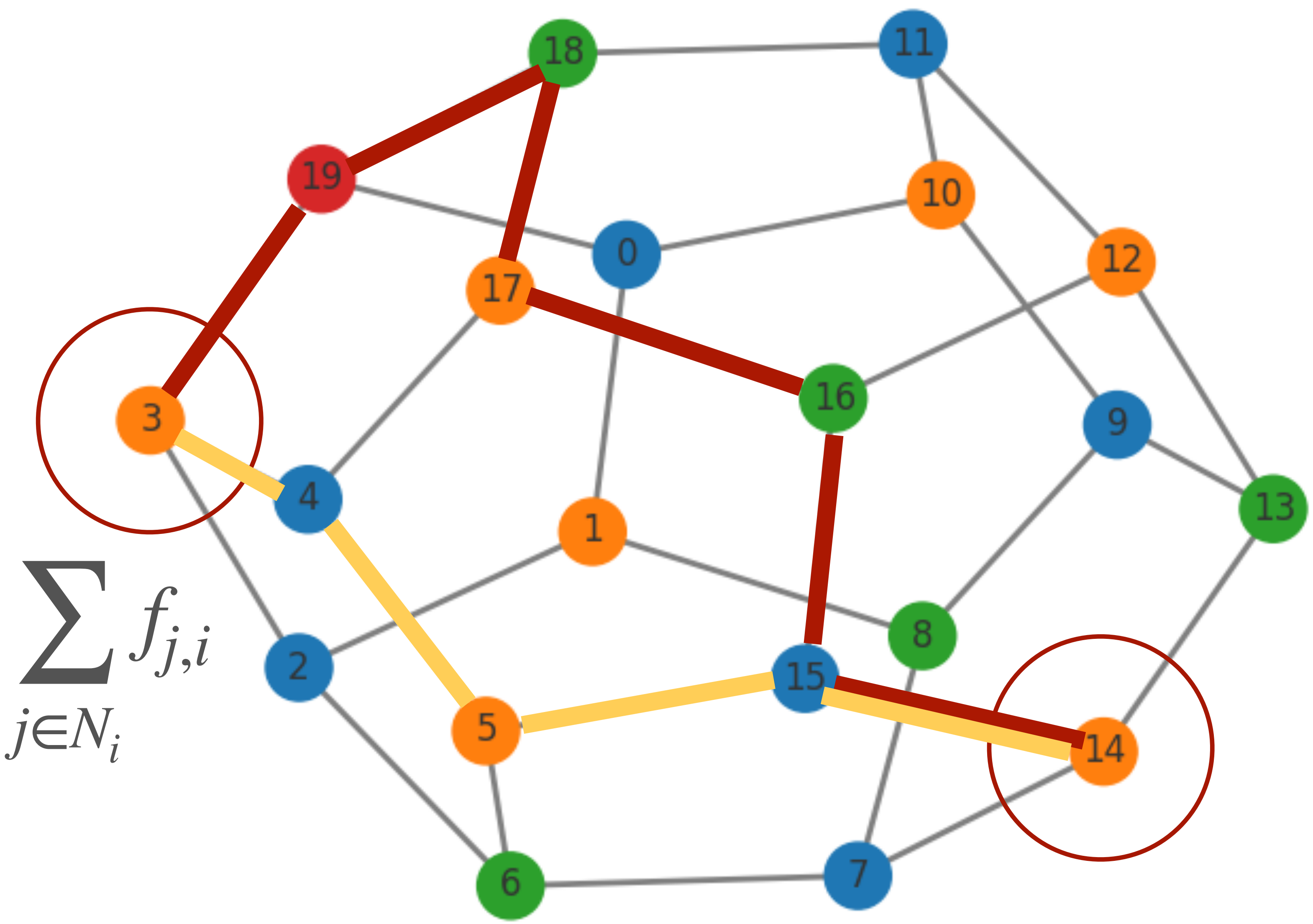
$$OF = \sum_{i,j \in N_i} D_{i,j} U_{i,j}$$

$$\forall_i \quad \sum_{j \in N_i} U_{i,j} = 1$$

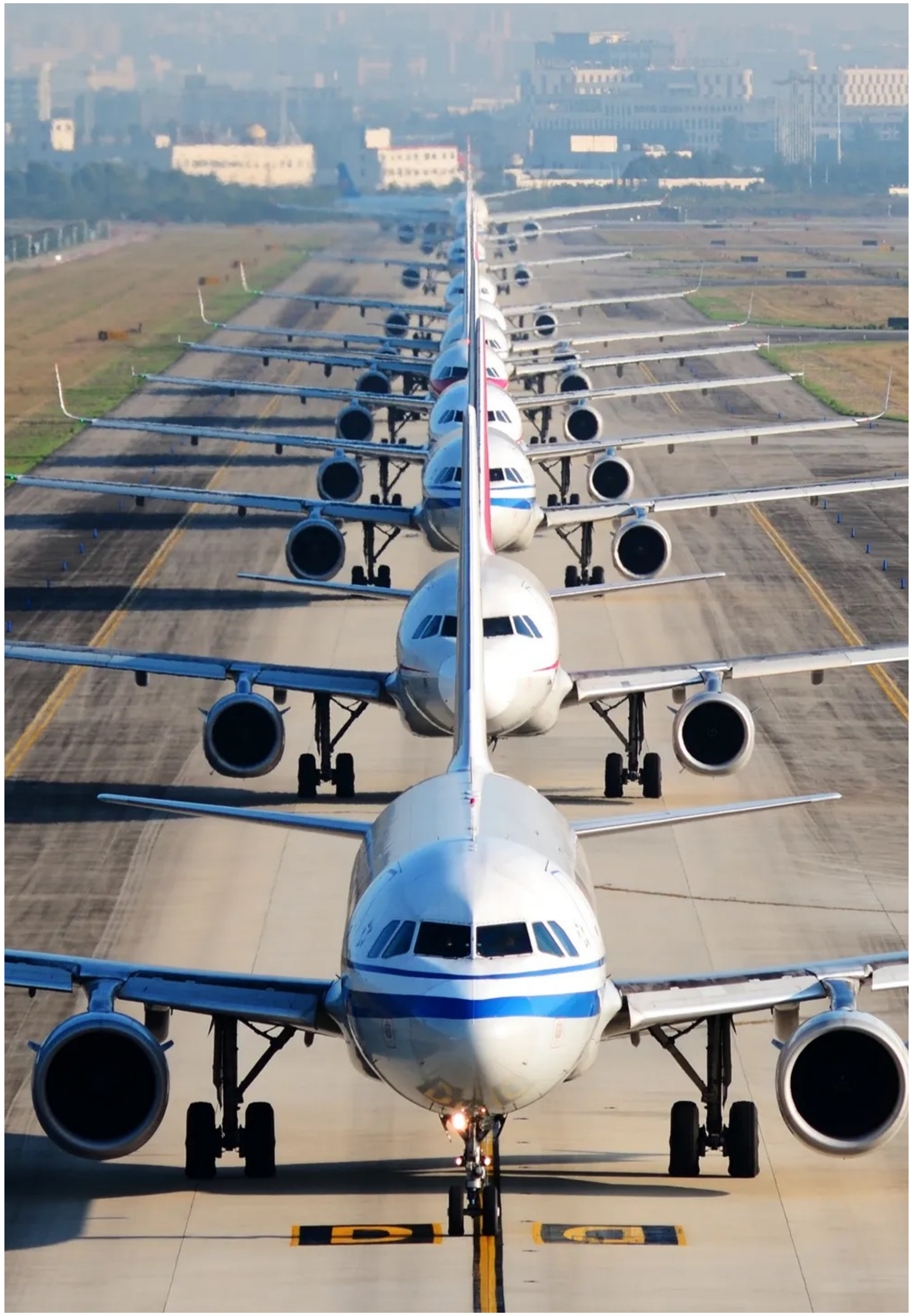
$$\forall_i \quad \sum_{j \in N_i} U_{j,i} = 1$$

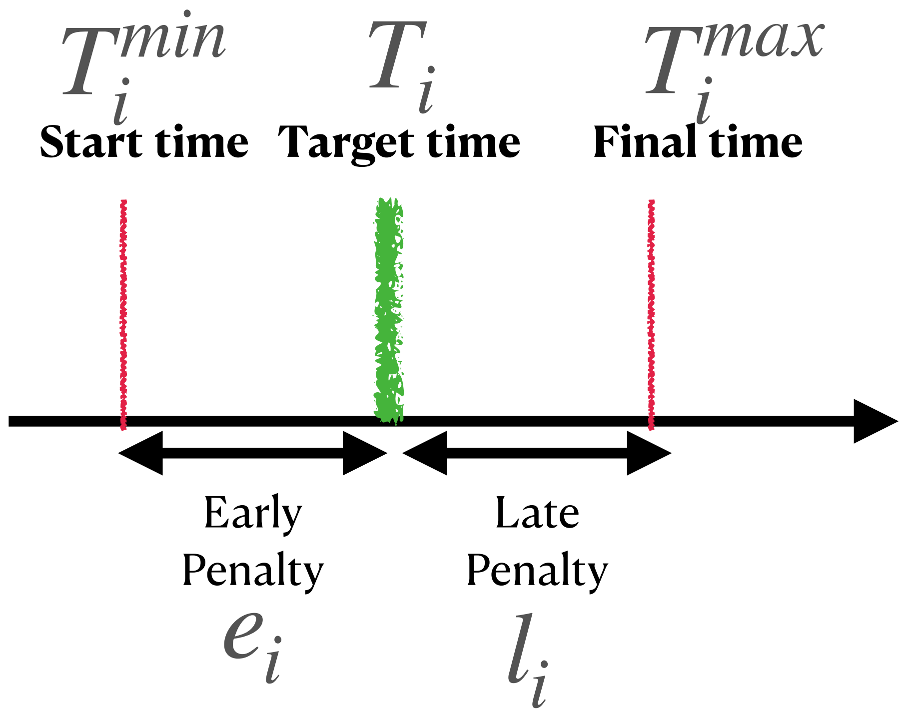
$$\forall_i \quad S_i - D_i = \sum_{j \in N_i} f_{i,j} - \sum_{j \in N_i} f_{j,i}$$

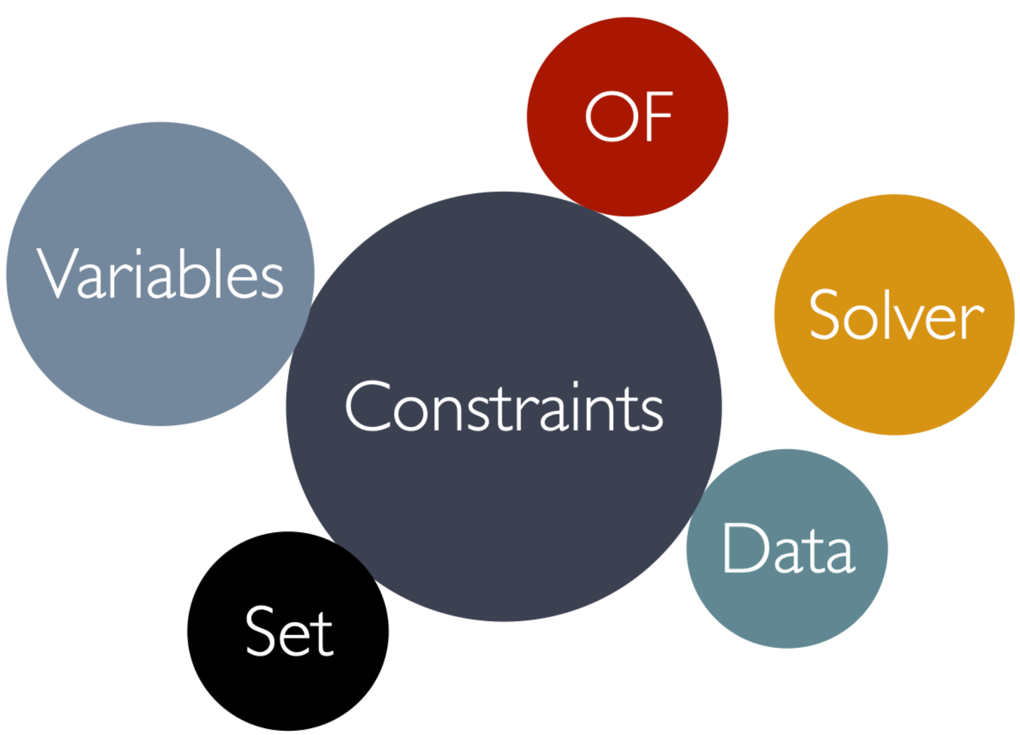
$$\forall_{i,j \in N_j} \quad f_{i,j} \leq M \times U_{i,j}$$

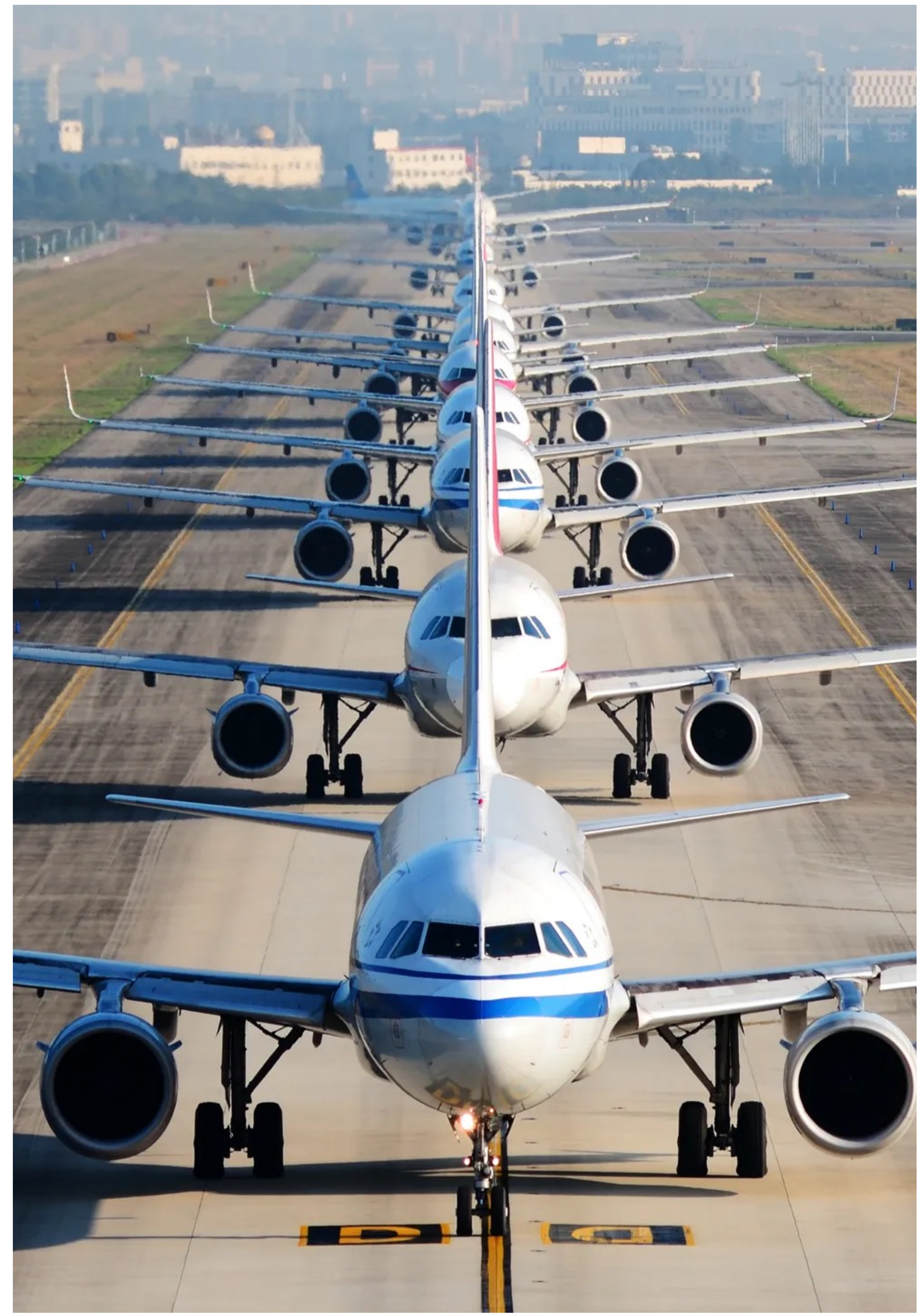


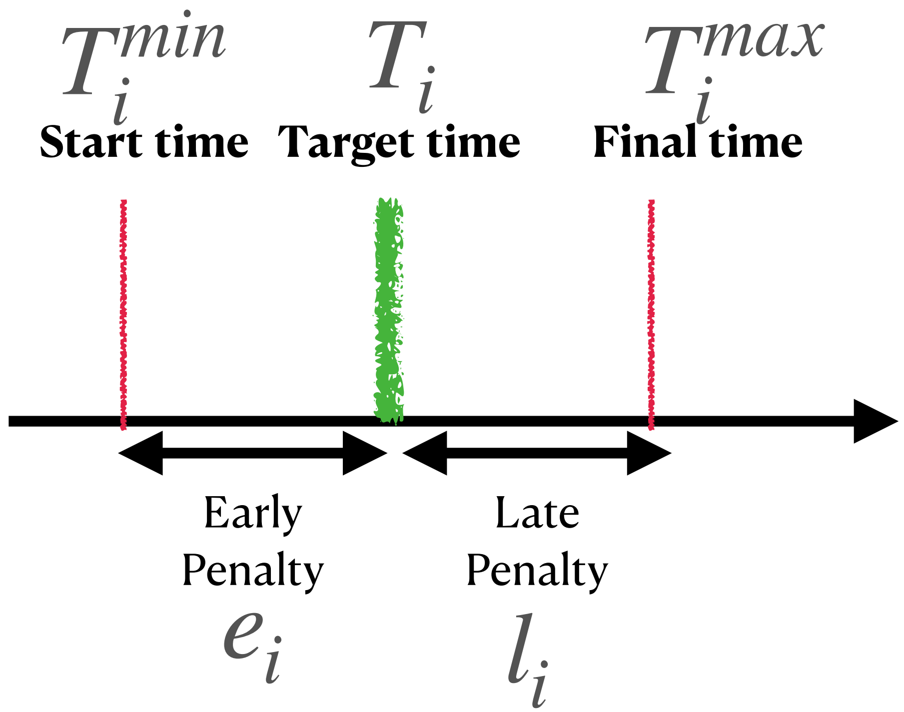
AIRCRAFT LANDING PROBLEM




T_i^{min}
Start time
T_i
Target time
T_i^{max}
Final time
Early Penalty
e_i
Late Penalty
l_i


OF
Variables
Solver
Constraints
Data
Set

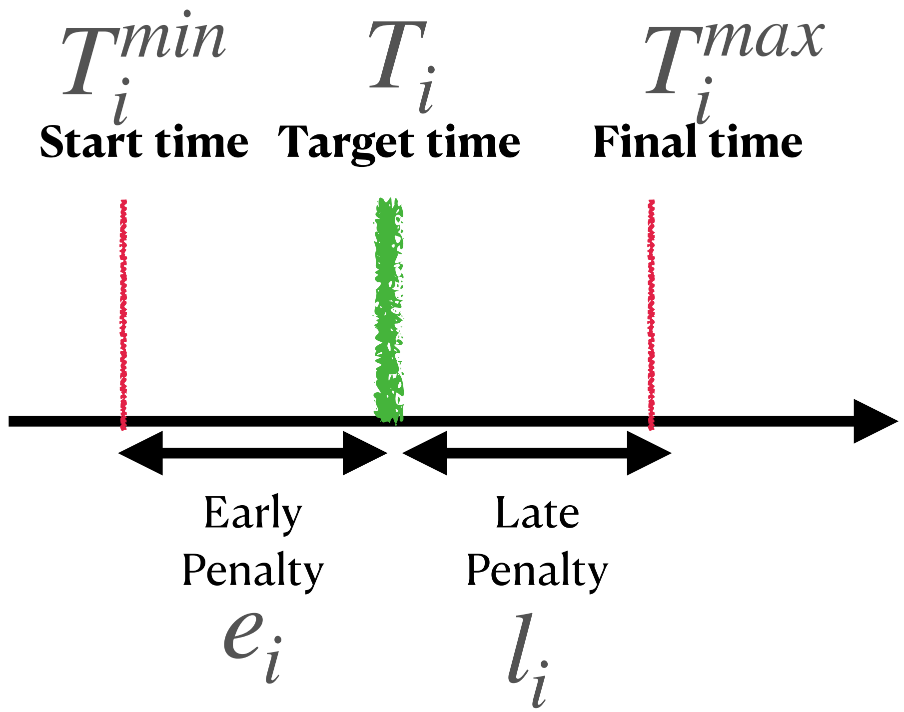




$S_{i,j}$ Separation time required after i lands before j can land

A_i Appearance time of airplane i

Prepare the optimal landing schedule



$S_{i,j}$ Separation time required after i lands before j can land

A_i Appearance time of airplane i

$U_{i,j}$ Airplane i lands before j

x_i Airplane i lands at x_i

$$x_i + S_{i,j} \leq x_j + M(1 - U_{i,j})$$

$$T_i^{min} \leq x_i \leq T_i^{max}$$

$$U_{i,j} + U_{j,i} = 1$$

Prepare the optimal landing schedule

$$OF = \sum_i Penalty_i^{late} + Penalty_i^{early}$$

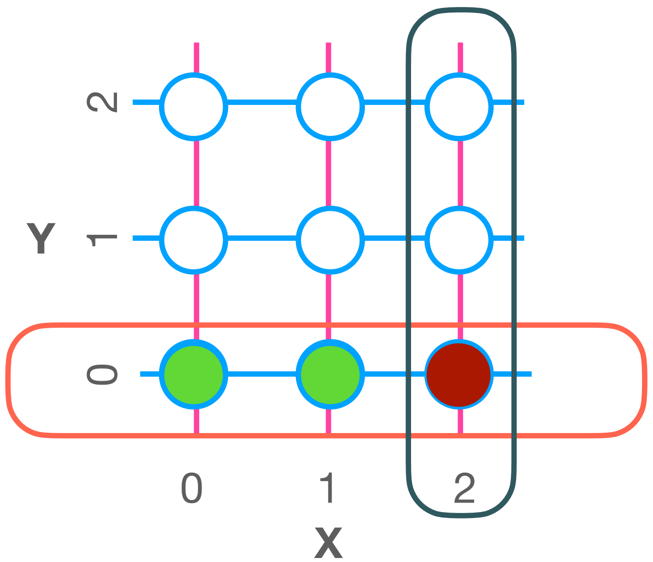
$$Penalty_i^{late} \geq l_i(x_i - T_i)$$

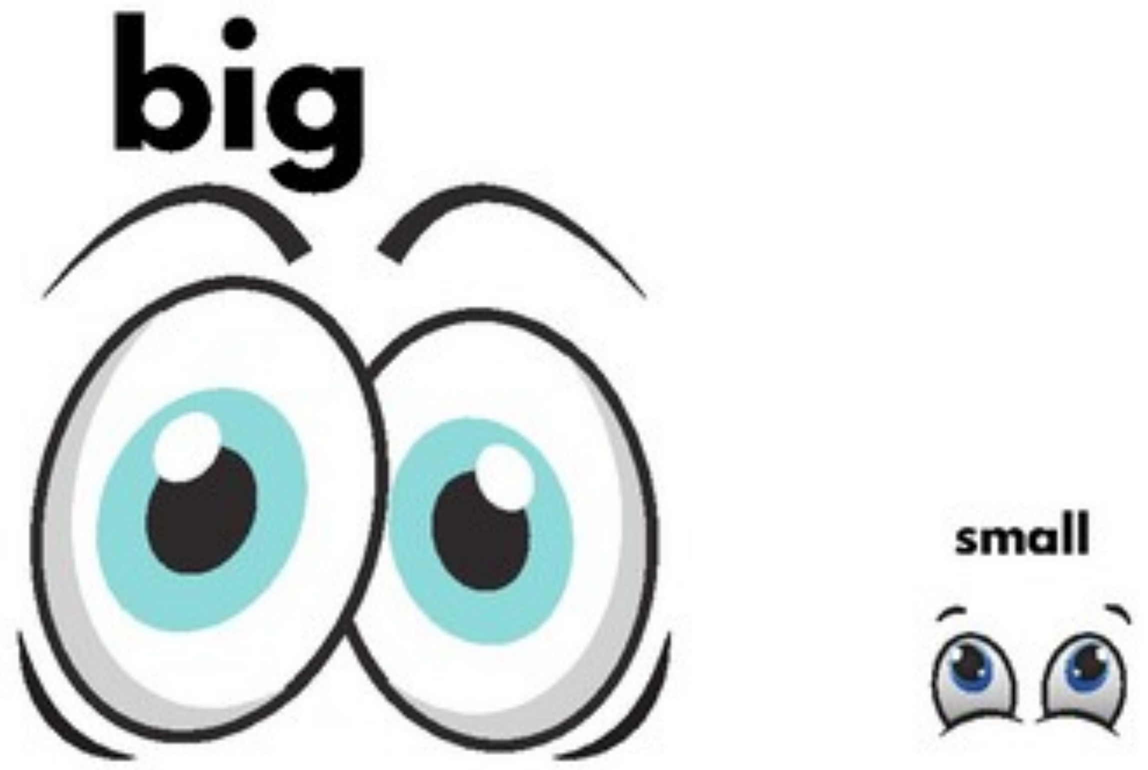
$$Penalty_i^{early} \geq e_i(T_i - x_i)$$

$$\max \quad OF = x + y$$

$$x + 5y \leq 2$$

$$0 \leq x, y \leq 2$$





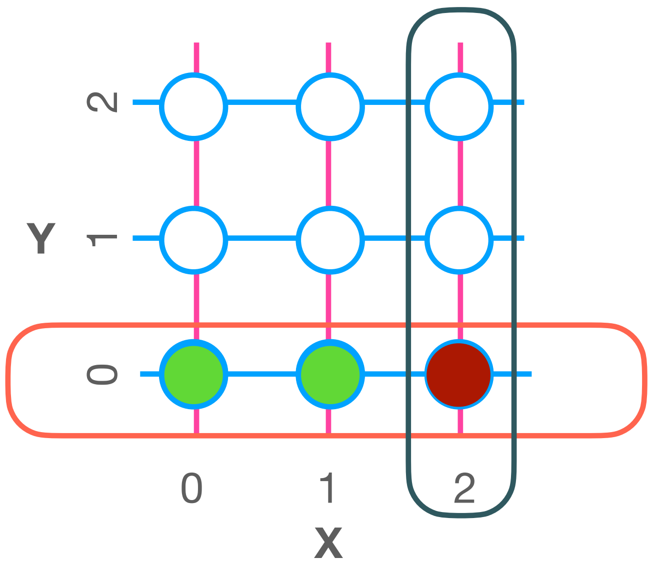
9 possible combinations

CODE 1

$$\max \quad OF = x + y$$

$$x + 5y \leq 2$$

$$0 \leq x, y \leq 2$$




```
!pip install ortools
from ortools.sat.python import cp_model

def main() -> None:
    # Creates the model.
    model = cp_model.CpModel()

    # Creates the variables.
    x = model.NewIntVar(0, 2, "x")
    y = model.NewIntVar(0, 2, "y")

    # Creates the constraints.
    model.add( x + 5 * y  <= 2)
    model.maximize(x + y)

    # Creates a solver and solves the model.
    solver = cp_model.CpSolver()
    status = solver.solve(model)

    if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE:
        print(f"Maximum of objective function:
{solver.objective_value}")
        print(f"x = {solver.value(x)}")
        print(f"y = {solver.value(y)}")
    else:
        print("No solution found.")

if __name__ == "__main__":
    main()
```

CODE 2

$$\min \quad OF = x + y + 4z$$

$$x + 5y \geq 2$$

$$x + 3y \leq 10z$$

$$0 \leq x, y \leq 2$$

$$z \in \{0,1\}$$

CODE 3

$$\min \; OF = x + y + 4z$$

$$x + 5y \geq 2$$

$$if \quad z = 1 \rightarrow x = 2y$$

$$0 \leq x, y \leq 2$$

$$z \in \{0,1\}$$