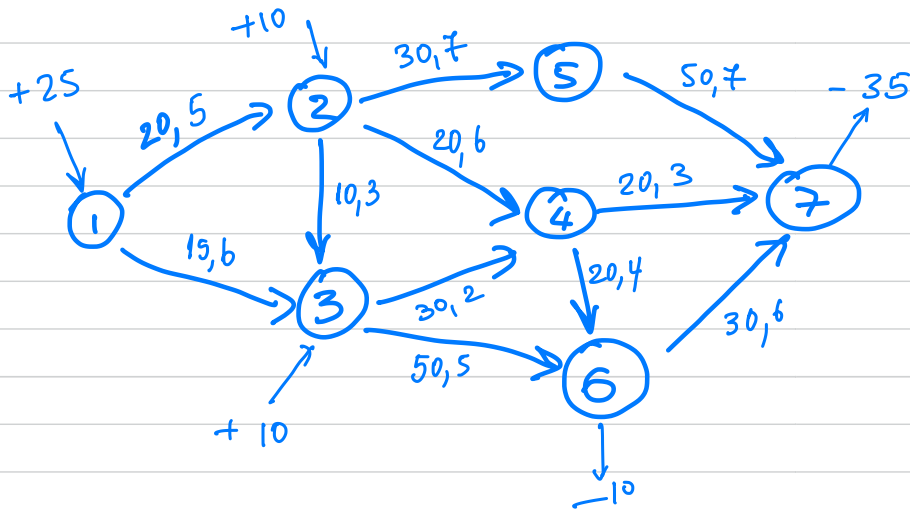


23/3/26



$$\min \sum_{(i,j) \in E} c_{ij} x_{ij}$$

$$x_{ij} \leq v_{ij} \quad (i,j) \in E$$

$$\sum_{(k,i)} x_{ki} + s_i = \sum_{(i,j)} x_{ij}$$

$$x_{ij} \geq 0 \quad (i,j) \in E.$$

Εισαγωγή στο model class:

Σύνολα δεδομένων

sets

$\left\{ \begin{array}{l} \text{nodes} = \{1, \dots, 7\} \\ \text{edges} = \{(), (), \dots\} \end{array} \right.$

$\text{nodes} = \text{range}(1, 8)$

$\text{edges} = \{ (1,2), (1,3), (2,3), \dots \}$ (length = 11)

(a)

cap_dict = { }
i = 0

for e in edges:

cap_dict.append(e: capacity[i])

i = i + 1

(b)

enumerate(edges)



0, (1,2)

1, (1,3)

⋮

10, (6,7)

cap_dict = { e: capacity[i]

for i, e in enumerate(edges)}

cost_dict = { e: cost[i], - - -

supply_dict = { i : supply[i-1] for i in nodes }

OK

m = Model()

m.add_set('nodes', nodes)

m.add_set('edges', edges)

m.add_parameter('supply',
supply_dict,
index = 'nodes')

m.add_parameter('cost', cost_dict,
index = 'edges')

⋮

('capacity', cap_dict,
index = 'edges')

Metabarnes $x_{ij} \quad (i,j) \in E$

m.add-variable('x', index='edges')

Objective

def objective(genmodel)

return sum(genmodel.cost[e] *
genmodel.x[e]

for e in genmodel.edges)

m.set-objective(objective, sense="min")

Constraints

capacity

def capacity-constr(g, e)

return g.x[e] <= g.capacity[e]

m.add_constraint('capacity', capacity_constr, index='edges')

Balance $\sum_{(k,i) \in E} x_{ki} : \frac{\text{αποεξόφηση που}}{\text{σώρι } i}$

$$\sum_{\substack{e \in E : \\ e = (k, i)}} : \sum_{\substack{e \in E \\ e[1] == i}}$$

↑

```
def in_arcs(g, i)
```

```
    return [e for e in g.edges if e[1] == i]
```

```
def out_arcs(g, i)
```

```
    return [e for e in g.edges if e[0] == i]
```

```
def flow_balance(g, i)
```

```
    return sum(g.x[e] for e in out_arcs(g, i))
```

```
        - sum(g.x[e] for e in in_arcs(g, i)) ==
```

```
        g.supply[i]
```

```
m.add_constraint("flow", flow_balance, index='nodes')
```

```
m.solve()
```

```
m.display()
```