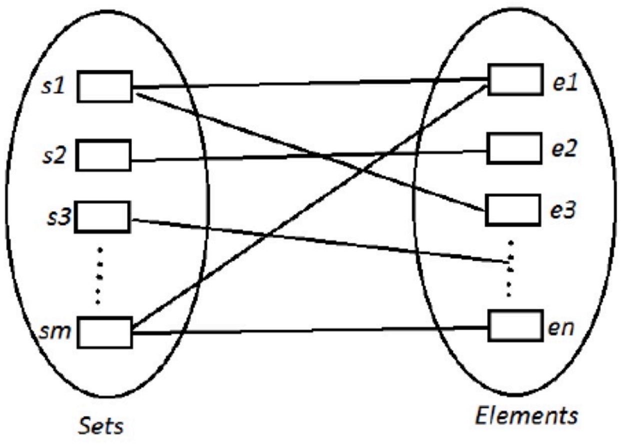
VG441 PS 3 周维翰 518021911039

1.



Decision variables: $x_t = 1$ if we choose set S_t, 0 otherwise.

$a_{it} = 1$ if elements e_i is in set S_t, 0 otherwise.

Objective: minimize $\sum_{t=1}^{m} x_t$

Constraints: $x_t \in \{0,1\} \quad \forall t = 1,2,3,\cdots,m$

$a_{it} \in \{0,1\} \quad \forall i = 1,2,3,\cdots,n \,;\ \forall t = 1,2,3,\cdots,m$

$\sum_{i=1}^{n} a_{it} x_t \geq 1 \quad \forall t = 1,2,3,\cdots,m;$

The code and result is on next page.

In [1]:

```
import numpy as np
from gurobipy import *
```

In [2]:

```
T = 5
m = Model()
x = m.addVars(T, lb=np.zeros(5), vtype = GRB.BINARY, name = "if_chosen")
m.setObjective(quicksum(x[t] for t in range(T)), GRB.MINIMIZE)

c1 = m.addConstr(x[0] + x[1] >= 1)
c2 = m.addConstr(x[0] + x[3] >= 1)
c3 = m.addConstr(x[2] >= 1)
c4 = m.addConstr(x[4] >= 1)
c5 = m.addConstr(x[1] + x[4] >= 1)
c6 = m.addConstr(x[1] + x[2] >= 1)
c7 = m.addConstr(x[3] >= 1)
c8 = m.addConstr(x[0] + x[3] >= 1)
```

```
Restricted license - for non-production use only - expires 2022-01-13
```

In [3]:

```
m.optimize()
m.printAttr('X')
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 8 rows, 5 columns and 13 nonzeros
Model fingerprint: 0x74c2bfc5
Variable types: 0 continuous, 5 integer (5 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 4.0000000
Presolve removed 8 rows and 5 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 4

Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%

    Variable            X
-------------------------
if_chosen[1]            1
if_chosen[2]            1
if_chosen[3]            1
if_chosen[4]            1
```

2. For n items, each item I_i has a value v_i and size S_i. The capacity is B.
First, we can re-index them such that $\frac{v_1}{S_1} \geq \frac{v_2}{S_2} \geq \cdots \geq \frac{v_n}{S_n}$

We can write the greedy solution as $G=(x_1, x_2, x_3, \cdots, x_n)$, where x_i indicates fraction of item I_i taken.

We can write any optimal solution as $O=(y_1, y_2, y_3, \cdots, y_n)$, where y_i indicates fraction of item I_i taken.

We can get $\sum_{i=1}^{n} x_i S_i = \sum_{i=1}^{n} y_i S_i = B$

For the first item I_a where the two solutions differ from each other, we can have $x_a > y_a$ since greedy alway takes as much as it can.

Then we consider a new solution $O' = (y_1', y_2', y_3', \cdots, y_n')$.

For $j<a$, we let $y_j' = y_j$.

For $j=a$, we let $y_j' = x_j$.

For $j>a$, we remove items of total size $(x_a - y_a) S_a$ and reset y_j'.

The total value of solution O' is no less than O.

Since O is an optimal solution, the total value of O' and O must equal to each other.

Therefore, O' is also an optimal solution.

If we continue this process, we can finally convert O into G without changing the total value.

Therefore, G is an optimal solution.