题面

> 某钢管零售商从钢管厂进货，将钢管按照顾客的要求切割后售出，从钢管厂进货时得到的原料钢管都是19m。
>
> 1. 现有一客户需要50根4m、20根6m和15根8m的钢管应如何下料最省(总余料最少或使用的钢管最少)?
> 2. 零售商如果采用的不同切割模式太多，将会导致生产过程复杂化，从而增加生产和管理成本，所以该零售商规定采用的不同切割模式不能超过3种。此外，如果客户还需要10根5m的钢管，又该如何下料?请分别针对上述两种情景需求建立数学规划模型，利用**Gurobi**求解。

Task 1

问题分析

由于原料钢管的长度为19m，客户将其切割成长度为4m、6m和8m的钢管。故一根钢管有**有限个**切割组合，如下表所示

	num. of 4m	num. of 6m	num. of 8m	余料 / 根 · m
组合1	4	0	0	3
组合2	3	1	0	1
组合3	2	0	1	3
组合4	1	1	1	1
组合5	1	2	0	3
组合6	0	3	0	1

一共有6中组合。因而只需要设置6个决策变量（Integer型）,并以总余料最小为问题的目标即可。

变量约定

设采取的不同的组合数分别为 $$ x_i,\ i=1,2,\ldots,6 $$ 相应的余料为 $$ c_i,\ i=1,2,\ldots,6 $$ 总余量为 $$ z $$

约束条件

$$ 4x_1+3x_2+2x_3+x_4+x_5=50 $$

$$ x_2+x_4+{2x}_5+{3x}_6=20 $$

$$ x_3+x_4=15 $$

目标函数

$$ \min z= \sum_{i=1}^{6} c_i x_i $$

编程求解

```python
import gurobipy as gp
from gurobipy import GRB

model = gp.Model("steel_cutting")

x = model.addVars(6, vtype=GRB.INTEGER, name="x")
c = [3, 1, 3, 1, 3, 1]  # 每种组合对应的余料

model.addConstr(4*x[0] + 3*x[1] + 2*x[2] + x[3] + x[4] == 50, "")
model.addConstr(x[1] + x[3] + 2*x[4] + 3*x[5] == 20, "")
model.addConstr(x[2] + x[3] == 15, "")

obj = sum(c[i] * x[i] for i in range(6))
model.setObjective(obj, GRB.MINIMIZE)

model.optimize()

if model.status == GRB.OPTIMAL:
    print("最优解为:")
    for i in range(6):
        print(f"x[{i}] = {x[i].x}")
    print(f"总余料最小为: {model.objVal}")
else:
    print("未找到最优解")
```

结果为

即10根采用组合2（3根4m和1根6m的钢管），5根采用组合3（2根4m和1根8m的钢管），10根采用组合4（1根4m、1根6m和1根8m的钢管）。得到的最小总余料为35.0m。

Task2

问题分析

如上文所述，我们提出的解决方案一共有6中切割方式。要想限制切割方式，只需要针对每一个切割方式 x 引入新的0-1变量，并确保这16个0-1的和不超过3。

在问题2中，客户还需要10根5m的钢管，故而原有的切割模式需要进行改变。最终得到下表.

切割方式	num. of 4m	num. of 5m	num. of 6m	num. of 8m	余料
1	0	0	0	2	3
2	0	0	3	0	1
3	0	1	1	1	0
4	0	1	2	0	2
5	0	2	0	1	1

切割方式	num. of 4m	num. of 5m	num. of 6m	num. of 8m	余料
6	0	2	1	0	3
7	1	0	1	1	1
8	1	0	2	0	3
9	1	1	0	1	2
10	1	3	0	0	0
11	2	0	0	1	3
12	2	1	1	0	0
13	2	2	0	0	1
14	3	0	1	0	1
15	3	1	0	0	2
16	4	0	0	0	3

*注：推算所有切割方式的代码见附录。

变量约定

设采取的不同的组合数分别为 $$ x_i,\ i=1,2,\ldots,16 $$ 相应的余料为 $$ c_i,\ i=1,2,\ldots,16 $$ 另设启用标志（0-1变量） $$ k_i,\ i=1,2,\ldots,16 $$ 总余量为 $$ w $$

约束条件

（展开） $$ k_{7}x_7+k_{8}x_8+k_{9}x_9+ k_{10}x_{10} +2k_{11}x_{11}+2k_{12}x_{12}+2k_{13}x_{13}+3k_{14}x_{14}+3k_{15}x_{15} +4k_{16}x_{16}=50 $$

$$
k_{3}x_3+k_{4}x_4+2k_{5}x_5+2k_{6}x_6+k_{9}x_9+3k_{10}x_{10}+k_{12}x_{12}+2k_{13}x_{13}+k_{15}x_{15}=10
$$

$$ 3k_{2}x_2+k_{3}x_3+2k_{4}x_4+k_{6}x_6+k_{7}x_7+2k_{8}x_8+k_{12}x_{12}+k_{14}x_{14}=20 $$

$$ 2k_{1}x_1+k_{3}x_3+k_{5}x_5+k_{7}x_7+k_{9}x_9+k_{11}x_{11}=15 $$

$$ \max \sum_{i=1}^{16}{k_i=3}
$$

目标函数

$$ \min w= \sum_{i=1}^{16} c_i x_i $$

编程求解

```
from gurobipy import *
m = Model("steel_cutting")

c = [3, 1, 0, 2, 1, 3, 1, 3, 2, 0, 3, 0, 1, 1, 2, 3]
x = m.addVars(16, vtype=GRB.INTEGER, name="x")
k = m.addVars(16, vtype=GRB.BINARY, name="k")

m.addConstr(quicksum(k[i] * x[i] for i in range(16)) == 50)
m.addConstr(k[2] * x[2] + k[3] * x[3] + 2 * k[4] * x[4] + 2 * k[5] * x[5] +
k[8] * x[8] + 3 * k[9] * x[9] + k[11] * x[11] + 2 * k[12] * x[12] + k[14] *
x[14] == 10)
m.addConstr(3 * k[1] * x[1] + k[2] * x[2] + 2 * k[3] * x[3] + k[5] * x[5] +
k[6] * x[6] + k[7] * x[7] + 2 * k[8] * x[8] + k[12] * x[12] + k[13] * x[13]
== 20)
m.addConstr(2 * k[0] * x[0] + k[2] * x[2] + k[4] * x[4] + k[6] * x[6] +
k[8] * x[8] + k[10] * x[10] == 15)
m.addConstr(quicksum(k[i] for i in range(16)) <= 3)

m.setObjective(quicksum(c[i] * x[i] for i in range(16)), GRB.MINIMIZE)
m.optimize()

for i in range(16):
    print(f"x[{i}] = {x[i].x}, k[{i}] = {k[i].x}")
print(f"Objective Value: {m.objVal}")
```

求解结果为

结论

对于Task1.最优方案为10根采用组合2（3根4m和1根6m的钢管），5根采用组合3（2根4m和1根8m的钢管），10根采用组合4（1根4m、1根6m和1根8m的钢管）。得到的**最小总余料为35.0m。**

对于Task2.最优方案为**启用切割方案7、方案12和方案15**，相应的切割参数如图所示。得到的**最小总余料为110m。**

附录

Task 2中生成最优切割方案的**暴力解法**：

```
ans=[]
for i in range(5):
    for j in range(5):
        for h in range(5):
            for g in range(5):
                if 4*i+5*j+6*h+8*g<=19 and 19-4*i-5*j-6*h-8*g<4:
                    ans.append([i,j,h,g,19-4*i-5*j-6*h-8*g])
__=0
for _ in ans:
    print(__, *_)
    __+=1
```