数学实验 Exp10

赵晨阳 计 06 2020012363

10.8

问题分析、模型假设与模型建立

我们的目标是在满足每个时间段服务员数量要求的前提下，最小化雇佣成本。

全时的服务员将在在 12 时到 13 时之间任意时刻开始吃 1 小时的午饭。考虑到题目按照每小时为单位给出服务员数量下限要求的，全时员工若在非整点开始吃午饭，否则会降低员工的总可服务时长，故而在非整点开始吃午饭的全时服务员应调整至整点开始吃午饭。

从而，将全时服务员分为两类：第一类全时服务员在 12 时到 13 时之间吃午饭，第二类全时服务员在 13 时到 14 时之间吃午饭。我们分别用 x_1 和 x_2 表示这两类服务员的数量。

同样地，对于半时服务员，我们将所有非整点开始服务的半时员工调整至从整点开始工作。故而，总共有五类半时服务员，每一类都工作 4 小时：第一类 9 时开始工作，第二类 10 时开始，第三类 11 时开始，第四类 12 时开始，最后一类 13 时开始，用 y_1, y_2, y_3, y_4, y_5 表示五类服务员的数量。

总的雇佣成本为：

$$F = 100(x_1 + x_2) + 40(y_1 + y_2 + y_3 + y_4 + y_5) \tag{1}$$

注意，所有变量均为非负整数，符合整数规划的实际含义。

针对每个时间段，我们可以列出服务员数量的下界。

$$
\begin{aligned}
x_1 + x_2 + y_1 &\geq 4 \\
x_1 + x_2 + y_1 + y_2 &\geq 3 \\
x_1 + x_2 + y_1 + y_2 + y_3 &\geq 4 \\
x_2 + y_1 + y_2 + y_3 + y_4 &\geq 6 \\
x_1 + y_2 + y_3 + y_4 + y_5 &\geq 5 \\
x_1 + x_2 + y_3 + y_4 + y_5 &\geq 6 \\
x_1 + x_2 + y_4 + y_5 &\geq 8 \\
x_1 + x_2 + y_5 &\geq 8
\end{aligned} \tag{2}
$$

此外，每天最多雇佣 3 名半时服务员，则有：

$$
y_1 + y_2 + y_3 + y_4 + y_5 \leq 3 \tag{3}
$$

如果每天不雇佣半时服务员，则有：

$$
y_1 + y_2 + y_3 + y_4 + y_5 = 0 \tag{4}
$$

如果没有半时服务员数量的限制，则没有相关限制。

算法设计

在上述模型中，费用作为目标函数，以及约束条件都是线性的。此外，所有的规划变量都必须是非负整数。我们的目标是求解一个典型的整数线性规划问题，即找到使费用最小化的有效解决方案。为了解决这个问题，我们可以将模型直接输入到 Python 的 PuLP 中进行求解。

代码

```python
# Import necessary libraries
from pulp import *
import numpy as np


# Define the problem to be solved
def solve(half_limit=None):
    # Define the problem
    prob = LpProblem("Employment", LpMinimize)

    # Set the parameters
    work_time = list(range(1, 9))  # Work hours
```

```
    n = np.array([4, 3, 4, 6, 5, 6, 8, 8])  # Number of servers needed in each time slot
    s_f = 100  # Daily salary of full-time server
    s_h = 40  # Daily salary of part-time server

    # Define the constraints
    lunch_time = [4, 5]
    l = LpVariable.dicts("l", lunch_time, lowBound=0, cat="Integer")
    half_start_time = list(range(1, 6))
    h = LpVariable.dicts("h", half_start_time, lowBound=0, cat="Integer")

    # Define the objective function
    prob += lpSum(
        [s_f * l[i] for i in lunch_time] + [s_h * h[i] for i in half_start_time]
    )

    # Add the constraints
    for i in work_time:
        prob += (
            lpSum([l[j] for j in lunch_time if j != i])
            + lpSum([h[j] for j in half_start_time if j <= i <= j + 3])
            >= n[i - 1]
        )

    if half_limit is not None:
        prob += lpSum([h[i] for i in half_start_time]) <= half_limit

    # Solve the problem
    prob.solve()

    # Print the results
    print("Solution:")
    for i in lunch_time:
        print(f"l[{i}] = {value(l[i])}")
    for i in half_start_time:
        print(f"h[{i}] = {value(h[i])}")
    print("Objective function value: ", value(prob.objective))
    print("\n")


```

```
# Call the function to solve the problem with various parameters
solve(3)
solve(0)
solve(None)
```

结果、分析与结论

如下只展示有效部分的输出：

```
Solution:
l[4] = 2.0
l[5] = 5.0
h[1] = 0.0
h[2] = 0.0
h[3] = 0.0
h[4] = 2.0
h[5] = 1.0
Objective function value:  820.0

Solution:
l[4] = 5.0
l[5] = 6.0
h[1] = 0.0
h[2] = 0.0
h[3] = 0.0
h[4] = 0.0
h[5] = 0.0
Objective function value:  1100.0

Solution:
l[4] = 0.0
l[5] = 0.0
h[1] = 4.0
h[2] = 0.0
h[3] = 0.0
h[4] = 2.0
h[5] = 8.0
Objective function value:  560.0
```

根据解决方案的结果，我们可以得出以下结论：

如果每天最多可以雇佣 3 名半时服务员，那么最小花费为 820 元，此时的员工安排是：全时服务员 $x_1 = 2$，$x_2 = 5$，半时服务员 $y_1 = 0$，$y_2 = 0$，$y_3 = 0$，$y_4 = 2$，$y_5 = 1$。

如果每天不雇佣半时服务员，那么最小花费为 1100 元，此时的员工安排是：全时服务员 $x_1 = 5$，$x_2 = 6$，半时服务员 $y_1 = y_2 = y_3 = y_4 = y_5 = 0$。

如果没有半时服务员数目的限制，那么最小花费为 560 元，此时的员工安排是：全时服务员 $x_1 = x_2 = y_2 = y_3 = 0$，半时服务员 $y_1 = 4$，$y_4 = 2$，$y_5 = 8$。

从结果可以看出，半时服务员的人数上限对最终成本影响很大。

进一步分析可以发现，半时服务员的时薪是 10 元，而全时服务员的时薪是约为14.29 元。因此，从小时薪水的角度来看，半时服务员更划算。

另一方面，半时服务员的工作时间更灵活，可以根据工作人数的需求进行更好的调整。例如，雇佣 2 名半时服务员（分别在 9 时和 13 时工作，覆盖时间为 9-13 和 13-17）相当于雇佣一个不需要午餐时间的全时服务员，并且花费更少。因此，在没有对半时员工数量限制的情况下，不需要全时服务员。

模型中每个变量的实际含义是根据不同雇佣计划下所需雇佣的服务员数量。在实际场景中，还需要考虑可能出现的紧急情况，这将使问题更加复杂。

10.9

问题分析、模型假设与模型建立

首先，我们考虑建立原油采购模型。假设按照 10000 元 / t 购买的原油 A 的数量为 a，按照 8000 元 / t 购买的原油 A 的数量为 b，按照 6000 元 / t 购买的原油 A 的数量为 c。根据它们所属的价格段，有以下限制条件：$0 \leq a \leq 500$，$0 \leq b \leq 500$，$0 \leq c \leq 500$。同时，当 a 没有达到上限 500 时，必须有 $b = 0$，即 $(500 - a) \times b = 0$。类似地，$(500 - b)c = 0$。在完成购买后，原油A的总吨数为$500 + a + b + c$，原油 B 的总吨数为 1000。

接下来，假设用于加工汽油甲的原油 A 和原油 B 的数量分别为 x_1 和 x_2，用于加工汽油乙的原油 A 和原油 B 的数量分别为 y_1 和 y_2。显然，我们有约束条件：

$$
\begin{aligned}
x_1, x_2, y_1, y_2 &\geq 0 \\
x_1 + y_1 &\leq 500 + a + b + c \\
x_2 + y_2 &\leq 1000 \\
\frac{x_1}{x_1 + x_2} \geq \frac{50}{100} &\Rightarrow 100x_1 \geq 50(x_1 + x_2) \\
100y_1 &\geq 60(y_1 + y_2)
\end{aligned} \tag{5}
$$

注意到最后两条约束为两种产品油料的比例限制。

最后的优化目标为：

$$
F = 4800(x_1 + x_2) + 5600(y_1 + y_2) - 10000a - 8000b - 6000c \tag{6}
$$

连续规划模型对问题建模时，上述变量的限制条件是实数；整数规划模型对问题建模则限制条件将是整数。

算法设计

我们的目标是求解 $\arg\max_{\text{valid solutions}} F$，可以直接使用 LINGO 算法进行求解。在整数规划中，还需要对每个约束变量使用 `@GIN` 进行限制。

以下是使用 LINGO 进行求解的代码实现，其中整数规划中的约束变量使用了 `@GIN` 进行限制。

整数规划代码为 `10_9_1.lg4`，连续规划代码为 `10_9_2.lg4`。

代码

```
MODEL:

MAX = 4800 * (x_1 + x_2) + 5600 * (y_1 + y_2) - 10000 * a - 8000 * b -
6000 * c;
(500 - a) * b = 0;
(500 - b) * c = 0;
a <= 500;
b <= 500;
c <= 500;
100 * x_1 >= 50 * (x_1 + x_2);
100 * y_1 >= 60 * (y_1 + y_2);
x_1 + y_1 <= 500 + a + b + c;
```

```
x_2 + y_2 <= 1000;
@GIN(x_1); @GIN(x_2); @GIN(y_1); @GIN(y_2);
@GIN(a); @GIN(b); @GIN(c);

END
```

如果进行的是连续规划：

```
MODEL:

MAX = 4800 * (x_1 + x_2) + 5600 * (y_1 + y_2) - 10000 * a - 8000 * b -
6000 * c;
(500 - a) * b = 0;
(500 - b) * c = 0;
a <= 500;
b <= 500;
c <= 500;
100 * x_1 >= 50 * (x_1 + x_2);
100 * y_1 >= 60 * (y_1 + y_2);
x_1 + y_1 <= 500 + a + b + c;
x_2 + y_2 <= 1000;

END
```

输出

整数规划：

```
  Global optimal solution found.
  Objective value:                              5000000.
  Objective bound:                              5000000.
  Infeasibilities:                              0.000000
  Extended solver steps:                               14
  Total solver iterations:                           3429
  Elapsed runtime seconds:                           0.49

  Model Class:                                       PIQP

  Total variables:                       7
  Nonlinear variables:                   3
  Integer variables:                     7
```

```
14
15    Total constraints:                    10
16    Nonlinear constraints:                 2
17
18    Total nonzeros:                       25
19    Nonlinear nonzeros:                    2
20
21
22
23                                                        Variable
      Value
24                                                             X_1
      0.000000
25                                                             X_2
      0.000000
26                                                             Y_1
      1500.000
27                                                             Y_2
      1000.000
28                                                               A
      500.0000
29                                                               B
      500.0000
30                                                               C
      0.000000
31
32                                                             Row    Slack or
      Surplus
33                                                               1
      5000000.
34                                                               2
      0.000000
35                                                               3
      0.000000
36                                                               4
      0.000000
37                                                               5
      0.000000
38                                                               6
      500.0000
39                                                               7
      0.000000
```

```
                                                                8
0.000000
                                                                9
0.000000
                                                               10
0.000000
```

连续规划:

```
 Global optimal solution found.
 Objective value:                              5000002.
 Objective bound:                              5000002.
 Infeasibilities:                          0.8684668E-06
 Extended solver steps:                                3
 Total solver iterations:                            119
 Elapsed runtime seconds:                           0.14

 Model Class:                                        QP

 Total variables:                      7
 Nonlinear variables:                  3
 Integer variables:                    0

 Total constraints:                   10
 Nonlinear constraints:                2

 Total nonzeros:                      25
 Nonlinear nonzeros:                   2



                            Variable           Value        Reduced
Cost
                                 X_1        0.000000
900.0000
                                 X_2        0.000000
0.000000
                                 Y_1        1500.000
0.000000

                                 Y_2        1000.000
```

```
27                                          Y_2          1000.000
    0.000000
28                                            A          500.0000
    0.000000
29                                            B          499.9991
    0.000000
30                                            C       0.9319157E-03
    0.000000
31
32                                          Row    Slack or Surplus      Dual
    Price
33                                            1          5000002.
    1.000000
34                                            2          0.000000
    6.000011
35                                            3      -0.8684668E-06
    1073058.
36                                            4          0.000000
    0.000000
37                                            5       0.9319157E-03
    0.000000
38                                            6          499.9991
    0.000000
39                                            7          0.000000
    -26.00000
40                                            8          0.000000
    -35.00000
41                                            9          0.000000
    7000.000
42                                           10          0.000000
    3500.000
43
44
```

结果、分析与结论

无论是连续规划模型还是整数规划模型，最大的利润都为 500 万，具体方案是购买 1000 吨原油 A，并将全部 1500 吨原油 A 和 1000 吨原油 B 全部用于生产汽油乙。这个方案的利润最大化是可以理解的，因为原油 B 的使用不会带来额外成本，而且汽油乙的性价比高于汽油甲。

最终两个模型的结果一致，说明最优的整数解也是最优的实数解。然而，观察 LINGO 的决

虽然两个模型的结果一致，说明最优的整数解也是最优的实数解。然而，观察 LINGO 的迭代次数，可以发现解决整数规划模型需要三千余次迭代，而连续规划模型只需要三百余次。这符合预期，因为整数规划比连续规划更为困难，因此求解效率较低。因此，在实际求解整数规划问题时，可以首先使用连续规划算法求解，然后将其结果用于试探和预测。

上述解的实际意义可以对应到具体的原油采购和加工策略。在实际场景中，还需要考虑其他因素，特别是市场上的不确定因素。因此，此处得到的理想解可以作为实际情况下的参考。

10.11

问题分析、模型假设、模型建立与算法设计

我们假设恰好有四种切割模式（可能有某些切割模式不使用），设第 $i\quad(1\leq i\leq 4)$ 种切割模式切割出的 290mm、315mm、350mm、455mm 钢管的数量分别为 $x_{i1},x_{i2},x_{i3},x_{i4}$，同时一共有 y_i 根钢管使用了第 i 种切割模式，每个变量都需要是**非负整数**。

最后切割出的不同钢管的个数需要满足客户需求，因此我们有以下约束条件：

$$\begin{aligned}
y_1x_{11}+y_2x_{21}+y_3x_{31}+y_4x_{41}&\geq 15\\
y_1x_{12}+y_2x_{22}+y_3x_{32}+y_4x_{42}&\geq 28\\
y_1x_{13}+y_2x_{23}+y_3x_{33}+y_4x_{43}&\geq 21\\
y_1x_{14}+y_2x_{24}+y_3x_{34}+y_4x_{44}&\geq 30
\end{aligned}\tag{7}$$

同时，每根原料钢管最多可以生产 5 根产品，因此有：

$$\forall 1\leq i\leq 4, x_{i1}+x_{i2}+x_{i3}+x_{i4}\leq 5\tag{8}$$

此外，每根钢管浪费不超过 100mm，即一根钢管至少使用 1750mm，我们有：

$$\forall 1\leq i\leq 4, 1750\leq 290x_{i1}+315x_{i2}+350x_{i3}+455x_{i4}\leq 1850\tag{9}$$

在计算成本时，我们可以将一根原料钢管的基础价格设为 1。考虑了使用频率排序后，进一步假设 $y_1\geq y_2\geq y_3\geq y_4$，最终的优化目标位为：

$$F=1.1y_1+1.2y_2+1.3y_3+1.4y_4\tag{10}$$

进一步的表优化

进一步约束优化

1. 可行方案有限

考虑到题切割方案的限制非常严格，这样的可行方案并不多。于是先枚举出所有的可行方案（代码位于 `./codes/10_11_1.py` 中），如下：

```
MAX_VALUE = 5
MIN_VALUE = 1750
MAX_WEIGHT = 1850

for x1 in range(MAX_VALUE + 1):
    for x2 in range(MAX_VALUE + 1):
        for x3 in range(MAX_VALUE + 1):
            for x4 in range(MAX_VALUE + 1):
                total_weight = 290 * x1 + 315 * x2 + 350 * x3 + 455 * x4
                if x1 + x2 + x3 + x4 <= MAX_VALUE and total_weight <= MAX_WEIGHT and total_weight >= MIN_VALUE:
                    print(x1, x2, x3, x4)
```

合法方案有且仅有：

```
0 0 0 4
0 0 5 0
0 1 3 1
0 2 2 1
0 3 1 1
1 0 3 1
1 1 2 1
1 2 0 2
2 0 1 2
2 1 0 2
3 0 0 2
```

基于此，我们可以加入对 x_{ij} 的限制以减小 LINGO 算法的计算量，以期望加速程序运行效率。不过，由于本问题的规模较小，这样的优化效果有限。虽然提前进行参数搜索以加强限制在这道题中没有显著作用，但是当问题规模增大时可能会有奇效。

2. 总切割次数有限

之前的优化方案较为复杂，考虑到总切割次数必然有有下界，也即假设所有原料钢管的长度

之前的优化方案较为复杂，考虑到总切割次数必然存在下界，也即假设所有原料钢管的长度都能用于生产时，没有任何损耗，也需要至少：

$$\left\lceil \frac{15 \times 290 + 28 \times 315 + 21 \times 350 + 30 \times 455}{1850} \right\rceil = 19 \tag{11}$$

因此至少需要 19 根钢管，我们以此为约束：

$$\sum_{i=1}^{4} y_i \geq 19 \tag{12}$$

代码

代码位于 `./codes/10_11_2.lg4` 下：

```
MODEL:

MIN = 1.1 * y_1 + 1.2 * y_2 + 1.3 * y_3 + 1.4 * y_4;
1750 <= 290 * x_11 + 315 * x_12 + 350 * x_13 + 455 * x_14;
290 * x_11 + 315 * x_12 + 350 * x_13 + 455 * x_14 <= 1850;
1750 <= 290 * x_21 + 315 * x_22 + 350 * x_23 + 455 * x_24;
290 * x_21 + 315 * x_22 + 350 * x_23 + 455 * x_24 <= 1850;
1750 <= 290 * x_31 + 315 * x_32 + 350 * x_33 + 455 * x_34;
290 * x_31 + 315 * x_32 + 350 * x_33 + 455 * x_34 <= 1850;
1750 <= 290 * x_41 + 315 * x_42 + 350 * x_43 + 455 * x_44;
290 * x_41 + 315 * x_42 + 350 * x_43 + 455 * x_44 <= 1850;
y_1 * x_11 + y_2 * x_21 + y_3 * x_31 + y_4 * x_41 >= 15;
y_1 * x_12 + y_2 * x_22 + y_3 * x_32 + y_4 * x_42 >= 28;
y_1 * x_13 + y_2 * x_23 + y_3 * x_33 + y_4 * x_43 >= 21;
y_1 * x_14 + y_2 * x_24 + y_3 * x_34 + y_4 * x_44 >= 30;
y_1 >= y_2;
y_2 >= y_3;
y_3 >= y_4;
x_11 + x_12 + x_13 + x_14 <= 5;
x_21 + x_22 + x_23 + x_24 <= 5;
x_31 + x_32 + x_33 + x_34 <= 5;
x_41 + x_42 + x_43 + x_44 <= 5;
y_1 + y_2 + y_3 + y_4 >= 19;
@GIN(y_1); @GIN(x_11); @GIN(x_12); @GIN(x_13); @GIN(x_14);
@GIN(y_2); @GIN(x_21); @GIN(x_22); @GIN(x_23); @GIN(x_24);
@GIN(y_3); @GIN(x_31); @GIN(x_32); @GIN(x_33); @GIN(x_34);
@GIN(y_4); @GIN(x_41); @GIN(x_42); @GIN(x_43); @GIN(x_44);
```

```
@SIN(Y_1); @SIN(X_11); @SIN(X_12); @SIN(X_13); @SIN(X_14);

END
```

输出

```
  Global optimal solution found.
  Objective value:                              21.50000
  Objective bound:                              21.50000
  Infeasibilities:                              0.000000
  Extended solver steps:                             428
  Total solver iterations:                        127221
  Elapsed runtime seconds:                          8.42

  Model Class:                                      PIQP

  Total variables:                    20
  Nonlinear variables:                20
  Integer variables:                  20

  Total constraints:                  21
  Nonlinear constraints:               4

  Total nonzeros:                     94
  Nonlinear nonzeros:                 16



                             Variable           Value        Reduced
Cost
                                  Y_1        14.00000
 -0.1000000
                                  Y_2        4.000000
 0.000000
                                  Y_3        1.000000
0.1000000
                                  Y_4        0.000000
0.2000000
                                 X_11        1.000000
 0.000000


                                 X_12        2.000000
```

```
 0.000000
                                X_13        0.000000
 0.000000
                                X_14        2.000000
 0.000000
                                X_21        0.000000
 0.000000
                                X_22        0.000000
 0.000000
                                X_23        5.000000
 0.000000
                                X_24        0.000000
 0.000000
                                X_31        2.000000
 0.000000
                                X_32        0.000000
 0.000000
                                X_33        1.000000
 0.000000
                                X_34        2.000000
 0.000000
                                X_41        2.000000
 0.000000
                                X_42        1.000000
 0.000000
                                X_43        0.000000
 0.000000
                                X_44        2.000000
 0.000000

                                 Row    Slack or Surplus      Dual
Price
                                   1        21.50000
-1.000000
                                   2        80.00000
 0.000000
                                   3        20.00000
 0.000000
                                   4        0.000000
 0.000000

                                   5        100.0000
```

```
    0.000000
51                                   6        90.00000
    0.000000
52                                   7        10.00000
    0.000000
53                                   8        55.00000
    0.000000
54                                   9        45.00000
    0.000000
55                                  10        1.000000
    0.000000
56                                  11        0.000000
    0.000000
57                                  12        0.000000
    0.000000
58                                  13        0.000000
    0.000000
59                                  14        10.00000
    0.000000
60                                  15        3.000000
    0.000000
61                                  16        1.000000
    0.000000
62                                  17        0.000000
    0.000000
63                                  18        0.000000
    0.000000
64                                  19        0.000000
    0.000000
65                                  20        0.000000
    0.000000
66                                  21        0.000000
   -1.200000
```

结果、分析与结论

最终求解的最优解为 21.5 个单位花销。使用了三种切割模式，分别切割出了四种钢管，其数量分别为 $1,2,0,2$；$0,0,5,0$；$2,0,1,2$。同时，使用这三种切割模式的原料钢管数量分别为 $14,4,1$。

参考下方表格：

参考下方表格：

序号	x_i	x_{i1}	x_{i2}	x_{i3}	x_{i4}
1	14	1	2	0	2
2	4	0	0	5	0
3	1	2	0	1	2
4	0	0	0	0	0

注意到 $y_4 = 0$，这是可以理解的。从建模过程中可以看出，由于使用频率较低的切割模式会线性增大切割成本，所以最终选择了三种切割模式，具有一定合理性。

这一模型的现实意义是显然的，我们求出了每种切割模式以及具体的使用。当然，实际的加工过程不可能做到加工无损耗，所以在现实生活中还需要添加更多限制条件。

最后，在对整数规划问题的优化上，存在多种可行的优化方案，可以都进行尝试，选取简洁容易实现却又行之有效的方法，比如我的处理过程选用了第二种新增的优化限制。