作业 Chapter 2

姓名：范潇　学号：2254298　日期：2024 年 3 月 24 日

1. (1.3)

本题的目标是让利润最大化，也就是要让收入和支出之间的差值最大化。

在本题中，收入的唯一来源是所生产的产品，而支出则分为两部分：原料费和设备费用。

在解决本题的过程中，我作以下假设：

1. 产品生产数为整数
2. 一件产品的一道工序只能由一台设备完成
3. 任一设备的总台时数不能超出该设备的有效台时数
4. 设备费用与台时数成正比

由于假设 1，本题需要进行整数线性规划。由于假设 3，还需要引入与台时数相关的约束。同时，还要确保生产的产品数与各个设备加工的产品数之间是匹配的。

我在 MATLAB 的实施编辑器中完成了该题的求解，具体代码和输出在下一页中给出。

从结果中可以看出，最大利润约为 1146 元，对应的产品数以及各个设备的加工数分别由数组 sol.A_usage,sol.B_usage,sol.product_num 给出。

```
prob = optimproblem("Description","Factory Problem","ObjectiveSense","maximize")
```

```
x = optimvar("product_num",3,"LowerBound", 0,"Type","integer")%各个产品的总数
A = optimvar("A_usage",2,3,"LowerBound",0,"Type","integer")%各个产品分配给各个设备的个数
B = optimvar("B_usage",3,3,"LowerBound",0,"Type","integer")
```

```
AT = [5 10 1e10;
    7 9 12]%设置惩罚项
BT = [6 8 1e10;
    4 1e10 11;
    7 1e10 1e10]
```

```
ATMAX =[6000 10000]%设备有效台时
BTMAX = [4000 7000 4000]
```

```
AC = [300 321]%满负荷时的设备费用
BC = [250 783 200]
```

```
c = [0.25 0.35 0.5]
price = [1.25 2 2.8]
net = price - c%单件产品的净利润（不考虑设备费用）
```

```
ACPH = AC ./ ATMAX%假设设备费用和台时成正比
BCPH = BC ./ BTMAX
```

```
ATused = A .* AT
BTused = B .* BT
AT_total = sum(ATused,2)'%总台时
BT_total = sum(BTused,2)'
AC_total = sum(AT_total.*ACPH)%总设备费用
BC_total = sum(BT_total.*BCPH)
prob.Objective = sum(net.* x') - AC_total - BC_total%最大化"净利润-设备费用"
```

```
%约束 1：不超过设备有效台时
prob.Constraints.ATime = AT_total <= ATMAX
prob.Constraints.BTime = BT_total <= BTMAX
```

```
%约束 2：分配给各工序的产品数与生产的产品数匹配
prob.Constraints.ABNum = sum(A) == sum(B)
prob.Constraints.num = sum(A) == x'
```

```
[sol,optival] = solve(prob)
```

```
将使用 intlinprog 求解问题。
LP:                Optimal objective value is 1146.474138.

Cut Generation:    Applied 2 Gomory cuts.
                   Upper bound is 1146.414200.
                   Relative gap is 0.00%.


找到最优解。

Intlinprog 在根节点处停止，因为目标值在最优值的间隙容差范围内，options.AbsoluteGapTolerance = 0。intcon 变量是容差范围内的
sol = 包含以下字段的 struct:
        A_usage: [2×3 double]
        B_usage: [3×3 double]
    product_num: [3×1 double]
optival = 1.1464e+03
```

```
sol.A_usage
```

```
ans = 2×3
10^3 ×
    1.2000         0         0
    0.2300    0.5000    0.3240
```

```
sol.B_usage
```

```
ans = 3×3
         0  500.0000         0
  859.0000         0  324.0000
  571.0000         0         0
```

```
sol.product_num
```

```
ans = 3×1
10^3 ×
    1.4300
    0.5000
    0.3240
```

```
optival
```

```
optival = 1.1464e+03
```

2. (1.4)

本题的目标是让利润最大化。

题中的假设 1 告诉我们货物可以任意分配，只需确保各类货物的总量不要超过表 1.4 中给出的总数即可。因此本题不需要进行整数规划。

本题中的限制有:

1. 运输的各类货物重量不能超过该类货物的总量
2. 货舱中的货物总重量不能超过重量限制
3. 货舱中的货物总体积不能超过体积限制
4. 三个货舱装载的货物重量必须与其最大的容许量成正比

在用方程描述约束 4 时，为了将该问题限制在 linear programming 的范围内，应该将比值相等转化为乘积相等。

我在 MATLAB 的实施编辑器中完成了该题的求解，具体代码和输出在下一页中给出。

从结果中可以看出，最大利润约为 121520 元，各类货物的运输数以及分配给各个货舱的情况由数组 sol.item_weight 和 sol.weight_allocation 给出。

```
vlim = [6800 8700 5300]
wlim = [10 16 8]
w = [18 15 23 12]
v = [480 650 580 390]
p = [3100 3800 3500 2850]
```

```
prob = optimproblem("Description","Plane Transimission
Problem","ObjectiveSense","maximize")
```

```
x = optimvar("item_weight",4,"LowerBound",0,"UpperBound",w)
```

```
prob.Objective = sum(x'.*p)
```

```
A = optimvar("weight_allocation",4,3,"LowerBound",0)%4 种货物分配给三个位置的重量
prob.Constraints.Weight = x == sum(A,2)%重量要一致
```

```
prob.Constraints.plane_weight = sum(A)<= wlim%各仓的总重量不能超出限制
```

```
prob.Constraints.plane_volumn = sum(A.*([v' v' v']))<=vlim%各仓的总体积不能超出限制
```

```
total_weight = sum(A)
prob.Constraints.proportion1 = wlim(1)*total_weight(2)== total_weight(1)*wlim(2)
prob.Constraints.proprotion2 = wlim(2)*total_weight(3) == total_weight(2)*wlim(3)
```

```
[sol optval] = solve(prob)
```

```
optval
```

```
optval = 1.2152e+05
```

```
sol.item_weight
```

```
ans = 4×1
        0
   15.0000
   15.9474
    3.0526
```

```
sol.weight_allocation
```

```
ans = 4×3
        0         0         0
    7.0000         0    8.0000
    3.0000   12.9474         0
```

```
         0    3.0526         0
```