

作业 Chapter 3

姓名：范潇 学号：2254298 日期：2024 年 3 月 25 日

1. (3.1)

本题的目标是最小化费用。费用由两部分组成，生产费用和存储费用。

本题的约束由两部分组成：要满足需求以及受生产能力的约束。其中，满足需求对应的方程并不是三个季度的生产数量分别大于 40, 60, 80，而是到该季度的累计生产量大于到该季度的累计需求，因为可以超额生产并存储。

我在 MATLAB 的实时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，最低花费为 11280 元，此时三个季度生产的台数分别为 50, 60, 70。

```
prob = optimproblem("Description","Factory","ObjectiveSense","minimize")%最小化费用
min = [40 60 80]
x = optimvar("num",3,"UpperBound",100)%每季度生产的台数
c = 50*x + 0.2*x.*x%季度的生产费用
```

```
prob.Constraints.first = x(1)>=40
prob.Constraints.second = x(1)+x(2)>=100
prob.Constraints.third = x(1)+x(2)+x(3)>=180
storage_cost = (x(1)-40 + x(2)+x(1) - 100 ) *4
```

```
prob.Objective = storage_cost + sum(c)
```

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

将使用 quadprog 求解问题。

找到满足约束的最小值。

优化已完成，因为目标函数沿
可行方向在最优性容差值的范围内呈现非递减，
并且在约束容差值范围内满足约束。

<停止条件详细信息>

```
sol = 包含以下字段的 struct:
    num: [3×1 double]
optval = 1.1280e+04
```

```
sol.num
```

```
ans = 3×1
    50.0000
    60.0000
    70.0000
```

2. (3.2)

我在 MATLAB 的实时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。

```
prob = optimproblem("Description","plane","ObjectiveSense","minimize")
```

```
delta_theta = optimvar("delta_theta",6,"LowerBound",-pi/6,"UpperBound",pi/6)
```

```
x0 = [150 85 150 145 130 0]  
y0 = [140 85 155 50 150 0]  
theta0 = [243 236 220.5 159 230 52]
```

```
theta = theta0 + delta_theta'
```

```
v = 800%飞机飞行速度
```

```
thetas = repmat(theta,6,1)  
x0s = repmat(x0,6,1)  
y0s = repmat(y0,6,1)
```

```
a = 4*v*v*(sin((thetas-thetas')./2)).*(sin((thetas-thetas')/2))
```

```
b = 2*v*((x0s-x0s').*(cos(thetas)-cos(thetas'))+(y0s-y0s').*(sin(thetas)-  
sin(thetas')))
```

```
c = (x0s-x0s').*(x0s-x0s') + (y0s-y0s').*(y0s-y0s')- 64
```

```
Delta = b.*b-4*a.*c - diag(ones(6,1))*1e10  
%关于对角线对称，为了确保对角线也满足不等式约束，所以减去一个对角阵
```

```
prob.Constraints.delta = Delta <=0
```

```
prob.Objective = sum(delta_theta.*delta_theta)
```

```
prob =  
  OptimizationProblem - 属性:  
  
    Description: 'plane'  
  ObjectiveSense: 'minimize'  
    Variables: [1x1 struct] containing 1 OptimizationVariable  
    Objective: [1x1 OptimizationExpression]  
    Constraints: [1x1 struct] containing 1 OptimizationConstraint
```

请使用 show 查看问题表示。

```
guess.delta_theta = [0 0 0 0 0 0]
```

```
guess = 包含以下字段的 struct:  
    delta_theta: [0 0 0 0 0 0]
```

```
[sol optimval]=solve(prob,guess)
```

将使用 fmincon 求解问题。

警告：矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND = 2.946843e-17。

警告：矩阵接近奇异值，或者缩放不良。结果可能不准确。RCOND = 2.946843e-17。

找到目标函数值更低的可行点。

可能存在局部最小值。满足约束。

fmincon 已停止，因为当前步长小于
步长容差值并且在约束容差值范围内满足约束。

<停止条件详细信息>

sol = 包含以下字段的 struct:

delta_theta: [6×1 double]

optimval = 0.3968

```
sol.delta_theta
```

```
ans = 6×1  
-0.1982  
-0.1861  
0.1155  
-0.4772  
-0.0804  
0.2744
```

3. (3.3)

我在 MATLAB 的实时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。

由于 `max` 函数无法直接出现在 `optimexpr` 中，我使用 `fcn2optimexpr` 将该功能转化为可以使用的 `optimexpr`。同时，在调试的过程中，我发现如果惩罚项过大，则 MATLAB 无法进行求解，当我把惩罚项从 $1e20$ 调整为 $1e5$ 后便能正常求解。

```
prob = optimproblem("Description","plane","ObjectiveSense","minimize")
```

```
delta_theta = optimvar("delta_theta",6)
```

```
x0 = [150 85 150 145 130 0]  
y0 = [140 85 155 50 150 0]  
theta0 = [243 236 220.5 159 230 52]
```

```
theta = theta0 + delta_theta'
```

```
v = 800%飞机飞行速度
```

```
thetas = repmat(theta,6,1)  
x0s = repmat(x0,6,1)  
y0s = repmat(y0,6,1)
```

```
a = 4*v*v*(sin((thetas-thetas')./2)).*(sin((thetas-thetas')/2))
```

```
b = 2*v*((x0s-x0s').*(cos(thetas)-cos(thetas'))+(y0s-y0s').*(sin(thetas)-  
sin(thetas')))
```

```
c = (x0s-x0s').*(x0s-x0s') + (y0s-y0s').*(y0s-y0s')- 64
```

```
Delta = b.*b-4*a.*c - diag(ones(6,1))*1e10  
%关于对角线对称，为了确保对角线也满足不等式约束，所以减去一个对角阵
```

```
mymax = @(x)sum(sum((x>=0).*x))
```

```
prob.Objective = sum(delta_theta.*delta_theta)+...  
1e5*fcn2optimexpr(mymax,Delta)+1e5*fcn2optimexpr(mymax,delta_theta-pi/6)+...  
1e5*fcn2optimexpr(mymax,-pi/6-delta_theta)%惩罚项不宜过大
```

```
prob =  
  OptimizationProblem - 属性:  
  
    Description: 'plane'  
  ObjectiveSense: 'minimize'  
    Variables: [1x1 struct] containing 1 OptimizationVariable  
    Objective: [1x1 OptimizationExpression]  
   Constraints: [0x0 struct] containing 0 OptimizationConstraints
```

请使用 `show` 查看问题表示。

```
guess.delta_theta = [0 0 0 0 0 0]
```

```
guess = 包含以下字段的 struct:  
    delta_theta: [0 0 0 0 0 0]
```

```
[sol optimval]=solve(prob,guess)
```

将使用 fminunc 求解问题。

找到局部最小值。

优化已完成，因为梯度大小小于
最优性容差的值。

<停止条件详细信息>

```
sol = 包含以下字段的 struct:  
    delta_theta: [6x1 double]  
optimval = 0.1982
```

```
sol.delta_theta
```

```
ans = 6x1  
    -0.1475  
         0  
     0.2936  
    -0.0412  
     0.2861  
         0
```


4. (3.4)

我在 MATLAB 的实时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。
程序给出的最大值为 18.0833。

```
prob = optimproblem("Description","function","ObjectiveSense","max")
x1 = optimvar("x1")
x2 = optimvar("x2")
x3 = optimvar("x3")
```

```
prob.Objective = 2*x1+3*x1*x1 + 3*x2+ x2*x2 + x3
```

```
prob.Constraints.inq1 = x1 + 2*x1*x1 + x2 + 2*x2*x2 + x3 <= 10
prob.Constraints.inq2 = x1 + x1*x1 + x2 + x2*x2 - x3 <= 50
prob.Constraints.inq3 = 2*x1 + x1*x1 + 2* x2 + x3 <= 40
prob.Constraints.enq1 = x1*x1 + x3 == 2
prob.Constraints.inq4 = x1 + 2*x2*x2 >= 1
prob.Constraints.inq5 = x1>=0
```

```
initialGuess.x1 = 1
```

```
initialGuess = 包含以下字段的 struct:
    x1: 1
    x2: 10
    x3: 1
```

```
initialGuess.x2 = 10
```

```
initialGuess = 包含以下字段的 struct:
    x1: 1
    x2: 10
    x3: 1
```

```
initialGuess.x3 = 1
```

```
initialGuess = 包含以下字段的 struct:
    x1: 1
    x2: 10
    x3: 1
```

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

将使用 fmincon 求解问题。

找到满足约束的局部最小值。

优化已完成，因为目标函数沿可行方向在最优性容差值范围内呈现非递减，并且在约束容差值范围内满足约束。

<停止条件详细信息>

```
sol = 包含以下字段的 struct:
    x1: 2.3333
    x2: 0.1667
    x3: -3.4444
optval = 18.0833
```