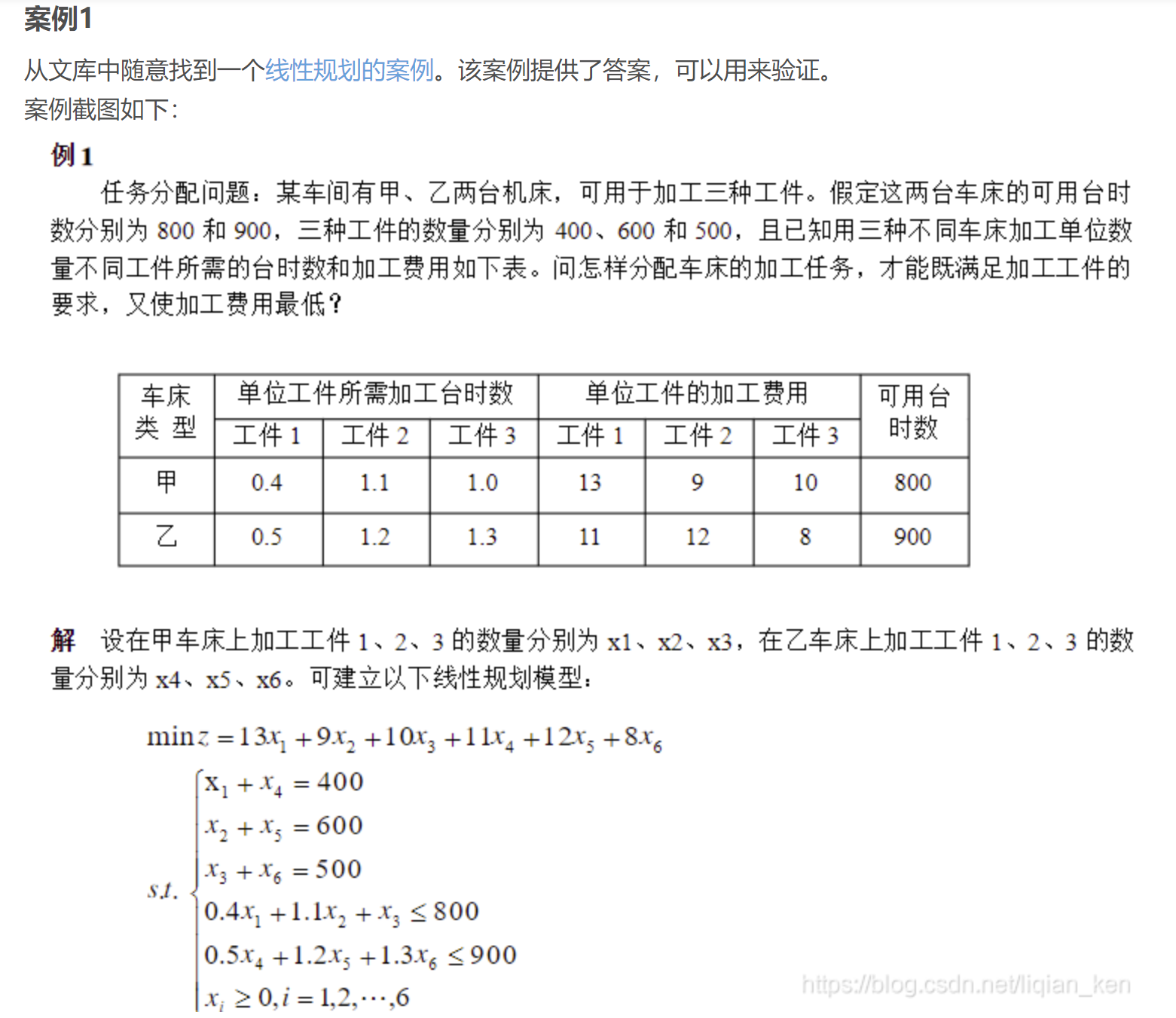
线性规划

例题：



import cvxpy as cp  
import numpy as np  
  
# ...把前面三个约束、向量c的代码加上  
# 约束1（相等约束）  
A1 = np.array([[1, 0, 0, 1, 0, 0],  
 [0, 1, 0, 0, 1, 0],  
 [0, 0, 1, 0, 0, 1],  
 ])  
b1 = np.array([400, 600, 500])  
  
# 约束2(小于等于约束)  
A2 = np.array([[0.4, 1.1, 1, 0, 0, 0],  
 [0, 0, 0, 0.5, 1.2, 1.3],  
 ])  
b2 = np.array([800, 900])  
  
# 约束3（大于等于约束）  
A3 = np.array([[1, 0, 0, 0, 0, 0],  
 [0, 1, 0, 0, 0, 0],  
 [0, 0, 1, 0, 0, 0],  
 [0, 0, 0, 1, 0, 0],  
 [0, 0, 0, 0, 1, 0],  
 [0, 0, 0, 0, 0, 1],  
 ])  
b3 = np.array([0, 0, 0, 0, 0, 0])  
  
# 用于目标函数  
c = np.array([13, 9, 10, 11, 12, 8])  
  
# 定义自变量  
n = 6 # 有6个自变量：x1,x2,x3,x4,x5,x6  
x = cp.Variable(n)  
  
# 定义问题，把三个约束条件添加上  
prob = cp.Problem(cp.Minimize(c.T @ x),  
 [A1 @ x == b1, A2 @ x <= b2, A3 @ x >= b3])  
  
# 解决问题  
prob.solve()  
  
# 输出结果  
print("\n目标函数的最小值", prob.value)  
print("所求的x矩阵")  
print(x.value)  
# 对x向量各元素取整数后再输出  
for item in x.value:  
 print(round(item))

