**重 庆 大 学**

**学 生 实 验 报 告**

**实验课程名称 数学实验**

**开课实验室 数学实验教学实验中心**

**学 院 计算机学院 年级2022 专业班 信息安全02**

**学 生 姓 名 姚凡 学 号 20221879**

**开 课 时 间 2023 至 2024 学年第 二 学期**

|  |  |
| --- | --- |
| **总 成 绩** |  |
| **教师签名** |  |

**数 学 与 统 计 学 院 制**

**开课学院、实验室：数学与统计学院、数学实验教学实验中心 实验时间 ： 2024年 4月13日**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课程**  **名称** | 数学实验 | **实验项目**  **名 称** | 数学规划实验 | **实验项目类型** | | | | |
| **验证** | **演示** | **综合** | **设计** | **其他** |
| **指导**  **教师** | 龚劬 | **成 绩** |  |  |  |  |  | **√** |
| 实验目的  学会运用Matlab和lingo解决数学规划问题。  基础实验1  问题重述  某车间有三台机床甲、乙、丙，可用于加工四种工件。设机床甲、乙和丙加工工件j（j=1,2,3,4）的加工费用分别为a1j、a2j和a3j，机床甲、乙和丙加工工件j（j=1,2,3,4）所需的加工台时数分别为b1j、b2j和b3j，机床甲、乙和丙的可用台时数分别为B1,B2和B3，工件j（j=1,2,3,4）的数量为Cj，问怎样分配机床的加工任务，才能既满足加工工件的要求，又使总加工费用最低？  （1）试建立求解该问题的数学模型;  （2）设A=[aij]3×4=[13,9,10,8;11,12,8,6;15,11,13,5];  B=[bij]3×4=[0.4,1.1,1,1.2;0.5,1.2,1.3,1.4;0.3,1,0.9,1.1]。 B1,B2和B3分别为600，700，800。Cj（j=1,2,3,4）分别为200，300，500，400。编写求解上述数学模型的MATLAB程序或Lingo程序。  实验过程  % （1）  **决策变量:**表示机床i加工工件j的数量。i=1、2、3分别指甲、乙、丙机床。  **目标函数:**最小化总加工费用。    **约束条件：**  **1.加工量满足需求:**每种工件的总加工数量需满足其需求量。  **2.机床可用时间不超限:**每台机床的使用总时间不得超过其可用时间。     1. **非负约束:**加工数量不能为负。   % （2）  A = [13, 9, 10, 8; 11, 12, 8, 6; 15, 11, 13, 5];  B = [0.4, 1.1, 1, 1.2; 0.5, 1.2, 1.3, 1.4; 0.3, 1, 0.9, 1.1];  B\_available = [600, 700, 800];  C = [200, 300, 500, 400];  % 构建线性规划的目标函数向量  f = A(:);  % 构建等式约束矩阵和向量  Aeq = zeros(4, 12);  for j = 1:4  Aeq(j, j:4:end) = 1;  end  beq = C';  % 构建不等式约束矩阵和向量  Aineq = zeros(3, 12);  for i = 1:3  Aineq(i, (i-1)\*4+1:i\*4) = B(i, :);  end  bineq = B\_available';  % 定义变量的下界  lb = zeros(12, 1);  % 调用线性规划求解器  options = optimoptions('linprog', 'Algorithm', 'dual-simplex');  [x, fval] = linprog(f, Aineq, bineq, Aeq, beq, lb, [], options);  % 显示结果  disp('Total Cost:');  disp(fval);  disp('Distribution of tasks (x\_ij):');  disp(reshape(x, 4, 3)');  实验结果及分析  找到最优解。  Total Cost:  1.0864e+04  Distribution of tasks (x\_ij):  0 0 0 0  200.0000 0 0 354.5455  0 300.0000 500.0000 45.4545  基础实验2  问题重述  一家小型汽车租赁公司有101辆汽车供出租，分布在10个代理点。每个代理点的位置坐标(xi,yi)已知，单位为千米。假设两代理点之间的距离约为它们之间的欧氏距离的1.3倍。下表给出了10个代理点的坐标，以及第二天早晨汽车租赁的需求量和前一天晚上各个代理点拥有的汽车数。   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 代理点 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | X坐标 | 0 | 20 | 18 | 30 | 35 | 33 | 5 | 5 | 11 | 2 | | Y坐标 | 0 | 20 | 10 | 12 | 0 | 25 | 27 | 10 | 0 | 15 | | 需求量 | 10 | 6 | 8 | 11 | 9 | 7 | 15 | 7 | 9 | 12 | | 拥有量 | 9 | 14 | 5 | 9 | 13 | 3 | 15 | 11 | 15 | 7 |   如何在各个代理点之间调度分配汽车才能满足各处的需求，并使总里程数最小。  （1）试建立数学模型（用公式编辑器输入公式）；  （2）给出相应的MATLAB程序或Lingo程序。  实验过程  % （1）  决策变量:表示从代理点i到代理点j运送的汽车数量。  目标函数:最小化总运输里程。  是代理点i到代理点j的欧氏距离。  约束条件：  1.每个代理点的汽车需求需被满足,  其中，是代理点j的需求量，是代理点j前一晚的汽车数量。  2.调动的汽车数量不能为负。  % (2)  x\_coords = [0, 20, 18, 30, 35, 33, 5, 5, 11, 2];  y\_coords = [0, 20, 10, 12, 0, 25, 27, 10, 0, 15];  demand = [10, 6, 8, 11, 9, 7, 15, 7, 9, 12];  supply = [9, 14, 5, 9, 13, 3, 15, 11, 15, 7];  % 计算距离矩阵  numLocations = length(x\_coords);  distance = zeros(numLocations, numLocations);  for i = 1:numLocations  for j = 1:numLocations  distance(i, j) = sqrt((x\_coords(i) - x\_coords(j))^2 + (y\_coords(i) - y\_coords(j))^2);  end  end  distance = 1.3 \* distance;  f = distance(:);  Aeq = zeros(numLocations, numLocations^2);  for j = 1:numLocations  Aeq(j, j:numLocations:end) = 1; % 接收量  Aeq(j, (j-1)\*numLocations+1:j\*numLocations) = -1; % 发送量  end  beq = demand - supply;  lb = zeros(numLocations^2, 1);  options = optimoptions('linprog', 'Algorithm', 'dual-simplex');  [x, fval] = linprog(f, [], [], Aeq, beq, lb, [], options);  disp('Total Minimum Distance:');  disp(fval);  disp('Car distribution matrix:');  disp(reshape(x, numLocations, numLocations));  实验结果及分析  找到最优解。  Total Minimum Distance:  213.1098  Car distribution matrix:  列 1 至 6  0 0 0 0 0 0  0 0 0 0 0 0  0 3 0 0 0 0  0 1 0 0 1 0  0 0 0 0 3 0  0 4 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  列 7 至 10  0 0 1 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 1 0  0 0 4 0  0 5 0 0  基础实验3  问题重述  **求解无约束优化**  1) 画出该曲面图形, 直观地判断该函数的最优解;  2) 使用fminunc或fminsearch命令求解, 能否求到全局最优解?  实验过程  %(1)作图  x1=-5:0.05:5;  x2=-5:0.05:5;  [X,Y]=meshgrid(x1,x2);  Z=-20\*exp(-0.2\*sqrt(0.5\*(X.^2+Y.^2)))-exp(0.5\*(cos(2\*pi\*X)+cos(2\*pi\*Y)))+22.713;  mesh(X,Y,Z)  %(2)  f=@(x)(-20\*exp(-0.2\*sqrt(0.5\*(x(1)^2+x(2)^2)))-exp(0.5\*(cos(2\*pi\*x(1))+cos(2\*pi\*x(2))))+22.713);  x0=[0.5,0.5];%初始点  [x,fval]=fminunc(f,x0)  实验结果及分析（一般应包括调试情况记录、图表等， 实验结果及分析）  (1)作图观察在(0,0)是,f(x1,x2)最小,最小值为-0.00528183    图1 曲面图形示意图  (2)通过fminunc 计算出的结果与之前画图找出的结果一致  程序输出结果为：  x =  1.0e-15 \*  -0.2220 -0.2220  fval =  -0.0053  基础实验4  问题重述  **求解非线性规划,**  **试判定你所求到的解是否是最优?**  用MATLAB的fmincon或LINGO软件求解。  实验过程  options = optimset;  [x,y] = fmincon('fun1',rand(3,1),[],[],[],[],zeros(3,1),[36,5,125],'fun2',options)  %创建fun1.m文件  function f = fun1(x)  f=-0.201.\*x(1).^4.\*x(2).\*x(3)./10.^7;  End  %创建fun2.m文件  function [g,h]= fun2(x)  g = [-675+x(1).^2.\*x(2)  -0.419+x(1).^2.\*x(3).^2./10^7];  h = [];  end  实验结果及分析  x =  0.9134  0.6324  0.0975  y =  -8.6286e-10  应用实验  一、问题重述  **警力调度方案**  某重大刑事案件，需要调度32个派出所的警力，对15条交通要道快速全封锁。一个派出所的警力最多封锁一个路口，请给出警力合理的调度方案（派出所到交通要道的距离可以用[5,50]区间的随机整数表示）。  二、问题分析  为了建立一个合理的警力调度方案，我们可以将这个问题视作一个经典的分配问题，其中的目标是最小化警力从派出所到指定路口的总距离,利用线性规划解决该问题。  三、数学模型的建立与求解  **决策变量:**表示从派出所i到交通要道j是否调度警力，其中为0或1。  **目标函数:**最小化总调度距离  其中，是派出所i到交通要道j的距离。  **约束条件：**  **1.封锁约束:**每个路口只能由一个派出所封锁。  **2.派出所调度限制:**一个派出所最多只能封锁一个路口。  **3.二进制约束：**  四、实验结果及分析  Total Minimum Distance:  93  Police station to road assignment matrix:  列 1 至 6  0 0 1 0 0 0  0 0 0 0 0 0  1 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 1  0 1 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 1 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 1 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  列 7 至 12  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 1 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 0 1 0 0 0  0 0 0 0 0 0  0 0 0 0 0 1  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  0 1 0 0 0 0  1 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 1 0  0 0 0 0 0 0  0 0 0 0 0 0  列 13 至 15  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  1 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 0  0 1 0  0 0 0  0 0 0  0 0 0  0 0 0  0 0 1  0 0 0  五、附录（程序等）  %可重复性  rng(1);  % 派出所和交通要道  numPoliceStations = 32;  numRoads = 15;  % 生成距离矩阵  D = randi([5, 50], numPoliceStations, numRoads);  %目标函数  f = D(:); % 使用(:)将矩阵转换成向量  % 等式约束矩阵  Aeq = zeros(numRoads, numPoliceStations \* numRoads);  for j = 1:numRoads  Aeq(j, (j-1)\*numPoliceStations+1:j\*numPoliceStations) = 1;  end  beq = ones(numRoads, 1);  % 不等式约束矩阵  Aineq = zeros(numPoliceStations, numPoliceStations \* numRoads);  for i = 1:numPoliceStations  Aineq(i, i:numPoliceStations:end) = 1;  end  bineq = ones(numPoliceStations, 1);  % 上下界  lb = zeros(numPoliceStations \* numRoads, 1);  ub = ones(numPoliceStations \* numRoads, 1);  options = optimoptions('intlinprog', 'Display', 'off');  [x, fval] = intlinprog(f, 1:numPoliceStations\*numRoads, Aineq, bineq, Aeq, beq, lb, ub, options);  disp('Total Minimum Distance:');  disp(fval);  disp('Police station to road assignment matrix:');  X = reshape(x, numPoliceStations, numRoads);  disp(X);  教师签名  年 月 日 | | | | | | | | |