**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 DS1407**

**学 院 计算机学院 年级 2022级 专业班 06班**

**学 生 姓 名 楼洋 学 号 20221627**

**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： 计算机学院 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  [1] 掌握数学规划的基本概念和理论，包括线性规划、非线性规划和整数规划等；  [2] 学习并应用MATLAB中的优化工具箱，包括`linprog`、`fmincon`、`intlinprog`等函数来求解数学规划问题；  [3] 通过实例深入理解数学规划模型的建立和求解过程，并能够对模型结果进行分析和解释；  [4]培养使用MATLAB进行算法编程和数据可视化的能力，以便在实际问题中应用数学规划方法。  通过本实验的学习，使学生能够熟练运用MATLAB软件解决数学规划问题，掌握从模型建立到求解再到结果分析的全过程。这将有助于学生在未来的学习和工作中，更好地运用数学规划方法解决实际问题，提高解决复杂工程问题的能力。  基础实验1  问题重述  1．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  这是一个典型的线性规划问题，可以通过建立数学模型来解决。我们需要最小化总加工费用，同时满足机床的可用台时数和工件的数量要求。   1. 数学模型建立如下：   目标函数（最小化总加工费用）:  约束条件（满足台时数和工件数量要求）:  其中，( ) 表示机床i加工工件j的数量。  （2）代码如下：  % 定义加工费用矩阵A和加工台时数矩阵B  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];  % 定义可用台时数和工件数量  B1 = 600; B2 = 700; B3 = 800;  C = [200, 300, 500, 400];  % 定义线性规划的目标函数系数  f = [A(1,:),A(2,:),A(3,:)];  % 定义线性规划的等式约束  Aeq = [1,0,0,0,1,0,0,0,1,0,0,0;0,1,0,0,0,1,0,0,0,1,0,0;0,0,1,0,0,0,1,0,0,0,1,0;0,0,0,1,0,0,0,1,0,0,0,1];  beq = C;  % 定义线性规划的不等式约束  Aineq = [B(1,:), zeros(1,8); zeros(1,4), B(2,:), zeros(1,4); zeros(1,8), B(3,:)];  bineq = [B1; B2; B3];  % 定义变量的下界  lb = zeros(12,1);  INTCON = 1:12;  % 调用intlinprog函数求解  [x, fval] = intlinprog(f,INTCON ,Aineq, bineq, Aeq, beq, lb);  % 输出结果  X = reshape(x, 3, 4);  disp('分配方案为：');  disp(X);  disp(['总加工费用为：', num2str(fval)]);  实验结果及分析    图一:实验一运行结果  分析：  通过优化工具箱中的intlinprog函数解决线性规划问题  基础实验2  问题重述  2.一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  这个问题可以通过建立一个最小化总运输距离的数学模型来解决，同时满足每个代理点的汽车需求量。我们可以使用线性规划来建立这个模型。  （1）数学模型建立如下：  目标函数（最小化总运输距离）:  其中，() 是代理点i到代理点j的欧氏距离，() 是从代理点i运输到代理点j的汽车数量。  约束条件（满足每个代理点的需求量和供应量）:  是代理点i的拥有量，是代理点j的需求量  （2）代码如下：  % 代理点坐标  X = [0, 20, 18, 30, 35, 33, 5, 5, 11, 2];  Y = [0, 20, 10, 12, 0, 25, 27, 10, 0, 15];  % 需求量和拥有量  demand = [10, 6, 8, 11, 9, 7, 15, 7, 9, 12];  supply = [9, 14, 5, 9, 13, 3, 15, 11, 15, 7];  % 计算代理点之间的距离  distances = zeros(10, 10);  for i = 1:10  for j = 1:10  distances(i, j) = 1.3 \* sqrt((X(i) - X(j))^2 + (Y(i) - Y(j))^2);  end  end  % 决策变量  x = optimvar('x', 10, 10, 'LowerBound', 0, 'Type', 'integer');  % 目标函数  objective = sum(sum(distances .\* x));  % 创建优化问题  prob = optimproblem('Objective', objective);  % 添加约束条件  for i = 1:10  % 为每个代理点的供应创建单独的约束  supplyConstraint = sum(x(i, :)) <= supply(i);  prob.Constraints.("supply" + i) = supplyConstraint;  end  for j = 1:10  % 为每个代理点的需求创建单独的约束  demandConstraint = sum(x(:, j)) >= demand(j);  prob.Constraints.("demand" + j) = demandConstraint;  end  % 求解  [sol, fval, exitflag, output] = solve(prob);  % 输出结果  if exitflag == 1  % 解决方案  solution = sol.x;  fprintf('总运输距离为：%f 千米\n', fval);  for i = 1:10  for j = 1:10  if solution(i, j) > 0 && i~=j  fprintf('从代理点 %d 到代理点 %d 运输 %d 辆汽车\n', i, j, solution(i, j));  end  end  end  else  disp('没有找到解决方案');  end  实验结果及分析  实验结果：  图二：实验二运行结果  分析：  由于直接使用linprog函数求解线性规划问题涉及较多决策变量，这里我采用优化工具箱中的optimvar、optimproblem和solve函数求解，求解结果如图二所示。  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  % 画图  [x,y]=meshgrid(-5:0.1:5);  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);  % 调用fminunc函数求解，初始值[3,3]  [x,fval]=fminunc(@(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,[3,3]);  fprintf("使用fminunc的结果: x:%.1f,y:%.1f,z:%.6f", x(1),x(2),fval);  % 调用fminunc函数求解, 初始值[10,10]  [x,fval]=fminunc(@(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,[10,10]);  fprintf("使用fminunc的结果: x:%.1f,y:%.1f,z:%.6f", x(1),x(2),fval);  实验结果及分析  实验结果：  图三：实验三运行结果  分析：   1. 通过mesh函数画出图形如图三，从图形直观可得在大约(0,0)处得到最小值 2. 通过fminunc可以得到局部最优解，当迭代初始值接近全局最优解可以得到全局最优解，而当迭代初始值当较远时，只能得到局部最优解。   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  function [c,ceq] = nonlcon(x)  c= [x(1)^2\*x(2)-675,x(1)^2\*x(3)^2/(10^7)-0.419];  ceq= [];  end  [x,fval,exitflag,output]=fmincon(@(x) -0.201\*x(1)^4\*x(2)\*x(3)^2,[0,0,0],[],[],[],[],[0,0,0],[36,5,125],'nonlcon')  % 检查输出参数  if exitflag == 1  disp('找到一个解，且满足所有约束。');  elseif exitflag == 0  disp('达到函数评估次数限制。');  else  disp('优化过程未成功。');  end  实验结果及分析  实验结果：    图四：实验四运行结果  分析：  根据输出结果如图四，根据exitflag==1和output中迭代信息可以知道该解是最优解  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  这个问题可以通过建立一个0-1整数规划模型来解决。在这个模型中，我们将派出所的警力调度到交通要道的决策变量定义为0或1，其中1表示某个派出所的警力被分配到特定的交通要道，而0则表示没有分配。由于每个派出所的警力最多封锁一个路口，这就形成了我们模型中的约束条件。  三、数学模型的建立与求解  （1）数学模型建立如下：  目标函数（最小化总运输距离）:  其中，() 是派出所i到交通要道j的距离，() 是是否从派出所i派出警力到交通要道j。  约束条件:  （2）建完模型后，通过matlab中优化工具箱的optimvar、optimproblem和solve函数求解  四、实验结果及分析  实验结果：    图五：实验五运行结果  分析：  由于每个派出所到每条路的距离是一个随机数，所以每次生成的结果不一定一样，参考结果如图五，可以求出每次对应的最优解，并给出对应派出所封锁的路口。  五、附录（程序等）  代码如下：  % 参数设置  num\_police\_stations = 32; % 派出所数量  num\_roads = 15; % 交通要道数量  % 随机生成每个派出所到每条路的距离  distances = randi([5, 50], num\_police\_stations, num\_roads);  % 决策变量  x = optimvar('x', num\_police\_stations, num\_roads, 'Type', 'integer', 'LowerBound', 0, 'UpperBound', 1);  % 目标函数：最小化总距离  objective = sum(sum(distances .\* x));  % 创建优化问题  prob = optimproblem('Objective', objective);  % 添加约束条件  % 每个派出所最多封锁一个路口  for i = 1:num\_police\_stations  prob.Constraints.("station" + i) = sum(x(i, :)) <= 1;  end  % 每条路必须被封锁  for j = 1:num\_roads  prob.Constraints.("road" + j) = sum(x(:, j)) == 1;  end  % 求解  [sol, fval, exitflag, output] =solve(prob)  % 输出结果  if exitflag > 0  % 解决方案  solution = sol.x;  fprintf('总运输距离为：%f 千米\n', fval);  for i = 1:num\_police\_stations  for j = 1:num\_roads  if solution(i, j) == 1  fprintf('派出所 %d 封锁路口 %d\n', i, j);  end  end  end  else  disp('没有找到解决方案');  end  教师签名  年 月 日 | | | | | | | | |