LDO

Topic1

1.1 Difference between LP,IP,BIP,MIP

LP: all decision value are any value except negative value

IP: all decision value are non-negative Integer

BIP: all decision value only could be 0 or 1;

MIP: some decision value could be IP(include BIP), and some decision value could be any non-negative value

Topic2

2.1 functional constraints and non-negative constraints

Functional constraints: according to the problem and list the constraints (simplify: the number of constraints except the non-negative constraints)

Non-negative constraints: some constraints ask the decision value should be non-negative value, it calls the non-negative constraints(when calculate the number of this type constraints, we should count each decision value that presented >=0 because sometimes they will be showed in the one line.)

2.2 binding constraint

Definition: the constraint will be the equal formula at the optimal solution

For example: constraint: X1+X2<=10 optimal solution is X1=4, X2=6; so the X1+X2=10, it means this constraint is a binding constraint.

2.3 size of LP model

Consists: the number of decision variables and constraints+ the size of search space(the feasible region; It means the description of the set of possible solutions)

2.4 general LP formulation

Divisibility: the value of each decision variable includes integer and non-integer

Certainty: the data or parameters given in the model are known and certain, no random variables in the model.

Proportionality: each increasement or decrement of the value of the decision variable will have the effect on the objective function or constraints. (it can’t exist the different objective function because of the change of the value of one decision variable ; at the same time, the component of one decision variable should be same in the formulation)

Additivity: the increasement and decrement of one decision should be independent and shouldn't be affected by other decision variables. (e.g. shouldn't exist multiplying two decision variables like 2X1X2)

2.5 Graphical Method (use the website: http://www.phpsimplex.com/simplex/simplex.htm?l=en)

Step1: define the border of each decision variable

Step2: draw the constraints

Step3: define the infeasible and feasible region

Step4: draw the objective function in the feasible region

Step5: find the line to make the value of the objective function be greater or smaller.

Step6: try to use CPF(corner point feasible)solution as the optimal solution

Step7: use the optimal solution to verify the correction.

Topic3

3.1 Difference between IP and LP model

LP -> IP Model: restrict all decision variables to be integer value only.

LP have more search space than IP model, because there is less restriction of decision of LP model than IP model.

The optimal solution of LP model usually in the CPF(if it exists), but for IP model it usually isn't at the border of the feasible region.

3.2 IP solve

B&B: branch and bound method

Relaxed solution: Satisfy all functional constraints but not all non-negative constraints
Then, use B&B method to find optimal integer solution

3.2.1 B&B Method (build and explore the search tree(search tree include all feasible and infeasible solutions)

Step1: list all possibilities of one decision variable

Step2: use possible values of this decision variable to calculate the possible value of the other decision variable though solving these constraints.

Step3: make the second decision variable(need to calculate by the first decision variable) as the root of the search tree. Then use the first decision variable as the nodes of the tree. Use lines and mark the value to link these nodes.

Step4: It means list all possible solution of the model and identify feasible and infeasible solutions, then find the only optimal solution of the model.

3.2.2 LPR(LP Relaxation) for IP model

Definition: allow decision variable be non-integer or fractional value of IP

Situation 1: after LPR, the decision variables are integer, so it could be the optimal solution for this IP model

Situation2(usually): (can't find suitable optimal solution after LPR).
Step1: find the optimal solution of LPR, then up or down the value of one decision variable to the integer, then calculate the value of objective function, also can change the value of the objective function of LPR's optimal solution.

Step2: Rounding the LP optimal solution(find the CPF of LP model, then find the optimal solution of LP model, then change these value of the solution to Integer (e.g. make 441.176->441), so that can find a nearest IP optimal solution). May not effective

Notice: CPF maybe the optimal solution for IP, but it isn't always here

Upper Bound: For maximization objective function, LPR provides the highest value of optimal solution (It means the value of IP should lower than that)

Lower bound: For minimization, LPR provides the lowest value of Optimal solution (the value of IP should higher than that)

3.2.3 Text format(type algebraic notation, compact notation, expanded notation) (LDO lecture slides page26)

Sum: SUM[i =… to …](a_i b_i)=$\sum_{10}^{1} a_i b_i$

x_1 : x_1   use_ to present subscript

3.2.4 Inequality constraint -> equality

Using slack or surplus decision variables

Slack variable: as a variable to add the formula to make the <= convert to =, because the former formula is <= , so it needs the value to satisfy the equality.

Surplus variable: use it to satisfy the <= convert to =, because the value is >= X, so only minus surplus to make the formula covert to =.

Notice: If there are conflicting constraints in the model, using slack and surplus variables to reformulate the constraints , so that it can satisfy the optimization model.

Topic4

4.1 Post-optimality(Sensitivity analysis)

Sensitivity analysis: Only LP model, IP model(with LP Relaxation), MIP(fix integer decision value and then use LPR)

Slack: for non-binding constraints, use the slack to satisfy the constraint to equal the value.

If a constraint is not a binding constraint, it means it can't limit the objective function

Improvement of optimal solution: increase the binding resources/increase the coefficients of the objective function

4.2 Sensitivity Analysis

Final Value: the optimal solution

Reduced cost: there is a upper or lower bound for decision variables so that the value of objective function lost, the value of lost called reduced cost

Allowable increase and decrease: it means when change the coefficients of the objective function, the optimal solution of it will remained

So when the value of allowable increase and decrease is non-0, it means the optimal solution is single.

Changing the allowable increase or decrease of the coefficients of the objective function, the optimal solution will be remained but the value of objective function may be changed. And at the same time, it will generate multiple optimal soltions.

Shadow price: it means when the right value of the constraints changed, it may cause the value of objective function changed, so the maximum value of each unit changed of the right hand of the constraints called shadow price.

Allowable increase and decrease of the value of the right hand of the constraints: it means in the range of the value, when the value of right hand of the constraints changes, it can remain the same optimal solution and the value of the objective function.

Notice: only change the value of single constraint each time to keep the optimal solution.

Sensitivity report: when each decision variable appeared in the constraints report, it will produced fully, but if some decision variables lacked in the constraint report, it can't produce fully.

4.3 Guideline of Spreadsheet modelling(Lecture04 page36)

Topic5

5.1 TP(transportation problem)

1. Balance: the sum of supply nodes= the sum of demand nodes

If not balanced: use dummy node to meet them

2. solutions will be integer if all data are integer

3. Data: net flow of each node and the cost of each edge(连接线)

Decision variable: each link between supply node and demand node, mark it as the decision variable, S1->D1, the decision variable of this edge called X11

The objective Function: For transportation is cost(minimize)

Constraint: the sum of decision variable should meet the sum of the supply node and the demand node.

Notice:每个出发点衍生的 link 的决策变量的总和与出发点的值相等

每个接收点收到的 link 的决策变量值总和与每接收点相等。

4. General Form of TP:
Minimize: the sum of cost of each edge multiply each decision variable
Constraints: (1)the sum of decision variable of each supply node should meet the value of the sum of this supply node

(2)the sum of decision variables of each demand node should meet the value of the sum of this demand node

(3)each decision variable should >=0

5.2 MCFP(Minimum Cost Flow Problem)

1. have capacity constraints(Uij, 0<=decision variable<= Uij)

2. not balance: the sum of supply not as the same of the value of the demand

3. solutions will be integer if all data are integer

4.objective function: cost* decision variable

5. constraints:
(1)supply 直接发出的决策变量的和等于其本身的值
(2)每个节点发出的决策变量的值减去接收到的决策变量的值=其本身的值(b)
(3)每个决策变量的值如有 capacity, 0<=decision variable<= capacity(Uij)

5.3 MFP(Maximum Flow Problem)

1. have capacity constraints(Uij, 0<=decision variable<= Uij)

2. data: the capacity of each link

3.objective function: 出发点发出的 link 的决策变量相加 或者 最后接收点接收到的决策变量相加

4. constraints:
(1)除了开始以及接收的节点， 中间每个节点的发出的决策变量-接收到的决策变量=0

(2)每个决策变量都要在 0 和该路径的 capacity 之间。

Difference:
TP:cost
MCFP: Cost/Capacity
MFP:capacity

5.4 Shortest Path as a Flow Problem
路径图：出发点（只有发出），接收点（只有接收），中间节点（双向输入输出）

Objective function:
出发点（路径长度*发出的决策变量）+每个中间节点的（路径长度*（接收的决策变量+发出的决策变量））+接收点（路径长度*接受的决策变量）

约束：
（1） 出发点发出的决策变量相加=1
（2） 决策点接收的决策变量相加=1
（3） 各个中间发出的-接收的决策变量（包含双向）=0
（4） 各个决策变量 0<=Xij<=1

TOPIC6

6.1 Set covering problem（寻找符合一组能够覆盖题意的组合，并且最小化组员）

（1）BIP （C_ij 表示这个人是否有这个特征， X_i 表示这个人是否被选中）

1. 目标函数： 最小化- 选择组员变量相加之和，即 SUM(N_i)

2.约束条件：目标是否包含这个特征（C_ij）,与组员是否选中 X_i 相乘，求和使其大于等于 1 即满足题意

3. BIP 变量属于（0,1）

N 为组员个数

The depth of search tree: N

The size of search tree: 2^N

6.2 ASSIGNMENT PROBLEM

1. X_ij(工人 j 是否被分配任务 i)， C_ij(把任务 i 分配给工人 j 的代价)

2.目标函数：任务成本*是否被分配。 可以画图： 任务 I 工人 j

3.约束条件：每个任务要被完成， 每个工人要在任务中至少接一个活

Size of search tree: 2^(N*N)

The depth of search tree: (N*N)

TOPIC7

7.1 knapsack type problem(背包选择问题)：给定一个体积为 B 的背包，每个物体有不一样的大小和利润，放哪些物品进入背包使得利润最大

给定条件：一个背包的容量 B， N 件物品，每件物品的体积 B

目标函数： 最大化：使每件物品的利润*他是否被放进背包 相加

约束条件：物品*它的体积相加 小于 背包的容量

是否被放进是（0，1） bin 类型

7.2 multiple knapsack problem:多背包问题， 有多个体积不一样的背包，和 n 件不一样的物品，每个物品有自己的 size， 物品 i 在背包 j 中的利润， 让背包的利润最大化

目标函数： 物品 i 在背包 j 中的利润*是否被选中 遍历所有情况（物品 i 在背包 j 中）

约束函数： 背包 j 中所放的物品 i 的 size 相加要小于背包 j 的容量

每个物品只能被放进一个背包：即每个物品在不同背包 j 中的存在性相加=1

物品是否被选中， bin（0，1）

7.3 General assignment problem(工人分配任务)：每个工人有自己的可用时间 T，工人 j 做任务 i 的成本是 Cij, 需要的时间是 Tij，要使所有任务完成的成本最小

目标条件：最小化：分配任务 i 给工人 j 的成本 与 任务是否分配相乘　求和所有情况

约束条件：工人 j 被分配的任务的需要时间(需要的时间*任务是否被分配)相加 小于 该工人可用时间

一个任务只能有一个工人来做：即一个任务分配到不同工人 j 的可能性相加=1

7.4 bin packing type problem（用最少的垃圾箱放垃圾）：M 个垃圾箱的容量一样大，有 N 个垃圾且体积不同，使用的垃圾箱数量最小

目标条件：垃圾箱数量最小：　最小化：　垃圾箱 m 被选中的情况相加（被选中意味着垃圾箱数量+1）

约束条件：垃圾 i*垃圾的大小相加 <= 垃圾箱被选中的情况（垃圾箱的数量）与其容量相乘

每个垃圾只能被扔进一个垃圾箱：　垃圾 i 扔进垃圾桶 j 的可能性相加=1

垃圾箱被选中的情况（垃圾箱的数量）为 bin（0，1）

垃圾 i 扔进垃圾桶 j 的可能性 为 bin（0，1）

7.5 minimum spanning tree problem(在图中寻找路线权重相加最小的子图（子图是在保证没有闭环的情况下包含图中所有的线）)

目标条件：　最小化　所有的线的权重*线是否选中 相加

约束条件：所有的线选中情况相加要 <= 节点数-1（避免整张图闭环）

每次选出最小化的情况时，观察是否会出现闭环，如果在出现闭环的情况下增加新的约束条件, 直到最优条件下没有闭环：让闭环的线的选中情况相加 <= 闭环的节点数量-1

线条选中的情况为 bin（0，1）

7.6 Travelling salesman problem(寻找旅游所有城市一次，且最小化旅游距离)

目标函数：最小化旅游距离：最小化：所有的线的权重*是否选中 相加

约束函数：每个节点流入的线相加=1，每个节点流出的线相加=1

每个线选中的可能性=bin(0,1)

避免出现闭环的情况：即闭环时增加约束条件：闭环的线相加<=闭环的节点个数-1

7.7 Sysmmetric Travelling salesman problem（对称的 TSP 问题， 即线路没有箭头）

目标函数： 每个节点(单个节点的流进，流出的线条*权重相加， 即每个节点涉及到的线都算), 相加后 ÷2= 1/2（每个节点的流进流出*各自权重 相加）

约束条件：每个节点关联的线（延展的线，无论进入还是流出），选中的可能性相加=2

每个线选中的可能性=bin(0,1)

避免出现闭环的情况：即闭环时增加约束条件： 闭环的线相加<=闭环的节点个数-1

7.8 job sequencing problem(一台机器按时间完成多个任务，并合理分配任务，让超时惩罚最小化)

目标函数： 最小化： 任务的惩罚系数*收紧系数（超时的时间）

约束： 当任务 j 在任务 i 之前： xi-xj+MYij >= Tj

当任务 i 在任务 j 之前： -xi+xj-MYij>=Ti- M

Xi+SiE-SiL=Di-Ti

Xi,SiE,SiL >=0

Yij(任务 i 是否在任务 j 之前，在的话为 1，不在为 0) bin（0，1）， i，j 为对比的任务组合（i，j）=(1,2),(1,3)(2,3)之类的

Topic 8

8.1 多目标优化

定义：存在两个冲突的目标函数

方法：归一化将其转为单目标函数

方法 1：可行域内的有效点

1. **根据两个最大化目标画图得出相应的可行域的边点**
2. **列表，表明所有可行点（EP：efficient points）以及对应的目标函数值（ES:efficient solution）**

方法 2：Lexicographic Ordering

（1） 得出相应的 EP 和 ES， 将符合第一个优化函数的值带入目标函数，将其作为约束条件
（2） 加入新的约束条件后，重新计算变量和优化函数值

方法 3：权重法

（1）将两个目标函数加入各自的权重转为单一函数，最大化为权重系数为+，最小化为-

（2）加入权重后，计算得出有效方案，和有效点

方法 4：Goal Programming Technique

（1） 对相应的目标函数，设置一个目标值。
（2） 对目标函数加入可变变量
（3） 计算当前的方案

Topic 9(simplex method)

9.1 concept

(1)首先可以用图形法生成图，从而查看该模型的 CPF(corner point feasible 交点可行域解)

(2)其次，可以将 xi=0 的点作为 simplex method 初始化的第一个尝试的点（initial point）

(3)测试当前的 cpf 解是不是最优解，最优则不用转移到下一个解，否则转移到下一个相邻的 cpf 解

测试最优的方法：如果相邻的 cpf 对目标值的提升率为正则说明相邻 cpf 解更优

(4)如果当前 cpf 相连的边上的其他 cpf 对目标值的改进都不是正的，则说明目前的 cpf 为最优解。

(5)对各个约束加上松弛变量时，可以通过将 cpf 解的值带入约束，获得各个松弛变量的值，如果松弛变量为 0，则说明该约束 binding， 否则 non-binding

9.2 the algebraic simplex method

(1)non-variables:设置为 0，表示当前的 cpf 解已经被探索过

(2)basic variables: 表示需要代数计算的变量，当当前的 cpf 解需要通过其他约束更新时。

过程（slide14- slide25 ）：

1. **将初始的变量设为 0，开始初始化，得出松弛变量，得到初始化的 CPF(即包含松弛变量的所有变量的值)。**
2. **如果目标函数中的式子有为正的系数放到表达式前面，说明还有提升空间，继续更新**
3. **将第一个为正的系数用其他变量代替，并将其他约束和目标用当前的表达式合并替换。而不是直接带入**
4. **将第一个为正的系数用其他变量代替，同时将初始为 0 的其他变量带入，得出第一个为正的系数的变量的取值范围（选取小的作为其值,并将相关的变量作为 non-basic 变量设为 0）**

5. **将所有的变量相应表示后，代入约束后，获得一组新的 CPF(即所有变量包含松弛变量的值)**
6. **不停迭代，知道目标函数的表达式中，所有的变量的系数都为负数。 形式为 Z=….**

Topic 10

10.1 search tree（适用于 IP 或者 BIP 问题）

1.根据约束推出各个变量的范围

2. 根据变量的范围画出二叉树，即罗列所有可能的情况

3. 左侧子树的 upper bound 要比右侧的 upper bound 高

4. 在树中，如果该节点违反约束约束，将其值标为 Inf 并在最后子节点旁标出违反的约束号码

5. upper bound 由 LP-relaxation solution 得出。即放松情况下的目标函数的值

10.2 Branch and Bound(B&B)

分治（将问题分解为子问题）， 二叉树

Branching：将问题分割为子问题，即形成树，罗列出子问题情况

Bounding：寻找每一个子问题的 upper bound（最大化目标）， lower bound（最小化目标），使用 LP-relaxation

Conquering:**当以一个子问题被探索后，就不需要对该子问题树进行继续分支**

被征服的情况:

（1） **通过 LP-R，寻找到目前一个质量最好的整数解的解决方案**
（2） **子问题的边界不比当前整数方案更好**
（3） **用 LPR 解决问题不能找到可行方案。（违反约束）**

过程：

1.初始化，直接 LPR 整个问题，得到 bound

2.分支，子问题松弛情况下的解如果满足征服情况则不分支，否则则分支。并将上限安全设置成整数略小于松弛下的最大值（例如 16.3 B=16）

3.当到最后 solution 时结束，当