交通工程学
Introduction to Traffic Engineering
第 12 节　整数规划简介[1]

葛乾

西南交通大学　交通工程系
西南交通大学　系统科学与系统工程研究所

[1]基于美国亚利桑那州立大学 Angelia Nedich 教授讲义

本节目录

整数规划

- 如果线性规划问题中，要求某些或者全部变量的取值为整数，即为**线性整数规划问题**

$$\begin{aligned} \max \quad & 3x_1 + 4x_2 - 6x_3 \\ \text{s.t.} \quad & x_1 + x_2 - x_4 \geqslant 7 \\ & x_1, x_2, x_3 \geqslant 0 \\ & x_1, x_2, x_3 \text{ are integers} \end{aligned}$$

pure integer linear program

$$\begin{aligned} \min \quad & 2x_1 + 9x_2 - 5x_3 \\ \text{s.t.} \quad & 2x_1 + x_2 - 6x_3 \leqslant 6 \\ & x_1, x_2, x_3 \geqslant 0 \\ & x_1 \in \{0, 1\} \end{aligned}$$

mixed integer linear program

本节目录

投资项目选择

Five projects are being evaluated over a 3-year planning horizon. The following table gives the expected returns for each project and the associated yearly expenditures.

项目编号	每年投资	第 1 年	第 2 年	第 3 年	期末收益
1		5	1	8	20
2		4	7	10	40
3		3	9	2	20
4		7	4	1	15
5		8	6	10	30
投资预算		25	25	25	

Which projects should be selected to maximize the return?

整数线性规划模型

使用 x_j 表示对项目 $j, j = 1, \ldots, 5$ 的决策

$$x_j = \begin{cases} 1 & \text{选择项目}j \\ 0 & \text{反之} \end{cases}$$

则该问题可建模为

$$\begin{aligned} \max \quad & 20x_1 + 40x_2 + 20x_3 + 15x_4 + 30x_5 \\ \text{s.t.} \quad & 5x_1 + 4x_2 + 3x_3 + 7x_4 + 8x_5 \leqslant 25 \\ & x_1 + 7x_2 + 9x_3 + 4x_4 + 6x_5 \leqslant 25 \\ & 8x_1 + 10x_2 + 2x_3 + x_4 + 10x_5 \leqslant 25 \\ & x_j \in \{0, 1\} \quad \forall j \end{aligned}$$

其中三个不等式约束分别对应三年的投资预算

更多约束

根据项目自身特性，假设我们有以下额外的约束条件，应如何建模?

- **互斥选项**项目 1，3，5 中至多只能选择一个
  - $x_1 + x_3 + x_5 \leqslant 1$
- **依赖选项**项目 2 被选中，当且仅当项目 3 和 5 被选中
  - 引入约束 $x_3 + x_5 \leqslant 2x_2 + 1$ 与 $x_2 \leqslant x_3, x_2 \leqslant x_5$
  - 也可以考虑使用大 M 方法

更多约束

根据项目自身特性，假设我们有以下额外的约束条件，应如何建模？

- **互斥选项**项目 1，3，5 中至多只能选择一个
  - $x_1 + x_3 + x_5 \leqslant 1$
- **依赖选项**项目 2 被选中，当且仅当项目 3 和 5 被选中
  - 引入约束 $x_3 + x_5 \leqslant 2x_2 + 1$ 与 $x_2 \leqslant x_3, x_2 \leqslant x_5$
  - 也可以考虑使用大 M 方法

更多约束

根据项目自身特性，假设我们有以下额外的约束条件，应如何建模?

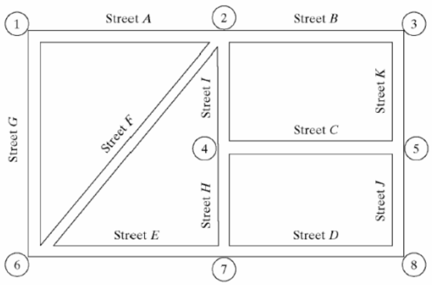
- **互斥选项**项目 1，3，5 中至多只能选择一个
  - $x_1 + x_3 + x_5 \leqslant 1$
- **依赖选项**项目 2 被选中，当且仅当项目 3 和 5 被选中
  - 引入约束 $x_3 + x_5 \leqslant 2x_2 + 1$ 与 $x_2 \leqslant x_3, x_2 \leqslant x_5$
  - 也可以考虑使用大 M 方法

求解分析

- 如果我们把整数约束松弛掉，即把 $x_j \in \{0,1\}$ 变为 $0 \leqslant x_j \leqslant 1, \forall j$，用上节课学习到的单纯性法，我们可求得 $x_1 \approx 0.5789, x_2 = x_3 = x_4 = 1, x_5 \approx 00.7368$。可知该结果并不满足 0-1 约束条件
- 如果我们把上面的结果四舍五入，则所有变量取值均为 1，显然也不是可行解
- 因此，四舍五入不是一种好的求解方式

集合覆盖问题

- To promote on-campus safety, a college's security department is in the process of installing cameras at selected locations. The department wants to install the minimum number of cameras while providing a surveillance coverage for each of the campus main streets. It is reasonable to place the cameras at street intersections so that each camera serves at least two streets. The eight candidate locations are given in the figure, indexed by 1 through 8
- Provide an ILP formulation of the problem.



使用 x_j 表示是否在位置 $j, j = 1, \dots, 5$ 处放置监控摄像头的决策

$$x_j = \begin{cases} 1 & \text{选择位置} j \text{ 以设置摄像头} \\ 0 & \text{反之} \end{cases}$$

我们的目标是最小化摄像头的个数；每条街道对应一个约束，即其至少被一个摄像头覆盖到。该问题可建模为

$$\begin{aligned}
\min \quad & \sum_{j=1}^{8} x_j \\
\text{s.t.} \quad & x_1 + x_2 \geqslant 1, \quad x_2 + x_3 \geqslant 1 \text{ 道路 A 与 B} \\
& x_4 + x_5 \geqslant 1, \quad x_7 + x_8 \geqslant 1 \text{ 道路 C 与 D} \\
& x_6 + x_7 \geqslant 1, \quad x_2 + x_6 \geqslant 1 \text{ 道路 E 与 F} \\
& x_1 + x_6 \geqslant 1, \quad x_4 + x_7 \geqslant 1 \text{ 道路 G 与 H} \\
& x_2 + x_4 \geqslant 1, \quad x_5 + x_8 \geqslant 1 \text{ 道路 I 与 J} \\
& x_3 + x_5 \geqslant 1, \quad \text{道路 K} \\
& x_j \in \{0, 1\} \ \forall j
\end{aligned}$$

固定费用的建模

- In some applications, we have two types of costs: an initial and a cost per unit of time. The initial cost (”set-up” cost, fixed charge) is incurred only at the start the activity/service. The cost per unit of time is incurred during the period of the activity/service
- Mathematically, given the fixed charge F_j and the unit cost c_j of some activity j, the cost of activity j over a period of time x_j is given by

$$\begin{cases} F_j + c_j x_j & \text{if } x_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

固定费用建模示例

- There are three mobile-phone service providers in a given area, ABell, BBell and CBell.
- ABell charge a flat fee of $16 per month plus $0.25 per minute.
- BBell charges $25 per month and $0.21 per minute.
- CBell charges $18 per month and $0.22 per minute.
- A customer needs to sign up for a service for an average of 200 minutes of monthly calls. Which company should the customer subscribe for the service to minimize the cost of the monthly phone bill?
- Formulate the problem as ILP.

使用 $1,2,3$ 分别表示 ABell、BBell、CBell，引入变量 y_j 变量对通讯公司的决策

$$y_j = \begin{cases} 1 & \text{选择通信公司}j\text{所提供的服务} \\ 0 & \text{反之} \end{cases}$$

同时，引入变量 x_i 表示服务时间的长度，即

$$x_j = \text{使用通信公司 j 的通话时长}$$

The fact that the customer will choose among the three providers for total of 200 minutes can be modeled as

$$x_1 + x_2 + x_3 = 200$$

The maximum subscribed time x_j is 200 minutes for provider j only if the provider is selected for the service

$$0 \leqslant x_j \leqslant 200 y_j \text{ for all } j$$

目标函数为

$$16y_1 + 0.25x_1 + 25y_2 + 0.21x_2 + 18y_3 + 0.22x_3$$

问题可建模为

$$\begin{aligned}
\min \quad & 16y_1 + 0.25x_1 + 25y_2 + 0.21x_2 + 18y_3 + 0.22x_3 \\
\text{s.t.} \quad & x_1 + x_2 + x_3 = 200 \\
& x_j - 200y_j \leqslant 0 \ \forall\ j \\
& x_j \geqslant 0 \ \forall\ j \\
& y_j \in \{0, 1\} \ \forall\ j
\end{aligned}$$

逻辑约束

- Suppose you have two constraints and you can choose only one of them.
- For example, the constraints may represent two alternative resource consumption relations
- The machine production times is given by

$$\begin{aligned} 3x_1 + 2x_2 &\leqslant 250 \qquad \text{machine 1} \\ x_1 + 4x_2 &\leqslant 600 \qquad \text{machine 2} \end{aligned}$$

- The company needs to purchase a machine and can afford to buy only one machine. How can we model this requirement within ILP setting?

大 M 方法

- 一种解决方案：引入 0-1 变量 y 表示是否选择机器 1

$$y = \begin{cases} 1 & \text{选择机器1} \\ 0 & \text{反之} \end{cases}$$

- 令 M 为一个非常大的数字，如 $M = 1000$，则上述的决策过程可建模为：

$$\begin{aligned} 3x_1 + 2x_2 &\leqslant 250 + M(1-y) \quad \text{machine 1} \\ x_1 + 4x_2 &\leqslant 600 + My \qquad \text{machine 2} \end{aligned}$$

- A similar idea can be used for selection of k out of m constraints, where k and m are given positive integers with $k < m$

全单模矩阵

- Some of the ILPs have a special structure that allows us to “ignore” integer constraints and solve them as LPs and still obtain integer solutions
- Such a class of ILP problems is rather small; one such problem that we have seen is
  - Transportation problem with integer supplies and demands
- Most of the ILP problems do not have the special structure
- These are solved using Branch-and-Bound (BB) algorithm

运输问题

	Denver	Miama	**Supply**
LA	80	215	1000
Detroit	100	108	1500
New Orleans	102	68	1200
Demand	2300	1400	

$$\min \quad 80x_{11} + 215x_{12} + 100x_{21} + 108x_{22} + 102x_{31} + 68x_{32}$$

$$x_{11} + x_{12} = 1000 \quad \text{LA}$$

$$x_{21} + x_{22} = 1500 \quad \text{Detroit}$$

$$x_{31} + x_{32} = 1200 \quad \text{New Orleans}$$

$$x_{11} + x_{21} + x_{31} = 2300 \quad \text{Denver}$$

$$x_{12} + x_{22} + x_{32} = 1400 \quad \text{Miami}$$

$$x_{ij} \geqslant 0 \; \forall \; i, j$$

- Let A be the matrix that captures the linearly independent constraints (the last row is linearly dependent on the first four rows)
- $Ax = b$ represents the first four constraints
- What makes the transportation model special is the structure of the matrix
- A corresponding to the independent "equality constraints" $Ax = b$.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- The matrix is **totally unimodular** (TU).

TU 的性质

Totally Unimodular

A matrix A is totally unimodular if every square nonsingular[a] submatrix has determinant +1 or -1.

[a] A square matrix is nonsingular if it is invertible.

A condition for an $m \times n$ matrix A to be totally unimodular is

The matrix A is totally unimodular if it has the following properties

- Every column of A contains at most two non-zero entries
- Every entry in A is 0, +1, or -1
- The rows of A can be partitioned into two disjoint index sets B and C, such that
  - If two non-zero entries in a column of A have the same sign, then the row of one is in B, and the other is in C;
  - If two non-zero entries in a column of A have opposite signs, then the rows of both are in B, or both in C

全单模矩阵示例

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- For our matrix A, we can see that the conditions are satisfied with $B = \{1, 2, 3\}$ and $C = \{4\}$.
- Not all ILP problems are with such a nice structure.
- To solve the general ILPs, we need a special algorithm that can deal with the integer nature of the variables

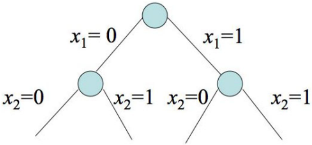
本节目录

- Suppose we are facing the problem

$$
\begin{aligned}
\max \quad & 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{s.t.} \quad & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leqslant 10 \\
& x_3 + x_4 \leqslant 1 \\
& -x_1 + x_3 \leqslant 0 \\
& -x_2 + x_4 \leqslant 0 \\
& x_j \in \{0, 1\} \ j = 1, \ldots, 4
\end{aligned}
$$

- There are 2^4 possibilities for the variables x_j. We can enumerate them all and find the best decision.

- One way to represent these options is through the use of a "binary" tree



- The places where the decision tree branches are referred as **nodes**.
- The nodes at the end of the branching tree are referred as leaf-nodes
- Our feasible points are at the leaf-nodes of this decision tree.
- We could build the whole tree and then explore the objective value at leaf-nodes
- But, this is not efficient.
- Branch-and-Bound (BB) algorithm is an efficient approach that builds the tree "as needed" and searches the nodes of the tree efficiently.

分支定界

- BB is an iterative algorithm that at each iteration **branches the tree and possibly prunes the tree** (在树上分叉以及剪枝) until the solution if found.
- Each node in a tree corresponds to a subproblem of the original problem. 每个节点对应原问题的一个子问题
- Each iteration involves the following operations:
  - *Branching* - deciding on from which node to branch among the "active" nodes in the tree 在活跃的节点中决定在哪个节点分叉
  - *Bounding* - operation that is performed at each created node to decide on which nodes to be fathomed (pruning the branches), It amounts to solving the corresponding subproblem at each node. 在每个节点处求解子问题，以测量在何处剪枝
  - *Fathoming* - in essence pruning the tree. Based on some test and the bound values at the "active" nodes, the algorithm decides on nodes that can be discarded. This step also includes a termination criterion. 根据在每个节点处的测试与界限值，确定舍弃那些节点。此处也需要判定算法是否收敛

初始化

- The initial node in the tree corresponds to solving the LP relaxation of the given problem
- LP relaxation is the problem resulting from the given ILP when we "ignore" integer constraints of the variables
- This node is "active" and it is the most recent.

（最大化问题[2]）的迭代步骤

- Branching. Among the most recently created nodes that are "active" select one (braking ties according to the largest bound). Branch from that node by fixing one of the "relaxed" variables to 0 and 1 (in the node subproblem)
- Bounding. At each new node, solve the corresponding LP problem and determine the optimal LP value. Round the non-integer value down (to the nearest integer). This is the bound of the subproblem at the node. （寻找一个上界）
- Fathoming. For each new node (subproblem) apply the following three tests:
  - Integer solution. If one of the new nodes has integer solution, its bound is compared to the bounds of other such nodes. If it does not have the best value - it is fathomed. If it has the best value it is fathomed and it is our current best solution (incumbent 迄今为止最好的解，为解的下界)
  - Bound value. If any of the new nodes has a bound smaller than currently the best bound - fathom the node.
  - Infeasibility. If LP at any of the new nodes has no solution (not feasible) - fathom the node.

[2]最小化问题中，上界与下界的准则反之

终止

- Stop when there are no more subproblems (no branching). The node with the best value provides the solution.
- Otherwise perform a new iteration.

示例

试用分支定界法求解以下问题:

$$
\begin{aligned}
\max \quad & 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{subject to} \quad & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leqslant 10 \\
& x_3 + x_4 \leqslant 1 \\
& -x_1 + x_3 \leqslant 0 \\
& -x_2 + x_4 \leqslant 0 \\
& x_j \in \{0, 1\} \quad j = 1, \ldots, 4
\end{aligned}
$$

初始化

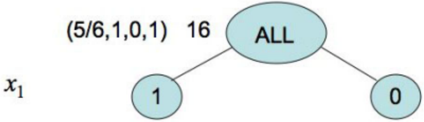
- At the initial node, we would solve its LP relaxation

$$
\begin{aligned}
\max \quad & 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
\text{subject to } & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leqslant 10 \\
& x_3 + x_4 \leqslant 1 \\
& -x_1 + x_3 \leqslant 0 \\
& -x_2 + x_4 \leqslant 0 \\
& 0 \leqslant x_j \leqslant 1 \; j = 1, 2, 3, 4
\end{aligned}
$$

- Using LP algorithm, we find the optimal value $(5/6, 1, 0, 1)$ and the optimal value $z = 16\frac{1}{2}$.
- If the solution and the optimal value were integers, we would stop.
- Since it is not, this value gives us 16 as the best lower bound on the optimal value of the ILP ($blb = 16$)（最好的下界即是目前在可行域所找到的与原问题的目标函数差值最小的解）

初始化示意

The initialization and moving to the first iteration



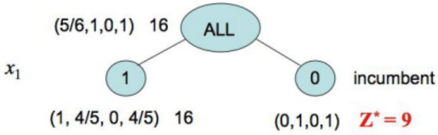
第一次迭代

We branch from the LP relaxation (ALL node) into two new nodes corresponding to $x_1 = 1$ and $x_1 = 0$.
At node $x_1 = 1$ we solve the following LP

$$\begin{aligned} \max \quad & 9 + 5x_2 + 6x_3 + 4x_4 \\ \text{s.t} \quad & 3x_2 + 5x_3 + 2x_4 \leqslant 4 \\ & x_3 + x_4 \leqslant 1 \\ & x_3 \leqslant 1 \\ & -x_2 + x_4 \leqslant 0 \\ & 0 \leqslant x_j \leqslant 1 \; j = 2, 3, 4 \end{aligned}$$

At node $x_1 = 0$ we solve the following LP

$$
\begin{aligned}
\max \quad & 5x_2 + 6x_3 + 4x_4 \\
\text{s.t} \quad & 3x_2 + 5x_3 + 2x_4 \leqslant 10 \\
& x_3 + x_4 \leqslant 1 \\
& x_3 \leqslant 0 \\
& -x_2 + x_4 \leqslant 0 \\
& 0 \leqslant x_j \leqslant 1 \; j = 2, 3, 4
\end{aligned}
$$



Node to the right is incumbent and fathomed (no branching there ever). Node to the left is the only active node and most recent

第二次迭代

We branch from the node $x_1 = 1$ going to nodes $x_2 = 1$ and $x_2 = 0$.
At node $x_2 = 1$ (also we have $x_1 = 1$), we solve the following LP

$$
\begin{aligned}
\max \quad & 14 + 6x_3 + 4x_4 \\
\text{s.t} \quad & 5x_3 + 2x_4 \leqslant 1 \\
& x_3 + x_4 \leqslant 1 \\
& x_3 \leqslant 1 \\
& x_4 \leqslant 1 \\
& 0 \leqslant x_j \leqslant 1 \; j = 3, 4
\end{aligned}
$$

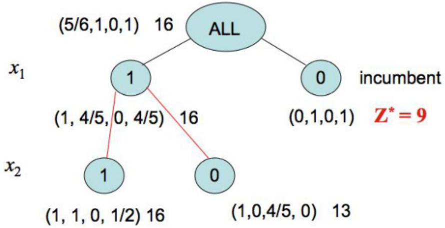
The solution is $(1, 1, 0, 1/2)$ and the optimal value is 16.

At node $x_2 = 0$ (and $x_1 = 1$) we solve the following LP

$$\begin{aligned} \max \quad & 9 + 6x_3 + 4x_4 \\ \text{s.t} \quad & 5x_3 + 2x_4 \leqslant 4 \\ & x_3 + x_4 \leqslant 1 \\ & x_3 \leqslant 1 \\ & x_4 \leqslant 0 \\ & 0 \leqslant x_j \leqslant 1 \; j = 3, 4 \end{aligned}$$

The solution is $(1, 0, 4/5, 0)$ and the optimal value is $13\frac{4}{5}$. We round this value down to 13

第二次迭代结果示意



At this point, we have two active nodes that are also the most recent (the two at the level x_2).

第三次迭代

We branch from the node $x_2 = 1$ since it has a larger bound of the two active nodes (16 and 13).
We create two new nodes $x_3 = 1$ and $x_3 = 0$.
At node $x_3 = 1$ (branch $x_1 = 1$ and $x_2 = 1$), we solve the following LP

$$\begin{aligned} \max \quad & 20 + 4x_4 \\ \text{s.t} \quad & x_4 \leqslant -2 \\ & x_4 \leqslant 0 \\ & x_4 \leqslant 1 \\ & 0 \leqslant x_4 \leqslant 1 \end{aligned}$$

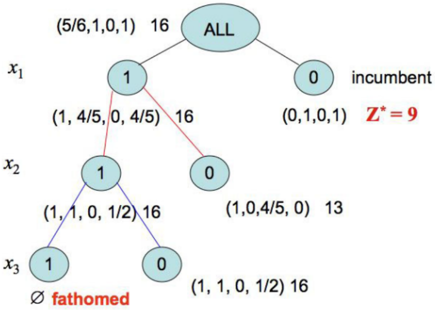
This LP has no solution - it is infeasible. We fathom this node.

We now create node $x_3 = 0$ ($x_1 = 1$,$x_2 = 1$). At this node we solve the following LP

$$\begin{aligned} \max \quad & 14 + 4x_4 \\ \text{s.t} \quad & 2x_4 \leqslant 1 \\ & x_4 \leqslant 1 \\ & 0 \leqslant x_4 \leqslant 1 \end{aligned}$$

The solution is $(1, 1, 0, 1/2)$ and the optimal value is 16. This node remains active.

第三次迭代结果示意



At this point, nodes $x_2 = 0$ and $x_3 = 0$ are active.

第四次迭代

We branch from the node $x_3 = 0$ since it is a more recent of the two remaining active nodes.
We create two new nodes $x_4 = 1$ and $x_4 = 0$.
At node $x_4 = 1$ the problem reduces to: having a point $(1, 1, 0, 1)$ and the "LP of the form"

$$\begin{aligned} \max \quad & 20 \\ \text{s.t.} \quad & 2 \leqslant 4 \\ & 1 \leqslant 0 \\ & 1 \leqslant 1 \end{aligned}$$

The LP has no solution - it is infeasible. We fathom this node.

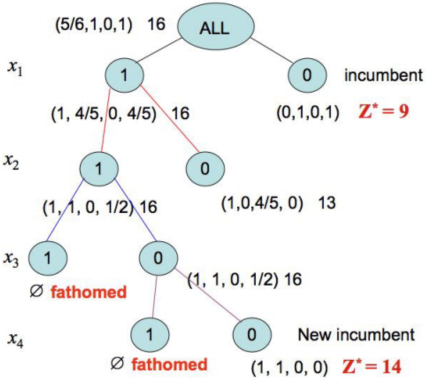
第四次迭代

We now create node $x_4 = 0$
At this node we have the following LP

$$\begin{aligned} \max \quad & 14 \\ & 0 \leqslant 1 \end{aligned}$$

and the point $(1, 1, 0, 0)$ being feasible. The optimal value is 14. This node becomes a new incumbent node. A new best value is $Z^* = 14$.

第四次迭代结果示意



At this point, only the node $x_2 = 0$ is active. Its value is smaller than the current best $Z^* = 14$, so we fathom this node. Since no node is active, the algorithm terminates. The optimal point and the value are given by the solution and the value at the best incumbent node, namely $(1, 1, 0, 0)$ and $Z^* = 14$

谢谢！