Ch 9 Integer Programming 整數規劃

9.1 有一些實際問題，決策變數必須限定為正整數，如人、商品、机器的數目。有關這些問題若採用 Linear Programming 求解，常會得到非整數解，有些人認為將非整數答案求出後用四捨五入即可將問題解決，但事實上許多問題的最適解不可以上述方法求得，例如

$$\text{Max } z = x_1 + x_2$$

st.

$$-x_1 + x_2 \le 3.5$$

$$x_1 + x_2 \le 16.5$$

用 LP 求解 $x_1^* = 6.5$ $x_2^* = 10$

x_1^* 取 6　　$-6 + 10 = 4 \nleq 3.5$ 與(1)式不合

x_1^* 取 7　　$7 + 10 = 17 \nleq 16.5$ 與(2)式不合

例如 $\text{Max } Z = x_1 + 5x_2$

st.

$$x_1 + 10x_2 \le 20$$

$$x_1 \le 2$$

用 LP 求解 $x_1^* = 2$, $x_2^* = 1.8$ $Z = 11$

為合於整數限制

$x_1^* = 2$, $x_2^* = 1$ 得到 $Z = 7$.

而其實最適解 $x_1^* = 0$, $x_2^* = 2$ Z 非

Integer Programming (IP)

$$\begin{aligned} \text{Max} \quad & Z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & x_1, x_2 \text{ integer} \end{aligned}$$

pure integer programming

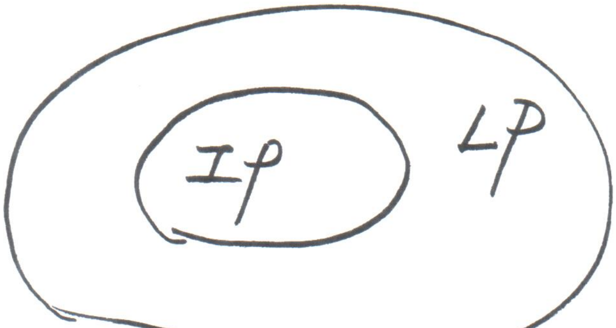
$$\begin{aligned} \text{Max} \quad & Z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & x_1, x_2 \in \{0, 1\} \end{aligned}$$

0-1 (binary) integer programming

$$\begin{aligned} \text{Max} \quad & Z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \le 6 \\ & x_1 \ge 0 \quad x_2 \text{ integer} \end{aligned}$$

mixed integer programming

Def: The LP obtained by omitting all integer or 0-1 constraints on variables is called <u>LP relaxation</u> of the IP.



Def: The optimal value of any relaxation of a <u>maximize</u> model yields <u>an upper bound</u> on the optimal value of the full model. The optimal value of any relaxation of a <u>minimize</u> model yields <u>a lower bound</u>

§資本預算問題§

有五項計劃供评估，我们需決定应投资那些計劃，每年可用資金是 25，25，25 百万元。

Project	Expenditure (million/year) 1	2	3	Returns
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
Available funds	25	25	25	

令 $x_j = \begin{cases} 1 & \text{代表計劃 } j \text{ 被選取} \\ 0 & \text{其他} \end{cases}$

$$\text{Max } Z = 20x_1 + 40x_2 + 20x_3 + 15x_4 + 30x_5$$

s.t.

$$5x_1 + 4x_2 + 3x_3 + 7x_4 + 8x_5 \le 25$$

$$1x_1 + 7x_2 + 9x_3 + 4x_4 + 6x_5 \le 25$$

$$8x_1 + 10x_2 + 2x_3 + 1x_4 + 10x_5 \le 25.$$

$$\forall x_i \in \{0, 1\}$$

用软体求解得

$x_1 = x_2 = x_3 = x_4 = 1$. $x_5 = 0$. $Z = 95$

Knapsack Problem 背包問題

是一种純整數規劃，只有一條限制式

Knapsack problem 起源於登山者要打包，他們必需選擇最重要的東西，因為登山者的負重有限。

§ Indy Car racing team 有 6 种方案可增加速度（可同時使用，不互斥）

	1	2	3	4	5	6
成本($,000)	10.2	6	23	11.1	9.8	31.6
Speed increase(mph)	8	3	15	7	10	12

假設希望表現最好，但預算僅有 35,000

令 $x_j = \begin{cases} 1 & \text{方案 } j \text{ 被使用} \\ 0 & \text{, otherwise} \end{cases}$

$$\text{Max } Z = 8x_1 + 3x_2 + 15x_3 + 7x_4 + 10x_5 + 12x_6$$

$$\text{st. } 10.2x_1 + 6x_2 + 23x_3 + 11.1x_4 + 9.8x_5 + 31.6x_6 \le 35$$

$$\forall x_i \in \{0, 1\} \quad i = 1, \dots, 6$$

$$x_1 = x_4 = x_5 = 1$$

假設希望成本最小，速度至少增加 30 mphrs.

$$\text{Min } Z = 10.2x_1 + 6x_2 + 23x_3 + 11.1x_4 + 9.8x_5 + 31.6x_6$$

$$\text{st. } 8x_1 + 3x_2 + 15x_3 + 7x_4 + 10x_5 + 12x_6 \ge 30.$$

$$\forall x_i \in \{0, 1\} \quad i = 1, \dots, 6$$

9.1.2 安裝安全電話 (Set Covering Problem)

某大學安全部门為保護學生安全，要安裝緊急電話，由於經費有限，希望安裝數目越少越好，但需滿足每一街道至少有一支電話。

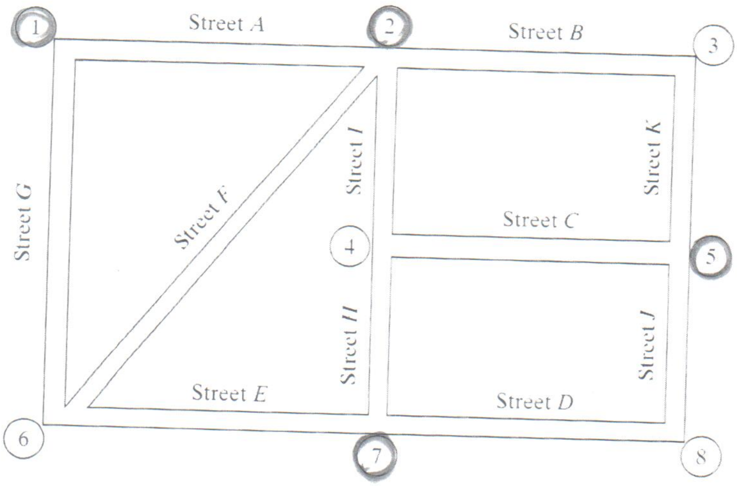


FIGURE 9.1
Street map of the U of A campus

The constraints of the problem require installing at least one telephone on each of the 11 streets (A to K). Thus, the model is

$$\text{Minimize } z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$$

subject to

$$\begin{array}{llll}
x_1 + x_2 & \geq 1 & \text{(Street } A\text{)} \\
x_2 + x_3 & \geq 1 & \text{(Street } B\text{)} \\
x_4 + x_5 & \geq 1 & \text{(Street } C\text{)} \\
x_7 + x_8 & \geq 1 & \text{(Street } D\text{)} \\
x_6 + x_7 & \geq 1 & \text{(Street } E\text{)} \\
x_2 + x_6 & \geq 1 & \text{(Street } F\text{)} \\
x_1 + x_6 & \geq 1 & \text{(Street } G\text{)} \\
x_4 + x_7 & \geq 1 & \text{(Street } H\text{)} \\
x_2 + x_4 & \geq 1 & \text{(Street } I\text{)} \\
x_5 + x_8 & \geq 1 & \text{(Street } J\text{)} \\
x_3 + x_5 & \geq 1 & \text{(Street } K\text{)}
\end{array}$$

$$x_j = (0, 1),\ j = 1, 2, \ldots, 8$$

交通便利性	5.28	3.3	5.28	1.98	1.98	1.98	2.64
運輸成本	1.77	2.66	3.25	2.66	2.66	4.01	2.66
原物料之取得	3.85	3.85	4.95	4.4	3.3	3.3	3.3
距離主要市場之遠近	3.3	3.3	2.2	1.1	1.1	2.2	3.3
風水偏好	1.74	1.74	1.8	1.77	2.84	0.06	0
社區對公司的態度	0.39	0.32	0.32	0.7	0.7	0.56	0.56
環境污染	0.04	0.08	0.12	0.14	0.08	0.19	0.19
氣候及天然現象	0.18	0.18	0.27	0.27	0.3	0.24	0.24
公共設施(醫院休憩場所等)	0.07	0.04	0.1	0.08	0.07	0.09	0.1
水電供應	0.05	0.05	0.09	0.05	0.05	0.06	0.06
總計	70.28	73.98	82.11	77.97	78.2	73.05	70.01

步驟 10 決定最佳的方案，應選擇分數最高的臺中。

I. 覆蓋模式(covering model)

多用於緊急公共設施如消防隊、警察局、醫院等

e.g.在凱洛伊郡有六個城市，凱洛伊郡必須決定在那些地方蓋消防隊，其目標是希望消防隊的數目越少越好並且至少有一消防隊可以在 15 分鐘內到達各城市，凱洛伊郡各城市間的行駛時間，如下表，試決定消防隊數目及其應蓋在何處。

	1	2	3	4	5	6
1	0					
2	10	0				
3	20	25	0			
4	30	35	15	0		
5	30	20	30	15	0	
6	20	10	20	25	14	0

$$\text{Minimize } Z = X_1 + X_2 + X_3 + X_4 + X_5 + X_6$$

$$\begin{aligned} \text{st} \quad & X_1 + X_2 \geq 1 \\ & X_1 + X_2 + X_6 \geq 1 \\ & X_3 + X_4 \geq 1 \\ & X_3 + X_4 + X_5 \geq 1 \\ & X_4 + X_5 + X_6 \geq 1 \\ & X_2 + X_5 + X_6 \geq 1 \\ & Xi = 0 \text{ or } 1 \ (i = 1,2,3,4,5,6) \end{aligned}$$

$$Z = 2,\ X_2 = X_4 = 1, X_1 = X_3 = X_5 = X_6 = 0$$

e.g. 固定費用問題

Mc Bell		\$16		\$0.25
Pa Bell	基本費率每月	\$25	加上每分鐘	\$0.21
BabyBell		\$18		\$0.22

假設每月使用電話平均為200分鐘，如果不使用該公司電話就不用付基本費率，可任選用三家電話公司，試問應如何使用三家公司，使每月帳單最低？

x_1 : Mc Bell

x_2 : Pa Bell 每月使用電話分鐘數目

x_3 : Baby Bell

$$y_1 = \begin{cases} 1 & \text{if } x_1 > 0 \\ 0 & \text{if } x_1 = 0 \end{cases} \qquad y_2 = \begin{cases} 1 & \text{if } x_2 > 0 \\ 0 & \text{if } x_2 = 0 \end{cases}$$

$$y_3 = \begin{cases} 1 & \text{if } x_3 > 0 \\ 0 & \text{if } x_3 = 0 \end{cases}$$

$$x_j \le M y_j \qquad j=1,2,3$$

M是一非常大正數，當 $x_j > 0$，$y_j = 1$

M非常大，$x_j \le M y_j$ 是多餘的不發生影响

在本例中，每月使用 200 minutes，$X_j \le 200$ $(j=1, 2, 3)$，令 $M=200$

完整的模式

$$\text{Min } Z = 0.25X_1 + 0.21X_2 + 0.22X_3 + 16y_1 + 25y_2 + 18y_3$$

s.t.

$$X_1 + X_2 + X_3 = 200$$

$$X_1 \le 200y_1$$

$$X_2 \le 200y_2$$

$$X_3 \le 200y_3$$

$$y_1, y_2, y_3 \in \{0, 1\}, \quad X_1, X_2, X_3 \ge 0$$

最佳解 $X_3 = 200$

$y_3 = 1$

$Z = 62$

希望在LA或SF蓋工廠(兩地均可蓋)，但只希望蓋一倉庫，而且倉庫一定要設在有工廠的地方，可用資金 10 million，希望NPV最大。

Decision	Yes or No Question	Decision Variables	NPV	Capital required
1	build factory in LA	x_1	9	6
2	build factory in SF	x_2	5	3
3	build warehouse in LA	x_3	6	5
4	build warehouse in SF	x_4	4	2

$$x_j \begin{cases} 1 & \text{if decision } j \text{ is Yes} \\ 0 & \text{if decision } j \text{ is No} \end{cases}$$

$$\text{Max } Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

$x_3 + x_4 = 1$ (只有一倉庫，mutually exclusive)

$6x_1 + 3x_2 + 5x_3 + 2x_4 \le 10$ (資金限制)

$x_1 - x_3 = 0$
$x_2 - x_4 = 0$ (contigent variables)

$x_1, x_2, x_3, x_4 \in \{0, 1\}$

$x_1 - x_3 = 0$ $\qquad$ $x_j \le M y_j$

$x_1, x_3 \in \{0, 1\}$ $\qquad$ x_j integer $\quad y_j \in \{0, 1\}$

Either-or-Constraints

Either $3x_1 + 2x_2 \le 18$

or $x_1 + 4x_2 \le 16$

令M為十分大之正數

$$\text{Either} \begin{cases} 3x_1 + 2x_2 \le 18 \\ \text{and} \quad x_1 + 4x_2 \le 16 + M \end{cases}$$

M有消去的效果.其他限制式滿足後.有M的限制式是多餘的,∵M是很大的正數

$$\text{or} \begin{cases} 3x_1 + 2x_2 \le 18 + M \\ x_1 + 4x_2 \le 16 \end{cases}$$

綜合上式兩式、令 $y \in \{0, 1\}$

$$\text{可得} \begin{cases} 3x_1 + 2x_2 \le 18 + My \\ x_1 + 4x_2 \le 16 + M(1-y) \end{cases}$$

$$\text{當 } y=0 \text{ 時} \begin{cases} 3x_1 + 2x_2 \le 18 \\ x_1 + 4x_2 \le 16 + M \end{cases}$$

$$y=1 \text{ 時} \begin{cases} 3x_1 + 2x_2 \le 18 + M \\ x_1 + 4x_2 \le 16 \end{cases}$$

9.2.5 (Either-or Constraints)

Jobco 使用一台机器處理叁工作

Job	處理時間 p_i	到期日 d_i	延後懲罰成本
1	5	25	19
2	20	22	12
3	15	35	34

本問題是決定工作順序使得延後懲罰成本最小。

令 x_j : 工作 j 的開始日期(從 0 開始衡量)

這類問題有兩類限制式 ① 沒有兩種工作同時進行 ② 到期日限制式

① 工作 i 和 j 有處理時間 p_i 和 p_j, 兩項工作不會同時進行 either

$$x_i \geq x_j + p_j \quad \text{or}$$

$$x_j \geq x_i + p_i$$

決定那項工作在前面進行, 令

$$y_{ij} = \begin{cases} 1 & \text{如果 } i \text{ 在 } j \text{ 前面} \\ 0 & \text{如果 } j \text{ 在 } i \text{ 前面} \end{cases}$$

就十分大的M而言，either-or 限制式轉換下列限制式

$$M y_{ij} + x_i \geq x_j + p_j$$

$$M(1-y_{ij}) + x_j \geq x_i + p_i$$

移項整理

$$x_i - x_j \geq p_j - M y_{ij} \quad \text{——(1)}$$

$$x_j - x_i - M y_{ij} \geq p_i - M \quad \text{——(2)}$$

如果 $y_{ij}=0$，則第(1)式有效，第(2)式無效

② 到期日限制式，給定 d_j 是工作 j 的到期日，令 S_j 是不受符號限制的變數，

$$x_j + p_j + S_j = d_j$$

如果 $S_j \geq 0$ 則滿足到期日

$S_j < 0$，則過期發生 penalty.

使用 $S_j = S_j^+ - S_j^-$，$S_j^+, S_j^- \geq 0$，

限制式變成

$$x_j + S_j^+ - S_j^- = d_j - p_j$$

延後的懲罰是和 S_i^- 成比例，本問題建構如下：

$$\text{Min } z = 19S_1^- + 12S_2^- + 34S_3^-$$

s.t.

$$x_1 - x_2 + My_{12} \ge 20$$
$$-x_1 + x_2 - My_{12} \ge 5 - M$$
$$x_1 - x_3 + My_{13} \ge 15$$
$$-x_1 + x_3 - My_{13} \ge 5 - M$$
$$x_2 - x_3 + My_{23} \ge 15$$
$$-x_2 + x_3 - My_{23} \ge 20 - M$$
$$x_1 + S_1^+ - S_1^- = 25 - 5$$
$$x_2 + S_2^+ - S_2^- = 22 - 20$$
$$x_3 + S_3^+ - S_3^- = 35 - 15.$$

$$y_{12}, y_{13}, y_{23} \in \{0, 1\},\ x_1, x_2, x_3, S_1^+, S_1^-, S_2^+, S_2^-, S_3^+, S_3^- \ge 0$$

最佳解 $x_1 = 20$ $x_2 = 0$ $x_3 = 25$.

代表工作順序 $2 \to 1 \to 3$

	開始	完成	到期日	懲罰成本
工作 2	0	20	22	12
工作 1	20	25	25	19
工作 3	25	40	35	34 ✓

$(40 - 35) \times 34 = 170$ 總成本

All-or-Nothing Constraint

$X_j = 0$ or U_j $\quad x_j = U_j y_j \quad y_j = \begin{cases} 1 \\ 0 \end{cases}$

$$\max \quad 18X_1 + 3X_2 + 9X_3$$
$$\text{st} \quad 2X_1 + X_2 + 7X_3 \le 150$$
$$0 \le X_1 \le 60$$
$$0 \le X_2 \le 30$$
$$0 \le X_3 \le 20$$

將 X_1, X_2, X_3 修改成為其上界或零.

$$\max \quad 18(60y_1) + 3(30y_2) + 9(20y_3)$$
$$\text{st} \quad 2(60y_1) + 30y_2 + 7(20y_3) \le 150$$
$$y_1, y_2, y_3 = 0 \text{ or } 1$$

$$\max \quad 1080y_1 + 90y_2 + 180y_3$$
$$\text{st} \quad 120y_1 + 30y_2 + 140y_3 \le 150$$
$$y_1, y_2, y_3 = 0 \text{ or } 1$$

If-Then Constraints

在許多情境我們希望

if $f(x_1, x_2, \ldots, x_n) > 0$ is satisfied.

then $g(x_1, x_2, \ldots, x_n) \ge 0$ must be satisfied.

while $f(x_1, x_2, \ldots, x_n) > 0$ is not satisfied,

then $g(x_1, x_2, \ldots, x_n) \geq 0$ may or may not be satisfied

To ensure this, we include the following constraints

$$-g(x_1, x_2, \ldots, x_n) \leq My \qquad \text{—— (1)}$$

$$f(x_1, x_2, \ldots, x_n) \leq M(1-y) \qquad \text{—— (2)}$$

$$y = 0 \text{ or } 1$$

當 $f>0$ 滿足，則 (2) 式中之 $y=0$ 才可滿足

$\Rightarrow$ (1) 式中 $-g(x_1, x_2, \ldots, x_n) \leq 0$

or $g(x_1, x_2, \ldots, x_n) \geq 0$ 亦即想要之結果

當 $f>0$ 不被滿足，(2) 式中之 y 可以 $=1$ or 0

選 $y=1$ (1) 式自动被滿足。$-g(x_1, x_2, \ldots, x_n) \leq M$

$\Rightarrow g(x_1, x_2, \ldots, x_n) \geq -M$.(可負的)

$y=0$ (1) 式 $-g(x_1, x_2, \ldots, x_n) \leq 0$

$\Rightarrow g(x_1, x_2, \ldots, x_n) \geq 0$

亦即 $f>0$ 不被滿足，

$g<0$ or $g \geq 0$ 均可能

If $X_{11}=1$ then $X_{21}=X_{31}=X_{41}=0$

$\forall X_{ij}=\{0, 1\}$

上式可重写成

If $X_{11}>0$ then $\underline{X_{21}+X_{31}+X_{41}} \leq 0$

$\Rightarrow -X_{21}-X_{31}-X_{41} \geq 0$

令 $f=X_{11}$, $g=-X_{21}-X_{31}-X_{41}$

∴使用(1)(2)式

$$X_{21}+X_{31}+X_{41} \leq My$$

$$X_{11} \leq M(1-y)$$

$$y=\{0, 1\}$$

since $X_{21}+X_{31}+X_{41}$ 不会大於 3.

$$X_{21}+X_{31}+X_{41} \leq 3y$$

$$X_{11} \leq 3(1-y)$$

$$y=\{0, 1\}$$

9.2 INTEGER PROGRAMMING ALGORITHMS

The ILP algorithms are based on exploiting the tremendous computational success of LP. The strategy of these algorithms involves three steps.

Step 1. Relax the solution space of the ILP by deleting the integer restriction on all integer variables and replacing any binary variable y with the continuous range $0 \leq y \leq 1$. The result of the relaxation is a regular LP.

Step 2. Solve the LP, and identify its continuous optimum.

Step 3. Starting from the continuous optimum point, add special constraints that iteratively modify the LP solution space in a manner that eventually renders an optimum extreme point satisfying the integer requirements.

Two general methods have been developed for generating the special constraints in step 3.

1. Branch-and-bound (B&B) method
2. Cutting-plane method

Neither method is consistently effective computationally. However, experience shows that the B&B method is far more successful than the cutting-plane method.

Maximize $z = 5x_1 + 4x_2$.

s.t.

$$x_1 + x_2 \leq 5$$
$$10x_1 + 6x_2 \leq 45$$

x_1, x_2 nonnegative integer.

§ Branch-and-Bound (B&B) Algorithm

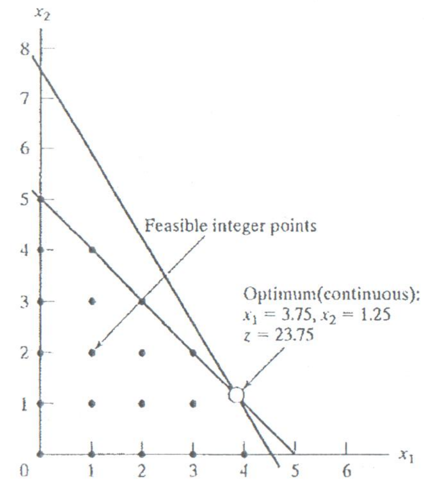


FIGURE 9.5
ILP solution space of Example 9.2-1

ILP optimum. First, we select one of the integer variables whose optimum value at LP0 is not integer. Selecting x_1 (=3.75) arbitrarily, the region $3 < x_1 < 4$ of the LP0 solution space contains no integer values of x_1 and can be eliminated as nonpromising. This is equivalent to replacing the original LP0 with two new LPs, LP1 and LP2, defined as

$$\text{LP1 space} = \text{LP0 space} + (x_1 \le 3)$$ 加入 constraint

$$\text{LP2 space} = \text{LP0 space} + (x_1 \ge 4)$$

Figure 9.6 depicts the LP1 and LP2 spaces. The two spaces contain the same feasible integer points of the original ILP, which means that, from the standpoint of the integer solution, dealing with LP1 and LP2 is the same as dealing with the original LP0.

If we *intelligently* continue to remove the regions that do not include integer solutions by imposing the appropriate constraints (e.g., $3 < x_1 < 4$ at LP0), we will eventually produce LPs whose optimum extreme points satisfy the integer restrictions. In effect, we will be solving the ILP by dealing with a succession of (continuous) LPs.

The new restrictions, $x_1 \le 3$ and $x_1 \ge 4$, are mutually exclusive, so that LP1 and LP2 must be dealt with as separate LPs as Figure 9.7 shows. This dichotomization gives rise to the concept of **branching** in the B&B algorithm with x_1 being the branching variable. 分枝變數

The optimum ILP lies in either LP1 or LP2. Hence, both subproblems must be examined. We arbitrarily examine LP1 (associated with $x_1 \le 3$) first. 任意取

$$\text{Maximize } z = 5x_1 + 4x_2$$

subject to

$$\begin{aligned} x_1 + x_2 &\le 5 \\ 10x_1 + 6x_2 &\le 45 \\ x_1 &\le 3 \\ x_1, x_2 &\ge 0 \end{aligned}$$

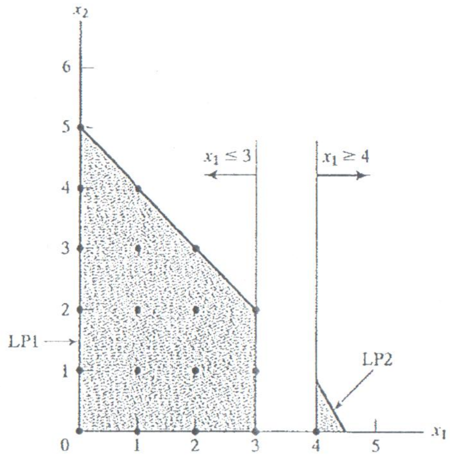


FIGURE 9.6
Solution spaces of LP1 and LP2 for Example 9.2-1

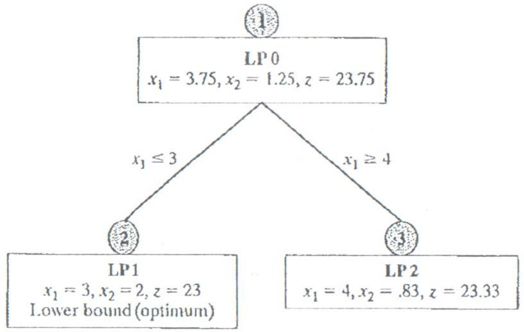


FIGURE 9.7
Using branching variable x_j to create LP1 and LP2 for Example 9.2-1

Max → Lower bound

Min → Upper bound

The solution of LP1 (which can be solved efficiently by the upper-bounded algorithm of Section 7.3) yields the optimum solution

$$x_1 = 3, x_2 = 2, \text{ and } z = 23$$

The LP1 solution satisfies the integer requirements for x_1 and x_2. Hence, LP1 is said to be **fathomed**. This means that LP1 need not be investigated any further because it cannot yield any *better* ILP solution.

We cannot at this point say that the integer solution obtained from LP1 is optimum for the original problem because LP2 may yield a better integer solution (with a higher value of z). All we can say is that $z = 23$ is a **lower bound** on the optimum (maximum) objective value of the original ILP. This means that any unexamined subproblem that cannot yield a better objective value than the lower bound must be discarded as nonpromising. If an unexamined subproblem produces a better integer solution, then the lower bound must be updated accordingly.

fathomed 三條件, p 378

Given the lower bound $z = 23$, we examine LP2 (the only remaining unexamined subproblem). Because optimum $z = 23.75$ at LP0 *and all the coefficients of the objective function happen to be integers,* it is impossible that LP2 (which is more restrictive than LP0) will produce a better integer solution. As a result, we discard LP2 and conclude that it has been *fathomed.*

$x_2 = 0$ or 1 (違反限制式)

The B&B algorithm is now complete because both LP1 and LP2 have been examined and fathomed (the first for producing an integer solution and the second for showing that it cannot produce a better integer solution). We thus conclude that the optimum ILP solution is the one associated with the lower bound—namely, $x_1 = 3$, $x_2 = 2$, and $z = 23$.

$x_1 = 4$, $x_2 = 0$, $z = 20 < 23$

Two questions remain unanswered regarding the procedure:

1. At LP0, could we have selected x_2 as the *branching variable* in place of x_1?
2. When selecting the next subproblem to be examined, could we have solved LP2 first instead of LP1?

The answer to both questions is "yes." However, ensuing computations could differ dramatically. Figure 9.8, in which LP2 is examined first, illustrates this point. The optimum LP2 solution is $x_1 = 4$, $x_2 = .83$, and $z = 23.33$ (verify using TORA LP module). Because x_2 (=.83) is noninteger, LP2 is investigated further by creating subproblems LP3 and LP4 using the branches $x_2 \le 0$ and $x_2 \ge 1$, respectively. This means that

$$\begin{aligned}\text{LP3 space} &= \text{LP2 space} + (x_2 \le 0)\\ &= \text{LP0 space} + (x_1 \ge 4) + (x_2 \le 0)\\ \text{LP4 space} &= \text{LP2 space} + (x_2 \ge 1)\\ &= \text{LP0 space} + (x_1 \ge 4) + (x_2 \ge 1)\end{aligned}$$

We have three "dangling" subproblems that must be examined: LP1, LP3, and LP4. Suppose that we arbitrarily examine LP4 first. LP4 has no solution, and hence it is fathomed. Next, let us examine LP3. The optimum solution is $x_1 = 4.5$, $x_2 = 0$, and $z = 22.5$. The noninteger value of x_1 (=4.5) leads to the two branches $x_1 \le 4$ and $x_1 \ge 5$, and the creation of subproblems LP5 and LP6 from LP3.

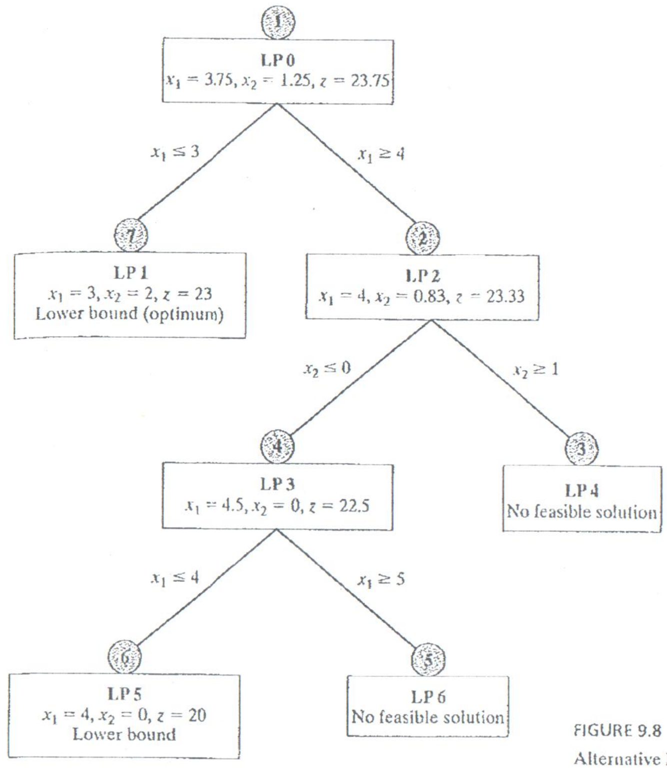
$$\text{LP5 space} = \text{LP0 space} + (x_1 \ge 4) + (x_2 \le 0) + (x_1 \le 4) \equiv \text{LP0 space} + (x_1 = 4) + (x_2 \le 0)$$

$$\text{LP6 space} = \text{LP0 space} + (x_1 \ge 4) + (x_2 \le 0) + (x_1 \ge 5) \equiv \text{LP0 space} + (x_1 \ge 5) + (x_2 \le 0)$$

Now, subproblems LP1, LP5, and LP6 remain unexamined. LP6 is fathomed because it has no feasible solution. Next, LP5 has the integer solution ($x_1 = 4$, $x_2 = 0$, $z = 20$) and, hence, yields a lower bound ($z = 20$) on the optimum ILP solution. We are left with subproblem LP1, whose solution yields a better integer ($x_1 = 3$, $x_2 = 2$, $z = 23$). Thus, the lower bound is updated to $z = 23$. Because *all* the subproblems have been fathomed, the optimum solution is associated with the most up-to-date lower bound—namely, $x_1 = 3$, $x_2 = 2$, and $z = 23$.

The solution sequence in Figure 9.8 (LP0 → LP2 → LP4 → LP3 → LP6 → LP5 → LP1) is a worst-case scenario that, nevertheless, may occur in practice. The example points to a principal weakness of the B&B algorithm: How do we select the next subproblem to be examined, and how do we choose its branching variable?

In Figure 9.7, we were lucky to "stumble" upon a good lower bound at the very first subproblem, LP1, thus allowing us to fathom LP2 without further computations and to terminate the B&B search. In essence, we completed the procedure by solving



fathomed [no feasible solution]

FIGURE 9.8

Alternative B&B tree for Example 9.2-1

one subproblem only. In Figure 9.8, we had to examine seven subproblems before the B&B algorithm could be terminated. Although there are heuristics for enhancing the ability of B&B to "guess" which branch can lead to an improved ILP solution (see Taha, 1975, pp. 154–171), there is no solid theory that will always yield consistent results, and herein lies the difficulty that plagues computations in ILP. Indeed in Section 9.2.2, Problem 1, Set 9.2b, demonstrates with the help of TORA the bizarre behavior of the B&B algorithm, even for a small 16-variable 1-constraint problem, where the optimum is found in 9 iterations (subproblems) but requires over 25,000 iterations to verify optimality. It is no wonder that to this day, and after four decades of research, available computer codes (commercial and academic alike) lack consistency (a la simplex method) in solving ILPs.

We now summarize the B&B algorithm. Assuming a maximization problem, set an initial lower bound $z = -\infty$ on the optimum objective value of ILP. Set $i = 0$.

Step 1. (Fathoming/bounding). Select LPi, the next subproblem to be examined. Solve LPi, and attempt to fathom it using one of three conditions.

(a) The optimal z-value of LPi cannot yield a better objective value than the current lower bound.
(b) LPi yields a better feasible integer solution than the current lower bound.
(c) LPi has no feasible solution.

Two cases will arise.

(a) If LPi is fathomed and a better solution is found, update the lower bound. If *all* subproblems have been fathomed, stop; the optimum ILP is associated with the current lower bound, if any. Otherwise, set $i = i + 1$, and repeat step 1.

(b) If LPi is not fathomed, go to step 2 for branching.

Step 2. (Branching). Select one of the integer variables x_j, whose optimum value x_j^* in the LPi solution is not integer. Eliminate the region

$$[x_j^*] < x_j < [x_j^*] + 1$$

(where $[v]$ defines the largest integer $\leq v$) by creating two LP subproblems that correspond to

$$x_j \leq [x_j^*] \text{ and } x_j \geq [x_j^*] + 1$$

Set $i = i + 1$, and go to step 1.

The given steps apply to maximization problems. For minimization, we replace the lower bound with an upper bound (whose initial value is $z = +\infty$).

The B&B algorithm can be extended directly to mixed problems (in which only some of the variables are integer). If a variable is continuous, we simply never select it as a branching variable. A feasible subproblem provides a new bound on the objective value if the values of the discrete variables are integer and the objective value is improved relative to the current bound.

PROBLEM SET 9.2A

1. Solve the ILP of Example 9.2-1 by the B&B algorithm starting with x_2 as the branching variable. Solve the subproblems with TORA using the MODIFY option for the upper and lower bounds. Start the procedure by solving the subproblem associated with $x_2 \leq [x_2^*]$.
2. Develop the B&B tree for each of the following problems. For convenience, always select x_1 as the branching variable at node 0.

   (a) Maximize $z = 3x_1 + 2x_2$
   subject to
   $$2x_1 + 5x_2 \leq 9$$
   $$4x_1 + 2x_2 \leq 9$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$

   (b) Maximize $z = 2x_1 + 3x_2$
   subject to
   $$5x_1 + 7x_2 \leq 35$$
   $$4x_1 + 9x_2 \leq 36$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$

   (c) Maximize $z = x_1 + x_2$
   subject to
   $$2x_1 + 5x_2 \leq 16$$
   $$6x_1 + 5x_2 \leq 27$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$

   (d) Minimize $z = 5x_1 + 4x_2$
   subject to
   $$3x_1 + 2x_2 \geq 5$$
   $$2x_1 + 3x_2 \geq 7$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$

   (e) Maximize $z = 5x_1 + 7x_2$
   subject to
   $$2x_1 + x_2 \leq 13$$
   $$5x_1 + 9x_2 \leq 41$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$

3. Repeat Problem 2, assuming that x_1 is continuous.
4. Show graphically that the following ILP has no feasible solution, and then verify the result using B&B.
   $$\text{Maximize } z = 2x_1 + x_2$$
   subject to
   $$10x_1 + 10x_2 \leq 9$$
   $$10x_1 + 5x_2 \geq 1$$
   $$x_1, x_2 \geq 0 \text{ and integer}$$
5. Solve the following problems by B&B.
   $$\text{Maximize } z = 18x_1 + 14x_2 + 8x_3 + 4x_4$$
   subject to
   $$15x_1 + 12x_2 + 7x_3 + 4x_4 + x_5 \leq 37$$
   $$x_1, x_2, x_3, x_4, x_5 = (0, 1)$$

9.2.2 TORA-Generated B&B Tree

TORA integer programming module is equipped with a facility for generating the B&B tree interactively. To use this facility, select User-guided B&B in the output screen

§ Cutting-Plane Algorithm

如同 B&B，切面法亦由 optimal LP solution 開始，借由特別的限制式（稱為 cut）加至解空間以產生整數解。

$$\text{Maximize } Z = 7x_1 + 10x_2$$

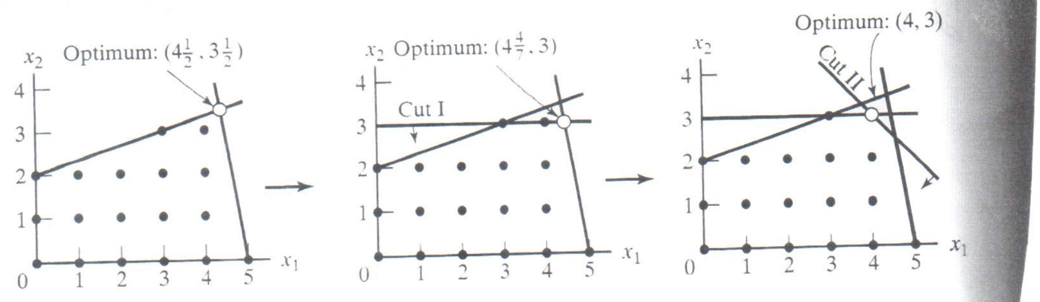
$$-x_1 + 3x_2 \le 6$$

$$7x_1 + x_2 \le 35$$

$$x_1, x_2 \ge 0 \text{ and integer.}$$

FIGURE 9.8

Illustration of the use of cuts in ILP



上圖示範了二個 cuts，首先由 LP optimum $x_1 = 4\frac{1}{2}$, $x_2 = 3\frac{1}{2}$, $Z = 66\frac{1}{2}$ 開始，接著加入 cut I，產生 LP optimum $x_1 = 4\frac{4}{7}$, $x_2 = 3$, $Z = 62$，最後再加入 cut II，產生了 integer LP optimum $x_1 = 4$, $x_2 = 3$, $Z = 58$.

切面法從解LP開始，在LP最佳解表格中選取一列，稱為來源列 source row，由於基本變數不為整數，想要的切割是由來源列的分數部分所構成，所以稱為分數切割（fractional cut）。

給定 x_3, x_4 是 Slack variables 與最佳解表格如下：

BASIC	x_1	x_2	x_3	x_4	Solution
Z	0	0	$\frac{63}{22}$	$\frac{31}{22}$	$66\frac{1}{2}$
x_2	0	1	$\frac{7}{22}$	$\frac{1}{22}$	$3\frac{1}{2}$
x_1	1	0	$\frac{-1}{22}$	$\frac{3}{22}$	$4\frac{1}{2}$

由上述資訊

$$Z + \frac{63}{22}x_3 + \frac{31}{22}x_4 = 66\frac{1}{2}$$

$$x_2 + \frac{7}{22}x_3 + \frac{1}{22}x_4 = 3\frac{1}{2}$$

$$x_1 - \frac{1}{22}x_3 + \frac{3}{22}x_4 = 4\frac{1}{2}$$

只要限制式 RHS 是分數就可以當成來源列。Z, x_2 equation, x_1 equation 均可當成來源列。

要構成分數切割是將非整數係數分成整數與分數兩部分，且分數部

分一定是正的。

例如 $\frac{5}{2} = 2 + \frac{1}{2}$

$-\frac{7}{3} = -3 + \frac{2}{3}$

於是 Z equation 就變成

$$Z + \left(2 + \frac{19}{22}\right)x_3 + \left(1 + \frac{19}{22}\right)x_4 = 66 + \frac{1}{2}$$

將所有整數移至左邊，分數移至右邊.

$$Z + 2x_3 + x_4 - 66 = -\frac{19}{22}x_3 - \frac{19}{22}x_4 + \frac{1}{2}$$

$\because$ x_3, x_4 是非負的 $-\frac{19}{22}x_3 - \frac{19}{22}x_4 + \frac{1}{2} \le \frac{1}{2}$

假設 z, x_3, x_4 是整數，則 $z + 2x_3 + x_4 - 66$ 也是整數，於是 $-\frac{19}{22}x_3 - \frac{19}{22}x_4 + \frac{1}{2}$ 是整數且 $\le \frac{1}{2}$，所以 $\Rightarrow -\frac{19}{22}x_3 - \frac{19}{22}x_4 + \frac{1}{2} \le 0$

這就是想要的分數切割。

再看 x_1 equation,

$$x_1 - \frac{1}{22}x_3 + \frac{3}{22}x_4 = 4\frac{1}{2}$$

$$x_1 + \left(-1 + \frac{21}{22}\right)x_3 + \left(0 + \frac{3}{22}\right)x_4 = 4 + \frac{1}{2}$$

$$x_1 - x_3 + 0x_4 - 4 = -\frac{21}{22}x_3 - \frac{3}{22}x_4 + \frac{1}{2}$$

伴隨的切割為

$$-\frac{21}{22}x_3 - \frac{3}{22}x_4 + \frac{1}{2} \le 0$$

同理 x_2-equation

$$x_2 + \frac{7}{22}x_3 + \frac{1}{22}x_4 = 3\frac{1}{2}$$

$$x_2 + (0+\frac{7}{22})x_3 + (0+\frac{1}{22})x_4 = 3 + \frac{1}{2}$$

伴隨之切割為 $-\frac{7}{22}x_3 - \frac{1}{22}x_4 + \frac{1}{2} \le 0$

上述三個切割均可被使用.

任選擇 $-\frac{7}{22}x_3 - \frac{1}{22}x_4 + S_1 = -\frac{1}{2}$, $S_1 \ge 0$ cut I.

BASIC	x_1	x_2	x_3	x_4	S_1	Solution
Z	0	0	$\frac{63}{22}$	$\frac{31}{22}$	0	$66\frac{1}{2}$
x_2	0	1	$\frac{7}{22}$	$\frac{1}{22}$	0	$3\frac{1}{2}$
x_1	1	0	$\frac{-1}{22}$	$\frac{3}{22}$	0	$4\frac{1}{2}$
S_1	0	0	$\frac{-7}{22}$	$\frac{-1}{22}$	1	$-\frac{1}{2}$

上表是 optimal 但 infeasible, 使用 dual simplex 獲得可行解.

BASIC	x_1	x_2	x_3	x_4	S_1	Solution
Z	0	0	0	1	9	62
x_2	0	1	0	0	1	3
x_1	1	0	0	$\frac{1}{7}$	$\frac{-1}{7}$	$4\frac{4}{7}$
x_3	0	0	1	$\frac{1}{7}$	$\frac{-22}{7}$	$1\frac{4}{7}$

上表中 x_1 與 x_3 仍為分數, 任選 x_1 equation,

$$x_1 + \left(0 + \frac{1}{7}\right) x_4 + \left(-1 + \frac{6}{7}\right) s_1 = \left(4 + \frac{4}{7}\right)$$

$$-\frac{1}{7} x_4 - \frac{6}{7} s_1 + \frac{4}{7} \le \frac{4}{7} \Rightarrow -\frac{1}{7} x_4 - \frac{6}{7} s_1 + \frac{4}{7} \le 0$$

$$-\frac{1}{7} x_4 - \frac{6}{7} s_1 + S_2 = -\frac{4}{7} \quad S_2 \ge 0, \quad \text{cut II.}$$

BASIC	x_1	x_2	x_3	x_4	s_1	s_2	Solution
Z	0	0	0	1	9	0	62
x_2	0	1	0	0	1	0	3
x_1	1	0	0	$\frac{1}{7}$	$\frac{-1}{7}$	0	$4\frac{4}{7}$
x_3	0	0	1	$\frac{1}{7}$	$\frac{-22}{7}$	0	$1\frac{4}{7}$
S_2	0	0	0	$\frac{-1}{7}$	$\frac{-6}{7}$	1	$-\frac{4}{7}$

應用 dual simplex 求解

BASIC	x_1	x_2	x_3	x_4	s_1	s_2	Solution
Z	0	0	0	0	3	7	58
x_2	0	1	0	0	1	0	3
x_1	1	0	0	0	-1	1	4
x_3	0	0	1	0	-4	1	1
x_4	0	0	0	1	6	-7	4