

Ch12 Deterministic Dynamic Programming

確定性動態規劃

12.1 動態規劃(DP)計算的遞迴特質

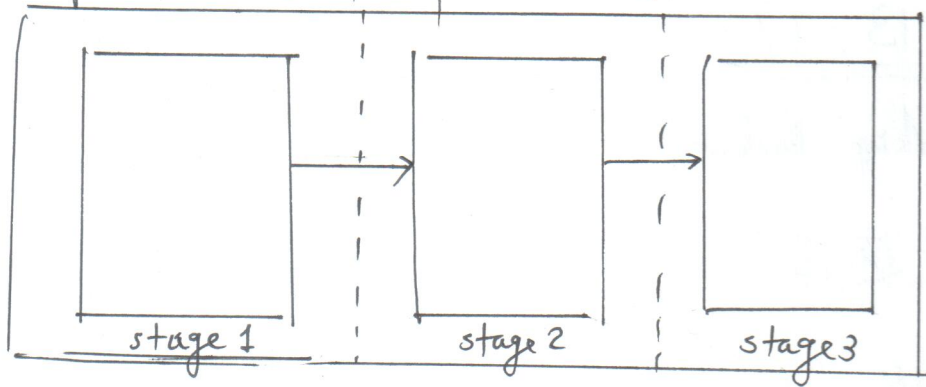
動態規劃是一種教學的方法，用於擬訂一連串相關的決策，提供有系統的程序，以訂立一組使全盤效用的決策。DP沒有一套標準的教學模式，而每一問題所用的方程式，必須適合個別的情形。

DP的計算是透過一連串的 Recursive Computations (將一大問題分成許多 stages 階段處理)。DP的名稱是因為他的決策通常包含不同的時間，有些問題，時間在問題中不是一主要因素，也可以用DP處理，所以它的另一名稱是 multistage programming 多階段規劃。

[遊戲] - 一次可數 1~3 個數字，最後數到 30 的人輸，就是一個 DP 的例子。

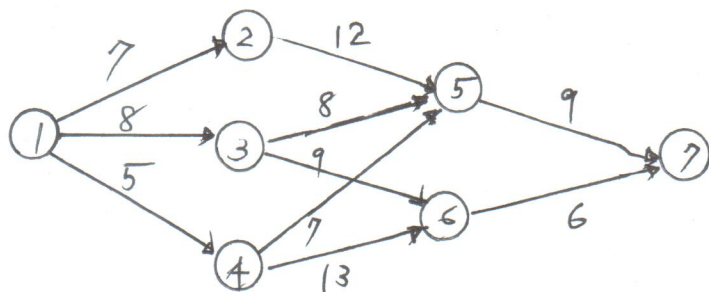
DP model 的名稱與性質

1. 問題可分成不同的 stages (階段). 在每一階段必需做一決策 (Decision), 所謂 stage 就是將問題分成很多部分, stage 就是其中的一部分, 這一部分包含了許多 mutually exclusive alternatives (互斥方案), 而我們必需去選擇一方案。
2. 每一階段有若干 states (狀態) 與之伴隨, 狀態, 系指在階段內的限制或可能方案, 適當的選擇狀態, 才能將不同階段連接在一起。
3. DP 遞迴計算性質



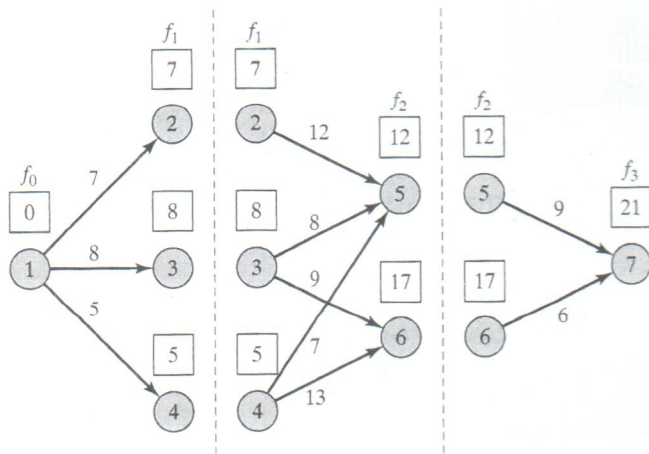
將大問題切成若干階段, 每一階段解決該子問題, 將其最佳解當成下一階段之 input, 再解決這一階段問題。並將最佳解當成下一階段之 input, 當解完最後一階段子問題時, 即所有問題之最佳解。

12.1-1 最短路徑問題



要找到 $1 \rightarrow 7$ 最短路徑，前面已經學過 Dijkstra algorithm，亦可用 exhaustive enumeration，但問題大 exhaustive 則變的不可行。可將問題分成三個 stages，然後在每一 stage 分別計算。

FIGURE 12.2
Decomposition of the shortest-route problem into stages



垂直虛綫代表不同階段，計算每一階段該節點之最短路徑，再將其當成下一階段之 input，然後計算該階段各節點之最短路徑。

由 12.2 可知最短距離為 21, 路徑為

$$1 \rightarrow 4 \rightarrow 5 \rightarrow 7$$

DP 的遞迴計算可以數學式表示, 也可以 tableau form (表格型式) 簡化計算.

令 $f_i(x_i)$ 表示階段 i , 至節點 x_i 的最短距離。 $d(x_{i-1}, x_i)$: 節點 x_{i-1} 至 x_i 的距離。 f_i 可由 f_{i-1} 計算而得.

$$f_i(x_i) = \min_{\substack{\text{all feasible} \\ (x_{i-1}, x_i) \text{ nodes}}} \{ d(x_{i-1}, x_i) + f_{i-1}(x_{i-1}) \}$$

$i=1, 2, 3$

定義 $f_0(x_0)=0$ — 開始距離為 0,

從 $i=1$, 開始。 i 代表階段。

x_i 代表階段 i 之狀態, 在本例中就是距離。(至該節點之最短距離)
將各階段連接在一起。

12.2 FORWARD AND BACKWARD RECURSION

上節的例子是 FORWARD, 而 BACKWARD 由 stage 3 開始, 結束於 stage 1, FORWARD 與 Backward 產生相同之結果, 但 DP 一般均使用 Backward 因為計算上較有效率。

Example 12.2-1

The backward recursive equation for Example 12.2-1 is

$$f_4(x_4 = 7) = 0$$

$$f_i(x_i) = \min_{\substack{\text{all feasible} \\ \text{routes } (x_i, x_{i+1})}} \{d(x_i, x_{i+1}) + f_{i+1}(x_{i+1})\}, i = 1, 2, 3$$

The order of computations is $f_3 \rightarrow f_2 \rightarrow f_1$.

Stage 3. Node 7 ($x_4 = 7$) is connected to nodes 5 and 6 ($x_3 = 5$ and 6) with exactly one route each. The following table summarizes stage 3 computations:

x_3	$d(x_3, x_4)$	Optimum solution	
	$x_4 = 7$	$f_3(x_3)$	x_4^*
5	9	9	7
6	6	6	7

Stage 2. Route (2, 6) does not exist. Given $f_3(x_3)$ from stage 3, we can compare the feasible alternatives as the following table shows:

x_2	$d(x_2, x_3) + f_3(x_3)$		Optimum solution	
	$x_3 = 5$	$x_3 = 6$	$f_2(x_2)$	x_3^*
2	$12 + 9 = 21$	—	21	5
3	$8 + 9 = 17$	$9 + 6 = 15$	15	6
4	$7 + 9 = 16$	$13 + 6 = 19$	16	5

The optimum solution of stage 2 reads as follows: For cities 2 and 4, the shortest routes pass through city 5, and for city 3, the shortest route passes through city 6.

Stage 1. From node 1, we have three alternative routes: (1, 2), (1, 3), and (1, 4). Using $f_2(x_2)$ from stage 2, we get

x_1	$d(x_1, x_2) + f_2(x_2)$			Optimum solution	
	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$f_1(x_1)$	x_2^*
1	$7 + 21 = 28$	$8 + 15 = 23$	$5 + 16 = 21$	21	4

Stage 1 solution links city 1 to city 4. Next, stage 2 solution links city 4 to city 5. Finally, stage 3 solution connects city 5 to city 7. The optimum route is $1 \rightarrow 4 \rightarrow 5 \rightarrow 7$, and the associated distance is 21 miles.

12.3 Selected DP Applications

當你學習每個 DP application 時，多注意 DP 三項基本元素。

1. 定義 stages
2. 每階段可行方案之定義
3. 每階段中狀態之定義

在上述三項中，以第三項如何定義狀態最為微妙，也最不容易（初學者最易犯錯）。

通常考慮，下列問題有助於寫出狀態，

1. 何種關係可以把不同階段連接在一起？
2. 我們在目前階段做決策而不需要去檢查前一階段的可行性，需要什麼訊息？

12.3.1. Knapsack / Fly-Away Kit / Cargo-Loading Model

Knapsack problem 背包問題起源於登山者要打包登山背包，他必需選擇最重要

的東西，因為背包能承受的重量有限。
背包問題可以下列模式代表

$$\text{Maximize } Z = r_1 m_1 + r_2 m_2 + \dots + r_n m_n$$

s.t.

$$w_1 m_1 + w_2 m_2 + \dots + w_n m_n \leq W$$

$m_1, m_2, \dots, m_n$ nonnegative integers

其中 W 背包之承重

m_i : 項目 i 之裝載數目

w_i : 項目 i 之重量

r_i : 項目 i 之單位收益

以 DP model 之三項元素為

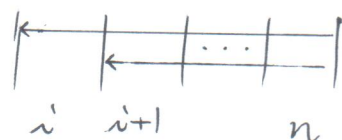
1. stage i : 貨物項目 i , $i=1, \dots, n$
 2. 在階段 i 的可行方案是項目 i 的裝載數目，以符號 m_i 表示。 $m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor$
 3. 階段 i 的狀態，以 X_i 符號代表，在本例中分配至階段 $i, i+1, \dots, n$ 的可裝載總重量。(backward)
- 定義 $f_i(X_i)$: 給定狀態 X_i ，在階段 $i, i+1, \dots, n$ 的最大收益。

再接再下來決定 recursive equation

$$f_i(x_i) = \max_{\substack{m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor \\ x_i \leq W}} \{ r_i m_i + f_{i+1}(x_{i+1}) \} \quad i=1, \dots, n$$

其中 $f_{i+1}(x_{i+1}) = 0$

為了確保 $f_i(x_i)$ 是 x_i 之函數， $\therefore f_{i+1}(x_{i+1})$ 必需稍做修改。令 $x_i - x_{i+1}$ 為階段 i 的裝載量



$$x_i - x_{i+1} = w_i m_i$$

$$\therefore x_{i+1} = x_i - w_i m_i$$

$$\text{於是 } f_i(x_i) = \max_{\substack{m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor \\ x_i \leq W}} \{ r_i m_i + f_{i+1}(x_i - w_i m_i) \} \quad i=1, \dots, n$$

Example 12.3-1

有一艘可載重 4 噸的船，可裝載一項或三項貨物。

項目	w_i (噸)	r_i 收益每單位千元
1	2	31
2	3	47
3	1	14

目標為決定每項貨物之裝載數目，使總收益最大。

Because the unit weight w_i and the maximum weight W are integers, the state x_i assumes integer values only.

Stage 3. The exact weight to be allocated to stage 3 (item 3) is not known in advance but can assume one of the values 0, 1, . . . and 4 (because $W = 4$ tons and $w_3 = 1$ ton). A value of m_3

²The definition of the state can be multidimensional. For example, the volume of the knapsack may pose another restriction. In general, a multidimensional state implies more complex stage calculations. See Section 12.4.

is feasible only if $w_3 m_3 \leq x_3$. Thus, all the infeasible values (with $w_3 m_3 > x_3$) are excluded. The revenue for item 3 is $14m_3$. Thus, the recursive equation for stage 3 is

$$f_3(x_3) = \max_{m_3=0,1,\dots,4} \{14m_3\}$$

The following tableau summarizes the computations for stage 3:

x_3	$14m_3$					Optimum solution	
	$m_3 = 0$	$m_3 = 1$	$m_3 = 2$	$m_3 = 3$	$m_3 = 4$	$f_3(x_3)$	m_3^*
0	0	—	—	—	—	0	0
1	0	14	—	—	—	14	1
2	0	14	28	—	—	28	2
3	0	14	28	42	—	42	3
4	0	14	28	42	56	56	4

Stage 2. $\max \{m_2\} = \left\lfloor \frac{4}{3} \right\rfloor = 1$, or $m_3 = 0, 1$, $f_2(x_2) = \max_{m_3=0,1} \{47m_2 + f_3(x_2 - 3m_2)\}$

x_2	$47m_2 + f_3(x_2 - 3m_2)$		Optimum solution	
	$m_2 = 0$	$m_2 = 1$	$f_2(x_2)$	m_2^*
0	$0 + 0 = 0$	—	0	0
1	$0 + 14 = 14$	—	14	0
2	$0 + 28 = 28$	—	28	0
3	$0 + 42 = 42$	$47 + 0 = 47$	47	1
4	$0 + 56 = 56$	$47 + 14 = 61$	61	1

Stage 1. $\max \{m_1\} = \left\lfloor \frac{4}{2} \right\rfloor = 2$ or $m_1 = 0, 1, 2$, $f_1(x_1) = \max_{m_1=0,1,2} \{31m_1 + f_2(x_1 - 2m_1)\}$

x_1	$31m_1 + f_2(x_1 - 2m_1)$			Optimum solution	
	$m_1 = 0$	$m_1 = 1$	$m_1 = 2$	$f_1(x_1)$	m_1^*
0	$0 + 0 = 0$	—	—	0	0
1	$0 + 14 = 14$	—	—	14	0
2	$0 + 28 = 28$	$31 + 0 = 31$	—	31	1
3	$0 + 47 = 47$	$31 + 14 = 45$	—	47	0
4	$0 + 61 = 61$	$31 + 28 = 59$	$62 + 0 = 62$	62	2

The optimum solution is determined in the following manner: Given $W = 4$ tons, from stage 1, $x_1 = 4$ gives the optimum alternative $m_1^* = 2$ —meaning that 2 units of item 1 will be loaded on the vessel. This allocation leaves $x_2 = x_1 - 2m_1^* = 4 - 2 \times 2 = 0$ for stages 2 and 3. From stage 2, $x_2 = 0$ yields $m_2^* = 0$, which leaves $x_3 = x_2 - 3m_2^* = 0 - 3 \times 0 = 0$ units for stage 3. Next, from stage 3, $x_3 = 0$ gives $m_3^* = 0$. Thus, the complete optimal solution is $m_1^* = 2$, $m_2^* = 0$, and $m_3^* = 0$. The associated return is $f_1(4) = \$62,000$.

12.3.3 Equipment Replacement Model

機器使用越久維修費用會越高，生產力降低，當機器到達某一年限重置該機器或許更經濟。

假設機器重置問題可展延至 n 年，在每年開始時，我們決定保留機器再用一年或是重置新的機器。

令 $r(t)$ 、 $c(t)$ 分別代表 t 年機器的年收益與年成本。 $s(t)$ 代表 t 年機器的殘值。獲得新機器的成本是 I 。

則 DP 的元素為

1. stage i 代表年 i ， $i=1, \dots, n$
2. alternative 可行方案在 stage i (年 i)，是在第 i 年初保留或換新機器
3. 在 stage i 的 state 是在 i 年開始時機器的年齡。

定義

$f_i(t)$ = 給定在 i 年開始時，機器年齡是 t ，階段 $i, i+1, \dots, n$ 之最大收益

recursive equation

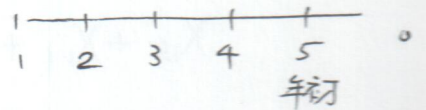
$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1) & \text{if keep} \\ r(0) + s(t) - I - c(0) + f_{i+1}(1) & \text{if replace} \end{cases}$$

其中 $f_{n+1}(\cdot) = 0$

$$f_n(t) = \max \begin{cases} r(t) - c(t) + s(t+1) & \text{if keep} \\ r(0) + s(t) + s(1) - I - c(0) & \text{if replace} \end{cases}$$

Example 12.3-3

一公司需決定最佳重置策略，目前機器年齡 3 years，未來四年之決策。



下表給予相關資料。公司要求機器 6 年必需重置，新機器成本 \$100,000

Age, t (yr)	Revenue, $r(t)$ (\$)	Operating cost, $c(t)$ (\$)	Salvage value, $s(t)$ (\$)
0	20,000	200	—
1	19,000	600	80,000
2	18,500	1200	60,000
3	17,200	1500	50,000
4	15,500	1700	30,000
5	14,000	1800	10,000
6	12,200	2200	5000

在每一階段決定機器可行的年齡，不是很顯而易見，Figure 12.6 的網路圖代表上述問題。

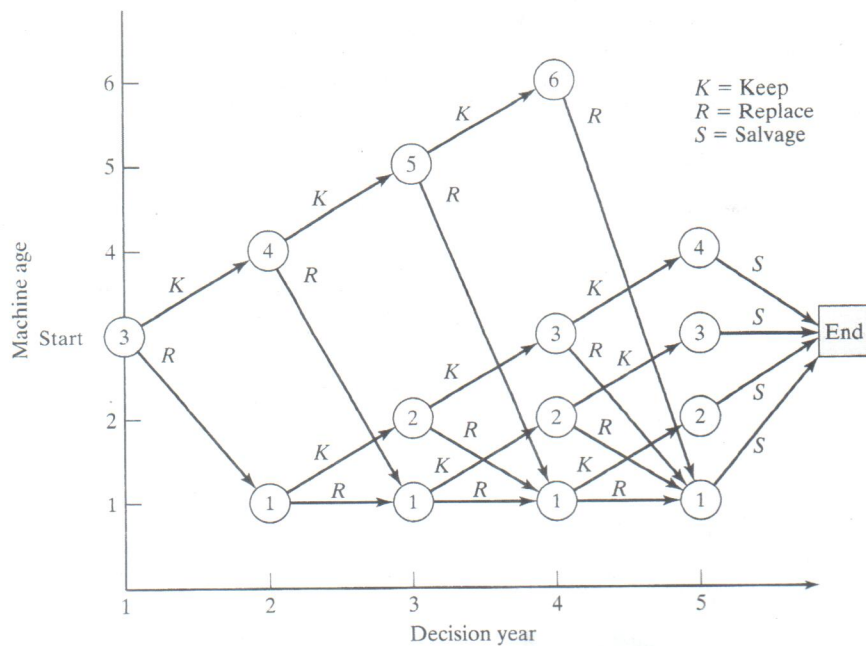


FIGURE 12.6

Representation of machine age as a function of decision year in Example 12.3-3

在第一年開始時，我們有一台機齡3年的機器，可以選擇保留或重置，如果保留機器在第二年初，機器之機齡為4，如果重置機器機器之機齡為1。相同的邏輯可應用至第2至第4年開始，在第四年初，6年的機器必須重置，在第四年底，則報廢(殘值)機器。

上述問題如同要尋找從第一年開始至第四年底的最長路徑。將使用下列表格型式，求解下列問題。

Stage 4.

t	K	R	Optimum solution	
	$r(t) + s(t+1) - c(t)$	$r(0) + s(t) + s(1) - c(0) - I$	$f_4(t)$	Decision
1	$19.0 + 60 - .6 = 78.4$	$20 + 80 + 80 - .2 - 100 = 79.8$	79.8	R
2	$18.5 + 50 - 1.2 = 67.3$	$20 + 60 + 80 - .2 - 100 = 59.8$	67.3	K
3	$17.2 + 30 - 1.5 = 45.7$	$20 + 50 + 80 - .2 - 100 = 49.8$	49.8	R
6	(Must replace)	$20 + 5 + 80 - .2 - 100 = 4.8$	4.8	R

Stage 3.

t	K	R	Optimum solution	
	$r(t) - c(t) + f_4(t+1)$	$r(0) + s(t) - c(0) - I + f_4(1)$	$f_3(t)$	Decision
1	$19.0 - .6 + 67.3 = 85.7$	$20 + 80 - .2 - 100 + 79.8 = 79.6$	85.7	K
2	$18.5 - 1.2 + 49.8 = 67.1$	$20 + 60 - .2 - 100 + 79.8 = 59.6$	67.1	K
5	$14.0 - 1.8 + 4.8 = 17.0$	$20 + 10 - .2 - 100 + 79.8 = 9.6$	17.0	R

Stage 2.

t	K	R	Optimum solution	
	$r(t) - c(t) + f_3(t+1)$	$R(0) + s(t) - c(0) - I + f_3(1)$	$f_2(t)$	Decision
1	$19.0 - .6 + 67.1 = 85.5$	$20 + 80 - .2 - 100 + 85.7 = 85.5$	85.5	K or R
4	$15.5 - 1.7 + 17.0 = 30.8$	$20 + 30 - .2 - 100 + 85.7 = 35.5$	35.5	R

Stage 1.

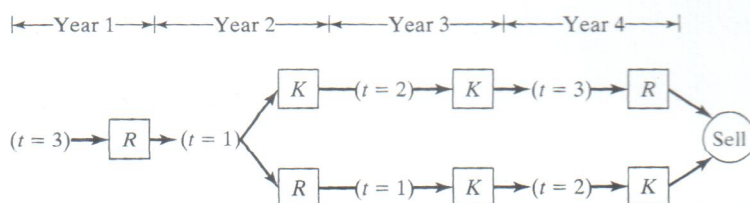
t	K	R	Optimum solution	
	$r(t) - c(t) + f_2(t+1)$	$R(0) + s(t) - c(0) - I + f_2(1)$	$f_1(t)$	Decision
3	$17.2 - 1.5 + 35.5 = 51.2$	$20 + 50 - .2 - 100 + 85.5 = 55.3$	55.3	R

Figure 12.7 summarizes the optimal solution. At the start of year 1, given $t = 3$, the optimal decision is to replace the machine. Thus, the new machine will be 1 year old at the start of year 2, and $t = 1$ at the start of year 2 calls for either keeping or replacing the machine. If it is replaced, the new machine will be 1 year old at the start of year 3; otherwise, the kept machine will be 2 years old. The process is continued in this manner until year 4 is reached.

The alternative optimal policies starting in year 1 are (R, K, K, R) and (R, R, K, K) . The total cost is \$55,300.

FIGURE 12.7

Solution of Example 12.3-3



12.4 Problem of Dimensionality

到目前為止，在任何 stage, state 均以一個變數代表。當 state 數目增加時，在每階段的計算亦會增大，隨著問題加大，計算困難增加。在文獻上稱之為 curse of dimensionality。下面一例子用來示範 curse of dimensionality (以 DP 求解 LP)

Example 12.4-1

Acme 生產兩種產品，產品 1 與產品 2 的每單位利潤分別為 \$2 與 \$5。Acme 每天製造時間為 430 minutes，產品 1 與產品 2 之製造時間分別為 2 minutes 與 1 minute，產品 2 每日最大需求為 230 單位。

$$\begin{aligned} \text{Max } Z &= 2x_1 + 5x_2 \\ \text{s.t. } 2x_1 + x_2 &\leq 430 \\ x_2 &\leq 230. \end{aligned}$$

$$x_1, x_2 \geq 0$$

DP 的元素如下

1. stage i : 產品 i ($i=1, 2$)
2. Alternative: x_i 是產品 i 的數量

3. state (V_2, W_2) : 在階段 2, 資源 1 與資源 2 的使用數量 (生產時間與需求限制)

4. state (V_1, W_1) : 在階段 1, 資源 1 與資源 2 的使用數量.

階段 2

定義 $f_2(V_2, W_2)$ 為給定 (V_2, W_2) , 在階段 2 之最大利潤 (產品 2).

$$f_2(V_2, W_2) = \max \left\{ 5x_2 \right. \\ \left. \begin{array}{l} 0 \leq x_2 \leq V_2 \\ 0 \leq x_2 \leq W_2 \end{array} \right.$$

於是 $x_2 = \min\{V_2, W_2\}$ $\because$ 限制式需同時成立

state	Optimum Solution	
	$f_2(V_2, W_2)$	x_2
(V_2, W_2)	$5 \min\{V_2, W_2\}$	$\min\{V_2, W_2\}$

階段 1

$$f_1(V_1, W_1) = \max \left\{ 2x_1 + f_2(V_1 - 2x_1, W_1) \right. \\ \left. 0 \leq 2x_1 \leq V_1 \right.$$

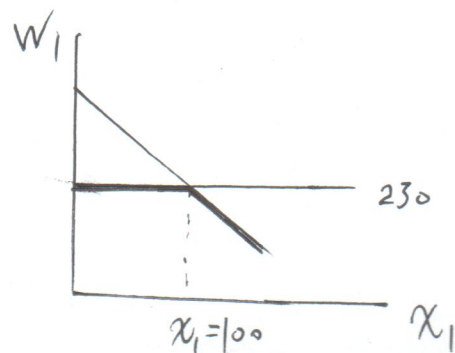
$$= \max \left\{ 2x_1 + 5 \min\{V_1 - 2x_1, W_1\} \right. \\ \left. 0 \leq 2x_1 \leq V_1 \right.$$

問題變成一種 minimax 的問題, 一般而言

求解是困難的。令 $V_1 = 430$, $W_1 = 230$
於是 $0 \leq 2x_1 \leq 430$

$$\therefore \min(430 - 2x_1, 230)$$

$$= \begin{cases} 230 & 0 \leq x_1 \leq 100 \\ 430 - 2x_1, & 100 \leq x_1 \leq 215 \end{cases}$$



$$\text{於是 } f_1(430, 230) = \max_{0 \leq x_1 \leq 215} \{2x_1 + 5 \min(430 - 2x_1, 230)\}$$

$$= \max_{x_1} \begin{cases} 2x_1 + 5 \cdot 230 & 0 \leq x_1 \leq 100 \\ 2x_1 + 5(430 - 2x_1) & 100 \leq x_1 \leq 215 \end{cases}$$

$$= \max_{x_1} \begin{cases} 2x_1 + 1150 & 0 \leq x_1 \leq 100 \\ -8x_1 + 2150 & 100 \leq x_1 \leq 215 \end{cases}$$

可以圖形驗證 $f_1(430, 230)$ 最佳解發生在

$x_1 = 100$ 時，於是得到

State	Optimum solution	
	$f_1(V_1, W_1)$	x_1
(430, 230)	1350	100

為決定 x_2 最佳解, $V_2 = V_1 - 2x_1 = 430 - 200 = 230$

$W_2 = W_1 - 0 = 230$ 於是 $x_2 = \min\{V_2, W_2\} = 230$.

最後解答 $x_1 = 100$, $x_2 = 230$ $Z = 1350$