4. 2 動態規劃法 (Dynamic Programming)

例 4-3:(多階圖網的最短路線問題 The Shortest Path Problem in A Multi-Stage Graph):

以下之圖網是較特殊的一種，它的結點可分為幾個群組，並可排出群組的順序，每一結點只能和相鄰群組的結點有連結，我們稱這種圖網為多階圖網(Multi-Stage Graph)，**由於它有很好的特性，. . . , 它也是很好的例子用來解釋動態規劃解法的原理。**

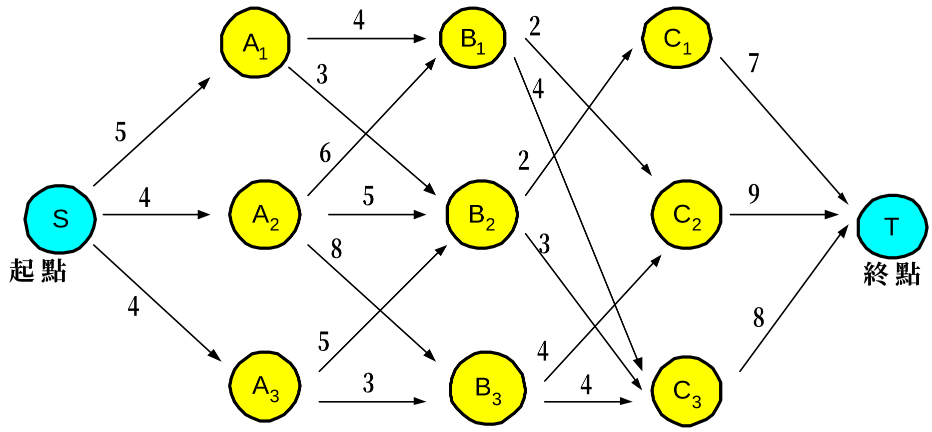

圖 一: 最短路徑問題

後推式解法(Backward approach) -- **令**

$g^*(X)$ = X **到** T **之最短路線長度**

$f(X, Y)$ = X **到** Y **之長度** (X , Y **相鄰**)

$g^*(S)$ = S **至** T **之最短路徑長度(最佳解)**

遞迴思考:

(1) **如果我們知道** $g^*(A_1)$, $g^*(A_2)$, $g^*(A_3)$, **則可得最佳解，則最短路徑長度可以下式的遞迴關係求得**

$$g^*(S) = \text{minimum} \{ 5 + g^*(A_1), \quad 3 + g^*(A_2), \quad 7 + g^*(A_3) \}$$

(2) **如何求得** $g^*(A_1)$, $g^*(A_2)$, $g^*(A_3)$?

$$g^*(A_1) = \text{minimum} \{ 4 + g^*(B_1), \quad 3 + g^*(B_2) \}$$

$$g^*(A_2) = \text{minimum} \{ 6 + g^*(B_1), \quad 4 + g^*(B_2), \quad 8 + g^*(B_3) \}$$

$$g^*(A_3) = \text{minimum} \{ 5 + g^*(B_2), \quad 3 + g^*(B_3) \}$$

(3) **如何求得** $g^*(B_1)$, $g^*(B_2)$, $g^*(B_3)$?

$$g^*(B_1) = \text{minimum}\ \{\ 2 + g^*(C_2),\quad 1 + g^*(C_3)\ \}$$

$$g^*(B_2) = \text{minimum}\ \{\ 2 + g^*(C_1),\quad 1 + g^*(C_3)\ \}$$

$$g^*(B_3) = \text{minimum}\ \{\ 1 + g^*(C_2),\quad 5 + g^*(C_3)\ \}$$

(4) **如何求得** $g^*(C_1)$, $g^*(C_2)$, $g^*(C_3)$?

很明顯 $g^*(C_1) = 7$, $g^*(C_2) = 7$, $g^*(C_3) = 12$, **所以倒推回去有由上述**(3), (2), (1)，**便可以求得由 S 到 T 的最短路徑**

由以上之分析, **我們可以由** (3) **到推到** (2) **而** (1), **便可以求得最短路線**, **以下我們以表格方式分四個階段求解**。

階段 1(Stage 1):

S_1	d_1	$f(S_1, d_1)$	$f(S_1, d_1) + g^*(S_0)$ $S_0 = T$, $g^*(S_0) = g^*(T) = 0$	$g^*(S_1)$	d_1^*
C_1	T	7	7 + 0	7	T
C_2	T	9	9 + 0	9	T
C_3	T	8	8 + 0	8	T

階段 2(Stage):

S_2	d_2	$f(S_2, d_2)$	$f(S_2, d_2) + g^*(S_1)$ $S_1 = d_2$	$g^*(S_2)$	d_2^*
B_1	C_2	2	2 + 9	11	C_2
	C_3	4	4 + 8		
B_2	C_1	2	2 + 7	9	C_1
	C_3	3	3 + 8		
B_3	C_2	4	4 + 9		
	C_3	5	4 + 8	12	C_3

階段 3(Stage 3):

S_3	d_3	$f(S_3, d_3)$	$f(S_3, d_3) + g^*(S_2)$ $S_2 = d_3$	$f^*(S_3)$	d_3^*
A_1	B_1	4	4 + 11		
	B_2	3	3 + 9	12	B_2
A_2	B_1	6	6 + 11		
	B_2	5	5 + 9	14	B_2
	B_3	8	8 + 12		
A_3	B_2	5	5 + 9	14	B_2
	B_3	3	3 + 12		

階段 4(Stage 4):

S_4	d_4	$f(S_4, d_4)$	$f(S_4, d_4) + g^*(S_3)$ $S_3 = d_4$	$g^*(S_4)$	d_4^*
S	A_1	5	5 + 12	17	A_1
	A_2	4	4 + 14		
	A_3	4	4 + 14		

由階段 4 **可知由** S **到** T **的最短路徑長度是** 17, **倒推回去便可求得最短路徑如下**:

$$S \rightarrow A_1 \rightarrow B_2 \rightarrow C_1 \rightarrow T$$

前推式遞迴思考:

令 $g^*(X)$ = S **至** X **之最短路線長度**

$g^*(T)$ = S **至** T **之最短路徑長度(最佳解)**

(1) **如果我們知道** $g^*(C_1)$, $g^*(C_2)$, $g^*(C_3)$, **則可得最佳解, 則最短路徑長度可以下式的遞迴關係求得**

$$g^*(T) = \text{minimum } \{ g^*(C_1) + 7,\quad g^*(C_2) + 9,\quad g^*(C_3) + 8 \}$$

(2) **如何求得** $g^*(C_1)$, $g^*(C_2)$, $g^*(C_3)$?

$$g^*(C_1) = \text{minimum } \{ g^*(B_2) + 2 \}$$

$$g^*(C_2) = \text{minimum } \{ g^*(B_1) + 2,\quad g^*(B_3) + 4 \}$$

$$g^*(C_3) = \text{minimum } \{ g^*(B_1) + 4,\quad g^*(B_2) + 3,\quad g^*(B_3) + 4 \}$$

(3) **如何求得** $g^*(B_1)$, $g^*(B_2)$, $g^*(B_3)$?

$$g^*(B_1) = \text{minimum } \{ g^*(A_1) + 4,\quad g^*(A_2) + 6 \}$$

$$g^*(B_2) = \text{minimum } \{ g^*(A_1) + 3,\quad g^*(A_2) + 5,\quad g^*(A_3) + 5 \}$$

$$g^*(B_3) = \text{minimum } \{ g^*(A_2) + 8,\quad g^*(A_3) + 3 \}$$

(4) **如何求得** $g^*(A_1)$, $g^*(A_2)$, $g^*(A_3)$?

很明顯 $g^*(A_1) = 5$, $g^*(A_2) = 4$, $g^*(A_3) = 4$, **所以倒推回去有由上述** (3), (2), (1) , **便可以求得由** S **到** T **的最短路徑**

由以上之分析，我們可以由 (3) 到推到 (2) 而 (1)，便可以求得最短路線，以下我們以表格方式分四個階段求解。

階段 1(Stage 1):

S_1	d_1	$f(d_1, S_1)$	$f(d_1, S_1) + g^*(S_0)$ $S_0 = T, \quad g^*(S_0) = 0$	$g^*(S_1)$	d_1^*
A_1	S	5	5 + 0	5	S
A_2	S	4	4 + 0	4	S
A_3	S	4	4 + 0	4	S

階段 2(Stage):

S_2	d_2	$f(d_2, S_2)$	$f(d_2, S_2) + g^*(S_1)$ $S_1 = d_2$	$g^*(S_2)$	d_2^*
B_1	A_1	4	4 + 5	9	A_2
	A_2	6	6 + 4		
B_2	A_1	3	3 + 5	8	A_1
	A_2	5	5 + 4		
	A_3	5	5 + 4		
B_3	A_2	8	8 + 5		
	A_3	3	3 + 4	7	A_3

階段 3(Stage 3):

S_3	d_3	$f(d_3, S_3)$	$f(d_3, S_3) + g^*(S_2)$ $S_2 = d_3$	$g^*(S_3)$	d_3^*
C_1	B_2	2	2 + 8	10	B_2
C_2	B_1	2	2 + 9	11	B_1
	B_3	4	4 + 7		
C_3	B_1	4	4 + 9		
	B_2	3	3 + 8	11	B_2
	B_3	5	5 + 7		

階段 4(Stage 4):

S_4	d_4	$f(d_4, S_4)$	$f(d_4, S_4) + g^*(S_3)$ $S_3 = d_4$	$g^*(S_4)$	d_4^*
T	C_1	7	7 + 10	17	C_1
	C_2	9	9 + 11		
	C_3	8	8 + 11		

可知由 S **到** T **的最短路徑長度是** 17, **由階段** 4 **倒推回去便可求得最短路徑如下**:

$$S \rightarrow A_1 \rightarrow B_2 \rightarrow C_1 \rightarrow T$$

4.2 動態規劃法的特點:

有許多尋優問題(Optimization Problem) 可以導出遞迴尋優關係式，同一問題也可能導出數個不同的遞迴關係式(見 4．節)，因此解遞迴尋優關係式也是尋優研究領域一個重要的主題，如 4.1 節介紹的方法, 我們可以結合猜方法將遞迴尋優關係式化解為一般之遞迴關係式，然再應用現有之方法解它，可惜的是很多尋優問題，其遞迴尋優關係式往往很複雜，牽涉的參數也很多，所以無法應用上述方法求解，很幸運的是 Bellman [ ] 在研究一些多階決策尋優問題解法過程中，理出一種有系統的方法，稱為動態規劃法，它可以用來解相當廣泛的遞迴尋優關係式，動態規劃法是解遞迴尋優關係式的一種解法，當然也有它的一些限制和優缺點，所以我們相信將來也有可能有新的方法被發現，是可以努力的方向。

以下我們要介紹動態規劃法一些特點: 階段(Stage), 狀態(State), 決策變數(Decision Variable), 遞迴尋優關係式(Recursive Optimization Relationship), 最佳化原理(Principle of Optimality)

(1) 問題可以分為數個階段(Stage)求解, 每一階段我們解一個或多個較單純的子問題, 並尋求子問題最佳決策, 可經由上一階段的資訊與遞迴尋優關係求得此最佳決策。 例如在例 4-3 的後推式解法，我們將問題劃分為四個階段，階段三要求 A_1, A_2, A_3 到 T 的最短路徑，要做的決策是找出 A_1, A_2, A_3 每一結點往前的最佳方向。

(2) 每一階段有數個狀態(State), 每一狀態就表示一種發生的可能性。例如在例 4-3 中, 階段三的三種狀態 A_1, A_2, A_3，就表示可能最短路徑有可能經過 A_1, A_2，或 A_3。

(3) 在一個階段的某一狀態, 如我們做一個決策(Decision), 我們將會對應到相鄰階段的一個狀態, 例如在例 4-3 階段三的狀態 A_3，如往 B_2 的方向，便會停留在第二階段的狀態 B_2。

$$\begin{array}{c} d_n \\ \downarrow \\ \cdots \rightarrow \boxed{S_n} \longrightarrow \boxed{S_{n-1}} \rightarrow \cdots \end{array}$$

圖 4.

(4) 動態規劃法在每一階段針對所有可能的狀態求解最佳決策, 進而在最後階段求出整個問題的最佳

決策, 例如在例 4-3 後推式解法解題的順序如下:

$$\begin{array}{ccccccccc} g^*(T)=0 & & g^*(C_1)=7 & & g^*(B_1) & & g^*(C_1) & & g^*(S) \\ & \rightarrow & g^*(C_2)=7 & \rightarrow & g^*(B_2) & \rightarrow & g^*(C_2) & \rightarrow & \\ \text{(初始條件)} & & g^*(C_3)=12 & & g^*(B_3) & & g^*(C_3) & & \text{(最佳值)} \end{array}$$

(5) 對每一階段 i, 我們必須有一個遞迴關係式, 以利用第 i-1 階段(或 1, . . . , i-1 階段)的最最佳決策, 以求出第 i 階段的最佳決策。例 4-3 第三階段就有以下遞迴關係式:

$g^*(A_1)$ = minimum { 4 + $g^*(B_1)$, 3 + $g^*(B_2)$ }

$g^*(A_2)$ = minimum { 6 + $g^*(B_1)$, 4 + $g^*(B_2)$, 8 + $g^*(B_3)$ }

$g^*(A_3)$ = minimum { 5 + $g^*(B_2)$, 3 + $g^*(B_3)$ }

動態規劃法所解的遞迴尋優問題和一般如例4.1, 不同之處是它的每一階段的遞迴尋優關係式可能都不一樣。 所以一般情況也就不易求得問題的公式解。

(6) 最佳化原理(Principle of Optimality) :

"

最佳化原理是動態規劃法的依據, 我們將問題劃分為許多小問題必須符合這個原理纔可以應用動態規劃法,

我們這一章的目的不只是要學習如何應用動態規劃法解問題, 同時也要知道以下的原理:

什麼問題可用動態規劃法解?

什麼問題不可以用動態規劃法解?

什麼問題用動態規劃法解較有效率? 為什麼?

什麼問題用動態規劃法解效率不好? 為什麼?

如何分析動態規劃法計算步驟與記憶體需求?

應用動態規劃法解題, 依問題決策(decision)的性質可分為二種情形(1) 確定性動態規劃法 (Deterministic Dynamic Programming) 與 (2): 隨機性動態規劃法(Probabilistic Dynamic

Programming)。**若依狀態的情況也可分為**(1) **離散型狀態**(Discrete State Space)**與** (2) **連續型狀態**(Continuous State Space), **以下我們將分別介紹**。

1. **確定性動態規劃法**(Deterministic Dynamic Programming): **在一狀態下做任一決策, 其結果是確定的, 例如在狀態** S_i **選擇決策** d_i, **將對應到第** i-1 **階段的唯一狀態** S_{i-1} (**見圖**

圖 4.

隨機性動態規劃法(Probabilistic Dynamic Programming): **在一狀態下做任一決策, 其結果有隨機性變化, 例如在狀態** S_i **選擇決策** d_i, **將有多個可能的結果,** p_j **的可能性對應到** $S_{i-1}^{(j)}$, $j = 1, \ldots, k$, $p_1 + \ldots + p_k = 1$ (**見圖**

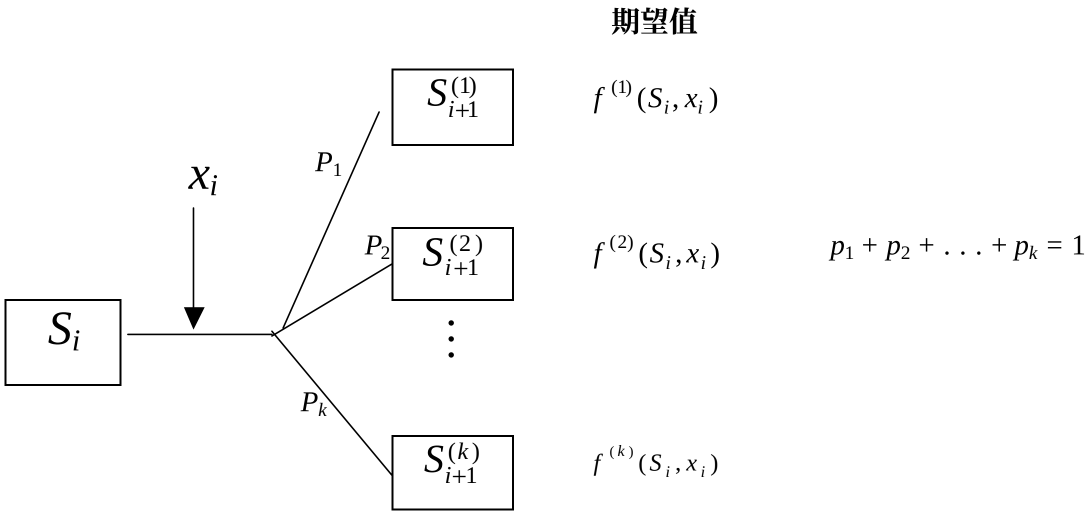

圖 4.

S_k = **階段** k (Stage k), **所有可能的狀態所組成的集合**

離散型狀態(Discrete State Space): **此時,** S_k **為一有限或是可數的集合, 例如,** $S_k = \{ a, c, d, e \}$, $S_k = \{1, 2, \ldots, n, \ldots\}$, $S_k = \{ 3, 6, 9, 12, \ldots, 3n, \}$ **等等, 因此對任一狀態** s, $s \in S_k$, **我們便要求解**

連續型狀態(Continuous State Space): **此時,** S_k **為無限且不可數的集合, 例如,** $S_k = [a, b]$, $a < b$, S_k = R(**所有實數的集合**), $S_k = [0, 1]\times[1, 3] = \{(x, y) | 0 \le x \le 1, 1 \le y \le 3 \}$ **等等, 因此對任一狀態** s, $s \in S_k$, **我們便要求解**

在一般的尋優問題, 我們通常對一個問題求得一個數值解, 但有些情況我們要求得一個最佳的函數解, 在變分學(Calculus of Variation)**與最佳控制**(Optimal Control),