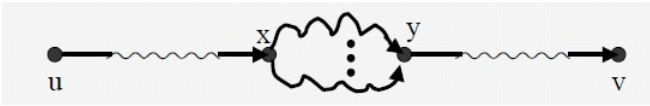
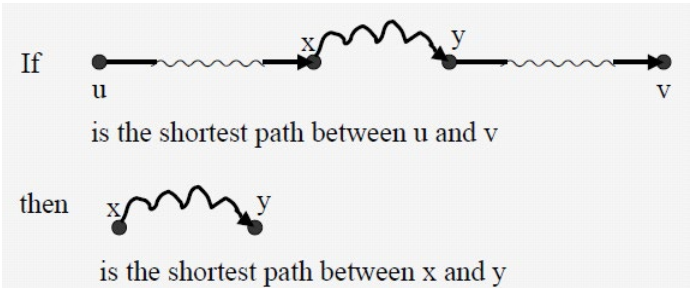
國立臺北大學自然資源與環境管理研究所
111 學年度第二學期『資源管理與環境系統分析』

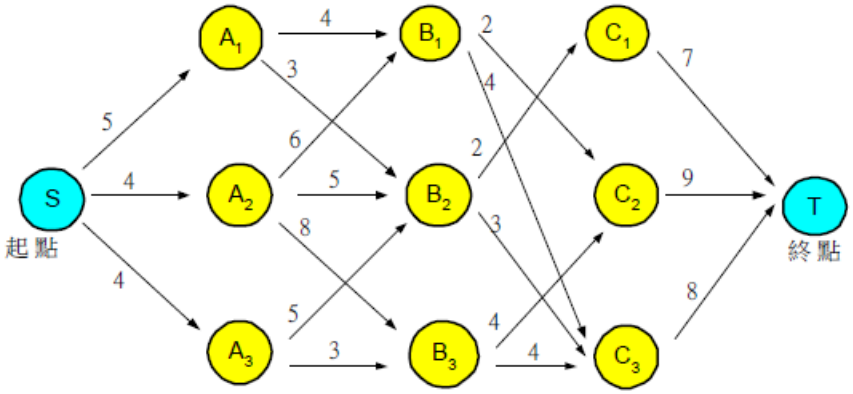
課程講義(04)：動態規劃與目標規劃
Dynamic Programming and Goal Programming

- INTRODUCTION TO DYNAMIC PROGRAMMING
  - □ Dynamic Programming = Divide and Conquer + Memorization
  - □ No Specific Forms or Formulations=> Principle of Optimality
  - □ Terminology: Stage, State, Decision, Return, Recursive Equation
  - □ Dynamic programming is a technique for solving problems with a recursive structure with the following characteristics:
    - ⇨ Optimal substructure (principle of optimality): An optimal solution to a problem can be decomposed into optimal solutions for sub-problems.
    - ⇨ A small number of sub-problems: The total number of sub-instances to be solved is small.
    - ⇨ Overlapping sub-problems: During the computation same instances are referred to repeatedly.





  - □ An Example of Dynamic Programming: The Shortest Path Problem
    - ⇨ Divide the problem into 4 subproblems (Stages)
    - ⇨ Find the optimal solution in the stage *i* and 'pass' into stage *i*+1.
    - ⇨ Formulate the recursive equation between stages.
    - ⇨ Backward vs. Forward => Can find the same optimal solution under deterministic conditions



- GOAL PROGRAMMING
  - □ Criteria for Decision-Making: Attribute, Objective, Target, and Goal
    - ⇨ The UN SDGs: Goals, Targets, and Indicators
  - □ Multiple Criteria Decision Making: Multiple Attribute and Multiobjective
  - □ Classification of Goal Programming: Non-Preemptive vs. Preemptive
  - □ Non-Preemptive Goal Programming
    - ⇨ Complementary relationship
    - ⇨ One-sided vs. Two-sided
  - □ Preemptive Goal Programming or Lexicographic GP
  - □ Drawbacks: Normalization and Weighting; Pareto Optimality?
  - □ An Example of Goal Programming: Expansion of Production Lines
    - ⇨ Deviational variables => slack vs. surplus => Minimize penalty weighted deviations
    - ⇨ Maximize the profit => allow overshooting and penalty on undershooting
    - ⇨ Remain employee level => penalty on both sides (may have different weights)
    - ⇨ Minimize the investment => penalty on overshooting and allow undershooting

例題 3-1：

某一家公司考慮製造三種產品取代目前的產品，決策者優先考慮了三個主要因子：長期利潤、勞動力之穩定性和投資資本額。目標如下：

(1) 三種產品之長期利潤至少 125,000,000 元。

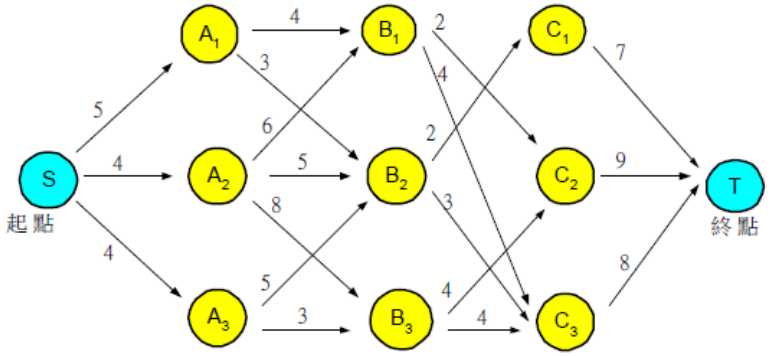
(2) 儘量維持現有雇用水準爲員工 4,000 人。

(3) 固定投資資本額少於 55,000,000 元。

所定的標準如下表 3.1：

表 3.1：例題 3-1 中之系統規劃條件

因子	產品之生產參數			標的 (單位)	懲罰權重 (penalty weight)
	1	2	3		
長期利潤 (10^3\$/產品)	12	9	15	≥125(百萬元/年)	5
雇用水準 (人)	5	3	4	=40(百名員工/年)	2(+)，4(-)
投資資本額 (10^3\$/產品)	5	7	8	≤55(百萬元/年)	3



階段1 (Stage 1):

S_1	d_1	$f(S_1, d_1)$	$f(S_1, d_1) + g^*(S_0)$ $S_0 = \text{T}, \quad g^*(S_0) = g^*(\text{T}) = 0$	$g^*(S_1)$	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 2):

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

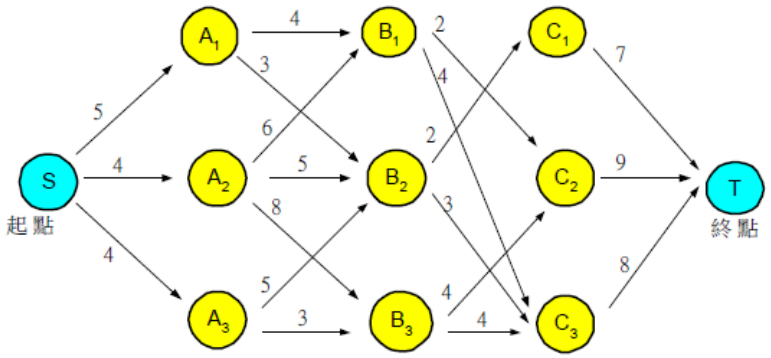
階段3 (Stage 3):

S_3	d_3	$f(S_3, d_3)$	$f(S_3, d_3) + g^*(S_2)$ $S_2 = d_3$	$g^*(S_3)$	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		

S → A_1 → B_2 → C_1 → T



階段1 (Stage 1):

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

階段2 (Stage 2):

S_2	d_2	$f(S_2, d_2)$	$g^*(S_1) + f(S_2, d_2)$ $S_1 = d_1$	$g^*(S_2)$	d_2^*
A_1	B_1	4	5 + 4		
	B_2	3	5 + 3	8	B_2
A_2	B_1	6	4 + 6		
	B_2	5	4 + 5	9	--
	B_3	8	4 + 8		
A_3	B_2	5	4 + 5		
	B_3	3	4 + 3	7	B_3

階段3 (Stage 3):

S_3	d_3	$f(S_3, d_3)$	$g^*(S_2) + f(S_3, d_3)$ $S_2 = d_2$	$g^*(S_3)$	d_3^*
B_1	C_2	2	--		
	C_3	4	--		
B_2	C_1	2	8 + 2	10	C_1
	C_3	3	8 + 3		
B_3	C_2	4	7 + 4	11	C_2
	C_3	5	7 + 5		

階段4 (Stage 4):

S_4	d_4	$f(S_4, d_4)$	$g^*(S_3) + f(S_4, d_4)$ $S_3 = d_3$	$g^*(S_4)$	d_4^*
C_1	T	7	10 + 7	17	T
C_2	T	9	11 + 9	20	T
C_3	T	8	--	--	--

T → C_1 → B_2 → A_1 → S