Université de Tiaret
Faculté des mathématiques et de l'informatique
Département informatique
1ère *Année Master « Génie Informatique+ Réseau&Télécom»*

Année 2019/2020 (CM : B. A/MALEK)

***Matière* : « ORO : Optimisation en Recherche Opérationnelle »**
Optimisation combinatoire (OC)

Fiche TD N° 01

(Les méthodes exactes « la programmation dynamique »)

I. Introduction

La programmation dynamique est une technique mathématique qui a pour objet d'aider à prendre des décisions séquentielles indépendantes les unes des autres.

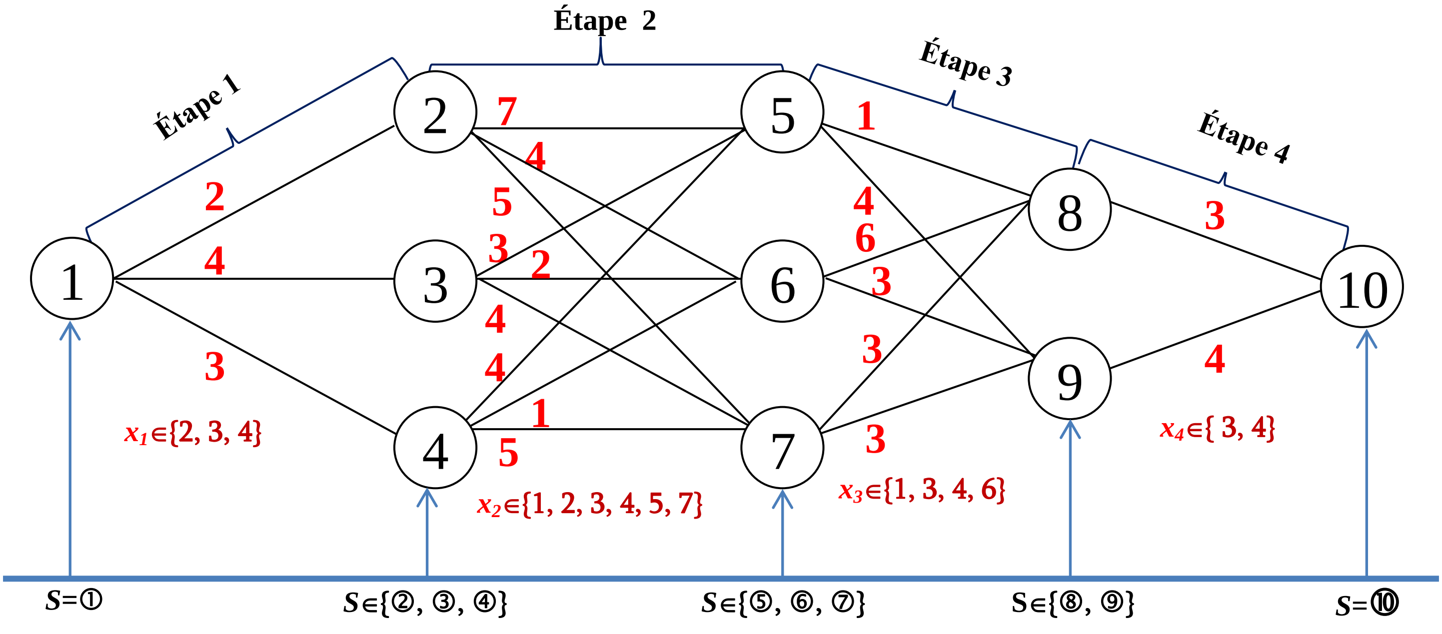
Contrairement à la programmation linéaire, il n'y a pas un formalisme mathématique standard. C'est une approche de résolution où les équations doivent être spécifiées selon le problème à résoudre.

II. Exemples.

II.1- Le problème du plus court chemin dans un graphe

Soit G= (S, U) avec S : l'ensemble des sommets et U : l'ensemble des arêtes

Question : Trouver le plus court chemin dans ce graphe G qui minimise la somme de distances qui nous amène du sommet de départ = ① au sommet d'arrivé = ⑩ ?



Exemple : Si le chemin trouvé est : ①→ ②→ ⑥→ ⑨→ ⑩, Alors la somme de distances du chemin = 13

Soit x_n (n=1, ..., 4) les variables de décisions relatives à chacune des 4 étapes. Le chemin à suivre est ①→ x_1→ x_2→ x_3→ x_4 avec x_4 = ⑩.

Soit $f_n(S, x_n)$ la distance à l'étape n sachant que nous somme à l'état S de l'étape n et que la destination choisie est x_n.

Etant donné S et n, soit x_n^* la valeur de x_n qui minimise $f_n(S, x_n)$ et soit $f_n^*(S)$ la valeur minimale de $f_n(S, x_n)$. ($f_n^*(S) = f_n(S, x_n^*)$).

L'approche de la programmation dynamique repose sur l'idée qu'un chemin ne peut être optimal que si chacune de ses composantes est-elle même optimale. La démarche de la programmation dynamique consiste à étudier d'abord les sous problèmes qui se situent chronologiquement les derniers et sur un principe de retour en arrière.

L'objectif est donc de trouver la valeur de $f_1^*(1)$. Pour l'avoir, la programmation dynamique va essayer d'évaluer avec une procédure de chaînage en arrière $f_4^*(s)$, $f_3^*(s)$, $f_2^*(s)$ et enfin $f_1^*(1)$.

A chaque étape, on va essayer d'évaluer pour chaque état s, la valeur de $f_n(S, x_n)$ pour chaque destination x_n possible, puis de retrouver la meilleure destination x_n^* (celle qui correspond au plus petite distance $f_n(S, x_n)$) et aussi la valeur de $f_n^*(s)$.

Etape 4 ($f_5^*(x_4) = 0$)

	$f_4(S, x_4) = C_s x_4 + f_5^*(x_4)$		
S \ x_4	⑩	$f_4^*(s)$	x_4^*
⑧	3	3	⑩
⑨	4	4	⑩

Etape 3

	$f_3(S, x_3) = C_s x_3 + f_4^*(x_3)$			
S \ x_3	⑧	⑨	$f_3^*(s)$	x_3^*
⑤	4(=1+3)	8(=4+4)	4	⑧
⑥	9	7	7	⑨
⑦	6	7	6	⑧

Etape 2

	$f_2(S, x_2) = C_s x_2 + f_3^*(x_2)$				
S \ x_2	⑤	⑥	⑦	$f_2^*(s)$	x_1^*
②	11	11	12	11	⑤ ou ⑥
③	7	9	10	7	⑤
④	8	8	11	8	⑤ ou ⑥

Etape 1

	$f_1(S, x_1) = C_s x_1 + f_2^*(x_1)$				
S \ x_1	②	③	④	$f_1^*(1)$	x_1^*
①	13	11	112	11	③ ou ④

La valeur de $f_1^*(①) = 11$ représente la somme de distances minimale du chemin. Le chemin optimal n'est pas unique puisque dès le départ on peut choisir $x_1^* =$ ③ ou ④, donc l'ensemble de ces chemins est:

① → ③ → ⑤ → ⑧ → ⑩

① → ④ → ⑤ → ⑧ → ⑩

① → ④ → ⑥ → ⑨ → ⑩

III. Caractéristiques d'un problème de programmation dynamique

Nous allons maintenant sur la base de l'exemple précédant analyser les propriétés communes aux problèmes de programmation dynamique.

1. Le problème peut être décomposé en étapes et une décision doit être prise à chaque étape.
Le dernier exemple est devisé en 4 étapes où à chaque étape, la décision à prendre est celle de la destination à choisir. Ces décisions sont interdépendantes et séquentielles.

2. A chaque étape correspond un certain nombre d'états. Dans l'exemple du plus court chemin, les états à chaque étape sont représentés par les sommets du graphe traité. Le nombre de ces états dans l'exemple est fini.

3. A chaque étape, la décision prise transforme l'état actuel en un état associé à l'étape suivante (dans certains cas avec une distribution de probabilité).
Dans l'exemple (II.1), se trouvant à un sommet donné, le décideur décide de se rendre à un autre sommet qui est un état de l'étape suivante.

4. Etant donné un état, une stratégie optimale pour les étapes restantes est indépendante des décisions prises aux étapes précédentes.

5. L'algorithme de recherche de la solution optimale commence par trouver la stratégie optimale pour tous les états de la dernière étape.

6. Une relation de récurrence identifie la stratégie optimale dans chaque état de l'étape *n* à partir de la stratégie optimale dans chaque état de l'étape *n+1*.

Dans l'exemple (II.1) la relation est $f_n^*(S) = \underset{x_n}{Min}\{C_s x_n + f_{n+1}^*(x_n)\}$

La stratégie optimale étant donné que nous sommes à l'état *S* de l'étape *n*, nécessite de retrouver la valeur de x_n qui minimise l'expression ci-dessus.

La relation de récurrence à toujours cette forme $f_n^*(S) = \underset{x_n}{Max} \text{ ou } \underset{x_n}{Min} \{f_n(S,x_n)\}$,

avec $f_n(S,x_n)$ est une expression en fonction de S, x_n et $f_{n+1}^*(\text{-})$

7. Utilisant cette relation de récurrence, l'algorithme procède à reculons étape par étape. Il détermine la stratégie optimale pour chaque état de chaque étape.

Dans tout problème de programmation dynamique, on peut construire à chaque étape un tableau analogue au suivant.

<u>Etape n</u>

	$f_n(S, x_n)$		
X_n / S	*états n+1*	$f_n^*(s)$	x_n^*
états n	*le coût ou la distance*	*La longueur du chemin optimal à partir de S jusqu'à l'état final*	*Le meilleur état de l'étape n+1 le long du chemin optimal final*

IV. Programmation dynamique déterministe

a. Introduction

Dans cette section on s'intéresse au problème dit déterministe, où la connaissance de l'état et de la décision à prendre suffisent pour savoir l'état à l'étape suivante.

Un problème dynamique déterministe est caractérisé par la détermination de la fonction objective. Cette fonction peut être le minimum de la somme de la contribution induite par le passage d'un état à un autre, ou le maximum d'une telle somme, ou le minimum du produit de ces termes… etc.

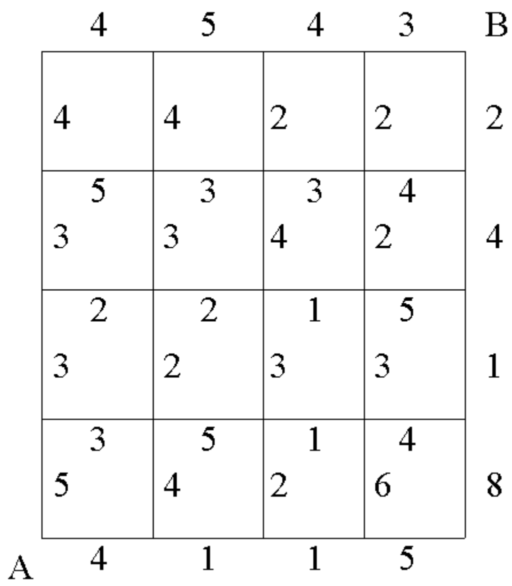
Il faut aussi déterminer la nature de l'ensemble des états dans chacune des étapes. Ces états S_n peuvent être représentés par des variables discrètes ou par des variables continues ou dans certains cas par un vecteur.

Étape n **Étape n+1**

$$S_n \xrightarrow{\text{Contribution de } x_n} S_{n+1}$$

La structure de base d'un problème dynamique déterministe

b. Problème du type plus court chemin



Quel est le plus court chemin de A à B ?

<u>Solution:</u>

On considère l'ensemble des états les point d'intersection sur les diagonales. Ainsi on a exactement 8 étapes.

$f_n(S, x_n) = C_s x_n + f_{n+1}(x_n), n=1,...., 8$

avec $f^*_{n+1}(s) = \underset{x_n}{Min} f_n(S, x_n)$ et $f_9(S) = 0$

Etape 8

S \ x_8	$f_8(S, B)$: B	$f_8^*(S)$	x_8^*
1	*3*	*3*	B
2	*2*	*2*	B

Etape 7

S \ x_7	$f_7(S, x_7) = C_s x_7 + f_8(x_7)$: 1	2	$f_7^*(s)$	x_7^*
1	*7*	-	*7*	1
2	*5*	*6*	*5*	1
3	-	*6*	*6*	2

Etape 6

S \ x_6	$f_6(S, x_6) = C_s x_6 + f_7(x_6)$: 1	2	3	$f_6^*(s)$	x_6^*
1	*12*	-	-	*12*	1
2	*9*	*8*	-	*8*	2
3	-	*7*	*11*	*7*	2
4	-	-	*7*	*7*	3

Etape 5

S \ x_5	$f_5(s, x_5) = C_s x_5 + f_6^*(x_5)$: 1	2	3	4	$f_5^*(s)$	x_5^*
1	*16*	-	-	-	*16*	1
2	*16*	*11*	-	-	*11*	2
3	-	*12*	*8*	-	*8*	3
4	-	-	*10*	*11*	*10*	3
5	-	-	-	*15*	*15*	4

Etape 4

S \ x_4	$f_4(s, x_4) = C_s x_4 + f_5^*(x_4)$: 1	2	3	4	5	$f_4^*(s)$	x_4^*
1	*20*	*16*	-	-	-	*16*	2
2	-	*14*	*10*	-	-	*10*	3
3	-	-	*11*	*11*	-	*11*	3,4
4	-	-	-	*16*	*20*	*16*	4

Etape 3

S \ x_3	$f_3(s, x_3) = C_s x_3 + f_4^*(x_3)$: 1	2	3	4	$f_3^*(s)$	x_3^*
1	*12*	*12*	-	-	*12*	2
2	-	*12*	*16*	-	*12*	2
3	-	-	*13*	*17*	*13*	3

Etape 2

S \ x_2	$f_2(s, x_2) = C_s x_2 + f_3^*(x_2)$: 1	2	3	$f_2^*(s)$	x_2^*
1	*15*	*15*	-	*15*	1,2
2	*2*	*16*	*14*	*14*	3

Etape 1

S \ x_1	$f_1(s, x_1) = C_s x_1 + f_2^*(x_1)$: 1	2	$f_1^*(s)$	x_1^*
A	*20*	*18*	*18*	2

On peut résumer les résultats de la façon suivante:

Etapes	1	2	3	4	5	6	7	8	9
Chemin optimal 1	A	2	3	3	3	3	2	1	B
Chemin optimal 2	A	2	3	3	4	3	2	1	B

La longueur du chemin optimal (1 ou 2) est égale à 18.

c. Répartition optimale des moyens

Un projet du gouvernement est étudié par 3 groupes de chercheurs. La probabilité que chacun de ces groupes 1, 2 et 3, n'arrive pas à terminer le projet est respectivement: 0,4; 0,6 et 0,8.
Si on ajoute à ces groupes deux nouveaux chercheurs, les probabilités d'échec sont données par ce tableau

	Probabilité d'échec		
Nbre de nouveaux chercheurs	*Groupes*		
	1	*2*	*3*
0	*0,4*	*0,6*	*0,8*
1	*0,2*	*0,4*	*0,5*
2	*0,15*	*0,2*	*0,3*

Le problème est de déterminer l'allocation optimale de ces deux chercheurs afin de minimiser la probabilité que les groupes de recherche échouent dans leur travail.

Solution:

- Etapes: 3 étapes ou les états représentent le nombre de chercheurs disponibles
- Variable de décision: x_n représente le nombre de chercheurs à allouer à l'équipe de recherche n, $n = 1, 2, 3$.
- Contribution de la décision x_n est la probabilité que l'équipe n échoue après avoir eu x_n chercheur de plus
- $f_n^*(S)$: c'est la probabilité minimale que les groupes $1,..., 3$ échouent dans leurs recherches :

  $f_n^*(S) = \underset{x_n \leq S}{Min}\, f_n(S, x_n)$ $n = 1, 2, 3$ avec $f_n(S, x_n) = p_n(x_n) \times f_{n+1}^*(S, x_n)$, $p_n(x_n)$ est la contribution de la décision x_n et $f_4^*(s) = 1$.

Étape n **Étape n+1**

$$S_n \xrightarrow{P_n(x_n)} S_n - x_n$$

$$f_n(S, x_n) = p_n(x_n) * f_{n+1}^*(S - x_n)$$

Etape 3

S \ x_3	$f_3(S, x_3) = p_3(x_3)$			$f_3^*(S)$	x_3^*
	0	1	2		
0	0,8	-	-	0,8	0
1	0,8	0,5	-	0,5	1
2	0,8	0,5	0,3	0,3	2

Etape 2

S \ x_2	$f_2(S, x_2) = p_2(x_2) * f_3^*(s - x_2)$			$f_2^*(S)$	x_2^*
	0	1	2		
0	0,48	-	-	0,48	0
1	0,3	0,32	-	0,3	0
2	0,18	0,2	0,16	0,16	2

Etape 1

$S \backslash x_1$	$f_1(S, x_1) = p_1(x_1) * f_2^*(s-x_1)$ 0	1	2	$f_1^*(S)$	x_1^*
2	*0,064*	*0,06*	*0,072*	*0,06*	*1*

La stratégie optimale est $x_1^* = 1$, $x_2^* = 0$ et $x_3^* = 1$. La probabilité d'échec des trois groupes de recherche est de 0,06.

Exercice : (Problème de gestion des Stocks)

Période i	**1**	**2**	**3**	**4**	**5**
Demande d_i	2	3	4	3	2
Prix unitaire P_i	13	15	20	11	12

Les contraintes du problème :

1- Les achats se font en début de période
2- On doit stocker les ventes de la période en cours
3- On commence et on finit avec un stock nul
4- Les quantités achetées sont nécessairement entière
5- La capacité de stockage est au maximum = 5

Question :
En utilisant la méthode de la programmation dynamique, trouver le coût total minimal d'achat ?