Programmation dynamique

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.
- L'approche de la programmation dynamique consiste alors à résoudre le problème stage par stage en tenant compte du lien entre les stages.

2

- C'est-à-dire qu'on va remplacer la résolution du problème « global » par la résolution d'une séquence de petits sous problèmes plus simples.
- **Principe d'optimalité de Bellman** *(R. Bellman 1957):* «Un ensemble de décisions optimales a la propriété que, quelque soit le sous ensemble de décisions correspondant en 1er stage, les décisions restantes doivent être optimales relativement au sous problème résultant des décisions du 1er stage ».

3

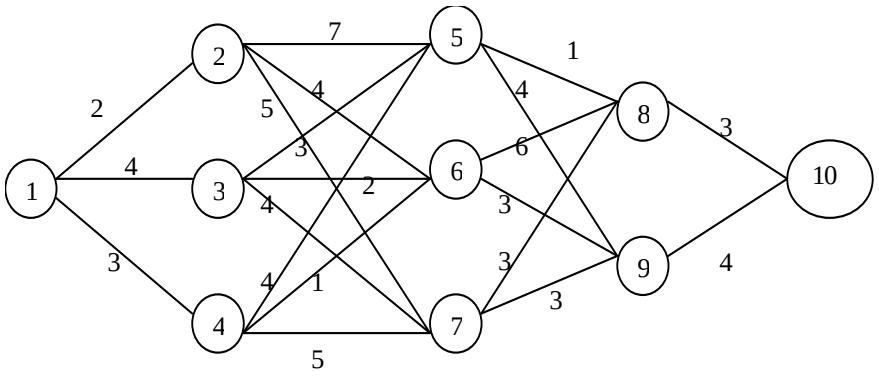
Exemple prototype: Le problème du voyageur

- Un postier décide d'effectuer un voyage entre différentes villes qu'il ne connaît pas. Son but est de choisir le chemin le moins dangereux. Pour choisir son chemin, il s'informe auprès des assurances sur les différentes valeurs des polices d'assurance de vie entre les différentes routes possibles du voyage.

4

- Le trajet du postier est composé de 4 étapes (voir figure) et il doit arriver à la ville numérotée 10 en partant de la ville numérotée 1. Les autres numéros représentent les villes susceptibles d'être traversées. Les arcs entre ces nœuds représentent les différents trajets possibles et les valeurs c_{ij} au-dessus des arcs représentent le prix de la police d'assurance vie relatif au trajet décrit par l'arc.
- Le chemin le moins dangereux est celui qui admet la plus petite valeur de la police d'assurance vie

5



Etape 1 Etape 2 Etape 3 Etape 4

6

- **<u>Exemple :</u>** Si le voyageur empreinte le trajet $1 \rightarrow 2 \rightarrow 6 \rightarrow 9 \rightarrow 10$, le coût total de la police d'assurance est 13
- Soit x_n *(n=1, ..., 4)* les variables de décisions relatives à chacune des 4 étapes. Le chemin à suivre par le voyageur est $1 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$ avec $x_4 = 10$.
- *Soit $f_n(S, x_n)$ le coût total de la police d'assurance vie pour le reste des étapes sachant que nous somme à l'état S de l'étape n et que la destination choisie est x_n.*

7

- 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 $f_n(S, x_n)$, i.e. $f_n^*(S) = f_n(S, x^*_n)$.
- L'approche de la programmation dynamique repose sur l'idée qu'un chemin ne peut être optimal que si chacune des 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.

8

- 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 $f(S, x_n)$ pour chaque destination x_n possible, puis de retrouver la meilleure destination (celle qui correspond au plus petit coût $f(S, x_n)$) et aussi la valeur de $f_n*(S)$.

9

- **<u>Etape 4</u>**

x_4 / S	$f_4(S,x_4)$	$f_4*(s)$	x_4*
8	3	3	10
9	4	4	10

10

- **Etape 3**

	$f_3(S,x_3)=C_s x_3+f_4^*(x_3)$			
x_3 / S	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

11

- **Etape 2**

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

12

- **<u>Etape 1</u>**

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

13

- La valeur de $f_1^*(1) = 11$ représente le coût total minimal de la police d'assurance vie. Le chemin optimal n'est unique puisque dès le départ on peut choisir $x_1^* = 3$ ou 4, donc l'ensemble de ces chemins est:

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 10$$
$$1 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 10$$
$$1 \rightarrow 4 \rightarrow 6 \rightarrow 9 \rightarrow 10$$

14

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.
- **Étape:** 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.

15

- **États:** A chaque étape correspond un certain nombre d'états. Dans l'exemple du voyageur, les états à chaque étapes sont représentés par les villes que le voyageur visité. Le nombre de ces états dans l'exemple est fini. Mais aussi il y a des problèmes où le nombre d'états est infinie ($x_n \in IN$) ou continue ($x_n \in IR$).

16

- **Politique:** 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 notre exemple, se trouvant à une ville donnée, le voyageur décide de se rendre à une autre ville qui est un état de l'étape suivante.

17

- **Politique optimale**: Étant donné un état, une stratégie optimale pour les étapes restantes est indépendante des décisions prises au étapes précédentes.
- Si le voyageur est dans un état quelconque de l'étape i, le chemin optimal entre cet état et l'état 10 sera indépendant de la façon dont il y est arrivé. En d'autres termes, l'état actuel contient toute l'information nécessaire aux décisions futures. Cette propriété est dite principe d'optimalité.

18

- **<u>Procedure de résolution:</u>** L'algorithme de recherche de la solution optimale commence par trouver la stratégie optimale pour tous les états de la dernière étape.
- **<u>Relation de récursivité</u>**: 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 $n+1$.

19

- Dans notre exemple la relation est $f_n(S,x_n)=Csx_n+f_{n+1}*(x_n)$ et $f_n*(S)=min_{xn}\{f_n(S,x_n)\}$
- 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 a toujours cette forme $f_n*(S)= max_{xn}$ *ou* $min_{xn}\{f_n(S,x_n)\}$,
  avec $f_n(S,x_n)$ est une expression en fonction de S, x_n et $f_{n+1}*(-)$
- Utilisant cette relation de récurrence, l'algorithme procède à reculons étape par étape. Il détermine la stratégie optimale pour chaque état de chaque étape.

20

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

<u>Étape n:</u>

	$f_n(S,x_n)$		
S \ x_n	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

21

Programmation dynamique déterministe

- Un problème est dit déterministe, lorsque la connaissance de l'état et de la décision à prendre suffisent pour savoir l'état à l'étape suivante.
- Par contre dans la programmation dynamique probabiliste, il y a une probabilité associée à chaque état possible.

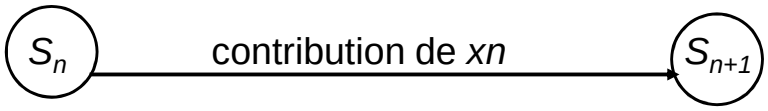
22

- 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.

23

étape n étape n+1



La structure de base d'un problème dynamique déterministe

- A l'étape n, on est à l'état S_n, on prend la variable de décision x_n puis on va à un état S_{n+1} à l'étape $n+1$. On calcule $f_{n+1}*(S)$. Ce qui nous donne avec la contribution de x_n la valeur de la fonction objective $f_n(S,x_n)$ lorsqu'on commence à l'étape n.

24

Programmation Dynamique

Exemples et Exercices

Exercice 1

- Une société industrielle recrute trois techniciens à fin de renforcer certain de ses quatre groupes de travail dont le but de maximiser le nombre totale de pièces fabriquées par l'ensemble de ses employés. La rentabilité de ces techniciens se mesure par le nombre de pièces additionnelles que va produire chaque groupe.

26

- Le tableau suivant nous donne le nombre de pièces additionnelles construites par chaque groupe pour chacune des affectations possibles des nouveaux techniciens.

Gpes / tech	1	2	3	4
1	20	40	50	30
2	30	50	80	40
3	80	85	90	100

Déterminer comment affecter ces trois techniciens de sorte à maximiser le rendement.

27

Exercice 2

- Pour que les gens puissent visiter la lune avec le plus possible de sécurité, le plan d'un gouvernement a necessité le resolution d'un certain problème d'ingenierie. Trois groupes de recherche sont entrain d'essayer trois différentes approches. Pour résoudre ce problème, on a estimer la probabilité d'échec de chacun des groupes 1,2 et 3 respectivement à 0,40 ; 0,60 et 0,80.

28

- Puisque l'objectif est de minimiser cette probabilité totale d'échec, deux autres scientifiques ont été désignés pour ce projet. La table suivante donne la probabilité estimée pour que le groupe 1, 2 et 3 ait un échec lorsque 0, 1 ou 2 scientifiques additionnelles sont ajoutés au groupe.
- Le problème est de déterminer comment affecter ces deux scientifiques de sorte à minimiser la probabilité que les trois groupes échouent.

29

Nombre de scientifiques additionnels	Probabilités d'échec		
	groupes		
	1	2	3
0	0,40	0,60	0,80
1	0,20	0,40	0,50
2	0,15	0,20	0,30

30