Chapitre 4

Programmation dynamique

Introduction

La programmation dynamique est une méthode complète applicable à de nombreux problèmes d'optimisation avec contraintes linéaires ou non linéaires, en variables continues ou discrètes, possédant une certaine propriété dite de décomposablité. Le terme « Programmation dynamique » provient du fait que la méthode a d'abord été appliquée à l'optimisation des systèmes dynamiques c.à.d. évoluant au cours du temps. Cependant le principe de la PD est plus général et peut s'appliquer à des problèmes d'optimisation où le temps n'intervient pas, comme pour la PLNE.

L'idée de base de la PD consiste à essayer de remplacer l'optimisation d'une fonction à n variables par la résolution d'un certain nombre de problèmes d'optimisation, plus simple à résoudre (par exemple l'optimisation d'un problème à une seule variable). En ce sens la méthode peut être reconsidérée comme une méthode de décomposition.

La PD est due à Bellman qui fut premier à formuler le principe d'optimalité : « Dans une séquence optimale de décisions quel que soit la première décision prise, les décisions subséquentes forment une sous-séquence optimale, compte tenu des résultats de la première décision »

Avant de présenter le formalisme de la programmation dynamique, nous présentons dans ce qui suit un exemple illustratif expliquant le principe du raisonnement de la programmation dynamique.

Exemple illustratif : Problème de gestion de stocks

Le tableau suivant donne, pour les 5 périodes qui font l'objet de l'étude, les quantités d'un certain bien qu'un marchant veut revendre, ainsi que les prix d'achat unitaires.

Période i	1	2	3	4	5
Demande d_i	2	3	4	3	2
Prix p_i	13	15	20	11	12

Sachant que ce marchant dispose d'une capacité de stockage de 5 unité (le coût de stockage est nul), le problème consiste à déterminer la politique d'achat optimal, c'est-à-dire qu'on cherche à déterminer les quantités à acheter à chaque période de manière à minimiser le coût total d'achat sur les 5 périodes. De plus on a :

1. Les achats se font en début de chaque 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 entières

Posons : x_j = quantité achetée à la j^{eme} période ($j \in$ N, j = 1,5)

a_j = quantité restant en stock à la fin de la j^{eme} période

a_j est appelée variable d'état. Elle décrit l'état du système à la fin de la j^{eme} période

x_j est appelé variable de décision. Les variables a_j et x_j doivent donc satisfaite :

i. $0 \leq a_j \leq 5 - d_j$ j=1,2,…,5 (1) et (2)
ii. $a_0 = a_5 = 0$
iii. $a_j = a_{j-1} + x_j - d_j$

Le problème précédent peut donc être formulé comme un PLNE :

$$(P)\begin{cases} Z(min) = \sum_{j=1}^{5} P_j x_j \\ 0 \leq a_j \leq 5 - d_j \quad j = \overline{1,5} \\ a_j = a_{j-1} + x_j - d_j \quad j = \overline{1,5} \\ a_0 = a_5 = 0 \\ a_j, x_j \in N, \; j = \overline{1,5} \end{cases}$$

Dans ce qui suit, nous résolvons (P) en décomposant le problème séquentiellement en plusieurs sous-problèmes de la manière suivante ; Nous nous intéressons dans un premier lieu à la première période seulement, puis nous cherchons une politique optimale d'achat sur les deux premières périodes, et ainsi de suite. A l'étape k de la résolution, nous nous intéressons au sous problème restreint aux k premières périodes. En d'autres termes, à l'‘étape k de résolution, le problème est restreint aux k premières variables seulement, jusqu'à arriver au problème initial.

$1^{ière}$ période :

En vertu de (i), la $1^{ière}$ période peut être terminée soit avec 0,1,2 ou 3 unités en stock

(ie a_1= 0,1,2 ou 3). Les coûts correspondants sont respectivement 26, 39, 52 et 65

Exemple : si a_1=0, x_1= 2 et d_1=2)

$2^{ième}$ période :

Si on veut terminer la deuxième période avec 0 unité en stock :

a_2=0 : on distingue alors 4 cas :

- x_1= 5 → x_2= 0 ($a_1 = 3$) → c= 65+0= 65
- x_1= 4 → x_2= 1 ($a_1 = 2$) → c= 52+15= 67
- x_1= 3 → x_2= 2 ($a_1 = 1$) → c= 39+30= 69
- x_1= 2 → x_2= 3 ($a_1 = 0$) → c= 26+45= 71

On voit bien que la meilleure décision à prendre si on veut terminer la 2ième période avec stock nul est d'acheter 5 unités à la 1ière période et 0 unité à la seconde.

De même, si on veut terminer la seconde période ace 1 unité $a_2 = 1$, la meilleure stratégie à prendre est la suivante : x_1=5 ($a_1 = 3$), x_2 =1 de coût c= 65+15=80

On obtient ainsi la sous-politique optimale à choisir dans la 2ième période est la suivante :

Si a_2=0 $(x_1, x_2) = (5{,}0)$ c=65

Si a_2=1 $(x_1, x_2) = (5{,}1)$ c=80

Si a_2=2 $(x_1, x_2) = (5{,}2)$ c=95

De la même façon on déduit la sous politique optimale à entreprendre lors de la 3ième période

(0≤ a_3 ≤ 1) :

- Si a_3=0 (x_1, x_2, x_3)= (5,2,2) c=95+40=135
- Si a_3=1 (x_1, x_2, x_4)= (5,2,3) c=95+60=155

Le tableau suivant résume les 5 sous-politiques optimales pour les 5 périodes. La politique optimale est bien sûr celle correspondante à la 5ième période (avec un stock nul bien-sûr)

Période	d_j	a_j	x_1	x_2	x_3	x_4	x_5	Coûts cumulés
1ière période	2	0 1 2 3	2 3 4 5	/ / / /	/ / / /	/ / / /	/ / / /	26 39 52 65
2ième période	3	0 1 2	5 5 5	2 1 2	/ / /	/ / /	/ / /	65+0=65 65+15=80 65+30=95
3ième période	4	0 1	5 5	2 2	2 3	/ /	/ /	95+40=135 95+60=155
4ième période	3	0 1 2	5 5 5	2 2 2	2 2 2	3 4 5	/ / /	135+33=168 135+44=179 135+55=190
5ième période	2	0	5	2	2	5	0	190+0=190

La politique optimale est donc : x_1=5, x_2=2, x_3=2, x_4=5, x_5=0

Le coût total= 190

Rappelons que problème précédent peut être formulé comme un PLNE :

$$(\mathrm{P})\begin{cases} Z(\min) = \sum_{j=1}^{5} P_j x_j \\ 0 \leq a_j \leq 5 - d_j \quad j = \overline{1,5} \\ a_j = a_{j-1} + x_j - d_j \quad j = \overline{1,5} \\ a_0 = a_5 = 0 \\ a_j, x_j \in N, \quad j = \overline{1,5} \end{cases}$$

Le dernier peut être décomposé (sur les 5 périodes) en cinq sous-problèmes $P_k(\alpha)$

$$P_k(\alpha)\begin{cases} Z_k(min) = \sum_{j=1}^{k} P_j x_j \\ 0 \leq a_j \leq 5 - d_j \quad j = \overline{1,k} \\ a_j = a_{j-1} + x_j - d_j \quad j = \overline{1,k} \\ a_0 = 0;\ a_k = \alpha \\ a_j, x_j \in N, \quad j = \overline{1,k} \end{cases}$$

Notons que la résolution de $P_k(\alpha)$ revient à chercher une sous-politique optimale sur les k premières périodes sachant qu'on veut terminer la $k^{\text{ième}}$ période avec un stock de α unités. Notre problème initial (P) est alors équivalent à $P_5(0)$.

Nous fixons donc d'abord la valeur de α, puis nous résolvons $P_k(\alpha)$ pour chaque valeur fixée de α.

Pour k=1 par exemple, en posant $Z_1(\alpha)$= valeur de la fonction objectif de $P_1(\alpha)$, on a $0 \leq \alpha \leq 3$, nous avons donc quatre sous problèmes :

$$P_1(\alpha)\begin{cases} Z_1(\alpha) = 13(\alpha + 2) \\ x_1 = \alpha + 2 \end{cases} \qquad \text{avec} \quad 0 \leq \alpha \leq 3$$

Notons que la méthode de résolution précédente est basée sur la relation de récurrence qui existe entre ces sous problèmes.

Résolution du problème de gestion de stock en utilisant les graphes : Algorithme de Bellman

Le problème de gestion de stock précédent peut être modélisé par un graphe orienté pondéré :

G= (V,E) tel que :

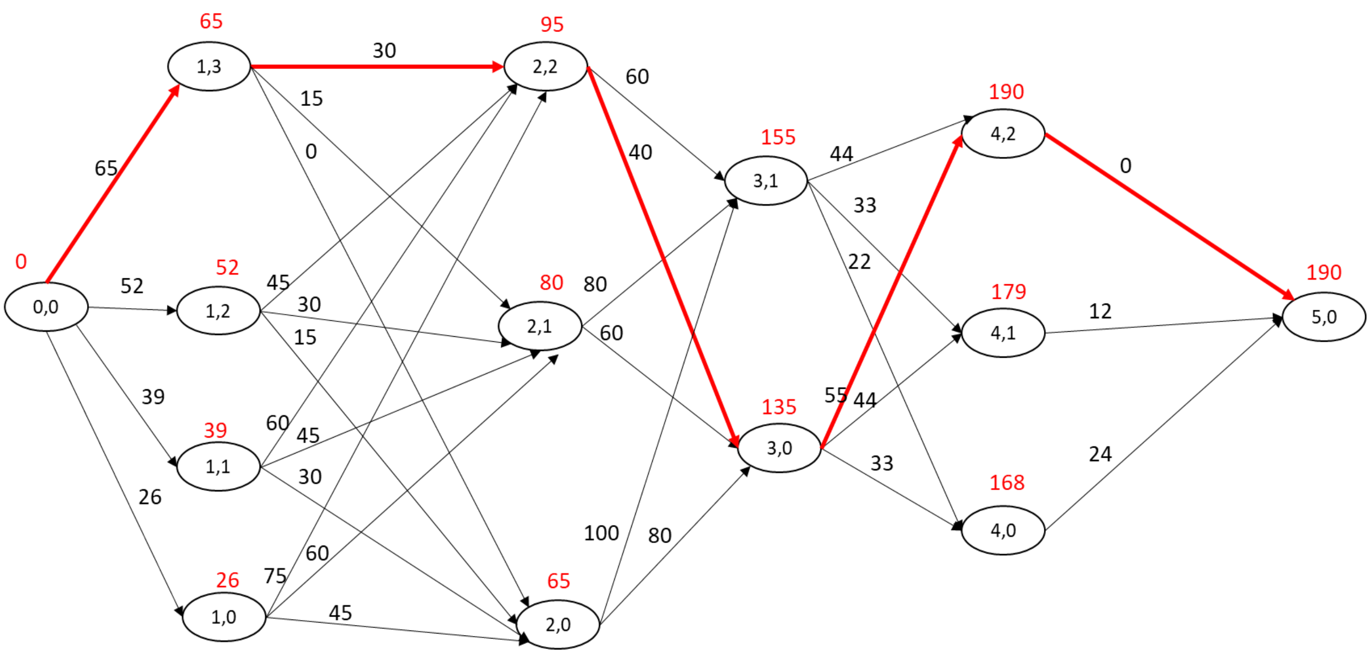
V ={(i, a_i) i=1,5 et $a_i \in \{0,1,2,3\}$} avec $a_0 = 0, a_5 = 0$ et $0 \leq a_j \leq 5 - d_j$

$((\text{i},a_i),(j,a_j)) \in E$ ssi j=i+1 et $x_j = a_j - a_{j-1} + d_j \leq 5 \quad j = \overline{1,k}$

Nous associons à chaque arc $((\text{i},a_i),(j,a_j))$ un poids $d_{ij} = p_j \times x_j = p_j \times (a_j - a_{j-1} + d_j)$.

Il s'agit bien sûr de trouver un plus cours chemin du sommet initial (0.0) vers le sommet final (5,0), qui correspond au problème initial.

Il est clair que le graphe ainsi construit est sans circuit. Nous appliquons l'algorithme de Bellman pour trouver un plus cours chemin du sommet initial (0.0) vers le sommet final (5,0).



Nous rappelons dans ce qui suit le principe de l'algorithme de Bellman :

<u>Algorithme de Bellman :</u> (graphe sans circuit)

Soit G=(V,E) un graphe orienté sans circuit pondéré muni d'une distance

$$d : E \rightarrow \mathbb{R}$$

$$e \rightarrow d_e \text{ (où } d_e \text{, poids associé à l'arête } e \text{)}$$

Soit s un sommet source sans prédécesseur.

$\pi(x)$ Représente (la plus courte) distance de s à x

Algorithme de Bellman
Début
Marquer le sommet s : $\pi(s) = 0$
Tant qu'il ∃ des sommets non marqués, dont tous les prédécesseurs ont été marqués faire :

- Marquer un tel sommet « le noter y »
- $\pi(y)_{x \in \Gamma^{-}(y)\{prédecesseurs\ de\ y\}}$ = Min { $\pi(x)$+ d(x,y)}

Fin Tant que
Fin

Dans notre cas : $\pi(i, a_i) = Min_{a_{i-1}} [\pi(i-1, a_{i-1}) + P_i(a_i - a_{i-1} + d_i)]$

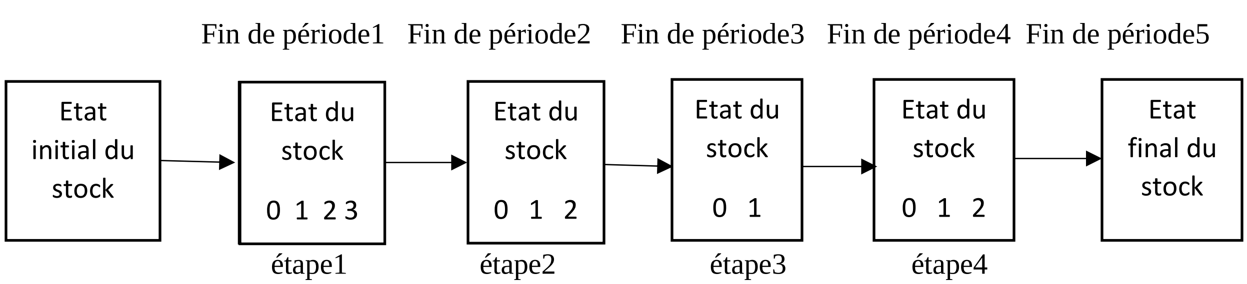
Le chemin optimal est : (0,0)-(1,3)-(2,2)-(3,0)-(4,2)-(5,0)

Qui correspond à la politique optimale : $x_1 = 5, x_2 = 2, x_3 = 2, x_4 = 5, x_5 = 0$

<u>Remarques :</u>

1. Le premier exemple de ce chapitre montre clairement que toute solution du problème posé est une suite de 5 décisions $(x_1, x_2, x_3, x_4, x_5)$, où chaque décision x_i représente l'achat pour la période *i*.

Par ailleurs, il est aussi clair que chaque décision prise à la période *i* ne dépend au fait que de l'état du stock à la fin de la période *i-1*. Donc on a ici un processus de décision séquentiel (PDS), où le futur ne dépend que du présent et non pas du passé, schématisé comme suit :



2. Nous supposons que dans toute la suite de ce chapitre on ne s'intéressera qu'aux problèmes d'optimisation combinatoire, dont la modélisation sous la forme d'un PDS est faisable, donc justifiable par la programmation dynamique.

Formalisation de la programmation dynamique

Quelques notions fondamentales associées à un PDS (systèmes évolutifs)

Considérons un PDS en n étapes modélisant un certain problème d'optimisation combinatoire dont le critère (objectif) est une minimalisation et évoluant d'un état initial e_0 vers un état final e_n

Pour chaque étape k, on définit deux ensemble X_k et E_k , ainsi que deux application C_k et T_k où :

X_k : ensemble des décisions que l'on peut prendre à l'étape k

E_k : ensemble des états dans lesquels le système se trouve à la fin de l'étape k

$C_k : E_{k-1} \times X_k \rightarrow \mathbb{R}$

$(e_{k-1}, x_k) \rightarrow C_k(e_{k-1}, x_k)$

$C_k(e_{k-1}, x_k)$ coût associé à la prise de décision x_k quand le système se trouve dans l'état e_{k-1} à la fin l'étape k-1.

$T_k : E_{k-1} \times X_k \rightarrow E_k$

$(e_{k-1}, x_k) \rightarrow e_k = T_k(e_{k-1}, x_k)$

T_k : étant la fonction de transition qui assure le passage du système d'un état à un autre quand une décision est prise

Application au problème de gestion de stock

Ces différentes notions appliquées à l'exemple introductif sont :

n=5 (cinq étapes : une pour chaque période d'achat)

Pour la période k :

X_k: achats possibles pour la période k

E_k : états possible du stock à la fin de la période k

Posons : $E_k = S_k = \{s_k\ stock\}$

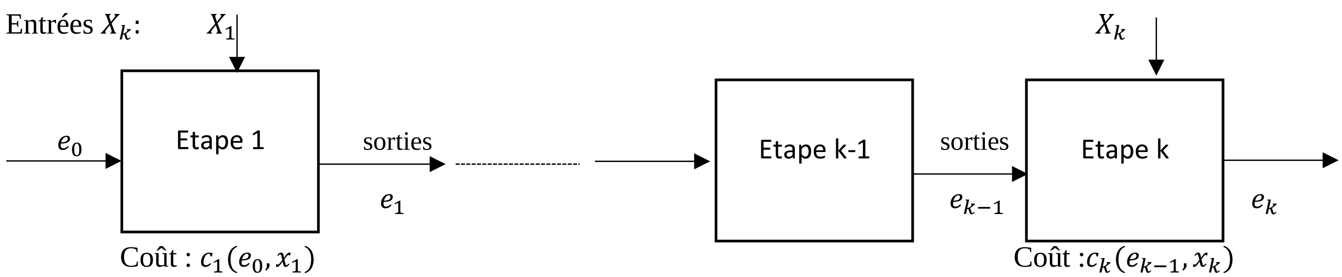
$T_k : S_{k-1} \times X_k \rightarrow S_k$

$(s_{k-1}, x_k) \rightarrow s_k = s_{k-1} + x_k - d_k$

$C_k : S_{k-1} \times X_k \to \mathbb{R}$

$(s_{k-1},\ x_k \ \to\ c_k(s_{k-1}, x_k) = p_k x_k + 0s_{k-1} = p_k x_k$

<u>Représentation d'un PDS par schéma</u>



Le but rechercher dans PDS est l'établissement d'une stratégie, basée sur une séquence de décision

$\{x_1^*, x_2^*, \dots \dots, x_k^*\}$ tel que $\sum_{k=1}^{n} c_k(e_{k-1}, x_k^*)$ soit minimale avec :

$$\begin{cases} e_0, e_n \ \textit{deux états connus} \\ \quad e_k = T_k(e_{k-1}, x_k^*) \end{cases}$$

Théorème

Toute sous-politique d'une séquence optimale est aussi optimale

Preuve

On utilise le raisonnement par l'absurde. Supposons le contraire (ie. Il existe une sous-politique d'une politique optimale, mais qui n'est pas optimale).

Soit $\wp^*(e_0, e_n) = \{x_1^*, x_2^*, \dots \dots, x_n^*\}$ une politique optimale d'un PDS à n étapes (problème de minimisation dans notre cas), et

$\wp^*(e_i, e_j) = \{x_i^*, x_{i+1}^*, \dots \dots, x_j^*\} \subset \{x_1^*, x_2^*, \dots \dots, x_n^*\}$ une sous-politique optimale de $\wp^*(e_0, e_n)$

Supposons maintenant qu'il existe une sous-politique non optimale $\wp(e_i, e_j) = \{x_i\ \ , x_{i+1}, \dots \dots, x_j\ \ \} \subset \{x_1^*, x_2^*, \dots \dots, x_n^*\}$ de $\wp^*(e_0, e_n)$.

Notons par $C(\wp(e_i, e_j))$ *le cout* d'une politique non optimale d'un état e_i *vers un état* e_j, et par $C(\wp^*(e_i, e_j))$, le cout d'une politique optimale d'un état e_i *vers un état* e_j . Alors an a :

$\mathrm{C}(\wp^*(e_0, e_n) = \mathrm{C}(\wp^*(e_0, e_i) + C(\wp^*(e_i, e_j)) + C(\wp^*(e_j, e_n)) = C(\wp^*(e_0, e_i) + C(\wp(e_i, e_j)) + C(\wp^*(e_j, e_n))$

Impossible puisque $C(\wp^*\ (e_i, e_j)) < C(\wp(e_i, e_j))$.

Remarque :

Un PDS est donc caractérisé par l'hypothèse de Markov (le futur ne dépend que du présent) et par la propriété de Markov, appelée aussi principe de l'optimalité (théorème précèdent).

Ces 2 caractéristiques facilitent le calcul du coût de la politique optimale par le biais de relations de récurrence dites : récurrence avant ou récurrence arrière.

Relations de récurrences :

Récurrence avant :

Notons par $AV_n\ (e_n)$ le coût d'une politique optimale d'un état e_0 vers un état e_n , qui est aussi le coût du plus court chemin d'un graphe approprié d'un nœud e_0 vers un nœud e_n.

$AV_n(e_n)$= C ($\wp^*(e_0, e_n)$)   où

$\wp^*(e_0, e_n) = \{x_1^*, x_2^*, \dots \dots, x_n^*\}$ et $e_i^* = T_i(e_{i-1}^*, x_i^*)$ $e_0^* = e_0, e_n^* = e_n$

Le calcul de $AV_n\ (e_n)$ se fera de façon récursive en utilisant la relation suivante dont la validité est due au principe d'optimalité d'un PDS

$$\begin{cases} AV_k(e_k) = Min_{\{(e_{k-1}, x_k)/e_k = T_k(e_{k-1}, x_k)\}}\{AV_k(e_{k-1}) + c_k(e_{k-1}, x_k\} \\ AV_0(e_0) = 0 \end{cases}$$

$AV_k(e_k)$: coût d'une politique optimale d'un état e_0 vers un état e_k

= solution optimale de la restriction du problème posé aux k premières étapes = solution optimale du sous-problème $\wp_k$ du problème initial.

Exemple :

Reprenons le premier exemple de ce chapitre, ainsi que son modèle linéaire.