

Programmation dynamique

Considérons les deux problèmes suivants de programmation en nombres entiers.

Problème du sac à dos

On désire charger un sac de capacité volumique b avec des objets de type $1, \dots, N$. Chaque objet de type k a un volume a_k et une valeur v_k . Combien d'objets de chaque type faut-il prendre pour maximiser la valeur du chargement ?

Notons u_k le nombre d'objets de type k à emporter. On doit donc résoudre le problème suivant de programmation linéaire en nombres entiers.

$$\text{Max } z = \sum_{k=1}^N v_k u_k$$

$$\text{s. c. } \sum_{k=1}^N a_k u_k \leq b$$

$$u_k \geq 0 \text{ entier, pour tout } k$$

Problème de gestion de stock

On désire gérer un stock sur N périodes ? On connaît l'état initial E_1 du stock au début de la période 1. Au début de chaque période, on peut commander un nombre $k \in \{0, 1, \dots, M\}$ d'unités du produit stocké. Le stock a une capacité maximale C . On connaît la demande d_k pour chaque période k . Le coût d'une commande de x unités est une fonction $c_k(x)$ qui peut dépendre de la période k . Si le stock est de x unités au début de la période k et qu'on décide de commander u unités, un montant $p_k(x, u)$ doit être payé pour le stockage ou la pénurie durant la période k . Aussi, on paie $p_{N+1}(x)$ pour un stock ou une pénurie de x unités à la fin des N périodes. Quelle est la politique optimale de commandes ?

Notons u_k le nombre d'unités commandées au début de la période k . On doit donc résoudre le problème de programmation en nombres entiers suivant.

$$\text{Min } z = \sum_{k=1}^N c_k(u_k) + \sum_{k=1}^N p_k(E_1 + \sum_{i=1}^{k-1} (u_i - d_i), u_k) + p_{N+1}(E_1 + \sum_{i=1}^N (u_i - d_i))$$

$$\text{s. c. } E_1 + \sum_{i=1}^k u_i - \sum_{i=1}^{k-1} d_i \leq C \quad \text{pour } k = 1, \dots, N \quad (\text{respect de la capacité})$$

$$E_1 + \sum_{i=1}^k u_i - \sum_{i=1}^{k-1} d_i \geq d_k \quad \text{pour } k = 1, \dots, N \quad (\text{si les pénurie sont interdites})$$

$$0 \leq u_k \leq M \text{ entier, pour } k = 1, \dots, N$$

Ces deux problèmes peuvent être résolus à l'aide de la même technique, appelée *programmation dynamique*. Le contexte est le suivant :

1. On doit gérer un système dynamique à temps discret
2. La fonction à optimiser est additive dans le temps

1. Le système dynamique

L'état du processus au fil des N périodes a la forme $x_{k+1} = f_k(x_k, u_k)$ où

- k numérote les périodes (ou étapes) du processus
- x_k est l'état du système au début de la période k (avant la prise de décision), et x_{N+1} est l'état du système en fin de processus. L'état x_1 est connu.
- u_k est la décision prise à la période k

Pour le problème du sac à dos,

- x_k est l'espace restant pour les objets de type $k, k+1, \dots, N$
- u_k est le nombre d'objets de type k à prendre.
- $f_k(x_k, u_k) = x_k - a_k u_k$

Pour le problème de gestion de stock,

- x_k est le nombre d'unités stockées au début de la période k ($x_1 = E_1$)
- u_k est le nombre d'unités commandées au début de la période k
- $f_k(x_k, u_k) = x_k + u_k - d_k$

2. La fonction additive à optimiser

La fonction à optimiser doit être additive, de la forme $\sum_{k=1}^N g_k(x_k, u_k) + g_{N+1}(x_{N+1})$ où $g_k(x_k, u_k)$ représente la valeur de la décision u_k étant donné l'état x_k et $g_{N+1}(x_{N+1})$ représente la valeur de l'état x_{N+1} atteint en fin de processus.

Pour le problème du sac à dos, $g_k(x_k, u_k) = v_k u_k$ et $g_{N+1}(x_{N+1}) = 0$, alors que pour le problème de gestion de stock, $g_k(x_k, u_k) = c_k(u_k) + p_k(x_k, u_k)$ et $g_{N+1}(x_{N+1}) = p_{N+1}(x_{N+1})$.

3. Stratégie optimale

Chaque variable d'état x_k fait partie d'un ensemble S_k d'états possibles, et chaque décision u_k doit être prise dans un ensemble $U_k(x_k)$ de choix restreint qui dépend de l'état x_k actuel.

Pour le problème du sac à dos, on a $S_k = \{0, 1, \dots, b\}$ et $U_k(x_k) = \{0, 1, \dots, \lfloor \frac{x_k}{a_k} \rfloor\}$.

Pour le problème de gestion de stock, si on n'accepte pas les pénuries, on a $S_k \subseteq \{r_k, r_k+1, \dots, C\}$, avec $r_k = \max\{0, d_k - M\}$. L'inclusion peut être stricte, car à titre d'illustration, si $M=2$, $d_2=1$ et $d_3 = 4$, on a $r_2 = 0$ alors qu'on doit avoir $x_2 \geq 1$. En effet, si $x_2 \leq 0$, on a $x_3 \leq 1$, ce qui implique une pénurie d'au moins une unité en fin de période 3. Aussi, si $E_1=3$ et $d_1 = 2$, on a $x_2 \geq 1$.

Toujours pour le problème de gestion de stock, on a $U_k(x_k) \subseteq \{a_k, a_k + 1, \dots, b_k\}$ avec $a_k = \max\{0, d_k - x_k\}$ et $b_k = \min\{M, C - x_k\}$. À nouveau, les inclusions peuvent être strictes. À titre d'illustration, si $M=2$, $C=2$, $d_1=1$, $d_2=1$, $d_3=4$ et $E_1=0$, on a $u_1=2$ car sinon $x_2=0$ et on a vu que cet état n'est pas acceptable si on interdit les pénuries. Pourtant, $a_1=1$ et $b_1=2$.

Une *politique de décision* est une suite de fonctions $\mu_1, \mu_2, \dots, \mu_N$ où μ_k associe à chaque état x_k de S_k une décision $u_k = \mu_k(x_k) \in U_k(x_k)$.

Le but est de déterminer une politique de décision qui optimise la fonction additive.

Propriété

- Si $\mu_1, \mu_2, \dots, \mu_N$ est une politique optimale pour les N périodes, et si on atteint l'état x_k à la période k en suivant cette politique, alors la politique partielle $\mu_k, \mu_{k+1}, \dots, \mu_N$ est optimale pour les N-k+1 dernières périodes, en débutant avec l'état x_k .

Algorithme de détermination d'une politique optimale

- Poser $C_{N+1}(x_{N+1}) = g_{N+1}(x_{N+1})$ pour tout x_{N+1} dans S_{N+1}
- Pour $k=N, N-1, \dots, 1$ et pour tout $x_k \in S_k$ faire
 Poser $C_k(x_k) = \underset{u_k \in U_k(x_k)}{\text{opt}} \{g_k(x_k, u_k) + C_{k+1}(f_k(x_k, u_k))\}$
 et stocker une décision $\mu_k(x_k)$ pour laquelle l'optimum est atteint
- Le coût optimal est $C_1(x_1)$ et une politique optimale consiste à prendre les décisions suivantes : $u_1 = \mu_1(x_1)$ et $u_k = \mu_k(f_{k-1}(x_{k-1}, u_{k-1}))$ pour $k=2, \dots, N$.

Exemple pour le problème du sac à dos avec 3 types d'objets, un volume disponible de 6, et les valeurs suivantes :

| Objet k | 1 | 2 | 3 |
|--------------|---|---|---|
| Valeur v_k | 5 | 3 | 7 |
| Volume a_k | 3 | 2 | 4 |

On a donc $C_4(x_4) = 0$.

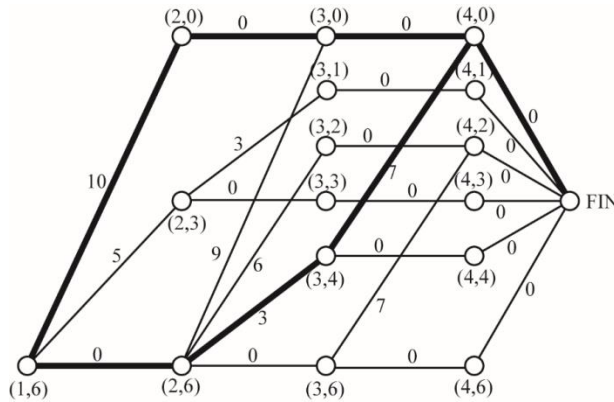
| x_3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------|---|---|---|---|---|---|----|
| $C_3(x_3)$ | 0 | 0 | 0 | 0 | 7 | 7 | 7 |
| $\mu_3(x_3)$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| x_2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| $C_2(x_2)$ | 0 | 0 | 3 | 3 | 7 | 7 | 10 |
| $\mu_2(x_2)$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

On a donc $C_1(x_1) = C_1(6) = \max\{0+10, 5+3, 10+0\} = 10$. Il existe donc deux politiques optimales : choisir $x_1=2$, et $x_2=x_3=0$ ou $x_1=0$, et $x_2=x_3=1$.

On peut représenter ce processus à l'aide d'un graphe :

- pour chaque période $k (1 \leq k \leq N+1)$ et pour chaque $x_k \in S_k$ on crée un sommet (k, x_k) ;
- pour chaque sommet (k, x_k) , et pour chaque décision $u_k \in U_k(x_k)$ on crée un arc de (k, x_k) vers $(k+1, f_k(x_k, u_k))$ de coût $g_k(x_k, u_k)$.
- On crée finalement un sommet FIN et, pour chaque $x_{N+1} \in S_{N+1}$ on crée un arc de $(N+1, x_{N+1})$ vers FIN de coût $g_{N+1}(x_{N+1})$.

On recherche alors un chemin de longueur optimale du sommet $(1, x_1)$ vers FIN.



Exemple pour le problème de gestion de stock sur 4 périodes avec stock initial nul, stock final devant être nul ou d'une unité, pénuries interdites, capacité maximale de stockage $C=7$, limite de commande M infinie, coût de stockage $p_k(x_k, u_k) = \max\left\{1, 2\left(x_k + u_k - \frac{d_k}{2}\right) - 1\right\}$, profit $p_5(x_5) = -4x_5$, coûts de commande $c_k(u_k)$ (identiques pour chaque période k) et demandes d_k comme indiqués ci-dessous.

| | | | | | | | | | | | | | |
|----------|---|---|----|----|----|----|----|----|-------|---|---|---|---|
| u_k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | k | 1 | 2 | 3 | 4 |
| $c(u_k)$ | 0 | 7 | 15 | 21 | 25 | 28 | 30 | 31 | d_k | 4 | 4 | 2 | 1 |

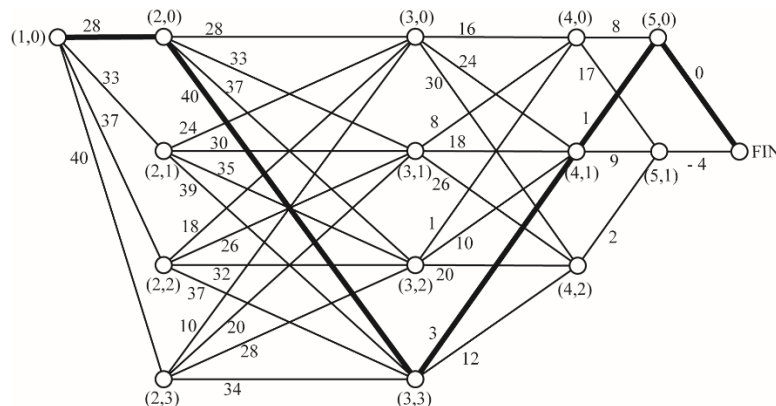
On a $x_{N+1} = x_5 = 0$ ou 1 et $C_5(0) = 0$ alors que $C_5(1) = -4$.

| | | | |
|--------------|---|---|----|
| x_4 | 0 | 1 | 2 |
| $C_4(x_4)$ | 8 | 1 | -2 |
| $\mu_4(x_4)$ | 1 | 0 | 0 |

| | | | | |
|--------------|----|----|---|---|
| x_3 | 0 | 1 | 2 | 3 |
| $C_3(x_3)$ | 24 | 16 | 9 | 4 |
| $\mu_3(x_3)$ | 2 | 1 | 0 | 0 |

| | | | | |
|--------------|----|----|-----|----|
| x_2 | 0 | 1 | 2 | 3 |
| $C_2(x_2)$ | 44 | 43 | 41 | 34 |
| $\mu_2(x_2)$ | 7 | 6 | 4,5 | 1 |

On a donc $C_1(0) = \min\{(25+3)+44, (28+5)+43, (30+7)+41, (31+9)+34\} = 72$, et la politique optimale est donc de commander 4 unités en première période, 7 en 2^{ème} et rien en 3^{ème} et 4^{ème} période.



Problème d'investissement. Une entreprise a un budget de 1 millions de dollars pour l'amélioration de ses 3 usines pour l'année à venir. On connaît pour chaque usine le bénéfice annuel espéré en fonction de la somme investie pour l'amélioration du site. On veut déterminer la politique optimale d'investissement pour maximiser les bénéfices annuels. On a

- x_k est le montant disponible pour les investissements sur les sites $k, k+1, \dots, N$, et x_{N+1} est la somme non investie.
- u_k est le montant investi pour l'amélioration du site k
- $g_k(x_k, u_k)$ est le bénéfice retiré d'un investissement u_k pour le site k
- $g_{N+1}(x_{N+1})$ est le montant non investi, c'est-à-dire x_{N+1} .
- $f_k(x_k, u_k) = x_k - u_k$.

Résolvons ce problème en supposant que les montants investis doivent être des multiples de 200 mille dollars. Les bénéfices annuels espérés selon les investissements sont les suivants :

| usine | Montants investis (à multiplier par 100 mille dollars) | | | | | |
|-------|---|---|----|----|----|----|
| | 0 | 2 | 4 | 6 | 8 | 10 |
| | Bénéfice annuel espéré | | | | | |
| 1 | 2 | 6 | 8 | 10 | 11 | 12 |
| 2 | 1 | 5 | 8 | 10 | 12 | 13 |
| 3 | 4 | 8 | 11 | 13 | 14 | 15 |

On a donc $C_4(x_4) = x_4$ pour $x_4 = 0, 2, 4, 6, 8$ ou 10 .

| x_3 | 0 | 2 | 4 | 6 | 8 | 10 |
|--------------|---|---|----|-----|-----|-----|
| $C_3(x_3)$ | 4 | 8 | 11 | 13 | 15 | 17 |
| $\mu_3(x_3)$ | 0 | 2 | 4 | 4,6 | 4,6 | 4,6 |

| x_2 | 0 | 2 | 4 | 6 | 8 | 10 |
|--------------|---|-----|----|-----|----|-----|
| $C_2(x_2)$ | 5 | 9 | 13 | 16 | 19 | 21 |
| $\mu_2(x_2)$ | 0 | 0,2 | 2 | 2,4 | 4 | 4,6 |

Ainsi, $C_1(10) = \max\{2+21, 6+19, 8+16, 10+13, 11+9, 12+5\} = 25$. La politique optimale est d'investir 200 millions dans la première usine, 400 millions dans la deuxième usine, et 400 millions dans la troisième. Sous forme de graphe, ça donne ceci :

