

INTRODUCTION
A
LA PROGRAMMATION DYNAMIQUE
STOCHASTIQUE

Partie 00: PROGRAMMATION
DYNAMIQUE STANDARD

MOTIVATION, MODELISATION ET RESOLUTION D'UN
PROBLEME

T. AL ANI

Laboratoire A²SI-ESIEE-Paris

e-mail: t.alani@esiee.fr

INTRODUCTION

EXEMPLE 1_: un problème de gestion de stocks (stochastique)

Un commerçant passe commande d'un des articles de son assortiment chaque fin de semaine et est livré le lundi matin.

Les ventes de chaque semaine sont aléatoires mais (supposées) indépendantes les unes des autres. Le commerçant dispose de modèles de prévision lui permettant d'anticiper les ventes des semaines à venir.

Le commerçant cherche à établir des règles de réapprovisionnement spécifiant la quantité à commander en fin de semaine en fonction de l'état de son stock. Ces règles doivent lui permettre de minimiser ses coûts de gestion pour les N semaines à venir. Ces coûts comprennent :

- des coûts de réapprovisionnement,
- des coûts de stockage,
- des coûts de pénurie.

EXEMPLE 2_: un problème du sac à dos (déterministe)

Un voyageur veut charger son sac à dos de capacité maximale N_{\max} . Pour cela, il peut choisir parmi N objets uniques et indivisibles numérotés de 1 à N .

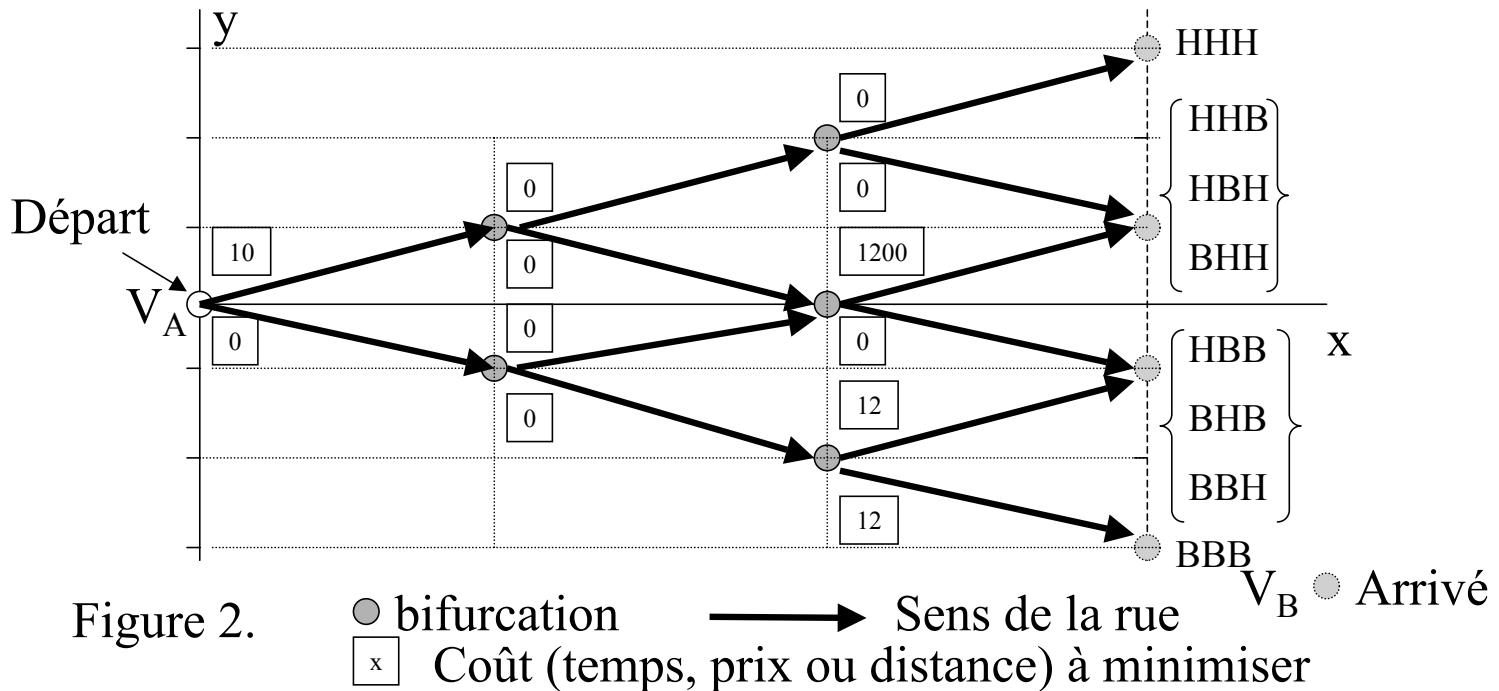
L'objet k a un encombrement a_k et une utilité c_k .

Le voyageur cherche à déterminer les objets à emporter de manière à maximiser l'utilité de sa sélection sans dépasser la capacité du sac.

Remarque. Le problème précédent est un problème de sac à dos binaire, chaque objet n'étant présent qu'en un seul exemplaire. On peut aussi considérer des situations où chaque objet est disponible en un nombre d'exemplaires fixés ou, même, arbitrairement grand. Le point important (et difficile) réside dans l'indivisibilité des objets susceptibles d'être sélectionnés.

EXEMPLE 3:

Déplacement d'un voyageur en sens diagonal unique d'une ville V_A vers une autre ville V_B en coût minimal- cas d'un réseau triangulaire



EXEMPLE 3 (suite)

Les réalisations des décisions sont considérées stochastiques mais nous supposons que les décisions H ou B amènent à un déplacement diagonal certain vers le Haut ou vers le Bas, faisant encourir un coût $c_H(x,y)$ si la décision est H et $c_B(x,y)$ si la décision est B.

EXEMPLE 3 (suite)

Position du problème

- Si nous conseillons au voyageur de prendre, à chaque intersection (nœud ou sommet), le sens diagonal (arc) vers le Haut (décision H) alors il se rappellera de ce conseil avec une probabilité de $3/4$ et il effectuera ce déplacement.
- Avec une probabilité de $1/4$ il oubliera notre conseil et il effectuera un déplacement en sens diagonal vers le Bas (décision B).

EXEMPLE 3 (suite)

- Si nous conseillons au voyageur de prendre, à chaque intersection, le sens diagonal vers le Bas (décision B) alors il se rappellera de ce conseil avec une probabilité de $3/4$ et il effectuera ce déplacement.
- Avec une probabilité de $1/4$ il oubliera notre conseil et il effectuera un déplacement en sens diagonal vers le Haut (décision H).

EXEMPLE 3 (suite)

Par exemple,

- la séquence de décision $i=1$ pour BHB possédant un coût minimal : $C_1=0+0+0=0$

et une probabilité :

$$P_1 = 3/4 \times 3/4 \times 3/4 = 27/64$$

si le chemin conseillé à suivre est : vers le Bas puis vers le Haut et enfin vers le Bas.

EXEMPLE 3 (suite)

- la séquence de décision $i=1$ pour HBH possédant un coût minimal :
 $C_1=10+0+1200=1210$ et une probabilité :

$$P_1 = 1/4 \times 1/4 \times 1/4 = 1/64$$

si le chemin conseillé à suivre est : vers le Bas puis vers le Haut et enfin vers le Bas.

EXEMPLE 3 (suite)

- la séquence de décision $i=1$ pour HHB possédant un coût minimal :

$$C_1 = 10 + 0 + 0 = 10$$

et une probabilité :

$$P_1 = 1/4 \times 3/4 \times 3/4 = 9/64$$

si le chemin conseillé à suivre est : vers le Bas puis vers le Haut et enfin vers le Bas.

EXEMPLE 3 (suite)

Ce voyageur se comportera de cette façon à chaque intersection sans tenir compte de sa bonne ou mauvaise mémoire a priori.

Par conséquent, quel que soit nos conseils, nous ne pouvons pas être sûrs de la trajectoire que notre voyageur effectuera, mais notre conseil déterminera certainement les probabilités de tous les résultats (sous-problèmes).

EXEMPLE 3 (suite)

Nous souhaitons minimiser le coût espéré (expected cost) de ce voyage ce que permet de minimiser le coût moyen (average cost) si le voyageur répète son voyage plusieurs fois avec leurs différents coûts à cause des trous de mémoire (s'il y en a).

EXEMPLE 3 (suite)

Un critère alternatif, parmi d'autres, que nous pourrions adopter serait de maximiser la probabilité que le chemin emprunté par notre voyageur coûte une valeur inférieure à un certain nombre Z .

Similarités entre les trois exemples

- Un système évoluant pendant un nombre fini d'étapes ordonnées.
- Une décision à prendre à chaque étape.
- Une évolution dynamique de l'état du système indépendante du passé (une fois l'état courant est connu).
- Une mesure de performances globales obtenues en sommant des mesures associées à chaque étape.

Notion	Gestion du stock	Sac à dos	Voyageur
étape	semaine	objet	déplacement
Décision de l'étape	Quantité à commander pour le début de la semaine	Sélectionner ou non l'objet	Aller vers le haut ou vers le bas
État en début de l'étape	Quantité en stock	Capacité du sac	Déplacement initial
Performance de l'étape	Coûts de gestion pendant une semaine	Utilité de l'objet (s'il est sélectionné)	Prix minimal ou distance minimale du parcours

Le terme *Programmation Dynamique*
(Dynamic Programming (DP))

Introduit par **Bellman** [Bel57] pour décrire les techniques qu'il a apportées pour étudier une classe de problèmes d'optimisation nécessitant des séquences de décisions.

Depuis, il y a eu beaucoup de développements et d'applications.

Programmation dynamique :

- Procédure d'optimisation applicable aux problèmes nécessitant une séquence de décisions corrélées (ou dépendantes). Une séquence de décisions produisent une séquence de situations : cherche à maximiser (ou minimiser) certaine valeur mesurée.
- Résoudre un grand nombre de sous-problèmes pour résoudre un problème complet donné.

Cette propriété n'est pas partagée par d'autres techniques.

Formulation du problème de la programmation dynamique en tant qu'un système dynamique à étapes discrètes

Deux caractéristiques principales du problème de base déterminent sa structure :

- (1) un systeme dynamique sous-jacent à étapes discrètes,
et
- (2) un coût fonctionnel qui est additif dans le temps.

Le **systeme dynamique** est de la forme

$$x_{k+1} = f_k(x_k, u_k, \omega_k); k = 0, 1, \dots, N-1, \text{ où}$$

k indice de l'étape discrète (e. g. temps discret),

x_k est l'**état** (supposé ici discret) du système qui récapitule l'information dans le passé appropriée pour la future optimisation.

A l'étape k , décrit la situation courante du système.

$u_k = \mu_k(x_k)$ est la **fonction de décision** à choisir à l'étape k avec la connaissance d'état x_k ;

ω_k est un paramètre aléatoire (également appelé **perturbation** ou **bruit**),

N est l'**horizon** ou le nombre fois de décisions appliquées.

f_k est la **fonction de transfert** de l'étape k .

Le coût immédiat ou fonctionnel est additif en un sens que le coût $g_k(x_k, u_k, \omega_k)$; est encouru à chaque étape k ; et le coût total le long d'un échantillon de n'importe quelle trajectoire de système est

$$g_N(x_N) + \sum_{k=0}^{N-1} g^k(x_k, u_k, \omega_k)$$

$g_N(x_N)$ est un coût terminal encouru à la fin du processus.

Déterministe vs stochastique

Pour les systèmes déterministes, c.-à-d. lorsqu'il n'y a pas de perturbations aléatoires (ou ce qui revient au même, lorsqu'elles ne peuvent prendre qu'une seule valeur), les règles de conduite optimales et les décisions associées sont obtenues en minimisant les coûts totaux.

Les méthodes de programmation dynamique basés sur des *modèles déterministes* supposent que le *coût* (cost) et le changement de l'*état* (state) résultants de chaque décision, même pour leurs valeurs futures, sont connus avec certitude [Dre77][Bat00].

Malgré la simplicité attirante que possède ces méthodes, ceci n'est le cas ni dans nos vies courantes ni dans le monde des affaires ni dans le domaine scientifique ou de l'ingénierie.

Dans ce cours nous introduirons une approximation plus proche de la réalité et nous supposerons que chaque décision peut produire un certain nombre de réalisations possibles.

Nous supposons que chacune de ces réalisations possède une probabilité connue a priori.

Pour un modèle plus réel, ces probabilités ne sont pas connues a priori mais peuvent être estimées.

Pour les systèmes stochastiques, les fonctions g_k dépendent des perturbations aléatoires w_k et sont donc des variables aléatoires elles aussi ! Il est alors impossible de minimiser les coûts totaux. L'approche traditionnelle consiste à minimiser leur *espérance*.

Nous sommes intéressés aux situations où des décisions sont prises par étapes. Les résultats de chaque décision ne sont pas entièrement prévisibles mais peuvent être observés avant que la prochaine décision soit prise. L'objectif est de réduire au minimum un certain coût - une expression mathématique de ce qui est considéré comme des résultats souhaitables.

Un aspect principal de tels problèmes est que des décisions ne peuvent pas être regardées indépendamment puisqu'on doit équilibrer le désir pour un coût actuel bas avec la possibilité que les coûts futurs élevés étant inévitables.

Cette idée est capturée dans la technique de programmation dynamique stochastique par laquelle on choisit à chaque étape une décision qui réduit au minimum la somme du coût de l'étape courante, et le meilleur coût qui peut être estimé des étapes futures.

Nous formulons donc le problème comme suit :
Nous souhaitons prendre des décisions u_0, u_1, \dots, u_{N-1} afin de réduire au minimum le coût espéré

$$E_{\omega_0, \dots, \omega_{N-1}} \left[\sum_{k=0}^{N-1} g^k(x_k, u_k, \omega_k) + g_N(x_N) \right]$$

où l'espérance est prise sur la distribution conjointe des perturbations ω_k .

PROCESSUS DE DECISIONS SEQUENTIELLES

Ce processus est formé

- d'un système dynamique à étapes discrètes évoluant pendant $N-1$ périodes selon le processus

$$x_{k+1} = f_k(x_k, u_k, \omega_k); k = 0, 1, \dots, N-1,$$

- d'une fonction additive au fil des périodes

L'objectif lors de l'étude d'un tel système consiste à établir des règles de conduite ou de gestion optimisant les performances globales du système (plus dans un instant sur ce point).

Conduite d'un système dynamique

- Au début de l'étape 1, le système est dans l'état initial x_0 .
- Successivement, pour chaque étape $k=0, 1, 2, \dots, N-1$
 - L'état x_k du système est observé et une décision u_k est prise d'après la fonction de décision $u_k = \mu_k(x_k)$ associant à chaque état (possible) une décision;
 - Il se produit ensuite une perturbation aléatoire ω_k dont la loi peut dépendre de x_k et/ou u_k ;
 - Des coûts sont encourus, s'élevant à $g_k(x_k, u_k, \omega_k)$;
 - Le système évolue vers son état suivant d'après sa fonction de transfert $x_{k+1} = f_k(x_k, u_k, \omega_k)$;
 - A l'étape N le système s'arrête dans l'état final x_N et des coûts terminaux $g_N(x_N)$ sont encourus.

EXERCICES

1. Établir un modèle pour le problème 1 de gestion de stock défini en page 1 : définir l'état x_{k+1} en fonction de (x_k, u_k, ω_k) et le coût global g_k , supposé additif, en fonction de (x_k, u_k, ω_k) .

Hypothèses :

- l'horizon de planification = N ;
 - la perturbation aléatoire ω_k de l'étape k représente la demande pendant la semaine k ;
 - en cas de pénuries, les clients sont d'accord d'attendre une semaine
 - le coût de commande : $r_k(u_k) = K\delta(u_k) + c_k u_k$
où $\delta(u_k) = 1$ si $u_k > 0$ et 0 autrement;
- K est une valeur constante, c_k est un coût immédiat
- le coût de stockage : $s_k(x_k, u_k, \omega_k) = h_k x_{k+1}^+ = h_k \max(x_{k+1}, 0)$
 - le coût de pénurie : $s_k(x_k, u_k, \omega_k) = h_k x_{k+1}^- = h_k \max(x_{k+1}, 0)$

EXERCICES

2. Établir un modèle pour le problème de sac à dos défini en page 2 : définir l'état x_{k+1} en fonction de (x_k, u_k, ω_k) et le coût global g_k supposé additif.

Hypothèses :

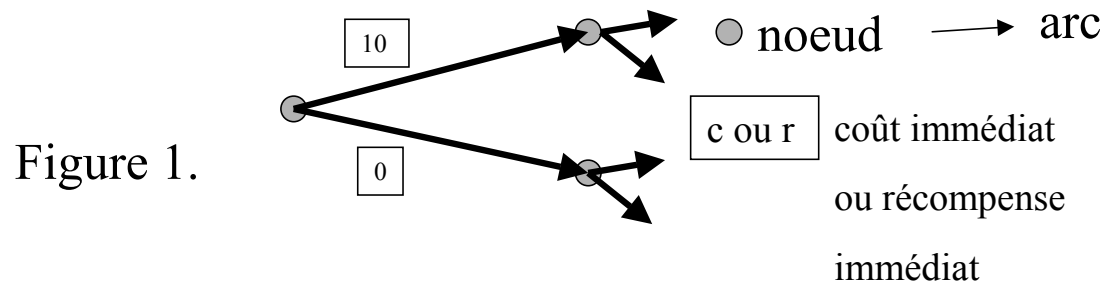
- L'état x_k du système au début de l'étape k représente la capacité réservée pour les objets k à N ;
- la décision de l'étape k est de prendre ou pas un seul objet;
- il n'y a pas de perturbation aléatoire ! : la capacité de l'étape $k+1$ = la capacité de l'étape k - $a_k u_k$;
- le coût d'une décision à l'étape k correspond à l'utilité de l'objet k s'il est sélectionné = $c_k u_k$, où c_k est le coût de l'utilité de l'objet k . Quelles sont les valeurs possibles de ce coût?

TERMINOLOGIE

Pour clarifier le développement de la suite de ce cours, nous introduirons quelques termes et quelques notations.

1. **Réseau** (network) : Plusieurs chemins (**arc** (arc)) ou **décisions** (decisions) liant un point de départ A : **solution initiale** à un point objectif B : **solution finale**. Chaque **bifurcation** est appelée **nœud** (vertex) de coordonnées (k, x_k) .

A chaque arc est associé un **coût immédiat** (cost) : temps, prix, distance, .. soit à une **récompense** (reward) : gain, augmentation de la production, ...



2. *Décision* : soit *commande en boucle ouverte*
soit *commande en boucle fermée*

En conformité avec la terminologie de l'ingénierie du contrôle, nous appellerons

- la solution spécifiée par une *séquence de décisions* :
commande en boucle ouverte (open-loop control)
- la solution spécifiée par une *politique* (policy) :
décision ou commande en boucle fermée
(feedback control).

La programmation dynamique produit une commande en boucle fermée.

3. Principe de l'optimalité de Bellman

(principle of optimality) :

Le meilleur chemin d'un point de départ A à un point objectif B possède la propriété suivante : Quelque soit la décision initiale à A, le chemin restant jusqu'à B, en commençant par le nœud qui suit A, doit être le meilleur chemin de ce nœud jusqu'à B.

4. **Sous-problème** ou **Sous-solution** $P_k(x_k)$ (subproblem) :
Un chemin de nœud (x_k) jusqu'au nœud B.

5. **Fonction optimale de la valeur espérée** (optimal expected value function) $J(k, x_k)$:

C'est une règle qui attribue des valeurs aux différents sous-problèmes). $J(k, x_k)$ est obtenue en optimisant (en minimisant ou maximisant) :

$$E_{\omega_0, \dots, \omega_{N-1}} \left[\sum_{k=0}^{N-1} g^k(x_k, u_k, \omega_k) + g_N(x_N) \right]$$

L'espérance est prise par rapport à la distribution conjointe des perturbations $\{\omega_k, \dots, \omega_{N-1}\}$.

Le **coût espéré** de la suite du processus si nous commençons au nœud (k, x_k) et utilisons la politique optimale de la commande en boucle fermée.

6. Argument de J : $a = (k, x_k)$: Indice correspond à un sous-problème particulier P_a . P_a indique le meilleur chemin à partir de nœud k jusqu'au nœud B.

7. Fonction politique optimale (optimal policy function) (μ_k^*) :

La règle qui associe la meilleur décision à chaque sous-problème.

8. **Relation de récurrence** (recurrence relation) :

C'est une une formule ou un ensemble de formules reliant différentes valeurs de J . Elle(s) est(sont) générée(s) par le principe d'optimalité.

9. *Conditions limites* (boundary conditions) : La valeur de J pour certains arguments est supposée évidente à partir de l'annoncé du problème et à partir de la définition de J sans la nécessité d'effectuer le calcul.

10. Technique de Programmation dynamique : Pour résoudre un problème par la technique de programmation dynamique, nous choisirons les arguments de la fonction J et définirons cette fonction de telle manière que le principe d'optimalité soit appliqué pour écrire une relation de récurrence.

La programmation dynamique déterministe détermine d'abord la politique π donnant une décision à chaque nœud, et ensuite en déduit la séquence optimale de décisions pour le nœud A.

Une politique et une séquence de décisions sont, dans ce cas, deux représentations équivalentes de la même solution optimale. La première permet de résoudre aussi d'autres problèmes tandis que la seconde est une représentation plus compacte et simple à communiquer.

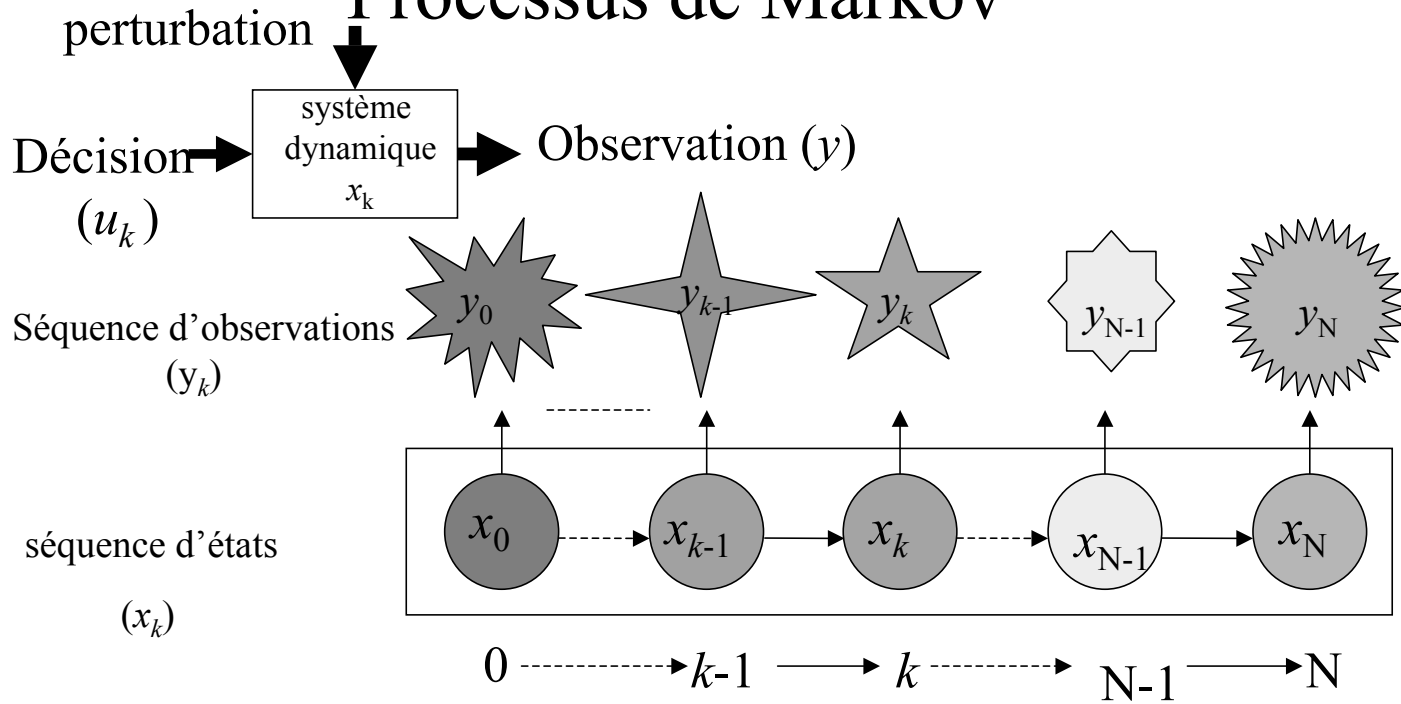
Dans le cas de la programmation dynamique stochastique, une politique et une séquence de décisions sont deux choses assez différentes puisque une séquence de décisions dicte la décision future d'une façon indépendante de la réalisation des décisions précédentes, tandis qu'une politique est dépendante de la réalisation.

DOMAINE DE DEFINITION DES GRANDEURS

- L'ensemble de tous les états x_k possibles à l'étape k est noté S_k , $k=0, 1, 2, \dots, N$. On suppose les ensembles finis.

Remarque. Les systèmes étudiés débutent à l'étape 0 dans l'état initial x_0 et évoluent pendant N étapes. Ils s'arrêtent donc au début de l'étape N dans l'état final x_N . Pendant cette étape terminale, il n'y a cependant ni décision ni perturbation aléatoire.

Processus de Markov



$\mathcal{P} = [p_1, p_2, \dots, p_{N_S}]$;
probabilités initiales

$$p_i = p(x_0 = i), i \in S_k$$

$u_0 \quad u_{k-1} \quad u_k \quad u_{N-1} \quad u_N$
Processus dynamique, N_s états ($x_k \in S_k = \{1, 2, \dots, N_S\}$)

$$a_{ij}(u_k) = p(x_{k+1} = j \mid x_k = i, u_k), A = [a_{ij}], i, j \in S_k$$

- L'ensemble de toutes les décisions u_k possibles à l'étape k est noté U_k , $k=0, 1, 2, \dots, N-1$.
- A une étape k donnée et dans un état x_k donné, toutes les décisions $u_k \in D_k$ ne sont pas forcément réalisables, on note $U_k(x_k)$ l'ensemble des **décisions admissibles** dans l'état x_k à l'étape k . On suppose les ensembles $U_k(x_k)$ finis.
- L'ensemble des valeurs que peuvent prendre les perturbations ω_k à l'étape k est noté Ω_k , $k=0, 1, 2, \dots, N-1$.
- La loi de probabilités de ω_k est $P_k : \Omega_k, \rightarrow \mathbb{R}$, $k=0, 1, 2, \dots, N-1$. Cette loi peut dépendre de l'état x_k et/ou de la décision u_k , auquel cas elle sera notée $P_k(\omega_k | x_k, u_k)$.
- Hypothèse : les perturbations ω_k sont indépendantes les une des autres.

POLITIQUES DE DECISION

Une *fonction de décision* à l'étape k est une fonction $\mu_k : S_k \rightarrow D_k$, $k=0, 1, 2, \dots, N-1$, qui associe à chaque état $x_k \in S_k$ une décision $u_k = \mu_k(x_k)$

Une fonction de décision μ_k est *admissible* si

$$\mu_k(x_k) \in U_k(x_k) \quad \forall x_k \in S_k$$

Une *politique* est une collection $\pi=(\mu_0, \dots, \mu_{N-1})$ de fonctions de décision.

Une politique π est *admissible* si chacune des fonctions de décision la formant l'est.

PERFORMANCES ET POLITIQUES OPTIMALES

Pour l'état initial x_0 , la *performance* (ou le coût) de la politique (admissible) π est égale à l'espérance des coûts totaux

$$J_{\pi}(x_0) = E_{\omega_0, \dots, \omega_{N-1}} \left[\sum_{k=0}^{N-1} g^k(x_k, \mu_k(x_k), \omega_k) + g_N(x_N) \right]$$

où $x_{k+1} = f_k(x_k, \mu_k(x_k), \omega_k); k=0, 1, \dots, N-1$

La politique π^* est optimale si,

$$\forall x_0 \in S_0, J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_{\pi}(x_0)$$

où Π est l'ensemble des politiques admissibles.

DETERMINISTIQUE VS. STOCHASTIQUE

Pour les programmes dynamiques déterministes, l'évolution de l'état du système est parfaitement prévisible. Ainsi, dès qu'une politique optimale π^* est connue, on peut appliquer chacune des fonctions de décision successivement et déterminer la *suite de décisions optimales* : partant de l'état initial x_0 , la première décision est $u^*_0 = \mu^*_0(x_0)$, on calcule alors $x_1 = f_0(x_0, u^*_0)$ puis $u^*_1 = \mu^*_1(x_1)$, et ainsi de suite...

En revanche, pour les programmes stochastiques, seule la première décision optimale $u^*_0 = \mu^*_0(x_0)$ peut être calculée immédiatement. Pour pouvoir déterminer l'état $x_1 = f_0(x_0, u^*_0, \omega_0)$ du système au début de l'étape 1, il faut que l'évènement ω_0 se réalise ! Ce n'est qu'après cette réalisation que l'état x_1 pourra être observé ou calculé et que la décision $u^*_1 = \mu^*_1(x_1)$, pourra être sélectionnée.

POLITIQUES RESIDUELLES ET PROBLEMES RESIDUELS

On note $P_k(x_k)$ le problème résiduel (ou partiel ou Sous-problème ou Sous-solution) débutant dans l'état x_k au début de l'étape k et se terminant au début de l'étape $N-1$.

De même, on note $\pi^{(k)}$ les politiques résiduelles ou partielles associées à $P_k(x_k)$:

$$\pi^{(k)} = (\mu_k, \dots, \mu_{N-1}), \quad k = 0, \dots, N-1.$$

La valeur minimale de l'espérance des coûts totaux pour le problème de décision $P_k(x_k)$ est notée $J_k^*(x_k)$ et la politique résiduelle $\pi^{(k)}$ est optimale si

$$J_k^*(x_k) = J_{\pi^{(k)}}(x_k) \quad \forall x_k \in S_k.$$

PRINCIPE DE RESOLUTION

On résout la famille de sous-problèmes emboîtés

$$\{P_k(x_k) \mid x_k \in S_k, k = N-1, \dots, 0\}$$

séquentiellement en partant du dernier, réduit à une seule étape. A l'itération i ($i = 2, 3, \dots, N+1$) correspondant à $k = N-1, N-2, \dots, 0$ de l'algorithme,

- on part d'une politique résiduelle optimale

$$\pi^{(N-i+1)} = \{\mu_{N-i+1}^*, \dots, \mu_{N-1}^*\}$$

pour les problèmes résiduels P_{N-k+1} ;

- on calcule une fonction de décision optimale μ_i^* afin d'obtenir une politique résiduelle optimale

$$\pi^{(N-i)} = \{\mu_{N-i}^*, \dots, \mu_{N-1}^*\}$$

pour des sous-problèmes P_{N-i} comportant une étape de plus.

PRINCIPE D'OPTIMALITE DE BELLMAN (1957)

Toute politique optimale est formée de politiques résiduelles optimales.

Remarque. Appliquée au problème du plus court chemin dans un réseau, ce principe s'énonce :

Si le plus court chemin du sommet A au sommet B passe par C, alors le sous-chemin de C à B est, lui aussi, un plus court chemin.

Remarque. Le principe d'optimalité n'est pas vérifié par tous les processus de décisions séquentielles. Les deux propriétés assurant sa validité pour les modèles étudiés ici sont

- la structure du système dynamique : l'évolution de l'état du système est indépendante du passé une fois l'état courant connu (*propriété Markovienne*);
- la forme de la fonction coût : somme de coûts associés à chaque étape.

RESOLUTION DANS LE CAS GENERAL

On considère un processus de décisions séquentielles formé

- d'un système dynamique évoluant pendant N étapes :

$$x_{k+1} = f_k(x_k, u_k, \omega_k) \quad k = 0, 1, \dots, N-1;$$

- d'une fonction coût additive associant à chaque 'étape un coût

$$g_k(x_k, u_k, \omega_k)$$

et pouvant inclure un coût terminal $g_N(x_N)$.

On cherche une politique de décision optimale, c.- c.-à-d. N fonctions de décision $\mu_0^*, \dots, \mu_{N-1}^*$, dont l'application successive aux périodes des $1, \dots, N$ minimise l'espérance des coûts totaux.

ALGORITHME DE PROGRAMMATION DYNAMIQUE STOCHASTIQUE

Données : Un processus de décisions séquentielles.

Résultats : Les tables contenant la politique optimale

$$\pi^* = \{\mu^*_0, \dots, \mu^*_{N-1}\}$$

et les espérance minimales J_k .

(1) Initialisation

$$J_N(x_N) = g_N(x_N) \quad \forall x_N \in S_N.$$

(2) Construction des tables optimales

Pour $k = N-1$ jusqu'à 0 faire

Pour $x_N \in S_k$ calculer

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E_{\omega_k} [g_k(x_k, u_k, \omega_k) + J_{k+1}(f_k(x_k, u_k, \omega_k))]$$

et stocker les valeurs $J_k(x_k)$ et les arguments $\mu^*_k(x_k)$

Théorème : Pour tout $k = 1, \dots, N$ et pour tout $x_k \in S_k$ on a

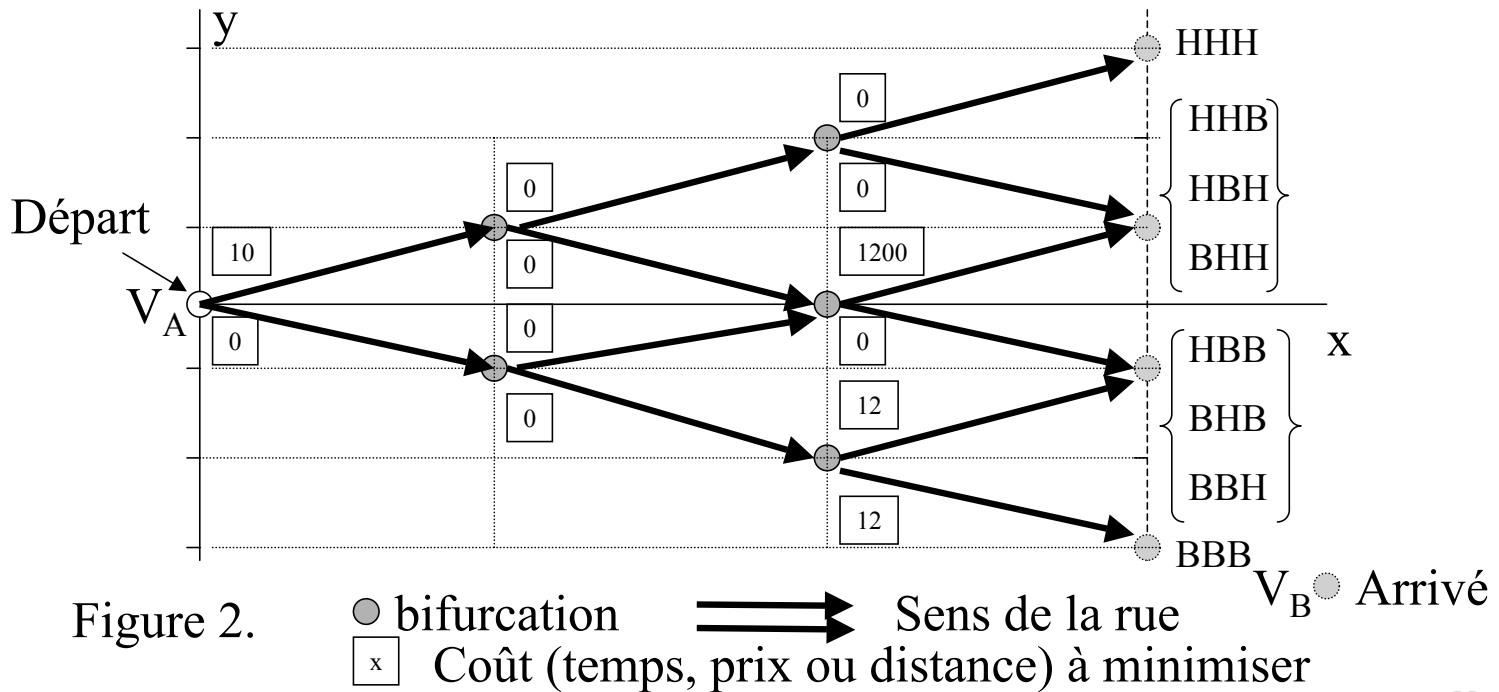
$$J^*_k(x_k) = J_k(x_k)$$

où $J_k(x_k)$ est obtenu en appliquant l'algorithme précédent. ■

Corollaire : L'algorithme de programmation dynamique permet de calculer une politique de décision optimale. ■

EXEMPLE 3:

Déplacement d'un voyageur en sens diagonal unique d'une ville V_A vers une autre ville V_B en coût minimal- cas d'un réseau triangulaire



SOLUTION DE L'EXEMPLE 3

Que constitue une solution?

Rappelons que dans le cas de la programmation dynamique stochastique, une politique et une séquence de décisions sont deux choses assez différentes puisque une séquence de décisions dicte les décisions futures d'une façon indépendante de la réalisation des décisions précédentes, tandis qu'une politique est dépendante de la réalisation.

SOLUTION DE L'EXEMPLE 3

A titre d'exemple, la solution (séquence) conseillée (peut être non optimale !) définie par la séquence : «Aller vers le Haut puis vers le Bas puis vers le Haut (HBH)» signifie que notre voyageur va essayer de se rappeler, à l'étape 2, d'aller vers le Bas (avec une probabilité $3/4$) quel que soit sa décision « H ou B » précédente par rapport à notre conseil « H ».

SOLUTION DE L'EXEMPLE 3

Si notre solution (politique) dit d'aller vers le haut « H » à l'étape 1. Ensuite, si vous êtes au (1,1), c'est-à-dire que vous êtes réellement allé vers le Haut « H » (en se rappelant de notre conseil) à la première étape, alors allez vers le bas « B ».

SOLUTION DE L'EXEMPLE 3

Mais si vous êtes au $(1,-1)$, c'est à dire que vous êtes réellement allé vers le bas « B » (en oubliant notre conseil) à la première étape, alors allez vers le haut « H ».

SOLUTION DE L'EXEMPLE 3

Ceci signifie que le voyageur doit essayer de se rappeler d'une façon différente en fonction de la situation courante.

SOLUTION DE L'EXEMPLE 3

Notre solution présentée comme une politique ou comme une séquence, dépend entièrement de ce que nous avons décidé de faire.

SOLUTION DE L'EXEMPLE 3

La séquence optimale est plus simple à exposer et elle ne nécessite pas que le voyageur observe où il est à chaque étape (c'est à dire observer l'état ou la coordonnée y). Cependant, la politique optimale produira toujours un coût moyen inférieur ou égal à celui de la séquence optimale puisqu'elle permet plus de flexibilité.

SOLUTION DE L'EXEMPLE 3

Nous développerons la solution spécifiée par une séquence de décisions optimales en boucle ouverte ou commande en boucle ouverte (open loop control) et la solution spécifiée par une politique de décisions optimales en boucle fermée ou commande en boucle fermée (closed loop control). La programmation dynamique par une politique de décisions optimales en boucle fermée sera la solution adoptée.

SOLUTION DE L'EXEMPLE 3

Solutions numériques

1. Séquence de décisions optimales en boucle ouverte

Pour déterminer la meilleur séquence de la commande en boucle ouverte, nous considérons tous les 8 séquences possibles, de 3 décisions chacune, et choisirons celle qui correspond à un coût minimal.

SOLUTION DE L'EXEMPLE 3

Solutions numériques (suite)

Le coût espéré si le chemin réel suivi à partir de $x=y=0$ est l :

$$V(0,0)_l = \sum_{i=1}^{N_c} [P_i \bullet C_i] = \sum_{i=1}^{N_c} \left[\left(\prod_{j=1, p_j \rightarrow a_j}^{N_d} p_j \right)_i \bullet \left(\sum_{j=1, c_j \rightarrow a_j}^{N_d} c_j \right)_i \right], 1 \leq l \leq N_c \quad (1a)$$

$$l = \underset{l}{\operatorname{arg\,min}} (V(0,0)_l) \quad (1b)$$

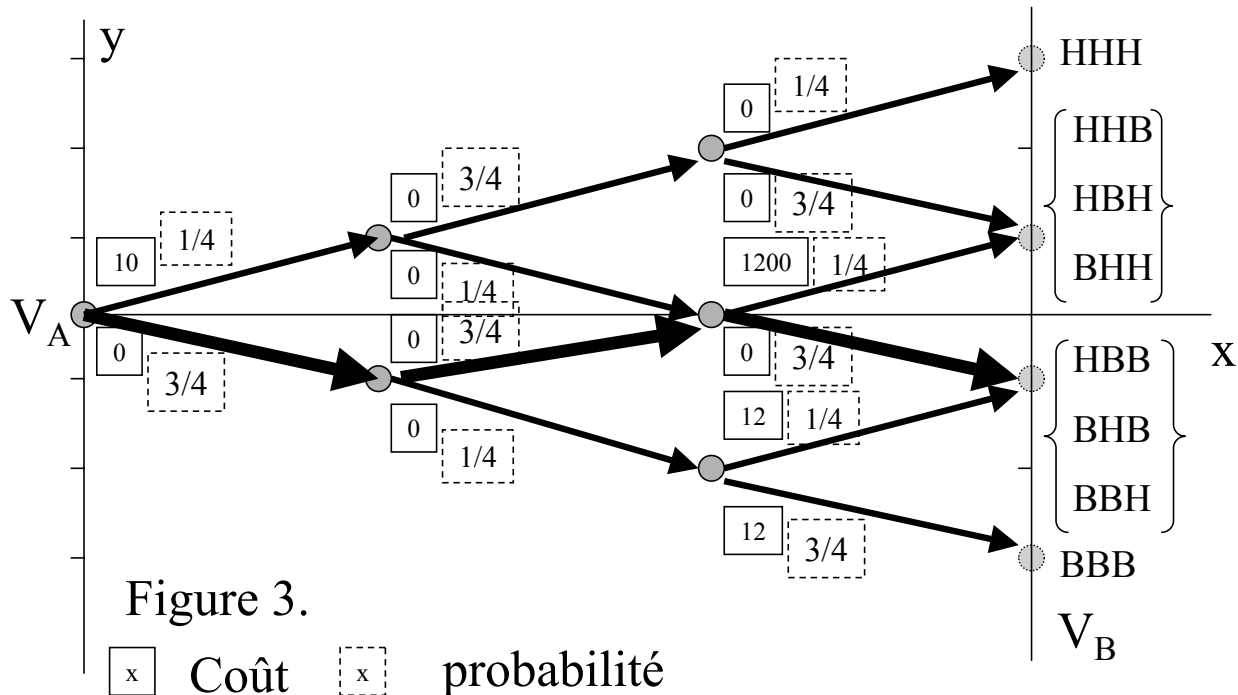
où c_j , p_j le coût et la probabilité respectivement correspondant à l'arc a_j ;

le nombre des étapes (décisions) possibles $N_d=3$,

le nombre maximal de chemins possibles $N_c = 2^{N_d} = 8$.

SOLUTION DE L'EXEMPLE 3

Par exemple, supposons que le chemin conseillé (naïvement d'après la somme des coûts immédiats !) à suivre est : vers le Bas puis vers le Haut et enfin vers le Bas.



SOLUTION DE L'EXEMPLE 3

Dans ce cas,

- la séquence de décision pour BHB possédant un coût minimal :
 $C_1=0+0+0=0$ et une probabilité : $P_1= 3/4 \times 3/4 \times 3/4=27/64$
- la séquence de décision pour HBH possédant un coût minimal :
 $C_1=10+0+1200=1210$ et une probabilité $P_1= 1/4 \times 1/4 \times 1/4=1/64$
- la séquence de décision pour HHB possédant un coût minimal :
 $C_1=10+0+0=10$ et une probabilité : $P_1= 1/4 \times 3/4 \times 3/4=9/64$
-

SOLUTION DE L'EXEMPLE 3









Chemin
conseillé



i	Chemin réels	P_i	C_i
1	BHB	27/64	0
2	BBB	9/64	12
3	BBH	3/64	12
4	HBB	3/64	10
5	BHH	9/64	1200
6	HBH	1/64	1210
7	HHB	9/64	10
8	HHH	3/64	10

$$V(0,0)_{BHB} = \frac{27}{64} \cdot 0 + \frac{9}{64} (10+12+1200) + \frac{3}{64} (12+10+10) + \frac{1}{64} 1210 = 192,25$$

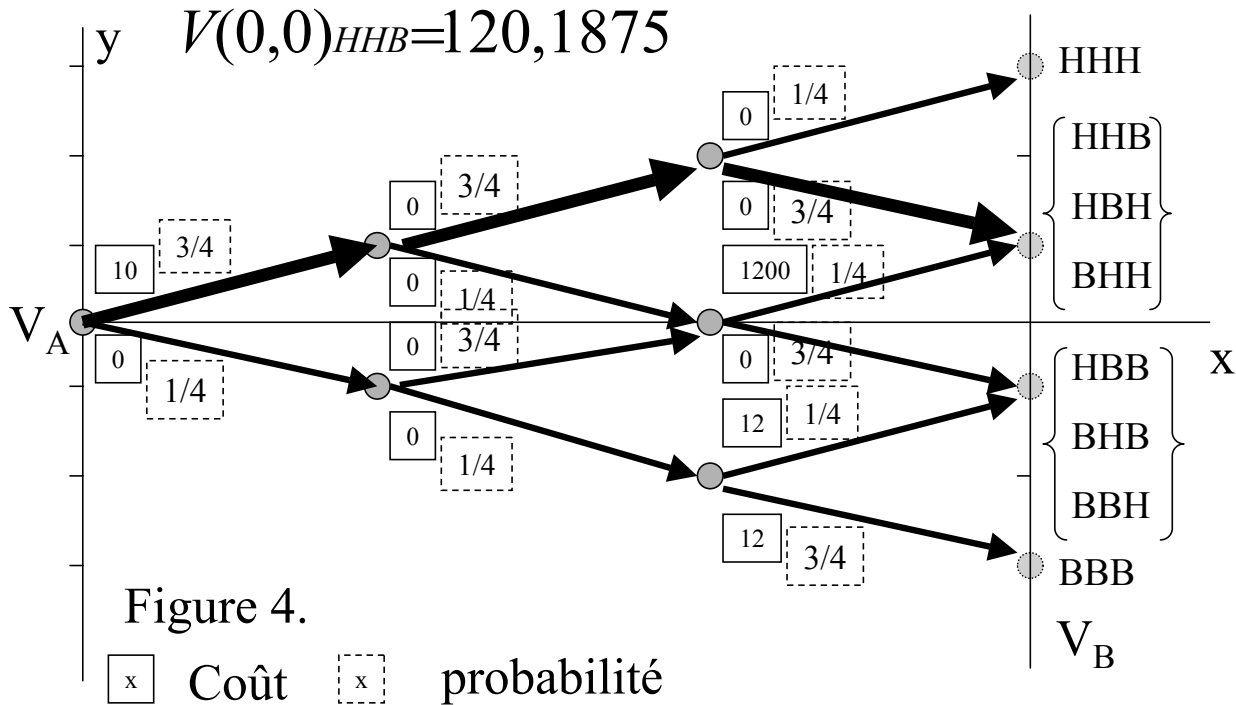
SOLUTION DE L'EXEMPLE 3

Chemin conseillé	i	Chemin réels	P_i	C_i
	1	BHB	P_{BHB}	C_{BHB}
	2	BBB	P_{BBB}	C_{BBB}
	3	BBH	P_{BBH}	C_{BBH}
	4	HBB	P_{HBB}	C_{HBB}
	5	BHH	P_{BHH}	C_{BHH}
	6	HBH	P_{HBH}	C_{HBH}
	7	HHB	P_{HHB}	C_{HHB}
	8	HHH	P_{HHH}	C_{HHH}

$V(0,0)_{BHB}$ $V(0,0)_{BBB}$ $V(0,0)_{BBH}$ $V(0,0)_{HBB}$ $V(0,0)_{BHH}$ $V(0,0)_{HBH}$ $V(0,0)_{HHB}$ $V(0,0)_{HHH}$

SOLUTION DE L'EXEMPLE 3

Il s'avère que la séquence de décisions HHB qui possède le coût espéré minimal (solution optimale) :



SOLUTION DE L'EXEMPLE 3

2. Politique de décisions optimales en boucle fermée

Cette politique peut être calculée récursivement comme suit :

1. Définir la fonction optimale de la valeur espérée (coût espéré) $J_k(x_k) = S(x,y)$ où l'étape k est la position x courante et l'état x_k est l'ordonné y de la bifurcation.
2. Définir les conditions limites.
3. Écrire la relation de récurrence appropriée.

SOLUTION DE L'EXEMPLE 3

- Si nous choisissons la décision « H » avec une probabilité 3/4 nous ferons une transition vers $(x+1,y+1)$, avec le coût d'une étape de $c_H(x,y)$ et le coût espéré restant $J_{k+1}(x_{k+1}) = S(x+1,y+1)$.
- Avec une probabilité 1/4 nous ferons une transition vers $(x+1,y-1)$, avec le coût d'une étape de $c_B(x,y)$ et le coût espéré restant $J_{k+1}(x_{k+1}) = S(x+1,y-1)$.
- Si nous choisissons la décision « B » avec une probabilité 3/4 nous ferons une transition vers $(x+1,y-1)$, avec le coût d'une étape de $c_B(x,y)$ et le coût espéré restant $J_{k+1}(x_{k+1}) = S(x+1,y-1)$.
- Avec une probabilité 1/4 nous ferons une transition vers $(x+1,y+1)$, avec le coût d'une étape de $c_H(x,y)$ et le coût espéré restant $J_{k+1}(x_{k+1}) = S(x+1,y+1)$.

SOLUTION DE L'EXEMPLE 3

Par une version stochastique du principe d'optimalité, nous avons un problème de programmation dynamique en arrière (Backward Dynamic Programming) :

$$S(x, y) = \min \left\{ \begin{array}{l} H : \frac{3}{4}(c_H(x, y) + S(x+1, y+1)) \\ \quad + \frac{1}{4}(c_B(x, y) + S(x+1, y-1)) \\ B : \frac{1}{4}(c_H(x, y) + S(x+1, y+1)) \\ \quad + \frac{3}{4}(c_B(x, y) + S(x+1, y-1)) \end{array} \right\} \quad (2)$$

Avec les conditions limites :

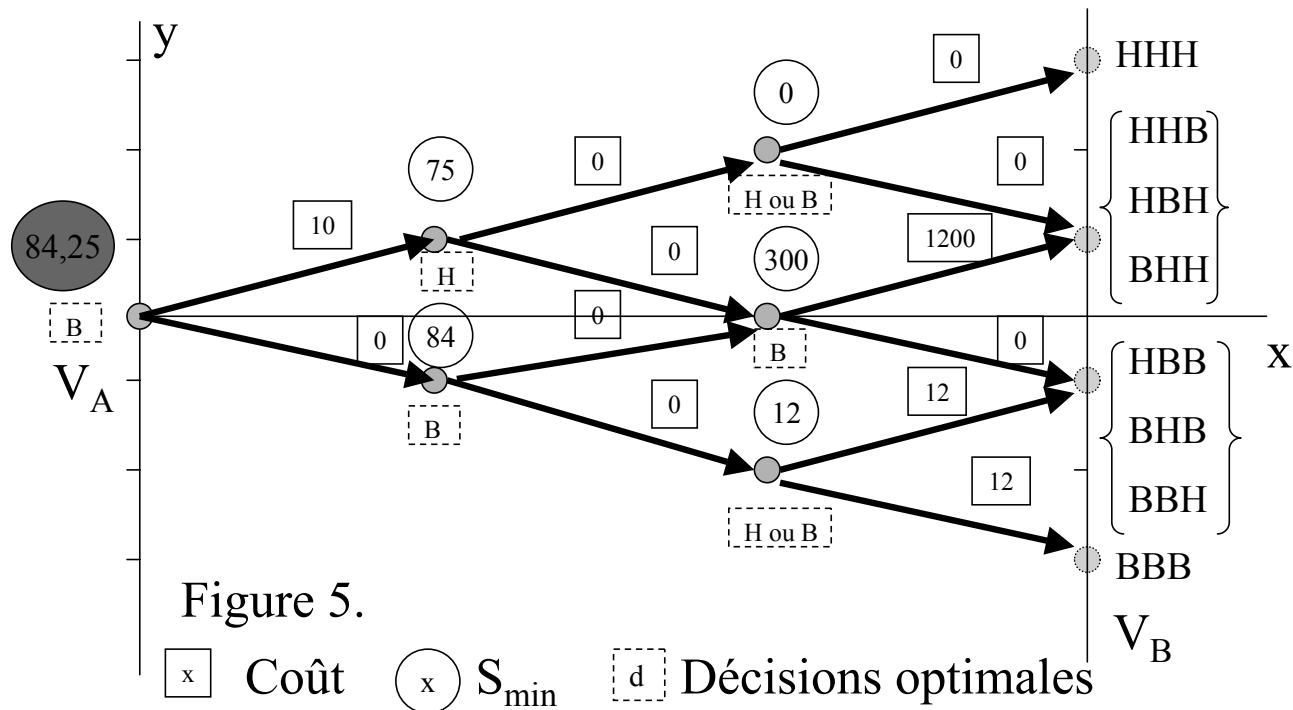
$$S(3,3) = 0, S(3,1) = 0, S(3,-1) = 0, S(3,-3) = 0 \quad (3)$$

SOLUTION DE L'EXEMPLE 3

La politique optimale en boucle fermée est l'ensemble de décisions, une par étape et par état, qui minimise le coût espéré $V(0,0)$ défini par l'équation 1b en commençant à $(0,0)$. Il est possible de démontrer, [Dre77], que $S(0,0)$ est le même que $V(0,0)$.

SOLUTION DE L'EXEMPLE 3

Le coût espéré en utilisant la politique de la commande optimale en boucle fermée = 84,25



SOLUTION DE L'EXEMPLE 3

Coût du calcul

Chaque application de (2) et (3) nécessite 4 additions, 4 multiplications, et une comparaison. Il y a $N+(N-1)+\dots+1$ nœuds pour un problème à N étapes. Ainsi $2N(N+1)$ additions, $2N(N+1)$ multiplications, et $N(N+1)/2$ comparaisons sont nécessaires pour la solution. En considérant chacune comme une opération, alors environ $9N^2/2$ opérations sont nécessaires.

Exercice : Comparer ce nombre au nombre approximatif nécessaire dans le cas déterministe.

SOLUTION DE L'EXEMPLE 3

Maximiser la probabilité que le coût est inférieur ou égal à une valeur donnée Z

Un critère alternatif, parmi d'autres, que nous pourrions adopter serait de maximiser la probabilité que le chemin emprunté par notre voyageur coûte une valeur inférieure à un certain nombre donné Z .

SOLUTION DE L'EXEMPLE 3

Considérons le cas général d'un problème de N_d étapes sur un réseau triangulaire.

Soit $c_H(x,y)$ et p_H le coût et la probabilité de la décision H respectivement et $c_B(x,y)$ et p_B la probabilité de la décision B respectivement.

L'objectif maintenant est d'utiliser la programmation dynamique pour déterminer la politique en boucle fermée qui maximise la probabilité que le coût soit inférieur ou égal à une valeur donné Z .

SOLUTION DE L'EXEMPLE 3

Définir $S(x,y,z)$ = la probabilité maximale que le coût total est inférieur ou égal à Z sachant que nous partons du nœud (x,y) avec le coût de A au nœud (x,y) étant égal à une valeur donnée z .

$$S(x, y, z) = \max \left\{ \begin{array}{l} H : p_H S(x+1, y+1, z + c_H(x, y)) + (1 - p_H) S(x+1, y-1, z + c_B(x, y)) \\ B : (1 - p_B) S(x+1, y+1, z + c_H(x, y)) + p_B S(x+1, y-1, z + c_B(x, y)) \end{array} \right\} \quad (4)$$

Avec les conditions limites :

$$S(N_d, y, z) = \begin{cases} 1, & z = 0, 1, 2, \dots, Z \\ 0, & \text{autrement} \end{cases} \quad (5)$$

La politique optimale à boucle fermée qui maximise la probabilité que le coût est inférieur ou égale à une valeur donnée z est donnée par $S(0,0,0)$.

CONCLUSION

1. La technique de la séquence de décisions optimales en boucle ouverte qui ne nécessite pas l'utilisation de l'information a posteriori sur la transition réelle produit généralement un coût espéré élevé, mais c'est la meilleure technique employée sans l'utilisation des informations supplémentaires.

2. La technique de la politique de décisions optimales en boucle fermée, où l'état est supposé connu quand la décision est prise, produit un coût espérée le plus petit puisqu'elle utilise toutes les informations disponibles.

PROBLEMES A TEMPS D'ARRET STOCHASTIQUE (Stochastic Stopping-Time Problem)

Dans beaucoup de problèmes de la vie et de la technique, la décision et la durée du processus sont de nature stochastique. Ainsi, dans ce paragraphe nous introduirons le problème d'un chemin à coût espéré minimal avec un temps d'arrêt stochastique (minimum-expected-cost path problem with uncertain stopping time)

EXEMPLE 4:

Processus à temps d'arrêt

stochastique cas d'un réseau triangulaire

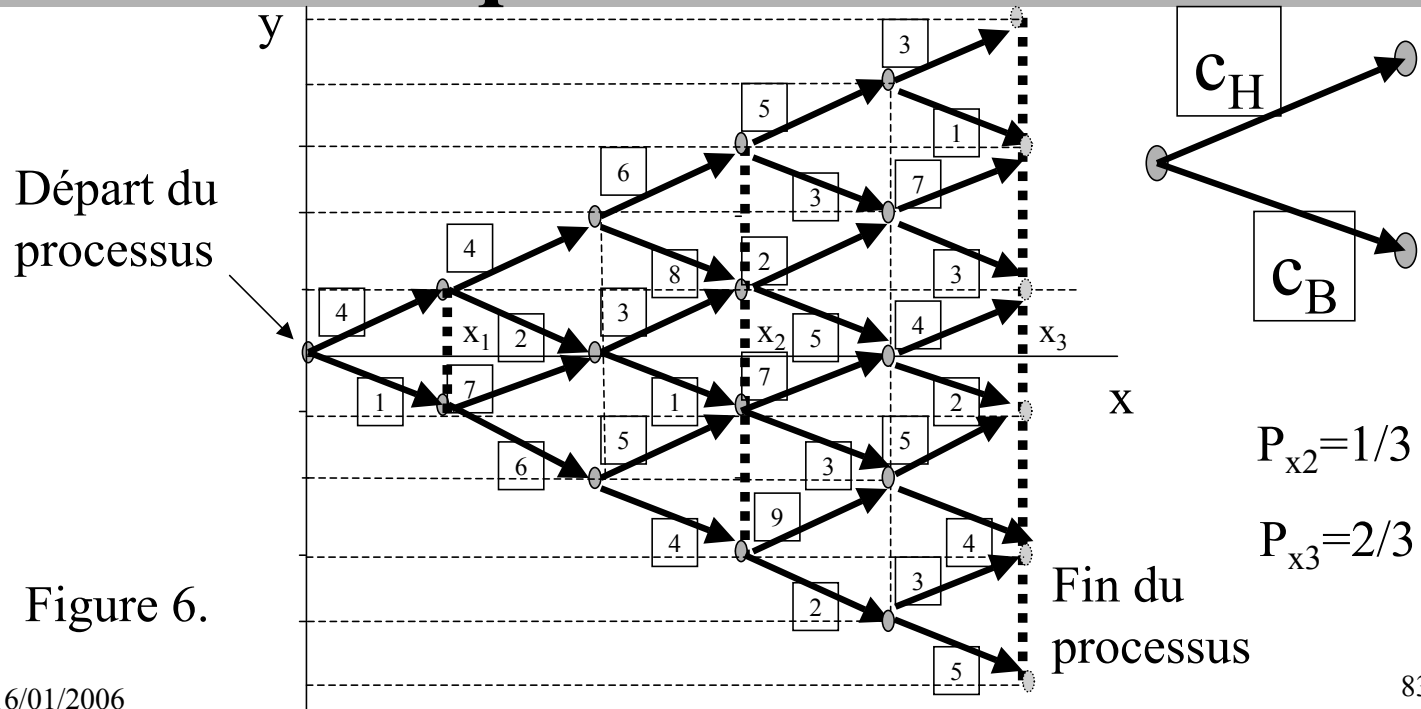


Figure 6.

Malgré que les réalisations des décisions peuvent être considérées de nature stochastique, néanmoins dans cet exemple nous supposons que les décisions H et B amènent à un déplacement diagonal certain vers le Haut et vers le Bas, faisant encourir un coût $c_H(x,y)$ si Haut et $c_B(x,y)$ si Bas.

Ce qui est nouveau dans cet exemple est que, quand le processus commence nous ne savons pas s'il termine à la ligne $x=x_2$ ou bien à la ligne $x=x_3$.

Cependant, ce que nous savons est que la probabilité de $x=x_2$ est p_2 et de $x=x_3$ est $p_3(1-p_2)$, et de plus nous supposons que quand nous atteignons la ligne $x=x_1$ et avant de prendre notre décision là, quelqu'un nous dit à quelle ligne le problème sera terminé.

Notre rôle comme conseiller à une certaine étape durant ce processus est de demander d'abord à notre voyageur les coordonnées x et y du nœud courant. Alors si l'étape est entre x_1 et x_2 , où x_1 et x_2 sont inclus, nous demanderions également quel est l'état de la ligne terminale (x_2 ou x_3) avait été donné à l'étape x_1 .

Si $x < x_1$, aucune information n'a encore été donnée, et si $x > x_2$, x_3 est évidemment la ligne terminale. Ainsi, en fonction de l'étape, nous avons différentes définitions de la fonction de la valeur optimale.

Pour $x < x_1$, ou $x \leq x_1 - 1$,

$F(x,y)$ = le coût minimal espéré du reste du processus en partant de (x,y) ; (6)

Pour $x_1 \leq x \leq x_2$,

$G(x,y,L)$ = le coût minimal espéré du reste du processus en partant de (x,y) et terminant à L où $L = x_2$ ou $L = x_3$; (7)

Pour $x \geq x_2$,

$H(x,y)$ = le coût minimal espéré du reste du processus en partant de (x,y) et terminant à $x = x_3$. (8)

Remarquons que c'est uniquement la première fonction F qui est le coût espéré puisque la durée est connue avec une certitude une fois les arguments, incluant L , sont donnés pour les fonctions des valeurs optimales G et H .

La relation de récurrence dépend de x et elle est pour $x \leq x_1 - 2$,

$$F(x, y) = \min \left\{ \begin{array}{l} H : c_H(x, y) + F(x+1, y+1) \\ B : c_B(x, y) + F(x+1, y-1) \end{array} \right\} \quad (9)$$

Avec les conditions limites :

$$F(x_1 - 1, y) = \min \left\{ \begin{array}{l} H : c_H(x_1 - 1, y) + p_2 G(x_1, y+1, x_2) + p_3 G(x_1, y+1, x_3) \\ B : c_B(x_1 - 1, y) + p_2 G(x_1, y-1, x_2) + p_3 G(x_1, y-1, x_3) \end{array} \right\} \quad (10)$$

Pour le cas particulier de la Figure 6 où $x_1=1$, l'équation 9 n'est jamais utilisée. Seule les conditions limites de l'équation 10 sont utilisées.

La relation pour $x_1 \leq x \leq x_2 - 1$,

$$G(x, y, L) = \min \left\{ \begin{array}{l} H : c_H(x, y) + G(x+1, y+1, L) \\ B : c_B(x, y) + G(x+1, y-1, L) \end{array} \right\} \quad (11)$$

Avec les conditions limites :

$$G(x_2, y, x_2) = 0 \quad \forall y, \quad G(x_2, y, x_3) = H(x_2, y) \quad (12)$$

La relation pour $x \geq x_2$,

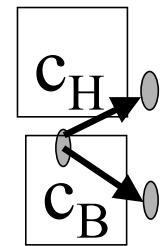
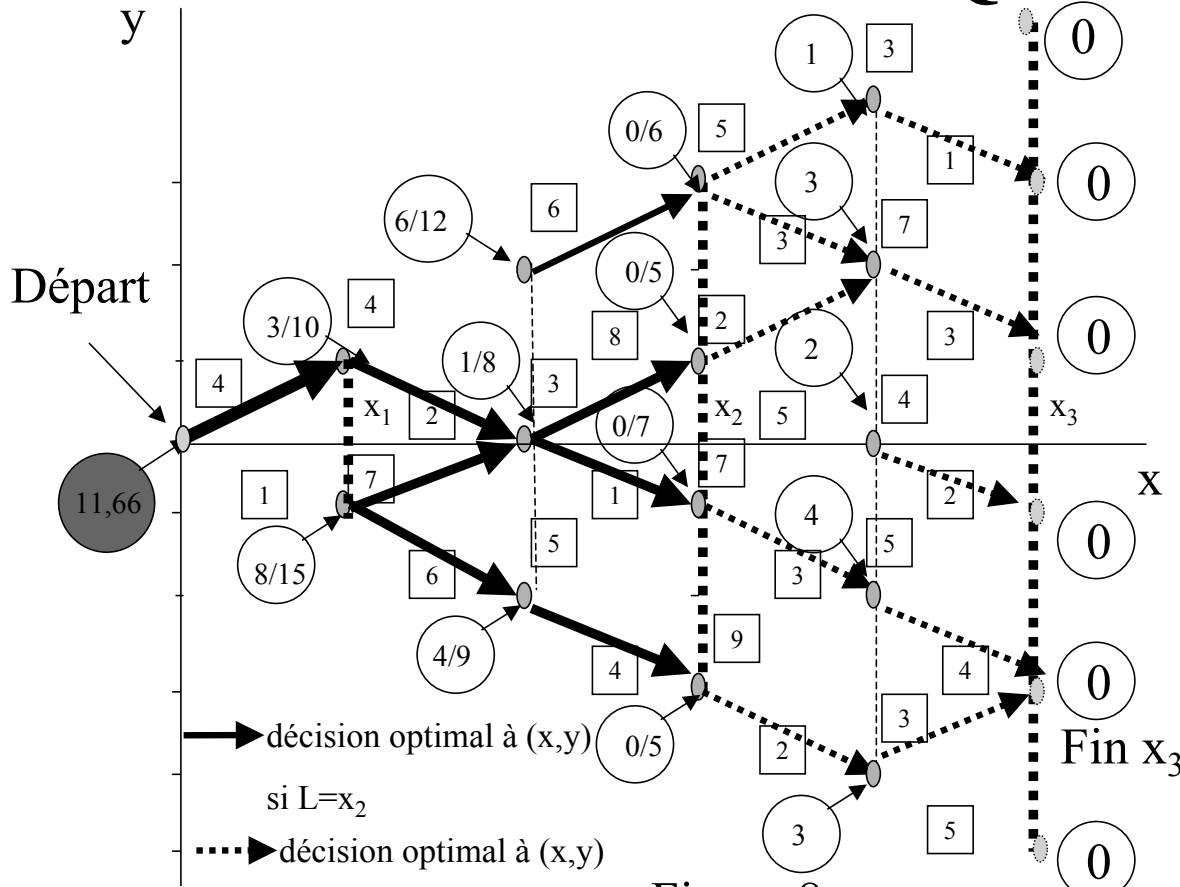
$$H(x, y) = \min \left\{ \begin{array}{l} H : c_H(x, y) + H(x+1, y+1) \\ B : c_B(x, y) + H(x+1, y-1) \end{array} \right\} \quad (13)$$

Avec les conditions limites :

$$H(x, y) = 0 \quad \forall y, \quad (14)$$

Le calcul s'effectue vers l'arrière (backward) à partir de x_3 , d'abord H, ensuite G (qui nous donne des coût différents et des décisions différentes possibles à chaque nœud de chacune des situations terminales), et finalement F.

SOLUTION NUMERIQUE



$$\frac{G(x,y,x_2)}{H(x,y)} \quad x=x_2$$

$$\frac{G(x,y,x_2)}{G(x,y,x_3)} \quad x < x_2$$

$$H(x,y) \quad x > x_2$$

→ décision optimal à (x,y)
 si $L=x_2$
 - - - → décision optimal à (x,y)
 si $L > x_2$

Figure 8.

Tarik AL ANI, A2SI-ESIEE – Paris

PROBLEMES AVEC UN RETARD TEMPOREL (Problems with Time-Lag or Delay)

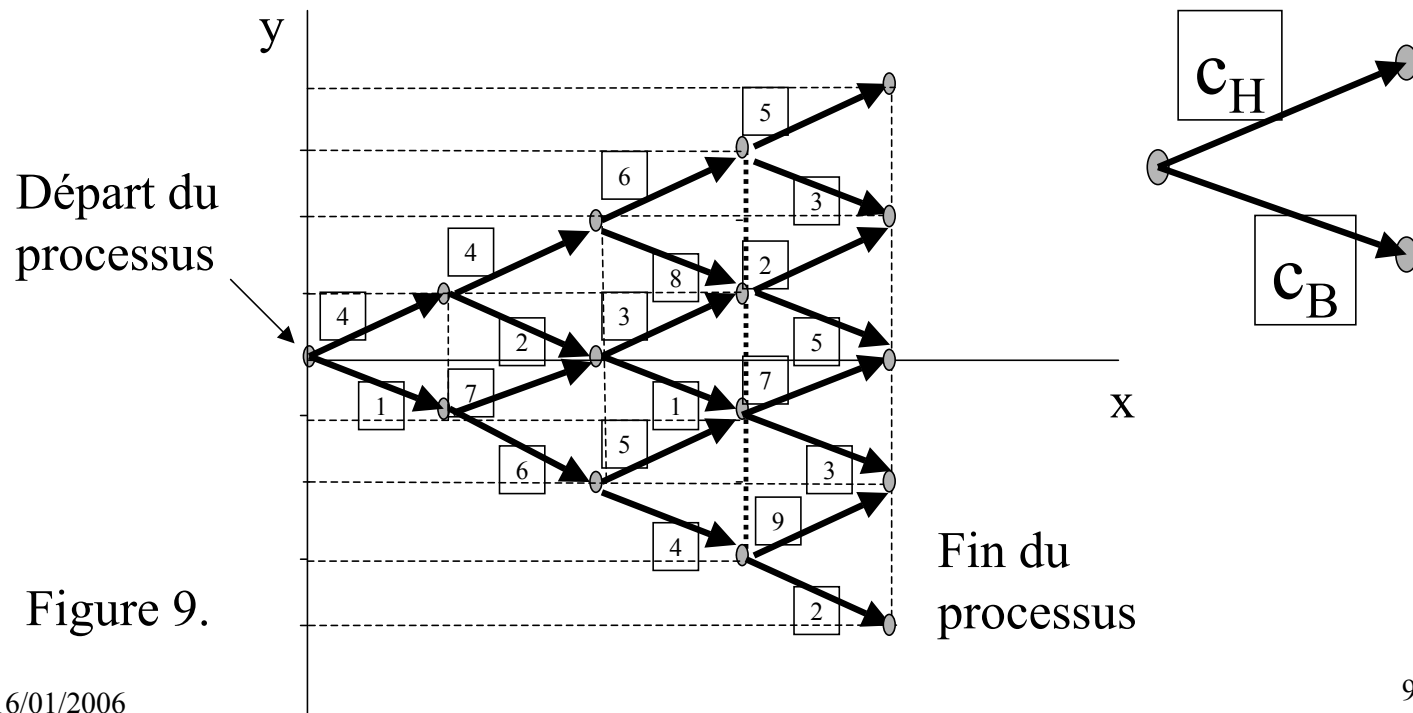
Pour beaucoup de problèmes, il existe un retard temporel entre l'instant de la prise de décision et l'instant de son exécution. Pour les problèmes de planification de la production, l'exécution d'une décision pour augmenter le niveau de la production nécessite de telles décisions à retard comme l'embauche et l'apprentissage d'un stagiaire.

Pour les problèmes d'inventaire, une commande et son arrivée sont séparées par un temps de transmission et de traitement de cette commande et par le temps de livraison et éventuellement par le temps de la fabrication.

Dans le cas déterministe , un tel retard ne présente pas de difficulté parce que si une décision d est optimale à l'étape K de la solution en boucle ouverte et si cette décision devient effective (exécutée) après Δ étapes, nous pourrions simplement décaler nos décisions de Δ étapes et prendre la décision d à l'étape $K-\Delta$.

Cependant, pour des processus stochastiques en boucle fermée, nous ne saurions pas quel serait l'état à l'étape K , ainsi ne nous ne pourrions pas déplacer une décision de Δ étapes.

EXEMPLE 5: PLANIFICATION



16/01/2006

Dans cet exemple, notre décision à l'étape K sera réalisé à l'étape $K+2$, quelque soit l'état à l'étape $K+2$. Nous employons une politique en boucle fermée puisque notre décision à l'étape K dépendra du nœud à l'étape K , avec lequel il y'a d'autres informations associées.

Décision H produit un déplacement en diagonale vers le Haut quand elle est exécutée deux étapes plus tard avec une probabilité $\frac{3}{4}$ et elle produit un déplacement en diagonale vers le Bas quand elle est prise deux étapes plus tard avec une probabilité $\frac{1}{4}$. Idem pour la décision B sauf que les probabilités sont inversées.

Puisque la décision à $(0,0)$ n'est pas exécutée jusqu'à l'étape 2 ($x=2$), nous devons de plus spécifier ce qui se passe aux étapes 0 et 1. Nous supposons qu'à $(0,0)$, $(1,1)$, et $(1,-1)$ la probabilité de se déplacer en diagonale vers le haut et la probabilité de se déplacer en diagonale vers le bas sont égales à 0,5 chacune.

Les décisions à $(0,0)$, $(1,1)$ et $(1,-1)$ sont exécutées deux étapes après , à cause du retard $\Delta=2$, à l'étape 2 et à l'étape 3. Aucune décision n'est prise pour les étapes 2 et 3 puisque le problème se termine avant leurs exécutions.

Notre rôle comme conseiller à une certaine étape durant ce processus est de demander d'abord les coordonnées x et y du nœud courant. De plus, nous pourrions demander la décision qui a été déjà prise à l'étape $x-2$ puisque cette décision sera exécutée au nœud courant et elle influencera les probabilités aux nœud deux étapes après et ainsi notre décision.

De la même façon, nous pourrions demander aussi de connaître la décision qui a été prise à l'étape précédente ($x-1$) puisque cette décision sera exécutée à l'étape suivante ($x+1$).

Ceci nous amène à la définition suivante de la fonction de la valeur optimale espérée pour la situation générale où notre demandeur a déjà pris les deux décisions précédentes et nous sommes amenés à prendre désormais des décisions (cette situation ne se produit jamais dans notre exemple particulier de quatre étapes uniquement).

$S(x,y,d_1,d_2)$ = le coût espéré de la suite du processus si nous commençons au nœud (x,y) , d_1 est la décision prise à l'étape $x-1$, et d_2 est la décision prise à l'étape $x-2$. (15)

Par le principe d'optimalité, nous pouvons écrire séparément chacun des quatre ensembles possibles des deux décisions précédentes.

$$S(x, y, H, H) = \frac{3}{4}c_H(x, y) + \frac{1}{4}c_B(x, y) + \min_{\substack{\dot{y} \\ \ddot{y} \\ \ddot{b}}} \begin{cases} H : \frac{3}{4}S(x+1, y+1, H, H) + \frac{1}{4}S(x+1, y-1, H, H) \\ B : \frac{3}{4}S(x+1, y+1, B, H) + \frac{1}{4}S(x+1, y-1, B, H) \end{cases} \quad (16a)$$

$$S(x, y, H, B) = \frac{1}{4}c_H(x, y) + \frac{3}{4}c_B(x, y) + \min_{\substack{\dot{y} \\ \ddot{y} \\ \ddot{b}}} \begin{cases} H : \frac{1}{4}S(x+1, y+1, H, H) + \frac{3}{4}S(x+1, y-1, H, H) \\ B : \frac{1}{4}S(x+1, y+1, B, H) + \frac{3}{4}S(x+1, y-1, B, H) \end{cases} \quad (16b)$$

$$S(x, y, B, H) = \frac{3}{4}c_H(x, y) + \frac{1}{4}c_B(x, y) + \min_{\substack{\dot{y} \\ \ddot{y} \\ \ddot{b}}} \begin{cases} H : \frac{3}{4}S(x+1, y+1, H, B) + \frac{1}{4}S(x+1, y-1, H, B) \\ B : \frac{3}{4}S(x+1, y+1, B, B) + \frac{1}{4}S(x+1, y-1, B, B) \end{cases} \quad (16c)$$

$$S(x, y, B, B) = \frac{1}{4}c_H(x, y) + \frac{3}{4}c_B(x, y) + \min_{\substack{\dot{y} \\ \ddot{y} \\ \ddot{b}}} \begin{cases} H : \frac{1}{4}S(x+1, y+1, H, B) + \frac{3}{4}S(x+1, y-1, H, B) \\ B : \frac{1}{4}S(x+1, y+1, B, B) + \frac{3}{4}S(x+1, y-1, B, B) \end{cases} \quad (16d)$$

Avec les conditions limites ($x = x_T$)

$$S(x_T, y, d_1, d_2) = 0, \quad \forall y, d_1, d_2$$

Remarquons que le 3ème et le 4ème termes de $S(x+1, y+1, \dots)$ et de $S(x+1, y-1, \dots)$, le coût espéré de la transition suivante, dépendent du quatrième argument de S , la décision prise deux étapes avant, et que la décision à l'étape x (H pour la première ligne et B pour la deuxième ligne après min) apparaît comme le 3ème argument de S à l'étape $x+1$, tandis que le 3ème argument de S à l'étape x devient le 4ème argument de S à l'étape $x+1$.

Pour des processus commençants à l'étape 0 ou 1, nous avons supposés que $P_H = P_B = 0,5$, ainsi

$$S(1, y, H, -) = \frac{1}{2}c_H(1, y) + \frac{1}{2}c_B(1, y) + \min \left\{ \begin{array}{l} H : \frac{1}{2}S(2, y+1, H, H) + \frac{1}{2}S(2, y-1, H, H) \\ B : \frac{1}{2}S(2, y+1, B, H) + \frac{1}{2}S(2, y-1, B, H) \end{array} \right\} \quad (17a)$$

$$S(1, y, B, -) = \frac{1}{2}c_H(1, y) + \frac{1}{2}c_B(1, y) + \min \left\{ \begin{array}{l} H : \frac{1}{2}S(2, y+1, H, B) + \frac{1}{2}S(2, y-1, H, B) \\ B : \frac{1}{2}S(2, y+1, B, B) + \frac{1}{2}S(2, y-1, B, B) \end{array} \right\} \quad (17b)$$

$$S(0, 0, -, -) = \frac{1}{2}c_H(0, 0) + \frac{1}{2}c_B(0, 0) + \min \left\{ \begin{array}{l} H : \frac{1}{2}S(1, 1, H, B) + \frac{1}{2}S(1, -1, H, -) \\ B : \frac{1}{2}S(1, 1, B, B) + \frac{1}{2}S(1, -1, B, -) \end{array} \right\} \quad (17c)$$

Dans le cas de cet exemple, les conditions limites à $x_T=4$ sont

$$S(4, y, d_1, d_2) = 0, \forall y, d_1, d_2 \quad (18)$$

Dans ce cas, la valeur de S sera calculée à l'étape 3 pour les différentes décisions à l'étape 2 où ces décisions sont inutiles et ne sont pas réellement prises, mais la réponse sera correcte.

SOLUTION NUMERIQUE

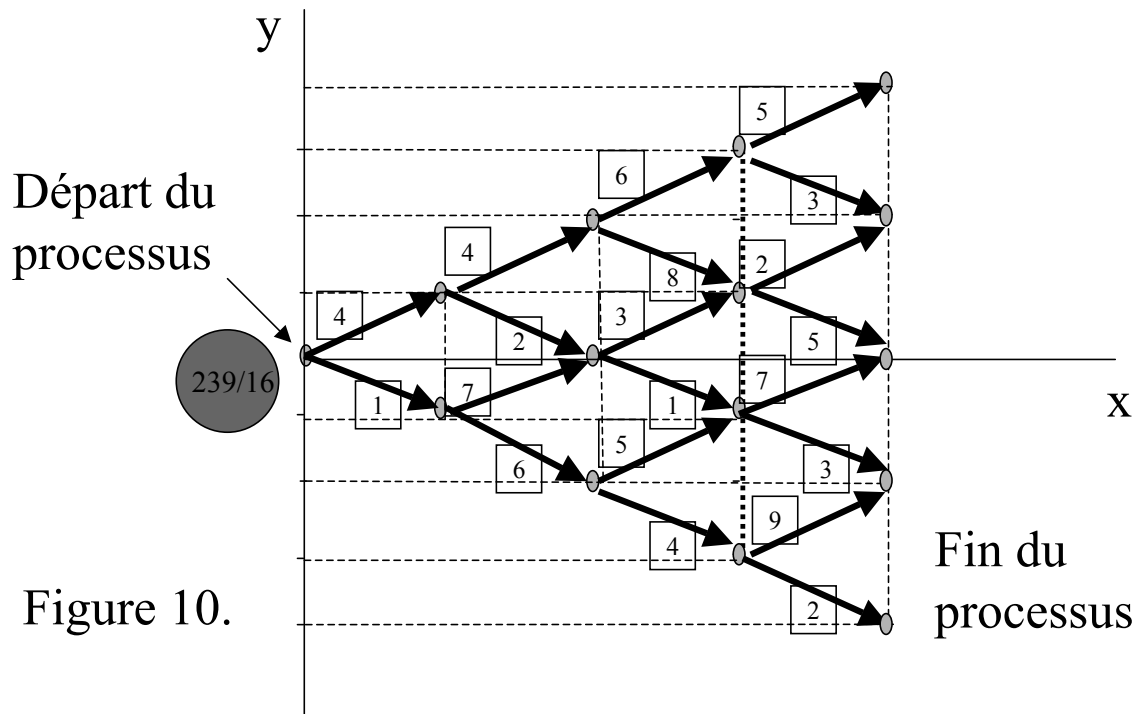
Par l'application des équations 16, 17 et 18 nous obtenons

$$\begin{array}{ll}
 S(2,2,H,H)=169/16, & S(2,0,H,H)=97/16, \\
 S(2,2,B,H)=163/16, & S(2,0,B,H)=107/16, \\
 S(2,2,H,B)=171/16, & S(2,0,H,B)=107/16, \\
 S(2,2,B,B)=185/16, & S(2,0,B,B)=89/16,
 \end{array}$$

Décision optimale

$$\begin{array}{lll}
 S(2,-2,H,H)=177/16, & S(1,1,H,_) = 181/16, & (H) \\
 S(2,-2,B,H)=139/16, & S(1,1,B,_) = 185/16, & (B) \\
 S(2,-2,H,B)=179/16, & S(1,-1,H,_) = 227/16, & (B) \\
 S(2,-2,B,B)=129/16, & S(1,-1,B,_) = 213/16, & (B) \\
 S(0,0,_,_) = 239/16, & & (B)
 \end{array}$$

La décision optimale à (0,0) est B, et quelque soit la première transition vers le haut ou vers le bas, la deuxième décision optimale est B.



NOTATIONS

$x_{k+1} = f_k(x_k, u_k, \omega_k);$

k

x_k

$u_k = \mu_k(x_k)$

ω_k

N

f_k

$g_k(x_k, u_k, \omega_k);$

$g_N(x_N)$

$$J_{\pi}(x_0) = E_{\omega_0, \dots, \omega_{N-1}} \left[\sum_{k=0}^{N-1} g_k(x_k, u_k, \omega_k) + g_N(x_N) \right]$$

$A = [a_{ij}], i, j \in S_k$

$U_k(x_k)$

Ω_k

$P_k : \Omega_k \rightarrow IR$

$\mu_k : S_k \rightarrow D_k$

système dynamique

indice de l'étape discrète $k = 0, \dots, N-1$

état (discret dans notre cours) du système

fonction de décision

perturbation

horizon du processus (nombre de décisions)

fonction de transfert de l'étape k

coût encouru à chaque étape k

coût terminal

coût espéré

matrice de transition d'états

décisions admissibles à l'étape k

L'ensemble des valeurs que peuvent prendre les perturbations ω_k à l'étape k

loi de probabilités de ω_k

fonction de décision admissible

NOTATIONS

D_k	ensemble de fonctions de décision admissibles à l'étape k
$\pi=(\mu_0, \dots, \mu_{N-1})$	politique (Une politique π est admissible si chacune des fonctions de décision la formant l'est.)
π^*	politique optimale
Π	ensemble des politiques admissibles
$P_k(x_k)$	problème résiduel (ou partiel ou Sous-problème ou Sous-solution) débutant dans l'état x_k au début de l'étape k et se
$\pi(k)=(\mu_k, \dots, \mu_{N-1})$	politiques résiduelles ou partielles associées à $P_k(x_k)$
$J^*_k(x_k)$	valeur minimale de l'espérance des coût totaux pour le problème de décision $P_k(x_k)$
$J^*\pi^*(x_0)$	coût espéré optimal

Bibliographie

- [Bat00] Bather J. Decision Theory. An Introduction to Dynamic Programming and Sequential Decisions. John Wiley & Sons, 2000.
- [Bel57] Bellman. R. Dynamic Programming. Princeton University Press, Princeton, N.J. 1992.
- [Dre77] Dreyfus S.E. and Law A. M. The Art and Theory of Dynamic Programming, Academic Press 1977.
- [Hêc05] J-F Hêche, Rosso-EPFL, Recherche opérationnelle.