
Programmation dynamique : définitions, notations et algorithme

Processus de décisions séquentielles

Un processus de décisions séquentielles est formé

- d'un système dynamique à temps discret évoluant pendant un nombre fini de périodes,
- d'une fonction coût additive au fil des périodes.

Système dynamique à temps discret

L'évolution de l'état x_k du système est régie par le processus

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

Le système débute son évolution dans l'état initial x_1 et s'arrête après N périodes ou étapes dans l'état final x_{N+1} .

VARIABLES

Symbole	Définition	Domaine de définition
x_k	état du système au début de l'étape k (avant la prise de décision)	$\in S_k, k = 1, \dots, N + 1$
u_k	décision prise pendant l'étape k	$\in C_k, k = 1, \dots, N$
ω_k	perturbation aléatoire de l'étape k (non connue lors de la prise de décision)	$\in \Omega_k, k = 1, \dots, N$

Les perturbations ω_k sont supposées indépendantes les unes des autres!

DOMAINES DE DÉFINITION

Symbole	Définition
S_k	ensemble des états possibles au début de l'étape $k, k = 1, \dots, N + 1$
C_k	ensemble de toutes les décisions possibles pendant l'étape $k, k = 1, \dots, N$
$U_k(x_k)$	ensemble des décisions admissibles dans l'état x_k au début de l'étape $k, k = 1, \dots, N, \emptyset \neq U_k(x_k) \subseteq C_k$
Ω_k	ensemble des valeurs possibles pour les perturbations aléatoires de l'étape $k, k = 1, \dots, N$

FONCTIONS

Symbole	Définition
$f_k(x_k, u_k, \omega_k)$	fonction de transfert de l'étape $k, k = 1, \dots, N$
$H_k(\omega_k)$	loi de probabilités des perturbations aléatoires ω_k de l'étape $k, k = 1, \dots, N$, peut dépendre de l'état x_k et/ou de la décision u_k

Fonction coût

Symbole	Définition	
$g_k(x_k, u_k, \omega_k)$	coût de l'étape k , $k = 1, \dots, N$	$g_k : S_k \times C_k \times \Omega_k \rightarrow \mathbb{R}$
$g_{N+1}(x_{N+1})$	coût terminal	$g_{N+1} : S_{N+1} \rightarrow \mathbb{R}$

Fonctions de décision et politiques

Symbole	Définition	
$\mu_k(x_k)$	fonction de décision de l'étape k , $k = 1, \dots, N$	$\mu_k : S_k \rightarrow C_k$
π	politique de décision	$\pi = \{\mu_1, \dots, \mu_N\}$
$\pi^{(k)}$	politique résiduelle ou partielle	$\pi^{(k)} = \{\mu_k, \dots, \mu_N\}$

La fonction μ_k est admissible si $\mu_k(x_k) \in U_k(x_k)$, $\forall x_k \in S_k$.

La politique π est admissible si chacune des fonctions μ_k la formant l'est. L'ensemble de toutes les politiques admissibles est noté Π .

La performance de la politique π pour l'état initial x_1 est

$$J_\pi(x_1) = E_{\{\omega_1, \dots, \omega_N\}} \left[\sum_{k=1}^N g_k(x_k, \mu_k(x_k), \omega_k) + g_{N+1}(x_{N+1}) \right]$$

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

La politique π^* est optimale si, quel que soit l'état initial $x_1 \in S_1$, on a

$$J_{\pi^*}(x_1) = \min_{\pi \in \Pi} J_\pi(x_1).$$

Algorithme

DONNÉES : Un processus de décisions séquentielles.

RÉSULTAT : Les tables contenant la politique optimale $\pi^* = \{\mu_1^*, \dots, \mu_N^*\}$ et les espérances minimales J_k .

(1) INITIALISATION

$$J_{N+1}(x_{N+1}) = g_{N+1}(x_{N+1}) \quad \forall x_{N+1} \in S_{N+1}.$$

(2) CONSTRUCTION DES TABLES OPTIMALES

Pour $k = N$ jusqu'à 1 faire

Pour tout $x_k \in S_k$ 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 $\mu_k^*(x_k) = u_k^*$ pour lesquels les minima sont atteints.