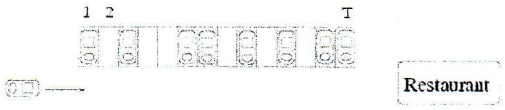
Partiel de Programmation Dynamique

Mercredi 7 décembre 2005

Aucun document autorisé

Exercice 1 *Le restaurant*
Ce soir, c'est le grand jeu, j'emmène ma copine au restaurant. Manque de chance, il pleut des cordes, et on a pas de parapluie. Je vais donc essayer de me garer le plus près possible de la porte du restaurant.
Les T places de parking sont disposées selon le schéma suivant :



Chaque place t est libre avec une probabilité p_t. Vu ce qu'il tombe, je ne peux voir si une place est libre que lorsque je suis devant, et je décide alors de me garer ou de continuer. Je ne peux jamais faire marche arrière. Bien sûr, plus je me rapproche du restaurant, moins ma copine sera trempée, et la place $1 \leq t \leq T$ m'apporte donc un profit (ou une récompense) t. Mais attention, si j'arrive devant le resto sans avoir réussi à me garer, mon profit devient nul.

1. Aidez moi ! Proposez moi une stratégie qui maximise mon profit espéré.
2. Appliquez votre solution aux données $T = 5, p_1 = 0.9, p_2 = 0.6, p_3 = 0.7, p_4 = 0.2, p_5 = 0.4$

Exercice 2 *Une variante du jeu de Nim*
Ca y est, on a bien mangé. Pour bien finir la soirée, nous avons décidé de faire un petit jeu. On a disposé N allumettes sur une table. On prend à tour de rôle 1,2 ou 3 alumettes. L'objectif pour moi est de forcer ma copine à prendre la dernière allumette (évidemment, elle essaye de faire pareil)

1. Formuler ce problème comme un programme dynamique déterministe.
2. Déterminer une stratégie optimale qui me permette de lui mettre la patée à coup sûr.
3. Pour corser un peu le jeu, on prend une variante : après chaque prise d'allumettes par un joueur, et s'il reste encore des allumettes sur la table, une pièce de monnaie non biaisée est lancée. Si le résultat est pile, le joueur qui vient de prendre des allumettes doit en ôter une de plus de la table. Formuler ce problème comme un programme dynamique.

Rappel : algorithme de programmation dynamique

Soit un problème d'optimisation séquentielle sur N périodes. Pour tout état initial x_1, le coût optimal $J^*(x_1)$ est égal à $J_1(x_1)$, où la fonction J_1 est donnée par la dernière étape de l'algorithme de programmation dynamique, qui procède depuis la période N jusqu'à la période 1, selon les étapes suivantes :

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

$$(\forall k \in \{N \cdots 1\}) J_k(x_k) = \min_{u_k \in U_k(x_k)} \mathrm{E}_{\omega_k} \left(g_k(x_k, u_k, \omega_k) + J_{k+1}(f_k(x_k, u_k, \omega_k)) \right)$$

Enfin, si $\mu_k^*(x_k) = u_k^*$ minimise cette espérance, pour tout x_k et tout k, la politique $\pi^* = (\mu_1^* \cdots \mu_N^*)$ est optimale