Processus décisionnels markoviens

Pour un contrôle stochastique optimal

Laurent Bougrain & Olivier Buffet
Université de Lorraine
Laurent.Bougrain@loria.fr

Bibliographie

SUTTON R. S. et BARTO A. G. : Reinforcement Learning :An Introduction, MIT Press, Cambridge, MA, 1998, A Bradford Book

version en ligne : https://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html

SIGAUD O. and BUFFET O. editeurs, *Processus décisionnels de Markov et intelligence artificielle*. Hermes, 2008

http://researchers.lille.inria.fr/~munos/papers/files/bouquinPDMIA.pdf

ROCHA-MIER L. E. : Apprentissage dans une INtelligence COllective NEuronale : application au routage de paquets sur Internet, thèse de Doctorat de l'INP Grenoble, 2002

RUSSELL S., NORVIG P. : Intelligence Artificielle, éd. Pearson, 2010

Laurent.Bougrain@loria.fr

Aperçu

Notions de base

- Processus stochastique
- Processus markovien
- Contrôle de processus markoviens

Problèmes décisionnels de Markov

Critère d'optimalité de Bellman

Algorithme de programmation dynamique

Equation de Bellman

Algorithme d'Itération sur les valeurs

Algorithme d'itération sur les politiques

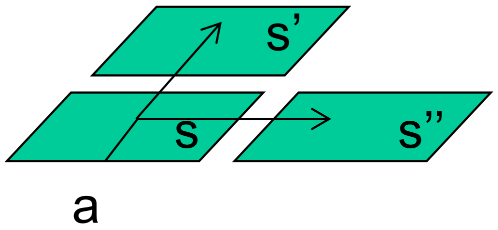
Laurent.Bougrain@loria.fr

Environnement

- ✔ **entièrement observable** : l'état courant est parfaitement connu
- • **partiellement observable** : l'état courant est mal connu

s : état

- • **déterministe** : l'état d'arrivée est unique et connu lorsqu'on effectue une action
- ✔ **stochastique** ou non déterministe : l'état d'arrivée est incertain lorsqu'on effectue une action. Une probabilité de transition est associée à chaque possibilité.



s, s', s'' : états

a : action

- ✔ **Transition markovienne** : La probabilité d'attendre un s' ne dépend que de l'état précédent s (et non des états plus anciens)

$$T(s,a,s') = P(s'|s,a)=0.8$$
$$T(s,a,s'')=P(s''|s,a)=0.2$$

Laurent.Bougrain@loria.fr

Processus de décision séquentiel

✔ Processus à **temps discret** : les instants de décision sont discrets

Processus à **temps continu** : les instants de décision sont continus

Problème à ***horizon fini*** *(ex.: Tetris avec un nombre fini de pièces) :* le nombre d'étapes de décision est fini et connu

✔ Problème à ***horizon infini*** *(ex.: Tetris à hauteur infinie) :* le nombre d'étapes de décision est infini

Problème à ***horizon indéfini*** *(ex.: Tetris à hauteur finie) :* le nombre d'étapes de décision est fini mais inconnu

Comment un système peut-il décider de sacrifier sa performance à court terme pour optimiser sa performance à long terme ?

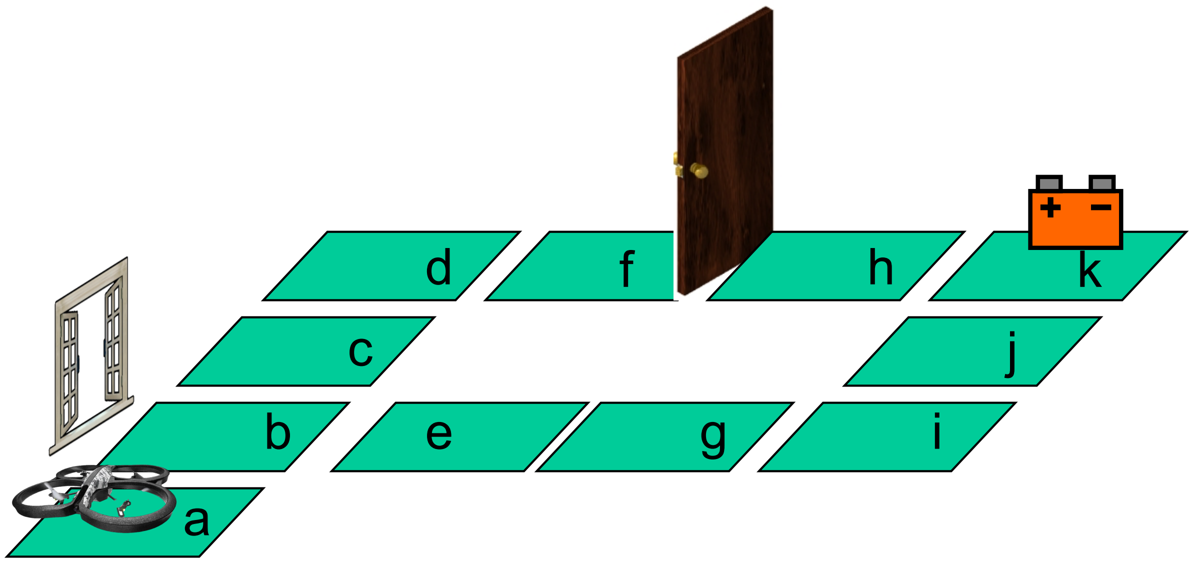
Laurent.Bougrain@loria.fr

Problème de décision séquentiel dans un milieu stochastique

Etats : a, b, c, d, e, f, g, h, i, j, k

Actions : A_a ={haut},…, A_f ={gauche, droite}…

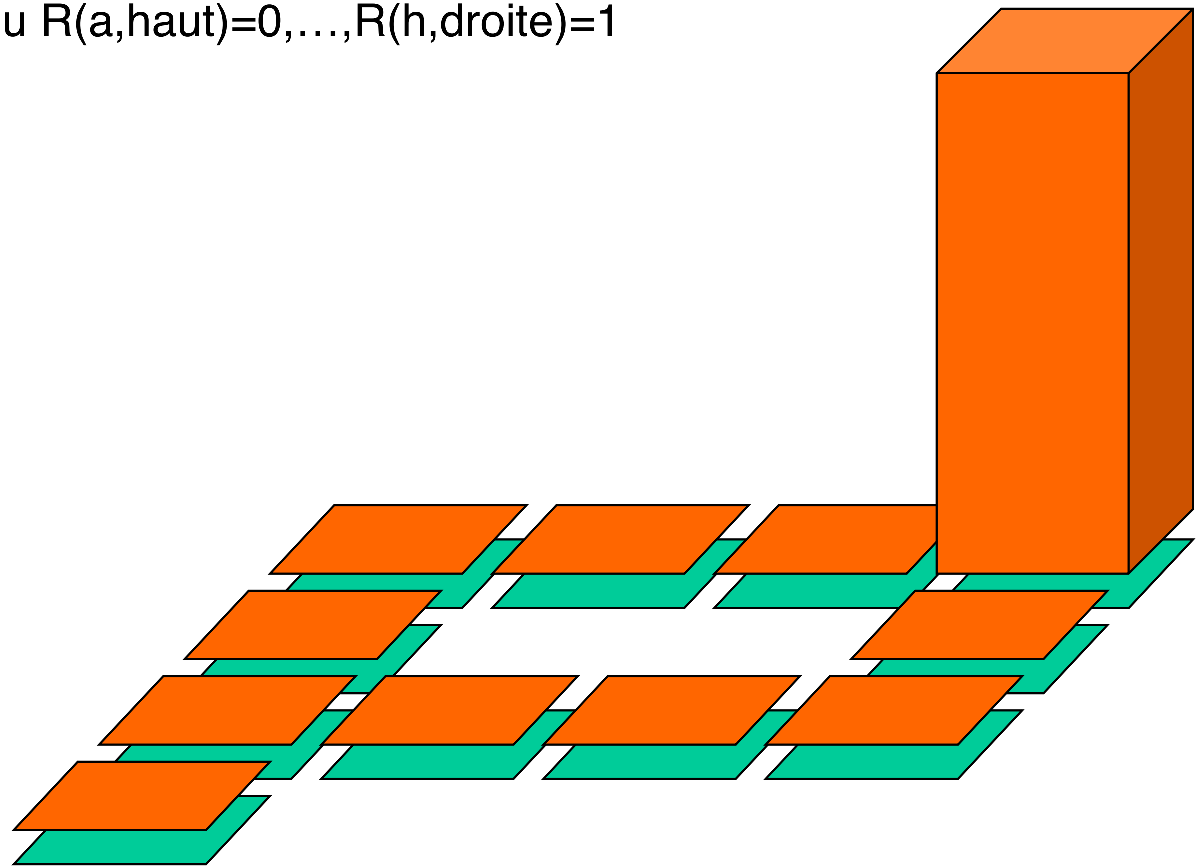
Probabilités de transition : P_{ab}(haut)=1,…,
P_{fh}(droite) = 0.8 et P_{ff}(droite) = 0.2 P_{fd}(droite) = 0…
P_{fd}(gauche) = 1…



Laurent.Bougrain@loria.fr

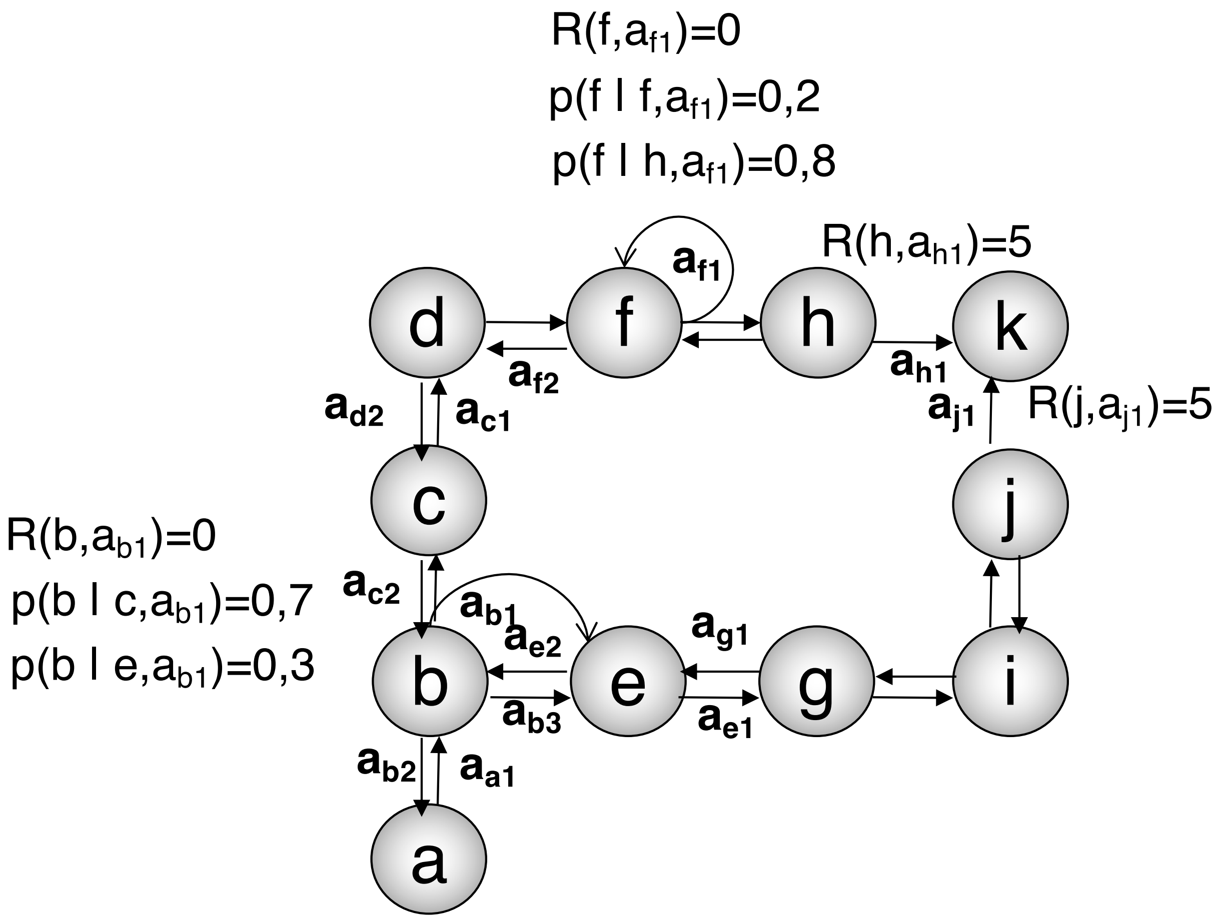
Problème de décision séquentiel dans un milieu stochastique

Récompenses : R(a) = 0,…, R(k)=1
ou R(a,haut)=0,…,R(h,droite)=1



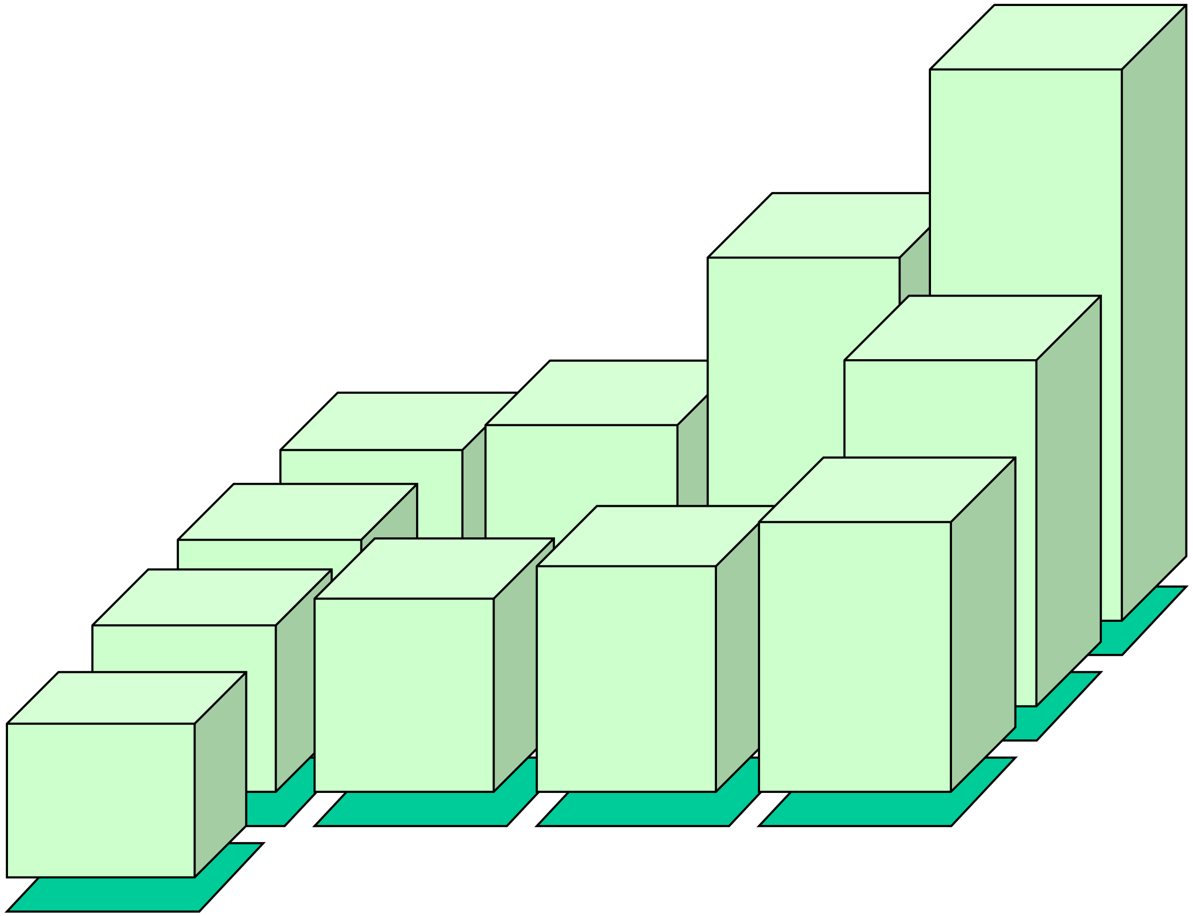
Laurent.Bougrain@loria.fr

Modélisation du Processus Décisionnel Markovien (MDP)



Laurent.Bougrain@loria.fr

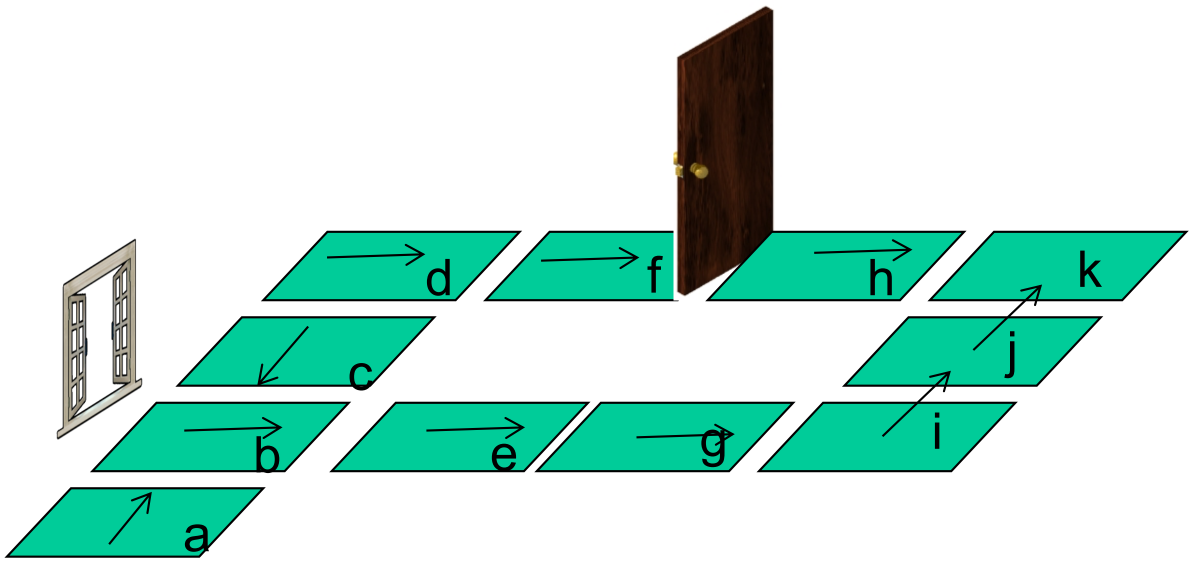
Fonction de valeur optimale : V(a) = 0,3 ,…, V(k) = 1



Laurent.Bougrain@loria.fr

Problème de décision séquentiel dans un milieu stochastique

Politique : μ(a) = haut, : μ(b) = droite, μ(c) = bas…

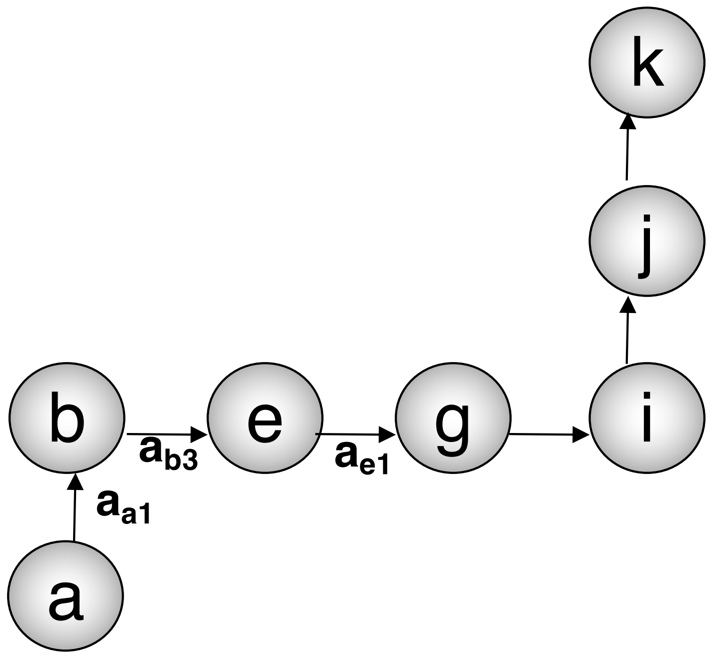


Laurent.Bougrain@loria.fr

Chaîne de Markov

Au regard de la politique : μ(a) = haut, : μ(b) = droite, μ(c) = bas…

on obtient la chaîne de Markov suivante :



Ici la politique est déterministe.

Laurent.Bougrain@loria.fr

Processus stochastique

Un ***processus stochastique*** { X_t : t $\in$ T } est caractérisé par une suite de variables aléatoires à valeurs dans un ensemble X des états indexées par t $\in$ T.

Si T est à valeurs discrètes, nous parlons de ***processus à temps discret***.

Si T est à valeurs continues, nous parlons de ***processus à temps continu***.

La probabilité d'être à l'état X_{t+1}, à partir de l'état X_t est notée :

$P(X_{t+1}|X_t, X_{t-1},...)$ où $X_t, X_{t-1},...$ représente l'historique

Un **processus est *déterministe*** lorsqu'il existe à l'instant t un état X_{t+1}, tel que :

$$P(X_{t+1}|X_t, X_{t-1},...) = 1$$

La notion de processus élargit la notion de variable aléatoire.

Une réalisation d'un processus stochastique est appelé ***trajectoire***. C'est donc la suite des valeurs de la variable aléatoire X_t.

Laurent.Bougrain@loria.fr

Processus markovien

Dans un processus stochastique, la valeur x(t) de la variable aléatoire X_t qui représente l'état du système à l'instant t dépendra des valeurs successives des variables aléatoires, c'est-à-dire de la trajectoire ou histoire du processus $\{X_t, X_{t-1}, X_{t-2}, \ldots\}$.

Si la partie de l'histoire d'un processus stochastique à prendre en compte à chaque instant t se réduit simplement à l'instant précédent t-1, alors nous pouvons nommer ce processus stochastique ***processus markovien***.

Plus précisément : $$P\left(X_{t+1} \middle| \{X_t, X_{t-1}, \ldots, X_1\}\right) = P\left(X_{t+1} \middle| \{X_t\}\right)$$

Une suite de variables aléatoires X_1, X_2, …, X_t, X_{t+1} constituera une ***chaîne de Markov*** si la probabilité lorsque le processus se trouve dans l'état X_{t+1} à l'instant t+1 dépend exclusivement de la probabilité de l'état précédent X_t, à l'instant t.

Laurent.Bougrain@loria.fr

Contrôle de processus markoviens

Soit $A_i = \{a_{i1}, a_{i2}, \ldots, a_{ik}\}$ l'ensemble des k actions possibles influençant la dynamique de l'environnement dans l'état i.

Nous associons au contrôle d'un système dynamique, un agent qui observe l'état courant de l'environnement et lui applique une action a_{ik}. De cette façon, l'agent influence la dynamique de son environnement et définit un processus décisionnel. Cette influence de l'agent sur la dynamique du système est résumée par la famille de distributions de probabilité :

$$p_{ij}(a_{ik}) = P\left(X_{t+1} = j \middle| X_t = i, A_i = a_{ik}\right)$$

qui représente la probabilité pour que le système passe, à l'instant t, d'un état i à un état j, lorsque l'agent effectue l'action a_{ik}.

Laurent.Bougrain@loria.fr

Contrôle de processus markoviens (2)

Ainsi, la probabilité de transition $p_{ij}(a_{ik})$ de l'état i à l'état j ne dépend que de l'état i et de l'action a_{ik}.

Ceci est la ***propriété de Markov***. Elle est très importante parce qu'elle indique que l'état actuel de l'environnement fournit l'information nécessaire à l'agent pour décider quelle action effectuer.

La probabilité de transition $p_{ij}(a_{ik})$ satisfait deux conditions qui sont imposées par la théorie de la probabilité :

$\forall i,j \; p_{ij}(a_{ik}) \geq 0$

$\forall i \; \Sigma_j \; p_{ij}(a_{ik}) = 1$

Les actions sont effectuées à certains instants appelés ***étapes de décision*** (ou ***séquences***).

Lorsque le nombre d'étapes de décision est fini, nous disons que le problème est à ***horizon fini*** *(ex.: Tetris à nombre de pièces fini)*, sinon nous parlons d'***horizon infini*** *(ex.: Tetris à hauteur infinie)*.

La suite des états de l'environnement, résultant des actions effectuées par l'agent à chaque étape de décision, constitue une **chaîne de Markov**.

Laurent.Bougrain@loria.fr

Stratégie de décision d'actions markovienne

Une ***stratégie*** pour la décision d'actions, $\pi = \{\mu_0, \mu_1, \dots\}$, est composée de plusieurs fonctions appelées ***politique d'action*** $\mu_t(i) \in A_i$, pour tout état $i \in X$ qui mettent en correspondance les états i,j,… et les actions A_{ik}.

Une politique est une règle utilisée par l'agent pour décider quelle action effectuer à partir de l'état actuel de l'environnement, à un instant précis t.

μ_t est une fonction qui met en correspondance l'état de l'environnement $X_t = i$ et l'action $A_i = a_{ik}$ à l'instant t = 0, 1, 2,…

Stratégie (ou politique)

- ***non stationnaire*** : les politiques d'action μ changent au cours du temps $\pi = \{\mu_0, \mu_1, \dots\}$
- **stationnaire** : les politique d'action ne changent pas $\pi = \{\mu, \mu, \dots\}$. Une stratégie stationnaire décidera donc de la même action à chaque instant lorsqu'un état en particulier est visité. Si une stratégie stationnaire μ est utilisée, la suite d'états ou trajectoire du système $X = \{X_t, X_{t-1}, X_{t-2}, \dots\}$ constituera une **chaîne de Markov** avec des probabilités de transition $p_{ij}(\mu(i))$, où $\mu(i)$ représente une action.

Laurent.Bougrain@loria.fr

Critère d'optimalité de Bellman

Principe : une stratégie optimale π^* est telle que, quel que soit l'état initial $X_0 = i$ et la décision initiale $a_{ik} \in A_t$, les décisions suivantes doivent constituer une sous-stratégie optimale, par rapport à l'état résultant de la première décision.

Une stratégie optimale π^* ne peut être formée que de politiques optimales $\{\mu_0^*, \mu_1^*, \ldots\}$.

C'est sur ce principe d'optimalité, démontré par l'absurde, que repose la programmation dynamique. On remplace par une optimisation séquentielle une optimisation globale de la fonction objectif.

Méthodologie pour construire une stratégie optimale :

1. Construire une politique optimale en prenant seulement en compte la dernière étape du système.
2. Prolonger la politique optimale en prenant en compte les deux dernières étapes du système.
3. Continuer pour le problème entier.

Laurent.Bougrain@loria.fr

Equation de Bellman

Pour étendre l'algorithme aux problèmes avec un horizon infini

•sous une stratégie stationnaire $\pi = \{\mu, \mu, \dots\}$

Equation d'optimalité de Bellman : $$V^*(i) = \min_{a \in A_i} \left\{ c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) V^*(j) \right\}$$

où $c(i,a)$ est **le coût de l'action** préconisée par la politique quand le système est dans l'état i et **γ est le taux d'atténuation**.

Le taux d'atténuation $0 \leq \gamma \leq 1$ est un moyen de contrôler les conséquences des actions de l'agent à court terme et à long terme. Si γ est petit, l'agent prendra uniquement en compte les conséquences immédiates de ces actions. A mesure que γ s'approche de 1, les coûts prennent une importance égale, y compris ceux obtenus en fin d'horizon.

Une fonction valeur d'un état X_t, représentée par V_t^{π}, indique le **coût total qu'un agent peut espérer accumuler dans le futur à partir de cet état** en suivant la stratégie π.

Si le problème d'optimisation consiste à **maximiser une récompense**, l'équation d'optimalité est

$$V^*(i) = \max_{a \in A_i} \left\{ R(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) V^*(j) \right\}$$

$R(i,a)$ représente alors la récompense de l'action.

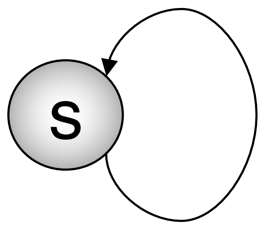
Laurent.Bougrain@loria.fr

Equation de Bellman et horizons fini et infini

L'équation de Bellman dans le cas d'un problème à **horizon fini** permet de calculer par itérations successives jusqu'au temps terminal les valeurs V à l'aide de méthodes de programmation dynamique (calcul en remontant le temps) plus efficace qu'une simple récurrence (calcul redondant si un état peut être atteint par plusieurs chemins).

Dans le cas d'un problème à **horizon infini**, l'équation de Bellman prend la forme d'un système de N équations avec une équation par état pour lequel il existe une unique solution de type point fixe. L'obtention des valeurs V peut se faire soit par itération, soit en résolvant un système d'équation. La solution de ce système détermine la fonction valeur optimale pour N états.

Pour transformer un problème à horizon fini en un problème à horizon infini, les états terminaux possèdent chacun une unique action réflexive avec un coût (ou une récompense) de 0.



$p(s|s,a)=p_{ss}(a)=1$

$c(s,a)=0$ ou $R(s,a)=0$

Laurent.Bougrain@loria.fr

Algorithme d'itération sur les valeurs (Value Iteration)

Voici deux méthodes pour calculer une politique optimale :

- La méthode d'itération sur les politiques (policy iteration)
- La méthode d'itération sur les valeurs (value iteration)

1. Q-valeurs

Soit μ une politique existante pour laquelle la fonction valeur $V^{\mu}(i)$ est connue pour tous les états i.

La Q-valeur $Q(i,a_{ik})$ pour chaque état $i \in X$ et action $a_{ik} \in A$ est définie comme le **coût immédiat** $c(i,a_{ik})$ plus la somme des coûts dégressifs de tous les états subséquents selon la politique μ :

$$Q^{\mu}(i,a_{ik}) = c(i,a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik}).V^{\mu}(j)$$

où l'action $a_{ik}=\mu(i)$.

Les Q-valeurs $Q(i,a_{ik})$ contiennent plus d'information que la fonction valeur $V^{\mu}(i)$.

En effet, les actions peuvent être classées en se basant seulement sur les Q-valeurs, alors que si elles étaient classées en se basant uniquement sur les valeurs d'état, il est nécessaire de connaître aussi les probabilités et les coûts de transition.

Laurent.Bougrain@loria.fr

Algorithme d'itération sur les valeurs (Value Iteration)

2. Algorithme
   1. Initialiser toutes les valeurs $V_0(i)$ pour état $i \in X = \{1, 2, \ldots, N\}$ avec une valeur arbitraire.

      $\varepsilon > 0$, $t = 0$, $\gamma \leq 1$
   2. Pour chaque valeur d'état $i \in \{1, 2, \ldots, N\}$, calculer $V_{t+1}(i)$ à partir de :

$$V_{t+1}(i) = \min_{a_{ik} \in A_i} \left\{ c(i, a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik}) V_t(j) \right\}$$

   3. Si $\left\| V_{t+1} - V_t \right\| < \varepsilon (1 - \gamma) / 2\gamma$ alors aller à l'étape 4 sinon t = t+1 et retourner à l'étape 2
   4. Pour tous les états $i \in X = \{1, 2, \ldots, N\}$ et toutes les actions $a_{ik} \in A_i$, calculer les Q-valeurs

$$Q^*(i, a_{ik}) = c(i, a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik}) V^*(j)$$

   5. Déterminer la politique optimale comme la politique gloutonne pour $V^*(i)$

$$\mu^*(i) = \arg\min_{a_{ik} \in A_i} Q^*(i, a_{ik})$$

pour tous les états $i \in X = \{1, 2, \ldots, N\}$

Laurent.Bougrain@loria.fr

Algorithme d'itération sur les valeurs (Value Iteration)

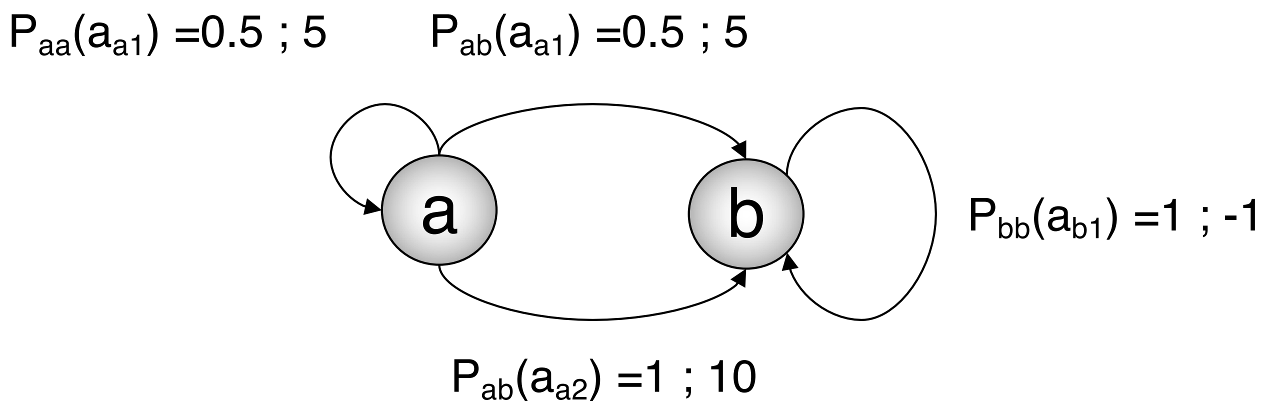
L'algorithme d'Itérations sur les valeurs est l'algorithme le plus utilisé pour résoudre des processus décisionnels markoviens en horizon infini.

Il permet de résoudre l'équation d'optimalité de Bellman en plusieurs itérations successives.

Pour tous les états, lorsque le nombre d'itérations tend vers l'infini, la fonction valeur du problème d'horizon fini converge uniformément vers la fonction valeur correspondante au problème d'horizon infini.

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les valeurs



Les probabilités de transition et les coûts sont supposés stationnaires.

Etapes de décision : $T = \{1,2,\ldots,k\}$ avec $K \leq \infty$

Etats de l'agent : $X = \{a, b\}$

Actions possibles : $A_a = \{a_{a1}, a_{a2}\}$, $A_b = \{a_{b1}\}$

Coût induits : $c(a,a_{a1}) = 5x0.5 + 5x0.5 = 5$, $c(a,a_{a2}) = 10$, $c(b,a_{b1}) = -1$

Probabilités de transition : $P_{ab}(a_{a1}) = 0.5$, $P_{ab}(a_{a1}) = 0.5$, $P_{aa}(a_{a2}) = 0$, $P_{ab}(a_{a2}) = 1$,
$P_{ba}(a_{b1}) = 0$, $P_{bb}(a_{b1}) = 1$

Nombre supposé d'étapes : $K = 2$

Stratégie : $\pi = \{\mu_1, \mu_2\}$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les valeurs

D'abord, initialisons ε=0.01 γ = 0.95.

Nous choisissons comme valeurs initiales $V_0(a) = V_0(b) = 0$.

Les itérations deviennent :

$$V_{t+1}(a) = \min_{a \in A_1} \{5 + \gamma(0.5V_t(a) + 0.5V_t(b)), \quad 10 + \gamma 1 V_t(b)\}$$

$$V_{t+1}(b) = \min_{a \in A_b} \{-1 + \gamma 1 V_t(b)\}$$

Les valeurs $V_t(a)$ et $V_t(b)$ convergent et après 169 itérations :

$$\|V_{t+1} - V_t\| < \varepsilon(1-\gamma) / 2\gamma = \frac{(0.01)(0.05)}{1.90} = 0.00026$$

$V_{169}(a)=-9$ et $V_{169}(b)=-20$. Elles représentent les valeurs ε-optimales.

Ensuite, il faut calculer les Q-valeurs pour tous les états et toutes les actions :

$$\begin{aligned} Q^*(a, a_{a1}) &= 5 + \gamma p_{aa}(a_{a1}) V^*(a) + \gamma p_{ab}(a_{a1}) V^*(b) \\ &= 5 + \gamma 0.5(-9) + \gamma 0.5(-20) \\ &= -8.77 \end{aligned}$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les valeurs

$$\begin{aligned} Q^*(a,a_{a2}) &= 10+\gamma p_{ab}(a_{a2})V^*(b) \\ &= 10+\gamma 1(-20) \\ &= -9 \end{aligned}$$

$$\begin{aligned} Q^*(b,a_{b1}) &= -1+\gamma p_{bb}(a_{b1})V^*(b) \\ &= -1+\gamma .1.(-20) \\ &= -20 \end{aligned}$$

Une fois les Q-valeurs calculées, nous pouvons déterminer la politique π-optimale :

$$\mu^*(a) = \arg\min_{a_{ak}\in A_a} \{Q^*(a,a_{a1}), Q^*(a,a_{a2})\}$$
$$\mu^*(a) = a_{a2}$$

$$\mu^*(b) = \arg\min_{a_{bk}\in A_b} \{Q^*(b,a_{b1})\}$$
$$\mu^*(b) = a_{b1}$$

Laurent.Bougrain@loria.fr

Algorithme d'itération sur les politiques (Policy iteration)

Plusieurs méthodes d'évaluation de la politique courante sont possibles : convergence vers un point fixe, résolution d'un programme linéaire, …

L'algorithme d'itérations sur les politiques est une autre méthode pour résoudre des PDM en horizon infini.

Cette méthode n'est pas faite pour résoudre des PDM en horizon fini.

Le fonctionnement d'itération sur les politiques est fait en deux étapes :

1. L'évaluation de la politique

   Dans cette étape, la fonction valeur et les Q-valeurs sont calculées pour tous les états et toutes les actions en utilisant la politique.

2. L'amélioration de la politique

   Dans cette étape, la mise à jour de la politique est faite de manière à être gloutonne par rapport à la fonction valeur calculée dans l'étape 1.

Laurent.Bougrain@loria.fr

Algorithme d'itération sur les politiques

1. Initialiser une politique arbitraire π^0, t = 0 et $\gamma \leq 1$
2. Résoudre le système d'équation de manière à obtenir les valeurs $V^{\mu n}(i)$ pour tout i de 1 à N

$$V^{\mu_n}(i) = \left\{ c(i, \mu(i) + \gamma \sum_{j=1}^{N} p_{ij}(\mu(i))V^{\mu_n}(j) \right\}$$

i.e.

$$\left(I - \gamma . P_{\mu_t}\right)\vec{V} = \vec{c}_{\mu_t}$$

3. Calculer les Q-valeurs pour chaque paire état-action (i,a) :

$$Q(i, a_{ik}) = c(i, a_{ik}) + \gamma \sum_{j=1}^{N} p_{ij}(a_{ik})V^*(j)$$

4. Faire la mise à jour de la politique comme suit (si on veut diminuer les coûts):

$$\mu_{t+1}(i) = \arg\min_{a_{ik} \in A_i} Q^{\mu_t}(i, a_{ik})$$

5. Si $\pi^{t+1} = \pi^t$ (i.e. $\forall i \in X = \{1, 2, \ldots, N\}$ $\mu_{t+1}(i) = \mu_t(i)$)

   Alors arrêter l'algorithme et mettre $\pi^* = \pi^t$

   Sinon incrémenter t = t + 1 et retourner au pas 2

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Choisissons une politique déterministe :

Pour l'étape 0 :

$\mu_0(a) = a_{a1}$

$\mu_0(b) = a_{b1}$

Etablissons la matrice de probabilités issue de la stratégie

$$P_{\mu_0} = \begin{bmatrix} p_{aa}(a_{a1}) & p_{ab}(a_{a1}) \\ p_{ba}(a_{b1}) & p_{bb}(a_{b1}) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}$$

et le vecteur de coûts également issu de la stratégie

$$\vec{c}_{\mu_0} = \begin{pmatrix} c(a, a_{a1}) \\ c(b, a_{b1}) \end{pmatrix} = \begin{pmatrix} 5 \\ -1 \end{pmatrix}$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les stratégies

Calculons les valeurs à partir du système pour γ=0,95 :

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \gamma \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix}\right) \cdot \begin{pmatrix} V_t(a) \\ V_t(b) \end{pmatrix} = \begin{pmatrix} 5 \\ -1 \end{pmatrix}$$

$$\begin{cases} V_t(a) - \gamma 0.5 V_t(a) - \gamma 0.5 V_t(b) = 5 \\ V_t(b) - \gamma V_t(b) = -1 \end{cases}$$

$$\Rightarrow \begin{cases} (1 - \gamma 0.5) V_t(a) = \gamma 0.5 V_t(b) + 5 \\ V_t(b) = -1 / (1 - \gamma) = -20 \end{cases}$$

$$\Rightarrow \begin{cases} V_t(a) = (\gamma 0.5 * -20 + 5) / (1 - \gamma 0.5) = -8.57 \\ V_t(b) = -20 \end{cases}$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Calculons les Q-valeurs :

$$
\begin{aligned}
Q(a,a_{a1}) &= 5+\gamma p_{aa}(a_{a1})V(a)+\gamma p_{ab}(a_{a1})V(b) \\
&= 5+\gamma 0.5*-8.57+\gamma 0.5*-20 \\
&= -8.57
\end{aligned}
$$

$$
\begin{aligned}
Q(a,a_{a2}) &= 10+\gamma p_{ab}(a_{a2})V(b) \\
&= 10+\gamma*1*-20 \\
&= -9
\end{aligned}
$$

$$
\begin{aligned}
Q(b,a_{b1}) &= -1+\gamma p_{bb}(a_{b1})V(b) \\
&= -1+\gamma*1*-20 \\
&= -20
\end{aligned}
$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Calculons la politique :

$$\mu_{t+1}(a) = \arg\min_{a_{ak} \in A_a} \left\{ Q^{\mu_t}(a, a_{a1}), Q^{\mu_t}(a, a_{a2}) \right\}$$

$$\mu_{t+1}(a) = \arg\min_{a_{ak} \in A_a} \{-8{,}57; -9\}$$

$$\mu_{t+1}(a) = a_{a2}$$

$$\mu_{t+1}(b) = \arg\min_{a_{bk} \in A_b} \left\{ Q^{\mu_t}(b, a_{b1}) \right\}$$

$$\mu_{t+1}(b) = \arg\min_{a_{bk} \in A_b} \{-20\}$$

$$\mu_{t+1}(b) = a_{b1}$$

La politique a changé

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Pour l'étape 1 :

$\mu_1(a) = a_{a2}$

$\mu_1(b) = a_{b1}$

Etablissons la matrice de probabilités issue de la stratégie

$$P_{\mu_1} = \begin{bmatrix} p_{aa}(a_{a2}) & p_{ab}(a_{a2}) \\ p_{ba}(a_{b1}) & p_{bb}(a_{b1}) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

et le vecteur de coûts également issu de la stratégie

$$\vec{c}_{\mu_1} = \begin{pmatrix} c(a, a_{a2}) \\ c(b, a_{b1}) \end{pmatrix} = \begin{pmatrix} 10 \\ -1 \end{pmatrix}$$

Et on recommence jusqu'à ce que la politique ne change plus

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Calculons les valeurs à partir du système :

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \gamma \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}\right) \cdot \begin{pmatrix} V_t(a) \\ V_t(b) \end{pmatrix} = \begin{pmatrix} 10 \\ -1 \end{pmatrix}$$

$$\begin{cases} V_t(a) - \gamma V_t(b) = 10 \\ V_t(b) - \gamma V_t(b) = -1 \end{cases}$$

$$\Rightarrow \begin{cases} V_t(a) = \gamma V_t(b) + 10 \\ V_t(b) = -1/(1-\gamma) = -20 \end{cases}$$

$$\Rightarrow \begin{cases} V_t(a) = (-\gamma * 20 + 10) = -9 \\ V_t(b) = -20 \end{cases}$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Calculons les Q-valeurs :

$$
\begin{aligned}
Q(a, a_{a1}) &= 5 + \gamma p_{aa}(a_{a1}) V(a) + \gamma p_{ab}(a_{a1}) V(b) \\
&= 5 + \gamma 0.5 * -9 + \gamma 0.5 * -20 \\
&= -8.775
\end{aligned}
$$

$$
\begin{aligned}
Q(a, a_{a2}) &= 10 + \gamma p_{ab}(a_{a2}) V(b) \\
&= 10 + \gamma * 1 * -20 \\
&= -9
\end{aligned}
$$

$$
\begin{aligned}
Q(b, a_{b1}) &= -1 + \gamma p_{bb}(a_{b1}) V(b) \\
&= -1 + \gamma * 1 * -20 \\
&= -20
\end{aligned}
$$

Laurent.Bougrain@loria.fr

Ex: application de l'algorithme d'itération sur les politiques

Calculons la politique :

$$\mu_{t+1}(a) = \arg\min_{a_{ak} \in A_a} \left\{ Q^{\mu_t}(a, a_{a1}), Q^{\mu_t}(a, a_{a2}) \right\}$$

$$\mu_{t+1}(a) = a_{a2}$$

$$\mu_{t+1}(b) = \arg\min_{a_{bk} \in A_b} \left\{ Q^{\mu_t}(b, a_{b1}) \right\}$$

$$\mu_{t+1}(b) = a_{b1}$$

La politique est stable, on s'arrête

Laurent.Bougrain@loria.fr

Stratégie non stationnaire

Politique déterministe :

Pour l'étape 1 :

$\mu_1(a) = a_{a1}$

$\mu_1(b) = a_{b1}$

Pour l'étape 2 :

$\mu_2(a) = a_{a2}$

$\mu_2(b) = a_{b1}$

Politique non déterministe :

Pour l'étape 1 :

$q_{\mu1(a)}(a_{a1}) = 0.7$

$q_{\mu1(a)}(a_{a2}) = 0.3$

$q_{\mu1(b)}(a_{b1}) = 1$

Pour l'étape 2 :

$q_{\mu2(a)}(a_{a1}) = 0.4$

$q_{\mu2(a)}(a_{a2}) = 0.6$

$q_{\mu2(b)}(a_{b1}) = 1$

Où $q_{\mu t(i)}(a_{ik})$ représente la probabilité de choisir l'action a_{ik} dans l'état i à l'instant t.

Laurent.Bougrain@loria.fr

En résumé

Problématique : contrôle optimal stochastique

Formalisation : processus décisionnel markovien

$$\langle S, A, T, R \rangle$$

états
actions
transition → $T(s,a,s')=P(s'|s,a)$
récompense → $R(s)$

On cherche une politique $\pi : S \rightarrow A$

qui maximise les récompenses sur le long terme $\sum_{t=0}^{\infty} \gamma^t . E[R(s_t)|s_0]$

On calcule la fonction de valeur optimale :

$$V^{\pi^*}(s) = \max_{a \in A} \left[R(s,a) + \gamma . \sum_{s' \in S} T(s,a,s').V^{\pi^*}(s') \right]$$

Laurent.Bougrain@loria.fr

Pour aller plus loin

Lorsque le nombre d'états est important, il n'est plus possible d'évaluer tous les états avec ces algorithmes.

•La fonction de valeur peut être approchée par un approximateur de fonction comme les réseaux de neurones.

•Le nombre d'états évalués peut être réduit en évitant de considérer les états qui ne devraient pas être visités par une politique optimale (cf A* pour les cas déterministes).

Si les fonctions de récompense et de transition ne sont pas connues au départ, elles peuvent être apprises par essais-erreurs à l'aide d'une apprentissage par renforcement.

Les PDM partiellement observables (PDMPO) s'attachent à modéliser un problème pour lequel l'état courant n'est pas connu avec certitude (les actions possibles permettent d'acquérir de l'information sur l'état et/ou de changer d'état. ex. diagnostic médical).

Il est également possible de travailler à temps continu.

Laurent.Bougrain@loria.fr