

UN ORDONNANCEMENT DYNAMIQUE DE TÂCHES STOCHASTIQUES SUR UN SEUL PROCESSEUR

ALI DERBALA¹

Communiqué par Jean-Pierre Crouzeix

Abstract. We show that a particular dynamic priority given to jobs in a multitasks operating system of computers is a deteriorating jobs or a delaying jobs scheduling. Under some assumptions we also show that it is an index rule. To do this, we present the tool of bandit processes to solve stochastic scheduling problems on a single machine.

Résumé. Nous montrons qu'une priorité dynamique particulière allouée aux tâches dans un système d'exploitation d'ordinateurs multitâches s'interprète comme deux problèmes d'ordonnancement particuliers, l'ordonnancement de tâches détériorantes à durée opératoires variables et de tâches en retard ou en attente de réparation de la machine. Deux propositions sur son comportement sont énoncées. Sous certaines conditions nous montrons qu'elle est une règle d'indice. Pour le faire, nous présentons l'outil des processus bandits pour la résolution des problèmes d'ordonnancement stochastiques sur une machine.

Mots-clés : Indices de Gittins, ordonnancement stochastique, processus bandit, stratégies préemptive et non préemptive.

Reçu en avril 2001.

¹ Université de Blida, Faculté des Sciences, Département de Mathématiques, BP. 270, route de Soumaa, Blida 009, Algérie; e-mail : aliderbala@yahoo.com

1. INTRODUCTION

Soit “ n tâches” à exécuter sur une seule machine. À chaque tâche “ i ” on associe une priorité instantanée

$$p_i(t) = \frac{M_i t_i^a(t) + m_i t_i^e(t)}{t} \quad (1)$$

pour $t = 1, 2, \dots$ et $p_i(0) \in [m_i, M_i]$. m_i et M_i sont des valeurs réelles. $t_i^e(t)$ et $t_i^a(t)$ sont respectivement les temps d’exécution qu’a reçu la tâche i et son temps d’attente pour s’exécuter durant l’intervalle $[0, t]$. En chaque date de décision, on exécute la tâche avec la plus grande priorité. En cas de conflit, la règle du Tourniquet ou Round Robin est utilisée pour départager les tâches. Ce schéma d’ordonnement de tâches qui alloue des priorités dynamiques répond à Browne et Yechiali [1] qui se demandaient s’il existe des fonctions de détérioration de tâches qui produisent une politique d’indice. En général une priorité est interprétée comme une urgence. Haro et Proust [11], concepteurs, ont défini cette priorité particulière et l’ont implémenté dans le noyau multitâches d’un système d’exploitation. Elle est définie pour améliorer les performances du système informatique.

Notre contribution au paragraphe trois consiste à montrer qu’une expression de la valeur de cette priorité s’interprète comme deux problèmes d’ordonnement connus. Les problèmes d’ordonnement de tâches détériorantes à durée opératoires variables et de tâches en retard ou en attente de réparation de la machine. Deux propositions sur son comportement sont énoncées. Sous certaines conditions nous montrons que cette priorité est une *règle d’indice*.

Pour le faire nous introduisons et présentons au paragraphe deux les processus bandits, les définitions des indices d’allocation dynamiques ou indices de Gittins, les théorèmes d’existence, de caractérisation et de détermination de ses indices.

2. LES PROCESSUS BANDITS

L’approche théorique prédominante pour l’ordonnement stochastique est la théorie des files d’attentes. Une approche entièrement différente a été développée par Gittins *et al.* [2, 4, 6]. Une façon d’ordonner les tâches est de leur attribuer une priorité dynamique appelée indices d’allocation dynamiques ou indices de Gittins. Ils les ordonnent selon un ordre croissant (ou décroissant) si l’objectif est de maximiser des gains (ou minimiser des pertes). Ces indices sont calculés à tout instant et durant l’exécution des tâches, permettant à l’ordonneur d’adapter dans son ordre les arrivées et les exécutions au fur et à mesure que l’information sur la tâche est recueillie. Les “processus bandits” est une traduction de ce qu’on appelle communément dans la bibliographie anglo-saxonne “Bandit Processes”. En leur octroyant des coûts ou des gains positifs, ces processus modélisent des tâches [6]. Les problèmes où les temps d’exécution des tâches sont incertains sont

dit stochastiques. Cet outil est présenté pour résoudre les problèmes sur une machine sous le critère de l'espérance des coûts linéaires, prévisionnels et en moyenne. Les temps d'exécution sont supposés aléatoires de lois connues.

Un processus bandit est un processus semi-markovien à deux décisions. Il est défini par la donnée d'un sextuplet (Ω, U, P, F, C, a) où Ω est l'espace d'états du processus et peut être fini, dénombrable ou continu (dans notre cas il est supposé fini), U est un ensemble de décisions consistant en deux éléments "c" (continue) et "w" (wait), P est la loi de transition entre les états, F est la fonction de répartition de la durée de transition τ (dans le cas markovien, F est une loi exponentielle), C est une fonction coût et "a" est un réel ($0 \leq a \leq 1$) appelé facteur de prévision. Une transition au temps t de l'état $x(t)$ à l'état $x(t+1)$ implique une augmentation du coût de $a^t c[x(t), x(t+1), u(t), t]$.

Les réalisations $\{x(t) : t = 0, 1, \dots\}$ du processus sont appelées trajectoires auxquelles on associe avec chaque ensemble de décisions et d'états, un coût.

Le coût total qui s'accroît si le processus décrit une trajectoire $\{x(t) : t = 0, 1, \dots\}$ est $\sum_{t=0}^{\infty} a^t c[x(t), x(t+1), u(t), t]$, où $c[x, y, u, t]$ est une fonction coût. Dans ce cas le processus est dit à fonction coût séparable.

On impose les conditions suivantes : si on n'a pas changé d'état, on est presque sur d'avoir utiliser la décision w . Utilisé la décision w , n'entraîne aucun coût.

Dans la littérature des automaticiens, ces processus sont appelés processus de contrôle adaptatifs. Le problème d'ordonnancement stochastique trouve un outil puissant pour sa résolution en donnant aux tâches des priorités instantanées. Gittins et Jones [2] ont énoncé le premier théorème d'existence de ces priorités appelées indices d'allocation dynamiques, notées Dai et appelées par Whittle [20] indices de Gittins. L'interprétation de ces derniers est qu'ils représentent un rapport de moyenne de coût sur une moyenne de temps utilisé. C'est aussi un coût d'arrêt du processus si on introduit la notion d'arrêt.

Dans la suite, une décision, une action, une commande ou un contrôle sont synonymes.

On appelle *famille de processus bandits alternatifs* une suite de processus $\{(\Omega_i, U, P_i, F_i, C_i, a) : 1 < i < n\}$ tel qu'en chaque instant t ($t = 0, 1, 2, \dots$), les processus évoluent simultanément. Ils ont les mêmes instants de décision. En tout instant la décision "c" est appliquée à un processus "i" et w est appliquée à tous les autres. Le facteur prévisionnel "a" étant le même pour tous les processus. Le problème est de trouver une stratégie d'espérance de coût total optimale.

Une *famille simple de processus bandits alternatifs* est une famille où tous les processus sont exécutés (la décision continue est appliquée) au moins une fois. Elle est notée SFABP.

De telle famille modélise aussi les tâches à exécuter par un processeur ou une machine.

Exemple 2.1. [6] "n" tâches sont à exécuter sur une machine. Un coût C_i est contracté à la fin de l'exécution de chaque tâche i et leurs temps de service S_i sont connus. Dans quel ordre les tâches doivent-elles être exécutées pour minimiser le coût total ? Un ordonnancement optimal est celui qui exécute les tâches dans l'ordre

croissant de C_i/S_i (qui n'est autre que la règle de Smith [17]). C_i/S_i représentent un indice de Gittins ou un ordre de priorité.

Dans le cas où les temps de service sont aléatoires de lois connues, le calcul des indices de Gittins nous fournit l'ordre dans lequel les tâches seront exécutées et les instants de préemption. Un processus bandit qui produit un gain positif en un temps aléatoire est appelé "tâche ou job". Une famille simple de tels processus définit le problème de l'ordonnancement de tâches sur une seule machine [6].

En général trois critères sont à considérer. Si un processus suit une trajectoire $\{x_t, t \in n\}$, pour toute politique π , on définit respectivement les critères de l'espérance du coût prévisionnel $E_\pi [\sum_{t=0}^\infty \alpha^t c(x_t, a_t)/x_0 = i]$, $i \geq 0$, des coûts positifs non prévisionnels $E_\pi [\sum_{t=0}^\infty c(x_t, a_t)/x_0 = i]$, $i \geq 0$, et du coût moyen $\lim_{n \rightarrow +\infty} \frac{E_\pi [\sum_{t=0}^n c(x_t, a_t)/x_0 = i]}{n+1}$.

Ce dernier est l'espérance de la plus longue exécution du coût moyen par unité de temps. Les coûts $c(x_t, a_t)$ sont supposés bornés et $\alpha < 1$. E_π désigne l'espérance conditionnelle sachant que la politique π est employée et que l'état du processus à l'instant zéro est i .

2.1. EXISTENCE DES INDICES D'ALLOCATION DYNAMIQUES

Au risque de se répéter, Gittins et Jones [2] ont énoncé le premier théorème d'existence des indices d'allocation dynamiques ou d'indices de Gittins notés Dai.

Théorème 2.1. [2] *Soit une famille de processus bandit alternatifs $\{(\Omega_i, U, P_i, F_i, C_i, a) : 1 < i < n\}$ alors il existe des fonctions $\{\nu_i : \Omega_i \rightarrow R, 1 \leq i \leq n\}$ tel que le choix $u_i(t) = c$ est optimal si et seulement si $\nu_i(x_i(t)) = \text{Min}_{1 \leq j \leq n} \nu_j(x_j(t))$.*

Les états de ces processus à l'instant t sont $x_i(t)$ ($1 \leq i \leq n$).

Pour une famille de processus bandits alternatifs, la fonction coût est minimale si et seulement si la stratégie des indices d'allocation dynamiques est exécutée.

2.2. CARACTÉRISATION DES INDICES D'ALLOCATIONS DYNAMIQUES OU DAI

Nash [12] a pu donner la première caractérisation de ces indices $\nu_i (1 \leq i \leq n)$. Une conséquence de la définition de $\nu_i(x_i)$ est qu'elle s'interprète comme un coût d'arrêt. On peut écrire pour tout état x_i , $\nu_i(x_i) = E\{\sum_{t=0}^{t^*-1} a^t C_i[x_i(t), x_i(t+1)] + a^{t^*} \nu_i(x_i)\}$ où $x_i(0) = x_i$. L'espérance est prise sur l'ensemble des réalisations du processus jusqu'à l'instant t^* , qui est un instant aléatoire défini par : $t^* = \text{Min}\{t, t > 0; \nu_i(x_i(t)) > \nu_i(x_i)\}$. Ce nombre qui représente un instant d'arrêt peut être infini.

Théorème 2.2. *Si $\nu_i(x_i)$ est défini pour tout état x_i , $x_i \in \Omega_i$, alors $\nu_i(x_i)$ est*

$$\text{l'unique solution finie des deux équations } \nu_i(x_i) = \frac{E\left[\sum_{t=0}^{t^*-1} a^t C_i[x_i(t), x_i(t+1)]\right]}{1-E[a^{t^*}]} \quad (N_1)$$

et $t^ = \text{Min}_{t>0}\{t, \nu_i(x_i(t)) \geq \nu_i(x_i)/x_i = x_i(0)\}$ (N₂).*

Une stratégie d'indice d'allocation dynamique est telle que dans une famille de processus bandit alternatifs $F = \{(\Omega_i, U, P_i, F_i, C_i, a) : 1 < i < n\}$, le processus i

est exécuté à l'instant t si et seulement si $\nu_i(x_i(t))$ vérifient le théorème d'existence et de caractérisation.

Beaucoup d'auteurs ont contribué à la caractérisation de ces indices de Gittins. Il s'agit de : Sevcik [15], Sevcik [16], Gittins et Glazebrook [3], Gittins [5], Nash [13], Glazebrook [8], Glazebrook [9], Glazebrook [10], Weiss [18], Nash [13]. Robinson [14] a donné un algorithme pour la détermination et le calcul des indices d'allocation dynamiques.

3. PRINCIPAUX RÉSULTATS

Les principaux résultats attribués à cette priorité dynamique allouée aux tâches dans un système d'exploitation d'ordinateurs multitâches sont cités ci-dessous. Ils constituent notre contribution. Un tel ordonnancement est appelé ordonnancement par priorité bornée en moyenne et noté OPBM.

3.1. OPBM : UN ORDONNANCEMENT STOCHASTIQUE DE TÂCHES DÉTÉRIORANTES OU À DURÉE OPÉRATOIRE VARIABLE SUR UNE SEULE MACHINE

Les tâches dont les durées opératoires ou d'exécution croissent avec le temps sont appelés des tâches détériorantes. Comme toutes les tâches sont supposées présentes dans l'atelier à l'instant 0, pour tout t , $t_i^e(t) + t_i^g(t) = t$. Pour tout t , $t = 0, 1, 2, \dots$, la formule de la priorité (1) devient : $p_i(t) = M_i + (m_i - M_i) \frac{t_i^e(t)}{t} = M_i - \Delta_i \frac{t_i^e(t)}{t}$.

Alors $A_i(t) = t \frac{p_i(t)}{m_i} = (1 + \alpha_i) t_i^e(t) - \alpha_i \cdot t = t_i^e(t) - \alpha_i(t - t_i^e(t))$ où $\alpha_i = -\frac{M_i}{m_i} \in R^+$.

On suppose que P_i est le temps d'exécution nécessaire pour la tâche i à l'instant $t = 0$, indépendant des P_j , $i \neq j$. La constante $\alpha_i = -\frac{M_i}{m_i}$ est le taux de détérioration de la tâche i . Si la tâche i n'a pas encore terminé son exécution à l'instant t , son temps d'exécution résiduel croît à un taux α_i .

$\alpha_i(t - t_i^p(t))$ est l'augmentation dans l'exécution résiduelle de la tâche i due à l'exécution des autres tâches durant $]0, t]$.

$A_i(t) = t \frac{p_i(t)}{m_i}$ s'interprète comme une "**réduction nette**" dans l'exécution résiduelle de la tâche i comme conséquence d'exécution des autres tâches sur $[0, t]$, où $p_i(t)$ est la priorité de la tâche i à l'instant t .

Plusieurs applications industrielles sont mentionnées pour ce modèle. L'exemple de la planification de la machine de maintenance ou de service dans la production d'acier où la matière coule durant les périodes d'attente. Dans la gestion des stocks où des produits périssent ou se détériorent, où la demande ou le manque croît. Des éléments se détériorent en stockage. Des éléments radioactifs qui se désintègrent dans le temps. Dans l'ordonnancement de fabrication des pneus, si la température d'un pneu de caoutchouc dans un espace d'attente entre le fourneau et la machine de moulage chute, alors il a besoin d'être réchauffé jusqu'à la température de moulage. Le réchauffement dépend du temps d'attente dans l'espace d'attente dont l'ordonnancement dépendra.

Soit une tâche qui n'a pas terminé son exécution. Si elle est élu pour s'exécuter à l'instant t , la prochaine date de décision ne peut être que l'instant $t + 1$ ou l'instant où une tâche j termine son exécution, soit le premier instant qui surviendra entre ces deux dates.

On définit l'état de la tâche i à la date de décision t par :

$$X_i(t) = \begin{cases} C \text{ (date de fin d'exécution)} & \text{si } i \text{ a terminé son exécution à l'instant } t \\ \{A_i(t), B_i(t)\} & \text{sinon} \end{cases}$$

$$B_i(t) = \text{Max}_{s \in D_t} A_i(t) = \text{Max}_{s \in D_t} t \frac{p_i(t)}{m_i} \text{ où } D_t = \{0, 1, \dots, t\}.$$

L'état du processus à l'instant t est $X(t) = \{X_1(t), \dots, X_N(t)\}$.

3.2. OPBM : UN ORDONNANCEMENT STOCHASTIQUE DE TÂCHES SOUMIS AU RETARD OU À L'ATTENTE DE LA RÉPARATION DE LA PANNE DE LA MACHINE

De la définition de l'OPBM, $t_i^a(t) = \frac{t}{\Delta_i} [P_i(t) - m_i]$ et $t_i^e(t) = \frac{t}{\Delta_i} [M_i - P_i(t)]$ sont respectivement le temps d'attente pour l'exécution et le temps d'exécution qu'a reçu la tâche i durant l'intervalle $[0, t]$. À l'instant zéro, chaque tâche nécessite P_i unités d'exécution. P_i est indépendant des P_j , si $j \neq i$.

Définissons le processus $Y(t) = (Y_1(t), \dots, Y_N(t))$, avec $Y_i(t) = (X_i(t), D_i(t), Z_i(t))$, où $X_i(t) = t_i^e(t)/N$, $D_i(t) = t_i^a(t)/N$ où N est le nombre de tâches dans l'atelier à l'instant 0. $Z_i(t)$ est un indicateur, qui prend la valeur 1 si la tâche i a terminé son exécution avant l'instant t et 0 sinon. On a aussi $X_i(0) = D_i(0) = Z_i(0) = 0, 1 \leq i \leq N$.

Le processus stochastique $\{D_i(t) = t_i^a(t)/N, t \in N\}$ est appelé processus à retard (delay process). Si la tâche i s'est donné une unité d'exécution à l'instant t , ayant reçu x unités d'exécution avant t , alors aucune exécution ne surviendra avant $t + 1 + D_i(x + 1) - D_i(x)$.

$$\begin{aligned} D_i(x + 1) - D_i(x) &= (1/N)[t_i^a(x + 1) - t_i^a(x)] \\ &= (1/N) \frac{(x + 1)}{\Delta_i} [P_i(x + 1) - m_i] - (1/N) \frac{x}{\Delta_i} [P_i(x) - m_i] \\ &= (1/N) \left\{ \frac{x}{\Delta_i} [P_i(x + 1) - P_i(x)] + \frac{1}{\Delta_i} [P_i(x + 1) - m_i] \right\}. \end{aligned}$$

D'où : $(1/N) \left\{ \frac{x}{\Delta_i} [P_i(x + 1) - P_i(x)] + \frac{1}{\Delta_i} [P_i(x + 1) - m_i] \right\}$ est interprété comme un retard causé par la $(x + 1)$ ième unité d'exécution reçu par la tâche i ou le temps de réparation d'une panne. Ce problème d'ordonnancement est un processus markovien de panne. Donnons ci-dessous deux propositions sur le comportement de la priorité.

Proposition 3.1. *Si la tâche j n'a pas terminé son exécution à l'instant t*

$$\text{Alors sa priorité } p_j(s) < \frac{m_j}{s} P_j, \quad \forall s \in]0, t].$$

Preuve. Si la tâche j n'a pas terminé son exécution à la date de décision t , alors P_j est plus grand que la plus grande réduction nette de l'exécution résiduelle de la tâche j observée durant $[0, t]$.

D'où $P_j > \text{Max}_{s \in [0, t]} A_j(s) \geq A_j(s), \forall s \in [0, t]. \forall s \in [0, t], P_j > \frac{s}{m_j} P_j(s)$. D'où le résultat. \square

Supposons que la tâche j est choisi pour exécution à l'instant t et que la prochaine date de décision surviendra à $t + \sigma$, où $\sigma \leq 1$.

Comme conséquence de cette exécution, une tâche k qui n'a pas terminé son exécution, $k \neq j$, se détériore. Son temps d'exécution augmentera linéairement à un taux α_k .

Proposition 3.2. *Soit k l'indice d'une tâche qui n'a pas terminé son exécution et qui ne s'exécute pas à l'instant t . Si la prochaine date de décision surviendra à $t + \sigma$, $\sigma \leq 1$ alors sa priorité est $p_k(t + \sigma) = \frac{t}{t + \sigma} p_k(t) - \alpha_k \cdot m_k \cdot \sigma \frac{1}{t + \sigma}$.*

Pour une tâche qui s'exécute à l'instant t et ne peut pas terminer son exécution dans un temps additionnel d'une unité alors sa priorité est

$$p_j(t + 1) = \frac{t}{t + 1} p_j(t) + \frac{m_j}{t + 1}.$$

Preuve. Soit $\text{Max}_{s \in D_t} (A_i(t)) = \text{Max}_{s \in D_t} \left(t \frac{p_i(t)}{m_i} \right) = B_i(t)$ où $D_t = \{0, 1, \dots, t\}$.

L'exécution résiduelle de i à t est stochastiquement identique à la distribution conditionnelle de $P_i - A_i(t)$ sachant que $P_i > B_i(t)$. Avec $A_i(0) = B_i(0)$, $i = 1, \dots, N$.

Se rappelant la définition de l'état de la tâche i à la date de décision t , l'état du processus à t est $X(t) = \{X_1(t), \dots, X_N(t)\}$. Supposons que la tâche i est choisie pour exécution à l'instant t . La prochaine date de décision surviendra à $t + \sigma$ où $\sigma \leq 1$. Comme conséquence de cette exécution, une tâche qui n'a pas terminé son exécution se détériore au taux α_k . Le nouvel état sera déterminé par :

$$A_k(t + \sigma) = A_k(t) - \alpha_k \sigma; B_k(t + \sigma) = B_k(t).$$

$A_i(t + 1) = A_i(t) + 1, B_i(t + 1) = B_i(t) + 1$. Le résultat découle de ces inégalités. \square

Théorème 3.1. *OPBM est une règle d'indice et $P_i(t) = M_i - \Delta_i \frac{t_i^e(t)}{t}$ est un indice d'allocation dynamique qu'on notera par "Dai".*

Preuve. Les tâches sont numérotées de 1 à n . La j -ième tâche est décrite par son indice, qui est connu, et de son temps de service actuel P_j , lequel est une réalisation tirée d'une loi de probabilité associée avec l'indice. Il ne sera connu qu'à la fin de l'exécution de la tâche. Notons par $f(\cdot)$ une "fonction allocation du processeur". Si $f(x) = j$, alors la tâche j est en service ou le processeur est alloué à la tâche j à l'instant x . À chaque tâche j , on associe "une fonction allocation de la tâche"

$$\sigma_j(t) = \begin{cases} 1 & \text{si } f(t) = j \\ 0 & \text{sinon} \end{cases}.$$

Au plus une tâche est servie en tout instant. Alors, $\forall t, \sum_{j=1}^n \sigma_j(t) \leq 1$.
 $t_j^p(t) = \min\{P_j, \sum_{y=0}^t \sigma_j(y)\}$ le temps passé à exécuter la tâche j durant $[0, t]$.
 $C_j = \text{Min}\{t, t_j^e(t) = P_j\}$ est la date de fin d'exécution de la tâche j .
 Alors le temps passé à exécuter la tâche j durant $[0, t]$.

$$P_i(t) = M_i - \Delta_i \frac{t_i^e(t)}{t} = M_i - \frac{\Delta_i}{t} \min \left\{ P_i, \sum_{y=0}^t \sigma_j(y) \right\}$$

$$= \text{Max} \left\{ M_i - P_i \frac{\Delta_i}{t}, M_i - \frac{\Delta_i}{t} \sum_{y=0}^t \sigma_j(y) \right\} = \frac{E \left[\sum_{t=0}^{t-1} a^t C_i[x_i(t), x_i(t+1)] \right]}{1 - Ea^t}$$

où $t = \text{Min}_{t>0}\{t, P_i(t) \geq P_i(0)\}$.

Soit t^* un instant d'arrêt d'espérance de $1 - (a^{t^*})$ égale à $t/2$. Cela signifie que la v.a $1 - (a^{t^*})$ suit une loi de distribution uniforme $U[0, t]$. Alors on en déduit que :

$$E \left[\sum_{t=0}^{t-1} a^t C_i[x_i(t), x_i(t+1)] \right] \times 1/2 = \text{Max} \left\{ M_i \cdot t - P_i \Delta_i, M_i \cdot t - \Delta_i \sum_{y=0}^t \sigma_j(y) \right\}.$$

Deux cas sont possibles :

- 1) si $\text{Max}\{M_i - P_i \frac{\Delta_i}{t}, M_i - \frac{\Delta_i}{t} \sum_{y=0}^t \sigma_j(y)\} = M_i - \frac{\Delta_i}{t} \sum_{y=0}^t \sigma_j(y)$ alors
 $M_i \cdot t - \Delta_i \sum_{y=0}^t \sigma_j(y) = E \left[\sum_{t=0}^{t-1} a^t C_i[x_i(t), x_i(t+1)] \right] \times 1/2 =$
 Il suffit de choisir par exemple une fonction coût : $E(a^y C_i[x_i(y), x_i(y+1)]) = -2\Delta_i \sigma_i(y)$ pour $y = 0, 1, \dots, t-2$ et $E(a^{t-1} C_i[x_i(t-1), x_i(t)]) = 2\{M_i \cdot t - \Delta_i[\sigma_i(t-1) + \sigma_i(t)]\}$. Le résultat découle ;
- 2) si $\text{Max}\{M_i - P_i \frac{\Delta_i}{t}, M_i - \frac{\Delta_i}{t} \sum_{y=0}^t \sigma_j(y)\} = M_i - P_i \frac{\Delta_i}{t}$

$$E \left[\sum_{t=0}^{t-1} a^t C_i[x_i(t), x_i(t+1)] \right] \times 1/2 = M_i \cdot t - P_i \Delta_i.$$

Par exemple, il suffit de définir $C_i[x_i(y), x_i(y+1)] = 0$, pour $y = 0, \dots, t-2$ et $E(a^{t-1} C_i[x_i(t-1), x_i(t)]) = 2(M_i \cdot t - P_i \Delta_i)$. □

4. CONCLUSION

Un schéma d'ordonnancement dynamique est défini. Nous montrons que cette priorité dynamique particulière allouée aux tâches par un ordonnanceur d'un système d'exploitation d'ordinateurs multitâches s'interprète comme deux problèmes d'ordonnancement bien connus (de tâches détériorantes ou tâches à durées opératoires variables et des ordonnancements à retard). Des résultats sur le comportement de la priorité sont obtenus. On a défini les processus bandits comme

outil pour la résolution des problèmes d'ordonnancement stochastiques sur une machine. Sous certaines conditions nous montrons qu'elle est une règle d'indice.

Remerciements. J'exprime ma gratitude au Professeur Christian Proust qui m'a fait intéressé à ce domaine, plus particulièrement aux processus bandits et leurs applications à l'ordonnancement stochastique.

RÉFÉRENCES

- [1] S. Browne et U. Yechiali, Scheduling deteriorating jobs on a single processor. *Oper. Res.* **38** (1990) 495-498.
- [2] J.C. Gittins et D.M. Jones, A dynamic index for the sequential Design of experiments, dans *Colloquia Mathematica Janes Bolai*, Vol. 9, *European meeting of statisticians*. Budapest, Hungary (1972) 241-266.
- [3] J.C. Gittins et K.D. Glazebrook, On Bayesian models in stochastic scheduling. *J. Appl. Probab.* **14** (1977) 556-565.
- [4] J.C. Gittins, Bandit process and dynamic allocation indices. *J. Roy. Statist. Soc. Sect. B* **41** (1979a) 148-177.
- [5] J.C. Gittins et D.M. Jones, A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika* **66** (1979b) 561-565.
- [6] J.C. Gittins, *Multi armed Bandit allocation indices*. John Wiley & Sons (1989).
- [7] K.D. Glazebrook, *Stochastic scheduling*, Ph.D. Thesis. Downing College, Cambridge (1976).
- [8] K.D. Glazebrook, Myopic strategies for Bayesian models in stochastic scheduling. *Oper. Res.* **19** (1982) 160-170.
- [9] K.D. Glazebrook, Optimal strategies for families of alternative bandit processes. *IEEE Trans. Automat. Control* **AC-28** (1983) 858-861.
- [10] K.D. Glazebrook, Evaluating the effects of machine breakdowns in stochastic scheduling problems. *Naval Res. Logist. Quarterly* **34** (1987) 319-335.
- [11] C. Haro et C. Proust, Un ordonnancement équitable par priorités bornées, dans *Colloque Méthodes et outils d'aide à la décision, MOAD'92*. Béjaïa, Algérie (1992) 46-49.
- [12] P. Nash, *Optimal allocation of resources between research projects*, Ph.D. Thesis. Cambridge University (1973).
- [13] P. Nash, A generalized bandit problem. *J. Roy. Statist. Soc. Sect. B* **42** (1980) 165-169.
- [14] D.R. Robinson, Algorithms for evaluating the dynamic allocation index. *Oper. Res. Lett.* **1** (1982) 72-74.
- [15] K.C. Sevcik, *The use of service time distributions in scheduling*, Ph.D. Dissertation. Comm. on Information Sciences, University of Chicago (1971).
- [16] K.C. Sevcik, Scheduling for minimum total loss using service time distributions. *J. Assoc. Comput. Mach.* **21** (1974) 66-75.
- [17] W.E. Smith, Various optimizers for single state production. *Naval Res. Logist. Quarterly* **3** (1956) 59-66.
- [18] G. Weiss, Turnpike optimality of Smith's rule in parallel machines stochastic scheduling. *Math. Oper. Res.* **17** (1992a) 255-270.
- [19] G. Weiss, *A tutorial in stochastic scheduling School of Isy*. E. Georgia Tech, Atlanta, GA 30332 (1992b).
- [20] P. Whittle, Multi-armed bandits and the Gittins index. *J. Roy. Statist. Soc. Sect. B* **42** (1980) 143-149.