

**UNIVERSITE SAAD DAHLAB DE BLIDA 1**

**Faculté des Sciences**

**Département de Mathématiques**

**THESE DE DOCTORAT**

en Mathématiques

Spécialité : Recherche Opérationnelle

ORDONNANCEMENT

DÉTERMINISTE ET STOCHASTIQUE

DE TÂCHES DÉTÉRIORANTES

Par

**Abdesselem KALI**

devant le jury composé de:

M. Blidia	Professeur, U. de Blida1	Président
A. Derbala	Maître de conférence, U. de Blida 1	Directeur de thèse
M. ABBAS	Professeur, U.S.T.H.B., Alger	Examineur
D. Aissani	Professeur, U. de Béjaïa	Examineur
M. Boudhar	Professeur, U.S.T.H.B., Alger	Examineur
M. Chellali	Professeur, U. de Blida 1	Examineur

Blida, 28 Juillet 2019

## RÉSUMÉ

Dans un atelier, « N » tâches sont à exécuter sur une machine. Leur durée opératoire augmente en fonction de leur date de début d'exécution. Elles sont dites «tâches détériorantes». De telle situation se pose par exemple pour les incendies de forêts, qui se propageant, requièrent un traitement d'extinction de plus en plus long. Nous formulons par un processus à 2-décisions semi-markovien un modèle des durées croissantes des interventions d'une seule ressource de lutte lorsque plusieurs incendies doivent être maîtrisés dans une forêt. Il prend en considération les différents phénomènes physiques de l'incendie qui se produisent à diverses échelles, les caractéristiques de la ressource de lutte ainsi que les risques liés à une défaillance éventuelle dans le confinement de l'incendie. Un nouvel outil appelé «règle ou priorité d'indices d'allocation dynamiques» est introduit pour résoudre ce problème. Sous le critère de l'espérance, cette règle optimise la somme des coûts de confinement sur un horizon de temps fini. Une caractérisation, la détermination et l'implémentation de cette priorité sont explicitées en détails. Son efficacité en temps et mémoire requise pour déterminer une solution optimale est constatée et prouvée. Les résultats expérimentaux effectués ont déterminé un cadre de lutte contre les incendies dont à l'intérieur la ressource atteint une plus grande efficacité.

**Mots-clés:** Ordonnancement, Une seule machine, Tâches détériorantes, Processus bandits, Indice d'allocation dynamique, Stratégie non préemptive.

## ABSTRACT

In a workshop, "N" jobs are to be carried out on a single machine. Their processing time increases according to their starting time. They are called "deteriorating jobs". Such a situation arises, for example, for forest fires containment operations, which will be longer as they are delayed. We formulate by a semi-Markovian 2-decision process a model of the increasing durations of a single fire-fighting resource interventions when there are several existing fires to be controlled in a forest. It takes into account different physical phenomena of the fire that occur at various scales, the characteristics of the fire-fighting resource as well as the risk involved in a potential fire containment failure. A new tool called "Dynamic Allocation Indices rule or priority" is introduced to solve this problem. The optimality of the rule is shown under the criterion of the expected total containment cost in a finite time horizon. Characterization, determination and implementation of this priority are explained in detail. Its efficiency in time and memory required to determine an optimal solution is established and proven. Experimental results depict the framework, and inside it, the forest fire-fighting resource achieves greater efficiency.

**Key-words:** Scheduling, Single machine, Deteriorating jobs, Bandit processes, Dynamic allocation indices, Strategies non pre-emptive.

## ملخص

في معمل، تقوم آلة بإنجاز "N" مهمة. تزداد مدة تنفيذ كل مهمة كلما تأخر وقت بدء إنجازها. توصف هذه المهام "بالمتهورة". هي حالة تطلق، على سبيل المثال، على عمليات احتواء حرائق الغابات، التي ستكون أطول على نحو متزايد كلما تأخر وقت انطلاق تنفيذها. نستعمل "تطور شبه مركوبي ثنائي القرار" لصياغة نموذج يقوم بحساب المدة التي يستغرقها مكافح وحيد في تدخلاته لإخماد مجموعة من حرائق الغابات نشبت في وقت واحد. يأخذ هذا النموذج في الاعتبار الظواهر الفيزيائية المختلفة للحريق في مختلف مستوياته، خصائص مكافح الحرائق والمخاطر المترتبة عند فشله المحتمل في احتواء الحريق. لحل هذه المشكلة نقوم بتقديم أداة جديدة تسمى "دلائل التخصيص الديناميكي". وبموجب معيار التوقع، تعمل قاعدة الدلائل على تخفيض إلى الحد الأدنى تكاليف احتواء الحرائق على أفق زمني محدود. يتم شرح طريقة تحديد هذه القاعدة بالتفصيل. كفاءتها من ناحية المدة و الذاكرة اللازمتين لحساب الحل الأمثل للمسألة واضحة ومبرهنة. أدت التجارب التي أجريناها إلى تحديد إطار للتحكم في الحرائق، يتسم المكافح فيه بالكفاءة العالية.

**الكلمات المفتاحية:** مسائل الجدولة، آلة واحدة، مهام متدهورة، تطور شبه مركوبي ثنائي القرار،

دلائل التخصيص الديناميكي، إستراتيجية الجدولة غير متقطعة.



## REMERCIEMENTS

Le travail présenté a été réalisé au Laboratoire de Modélisation et de Développement d'Algorithmes de la Recherche Opérationnelle (L.A.M.D.A-RO), Université Saad Dahlab Blida 1, au sein de l'équipe Ordonnancement, conduite et optimisation combinatoire.

Je tiens à remercier tout en premier lieu **DERBALA Ali**, mon directeur de thèse, Maître de Conférences à l'université Saad Dahlab de Blida 1, pour la confiance qu'il m'a témoigné. Sa convivialité et son support, ses encouragements et ses conseils, notamment pendant les premières années de chaos et confusion, m'ont permis de mener à bien ce travail. Il a su me guider et apporter un regard critique sur mes travaux tout en m'accordant une grande autonomie dans la conduite de mes recherches. Ce fut un réel plaisir de partager avec lui toutes ces heures de recherche sur des problèmes passionnants. J'ai acquis auprès de lui la rigueur scientifique qui faisait défaut dans mes présentations ainsi que mes rédactions. Qu'il trouvera ici toute ma reconnaissance et mon profond respect et mon admiration. J'espère que nous continuerons à travailler ensemble.

Mes remerciements vont également à **BLIDIA Mostafa**, Professeur de l'Université Saad Dahlab de Blida 1, d'avoir bien voulu examiner ce travail et présider ce jury.

Mes chaleureux et sincères remerciements s'adressent à **ABBAS Moncef**, Professeur de l'Université Houari Boumediene d'Alger, à **AISSANI Djamil**, Professeurs de l'Université Abderrahmane Mira de Béjaïa, à **BOUDHAR Mourad** Professeurs de l'Université Houari Boumediene d'Alger et à **CHELLALI Mustapha**, Professeur de l'Université Saad Dahlab de Blida 1, qui m'ont fait l'honneur d'accepter la lourde tâche d'être rapporteurs de ce travail et pour avoir analysé avec autant d'objectivité ce travail et avoir accepté d'être examinateurs.

Enfin, j'exprime ma reconnaissance à ma mère Yamina, à ma précieuse femme Aïcha et à mes deux fils Ziad et Awab, qui m'ont accompagnées dans cette aventure avec beaucoup de patience et surtout d'amour.

## TABLE DES MATIERES

RESUME .....	1
REMERCIEMENTS .....	4
TABLE DES MATIERES .....	5
LISTE DES FIGURES, TABLEAUX ET ALGORITHMES.....	7
INTRODUCTION .....	8
1. ORDONNANCEMENT A TACHES DÉTÉRIORANTES .....	12
1.1. Introduction .....	12
1.2. Pourquoi l'ordonnancement de tâches détériorantes est d'intérêt ? .....	13
1.3. Qu'est ce qui nous motive à étudier ce problème ? .....	14
1.4. De différents modèles d'ordonnancement à tâches détériorantes .....	15
1.5. Ordonnancer quand la durée d'exécution de la tâche augmente et sa valeur diminue avec le temps .....	17
2. LE CONCEPT DE TACHES DETERIORANTES DANS LA LUTTE CONTRE LES INCENDIES DE FORETS.....	18
2.1. Introduction .....	18
2.2. Notions de Pyrologie Forestière .....	18
2.3. Quatre modèles de propagation d'un incendie de forêt .....	20
2.4. Les mesures prises dans la gestion des incendies de forêts .....	21
2.5. Un modèle par tâches détériorantes pour la lutte contre les incendies forestiers .....	21
2.5.1. Présentation du problème .....	21
2.5.2. Notations et terminologies .....	22
2.5.3. Les hypothèses du modèle de RACHANIOTIS et PAPPIS.....	24
2.5.4. Construction du modèle de RACHANIOTIS et PAPPIS.....	25
2.6. Approche de solution .....	29
2.6.1. Rappels sur les procédures par séparation et évaluation.....	29
2.6.2. Borne supérieure .....	29
2.6.3. Règle de séparation .....	29
2.6.4. Borne inférieure .....	30
2.6.5. Règle de priorité .....	30
2.7. Problème à processeurs parallèles .....	31

3. UN ORDONNANCEMENT STOCHASTIQUE A UNE SEULE MACHINE DE TÂCHES DÉTÉRIORANTES : CAS DES INCENDIES DE FORÊTS.....	33
3.1. Introduction .....	33
3.2. Restrictions du modèle de RACHANIOTIS et PAPPIS .....	33
3.3. Un modèle stochastique pour le problème de la lutte contre les incendies de forêts .....	36
3.4. Un modèle semi-markovien pour le problème de confinement optimal.....	39
3.4.1. Les processus bandits .....	39
3.4.2. Existence des indices d'allocation dynamiques .....	41
3.4.3. Matrice stochastique de transition $P$ .....	41
3.5. Règle d'I.A.D pour la politique optimale de confinement.....	42
3.6. Détermination de la priorité de confinement optimale .....	43
3.7. Implémentation et expérimentation .....	45
CONCLUSION .....	49
LES ANNEXES	
A. Article de revue: Stochastic scheduling of single forest firefighting processor.....	52
B. Articles parus dans des colloques, symposium et congrès.....	59
C. Calcul de la superficie brûlée par un incendie .....	84
D. Code source Scilab pour calculer les I.A.D.....	86
REFERENCES.....	91

## LISTE DES FIGURES, TABLEAUX ET ALGORITHMES

Figure 2.1.	Allure exponentielle du taux de propagation $ROS_{ac}$ .....	26
Figure 3.1.	Évolution du front sur un terrain plat avec une végétation uniforme. Le front est illustré à toutes les 15 minutes .....	34
Figure 3.2.	Évolution du front sur une pente ayant une élévation de $\pi/16$ . Le front est illustré à toutes les 15 minutes .....	35
Figure 3.3.	Évolution du front sur une pente ayant une élévation de $\pi/4$ . Le front est illustré à toutes les 15 minutes .....	35
Figure 3.4.	Calcul de probabilité: $P_1$ et $P_2$ .....	38
Tableau 2.1.	Paramètres utilisés pour décrire le comportement d'un incendie .....	20
Tableau 3.1.	Résultats des expérimentations numériques .....	48
Algorithme 2.1.	Calcul de la borne inférieure initiale .....	30
Algorithme 2.2:	Heuristique pour une affectation parallèle des processeurs.....	32
Algorithme 3.1:	Calcul de l'IAD $\gamma_i(t)$ .....	46

## INTRODUCTION

Les problèmes d'ordonnancement sont posés dans de nombreux secteurs de l'industrie et des services. Ordonnancer est l'exécution d'un ensemble de tâches en leur attribuant les ressources nécessaires de manière à optimiser un ou plusieurs objectifs sous des contraintes temporelles et financiers.

Une bonne modélisation d'un problème d'ordonnancement est la partie indispensable dans l'élaboration d'un schéma de résolution. Dans la littérature, les modèles développés considèrent un environnement en général déterministe. Les données d'un problème d'ordonnancement sont souvent confrontées à une censure ou un manque d'information. Ces sources de perturbation sont nombreuses et variées.

Une tâche est un travail dont la réalisation nécessite un certain nombre d'unités de temps, sa durée d'exécution et d'unités de chaque ressource. Une ressource est un moyen, technique ou humain, dont la disponibilité limitée ou non est connue à priori. La connaissance de la durée opératoire ou de la durée d'exécution d'une tâche est importante et principale dans tout problème d'ordonnancement.

Une tâche est dite «détériorante» si sa durée opératoire augmente en fonction de sa date de début d'exécution. Ce type d'ordonnancement est d'intérêt. Il trouve au moins neuf exemples ou applications dans plusieurs secteurs de recherches et d'activités indiquées dans notre page 13 ci-dessous. La littérature est abondante. C'est le cas de la lutte contre les incendies [1] et les épidémies [2], l'exécution des protocoles médicaux sous la dégradation de l'état du patient [3], dans les maintenances industrielles, réparations et nettoyage [3], etc.

Notre motivation à étudier ce problème est la découverte de notre recherche bibliographique sur l'ordonnancement à une seule machine de tâches détériorantes, d'un nombre important d'articles publiés. Il est un créneau porteur. Dans leurs conclusions et perspectives, des problèmes ouverts ou des problèmes non encore étudiés sont relatés. Nous nous sommes inspirés de trois articles [4,5,6] qui constitueront notre motivation à faire cette étude. Nous les décrivons ci-dessous. Nous avons contribué à l'étude du phénomène de la lutte contre les incendies de forêts. Cette thèse a fait l'objet d'une publication dans la littérature [7].

Plusieurs modèles de demande ont été proposés pour déployer les ressources disponibles aux bons endroits de telle sorte qu'un incendie de forêt soit pris en charge dans une limite de temps déterminée. RACHANIOTIS et PAPPIS [4] ont abordé le problème d'allocation d'une seule ressource à l'aide du concept des tâches détériorantes. Le modèle établi détermine la perte de valeur qui augmente tant que les incendies ne sont pas maîtrisés et la durée croissante des interventions requises pour les éteindre.

Le modèle est déterministe et relie les paramètres physico-chimiques les plus probables de l'incendie et les paramètres de la ressource de lutte utilisée. Il est construit sur des bases mathématiques. Il est unanimement reconnu mais sa formulation complexe est difficile à utiliser pour un ordonnancement optimal. Dans la pratique, il est considéré comme une variable aléatoire. Malgré des multiples recherches, les propriétés d'un incendie de forêt ne sont pas entièrement établies. Il existe de nombreuses et complexes interactions entre les facteurs physiques du milieu naturel et les caractéristiques biologiques des combustibles.

La déclaration, le développement et la propagation d'un incendie font appel à des paramètres qu'il est judicieux de fixer. La modélisation stochastique est envisageable et la plus indiquée. La connaissance a priori des données du problème n'est pas toujours suffisante pour déduire correctement les lois de probabilités associées. Pour palier à cette difficulté, la littérature propose une utilisation de la distribution de la loi normale. Elle est utilisée pour maximiser la chance de succès avec un niveau de risque acceptable, ou pour minimiser le risque pour un niveau de succès désiré. Le modèle stochastique utilise les informations dont on peut raisonnablement disposer. Le modèle de RACHANIOTIS et PAPPIS [4] donne des approximations plus réalistes de l'espérance mathématique et de l'écart-type de la loi utilisée. Le développement d'un modèle généralisé reste toujours un défi permanent de recherche qualifié de problème ouvert.

Une approche théorique pour résoudre les problèmes d'ordonnancement stochastique est la théorie des «processus bandits», des processus de décision semi-markoviens [8]. Ils modélisent des tâches (les incendies) à exécuter sur une machine (la ressource de lutte). GITTINS et JONES [9] ont étudié les politiques d'ordonnancement de tâches stochastiques de type myope, backwards induction, forwards induction, politique d'indices etc. Si l'objectif de la résolution d'un problème d'ordonnancement stochastique est l'espérance des coûts linéaires et prévisionnels, ces auteurs ont montré que

la politique d'indices est optimale. On associe à chaque tâche une priorité dynamique appelé «indice d'allocation dynamique» et est noté I.A.D. En tout instant de décision, ces I.A.D sont calculés et la tâche qui a le plus grand I.A.D est exécutée. En cas de conflit ou d'égalité entre les plus grands I.A.D, on arbitre en choisissant une tâche selon généralement une règle d'ordonnancement connue. L'utilisation de ces I.A.D réduit rigoureusement la complexité en temps et mémoire requise pour déterminer une solution optimale. Cela se fait en réduisant un problème de dimension  $n$  en  $n$  problèmes unidimensionnels.

Pour une politique de confinement optimale des incendies, une priorité d'I.A.D est établie. Sa détermination s'avère d'une importance pratique. Les équations de Bellman apparaissent naturellement dans le calcul des I.A.D. Il y a un certain nombre de méthodes pour résoudre ces équations. Voir par exemple [10]. Dans le modèle décisionnel semi-markovien élaboré, la forme de la matrice des probabilités de transition permet de résoudre l'équation de Bellman correspondante d'une manière analytique efficace.

Le contenu de ce mémoire est organisé en trois chapitres distincts.

Le premier chapitre décrit le contexte général de notre travail. Il rappelle l'essentiel des définitions et notations utilisées dans les problèmes d'ordonnancement de tâches détériorantes où des exemples sont présentés. Les motivations qui nous ont incité à étudier ces problèmes sont aussi évoquées. Des différents modèles de détérioration de tâches sont fournis. Un aperçu sur l'ordonnancement quand la durée d'exécution de la tâche augmente et sa valeur diminue dans le temps est aussi exposé.

L'étude de trois articles publiés dans la littérature ont constitué notre motivation à réaliser ce travail. Des détails nécessaires à notre étude sont donnés dans le deuxième chapitre. Une introduction est faite. Les notions de pyrologie forestière, quatre modèles de propagation d'un incendie de forêt, les mesures prises dans la gestion des incendies de forêts, un modèle par tâches détériorantes pour la lutte contre les incendies forestiers, un premier algorithme de résolution utilisant la notion de séparation et évaluation et le problème à processeurs parallèles où un second algorithme heuristique sont présentés en détails.

Le troisième chapitre sera consacré à une présentation et une résolution optimale du problème stochastique de la lutte contre les incendies de forêts. Un modèle de prise de décision semi-markovien décrivant l'évolution de l'opération de confinement

de l'incendie en situation d'incertitude est construit. Une priorité d'I.A.D pour une politique optimale de lutte est établie. Sa détermination est indiquée et donnée rigoureusement. Des expérimentations numériques sont effectuées. Le degré d'efficacité de la ressource de lutte contre les incendies est analysé et est indiqué.

Des conclusions de ce travail entamé par nos soins ainsi que leurs perspectives sont présentées pour un travail ultérieur. Vous trouverez attachés l'article publié dans une revue, et les articles parus dans des colloques, symposium et congrès.



## 1. ORDONNANCEMENT À TÂCHES DÉTÉRIORANTES

### 1.1. Introduction

Dans un problème d'ordonnancement interviennent deux notions fondamentales : Les tâches et les ressources. Une ressource est un moyen, technique ou humain, dont la disponibilité limitée ou non est connue à priori. Une tâche est un travail dont la réalisation nécessite un certain nombre d'unités de temps, sa durée, et d'unités de chaque ressource. Les problèmes d'ordonnancement sont caractérisés par trois ensembles: L'ensemble de tâches  $J$ , l'ensemble de machines  $M$  et l'ensemble de ressources  $R$ .

« Ordonnancer, c'est attribuer les machines de l'ensemble  $M$  et des ressources de l'ensemble  $R$  aux tâches de l'ensemble  $J$  dans l'ordre de terminer l'exécution de toutes les tâches sous des contraintes imposées et d'atteindre un objectif ».

Il y a deux types de contraintes classiques. En tout instant chaque tâche est à exécuter au plus sur une machine et chaque machine n'est capable d'exécuter qu'une tâche à la fois. Pour de plus amples détails sur la problématique de l'ordonnancement, le lecteur peut se référer à l'ouvrage de PINEDO [11].

Dans les problèmes d'ordonnancement il est souvent supposé que les durées opératoires des tâches sont fixes et connues d'avance. Cette hypothèse s'avère très restrictive et peut correspondre à une image un peu simpliste de la réalité. En effet dans de nombreux cas, les conditions d'exécution changent, parfois jusqu'au moment de l'exécution des tâches, même durant leur exécution, ce qui suppose une résolution en partie différée.

On appelle «tâche détériorante» une tâche qui avec le temps, sa durée d'exécution augmente. De telle situation se pose par exemple 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.

## 1.2. Pourquoi l'ordonnancement à tâches détériorantes est d'intérêt ?

L'ordonnancement à tâches détériorantes trouve des applications dans plusieurs secteurs de recherches et d'activités. La littérature est abondante. Citons neuf exemples :

- Dans les banques [12].

Si le remboursement d'un prêt bancaire est retardé, on doit payer une amende ou une pénalité. L'acompte constitué de la valeur du prêt et des frais de pénalité augmente avec le temps.

- En sidérurgie [13].

Il faut exécuter une série d'opérations sur la matière en fusion, celle-ci se refroidit avec le temps, et la durée d'exécution des opérations s'allonge avec ce refroidissement.

- Dans la lutte contre les incendies [1].

Le temps ou l'effort nécessaire pour éteindre un incendie de forêt est fortement liés à l'instant de début de l'intervention des pompiers par des moyens aériens, terrestres, etc.

- Dans la lutte contre les fléaux médicaux [2].

Si une épidémie s'est répandue, tout retard dans le déclenchement de sa lutte signifie qu'il faut plus de temps pour la combattre.

- Chercher des personnes en danger [3].

Le temps nécessaire aux équipes de secours pour trouver les personnes disparues sous les conditions météorologiques difficiles ou sous une obscurité croissante s'allonge au fil du temps.

- Dans les services de soins intensifs [3].

L'exécution des protocoles médicaux sous la dégradation de l'état du patient augmente si celle-ci ne démarre pas au moment opportun.

- Dans les maintenances industrielles, des réparations et nettoyage [3].

Une procédure d'investigation doit commencer à une date prédéterminée  $t^*$  et dure un certain temps. Si Elle commence après  $t^*$ , sa durée s'allonge.

- La productivité d'une machine [14].

Une machine a le plus haut niveau de capacité de production au début de sa mise en service. Au fil du temps, elle perd de son efficacité. Une tâche exécutée plus tard requerra un temps de traitement plus long.

- Traitement d'un lot de pièces défectueuses [15].

Un lot de pièces produites avec des critères de qualité en dessous du standard défini est généralement mis en un processus de rebut et de récupération de matières premières pour augmenter les bénéfices de l'entreprise. Les pièces se détériorent pendant leur attente.

### 1.3. Qu'est ce qui nous motive à étudier ce problème ?

De notre recherche bibliographique sur l'ordonnancement à tâches détériorantes, un grand nombre d'articles sont parus. Dans leurs conclusions et perspectives, des problèmes ouverts ou des problèmes non encore étudiés sont relatés. Nous nous sommes inspirés de trois articles qui constitueront notre motivation à faire cette étude. Nous les décrivons ci-dessous.

RACHANIOTIS et PAPPIS [4] montrent que l'allocation d'une seule ressource de lutte pour contenir plusieurs incendies de forêts est un problème d'ordonnancement à une seule machine de tâches détériorantes. Le modèle détermine les durées des interventions de la ressource qui augmentent tant que les incendies ne sont pas contrôlés. Il prend en considération les différents phénomènes physico-chimiques de l'incendie qui se produisent à diverses échelles et éventuellement les caractéristiques de la ressource. A l'origine, il est théorique ou sur papiers. Il n'est pas basé sur des observations des incendies ou de simulations en laboratoire, mais plutôt sur les lois de la physique. L'objectif est de réduire au minimum la perte de valeur, monétaire ou d'une autre nature, dans les zones brûlées par les incendies. Cette fonction augmente quadratiquement avec la date d'achèvement des opérations de confinement. Un algorithme de type par séparation et évaluation est présenté dans [5] pour le résoudre exactement. L'allocation parallèle de plusieurs ressources de lutte est étudiée dans [6] et une heuristique de bonne performance est développée.

Les solutions proposées dans [4,6] s'avère être distantes des valeurs réelles des durées sur le terrain. Elles sont difficiles à utiliser pour un ordonnancement optimal. Si les paramètres d'un modèle ne peuvent être complètement prédits, il est clair que les démarches possibles pour la gestion des incertitudes sont utilisables.

Il constitue un intérêt d'investigation. Un cas intéressant est celui où une distribution de probabilité lui est associée. On se place donc dans un cadre stochastique. Elle sera abordée dans le troisième chapitre.

#### 1.4. De différents modèles d'ordonnancement à tâches détériorantes

Les premiers travaux sur l'ordonnancement à tâches détériorantes sont ceux de SHAFRANSKY [16] et MELNIKOV et SHAFRANSKY [17]. Ces problèmes ont reçu un intérêt considérable. La littérature s'est enrichie d'articles. [18,19,20,21,22,23,24] présentent des états de l'art sur la notion de détérioration de tâches.

La durée opératoire  $p_i$  d'une tâche détériorante  $J_i$  est une fonction croissante à valeurs positives dépendante de la date de début d'exécution de la tâche.  $p_i$  peut s'écrire sous forme  $g_i(s_i)$  où  $g_i$  est une fonction croissante à valeurs positives et  $s_i$  est l'instant de début d'exécution de  $J_i$ . Cette forme d'écriture n'est pas souvent indiquée pour déduire les variations de  $p_i$ .

Une deuxième façon d'écrire  $p_i$  est la forme translatée  $a_i + f_i(s_i)$ .  $a_i$  est une constante positive appelée «la durée intrinsèque» de la tâche  $J_i$ .  $f_i$  est une fonction croissante qui vérifie la condition  $f_i(0) = 0$ . Elle est appelée «la détérioration instantanée» de la tâche  $J_i$ .

La date de début d'exécution d'une tâche est une variable du temps.  $f_i(s_i)$  s'écrira  $f_i(t)$ . La variation temporelle de la durée opératoire de la tâche est anticipée avant le début de l'exécution. Dans la suite du chapitre 1 et dans le chapitre 2, nous considérons que l'ordonnancement déterministe.

La forme de la fonction de détérioration  $p_i$  est donnée par son domaine d'application. Les cas les plus fréquents sont les fonctions linéaires ou linéaires par morceaux. Les fonctions quadratiques, exponentielles, ou simplement non linéaires, sont très peu utilisées.

- Le premier modèle est celui des deux articles de SHAFRANSKY [16] et MELNIKOV et SHAFRANSKY [17]. La détérioration simple non linéaire  $p_i = a_i + f(t)$  est relatée.  $a_i$  est une constante strictement positive et  $f$  est une fonction quelconque

croissante à valeurs positives. Des cas généraux et quelques cas particuliers de cette détérioration ont été examinés.

- Si  $f_i(t) = b_i t$ , le modèle devient linéaire  $p_i = a_i + b_i t$ .  $a_i$  et  $b_i$  sont respectivement la durée intrinsèque et le «taux de détérioration» de la tâche  $J_i$ . Ce modèle est décrit dans les travaux de TANAEV et al. [25]. Aussi, des cas particuliers de cette fonction ont été considérés.
- MOSHEIOV [26] a étudié la détérioration  $p_i = 1 + b_i t$  où les durées opératoires intrinsèques des tâches sont toutes égales à une unité.
- MOSHEIOV dans [3] propose la fonction proportionnelle  $p_i = b_i t$ , dans laquelle les durées intrinsèques  $a_i$  sont nulles.
- KONONOV [27] considère la détérioration linéaire proportionnelle de type  $p_i = b_i(\alpha + \beta t)$ , avec  $\alpha$  et  $\beta$  sont des constantes positives.
- Les cas spéciaux  $b_i = b \neq 0$  ou  $a_i = a \neq 0$  à dates de disponibilité ou d'échéance, ont été traités par CHENG et DING dans [28, 29].
- KUO et YANG [30] se sont intéressés à un cas de détérioration non linéaire simple, de type  $p_i = a_i + \sum_{k=1}^m \lambda_k t^{r_k}$ , avec  $r_k \in [0; +\infty[$ .
- KONONOV [27] a proposé une autre forme de détérioration non linéaire simple, la détérioration proportionnelle à une fonction convexe ou concave  $h$  notée  $p_i = b_i h(t)$ .
- Pour les détériorations non linéaires, la fonction polynomiale  $p_i = \sum_{k=0}^m a_{ik} t^k$  est proposée par GUPTA et GUPTA [13].
- Les fonctions exponentielles de type  $p_i = a_i e^{b_i t}$  sont introduites initialement par ALIDAEI [31] et BROWNE et YECHIALI [32].
- Les fonctions  $p_i = a_i 2^{b_i(t-r_i)}$  sont proposées tardivement par JANIÁK et KOVALYOV [33].
- KUNNATHUR et GUPTA [1] ont proposé la détérioration  $p_i = a_i + \max\{0; b_i(t - d_i)\}$ .

– SUNDARARAGHAVAN et KUNNATHUR [2] et MOSHEIOV [34] ont introduis un

$$\text{autre modèle de type } p_i = \begin{cases} a_i & \text{si } t \leq d_i \\ a_i + b_i & \text{sinon} \end{cases} .$$

– Dans le même article de [34], une extension de ce modèle est indiquée, à savoir

$$\text{la fonction donnée par } p_i^k = \begin{cases} a_i^k & \text{si } d_i^{k-1} \leq t \leq d_i^k \\ a_i^{m+1} & \text{si } t > d_i^k \end{cases} ,$$

Où  $d_i^0 = 0$ ,  $a_i^1 < a_i^2 < \dots < a_i^{m+1}$  et  $d_i^1 < d_i^2 < \dots < d_i^{m+1}$ .

– KUBIAK et VAN DE VELDE [35] ont considère la fonction

$$p_i = \begin{cases} a_i & \text{si } t \leq d \\ a_i + b_i(t-d) & \text{si } d < t < D \\ a_i + b_i(D-d) & \text{si } t \geq D \end{cases} .$$

– CAI et al. [36] proposent d'étudier le modèle noté  $p_i = a_i + b_i f(t, t_0)$ , avec  $f(t, t_0) = 0$  si  $t \leq t_0$  et  $f(t, t_0)$  est croissante si  $t > t_0$ .

### 1.5. Ordonnancer quand la durée d'exécution de la tâche augmente et sa valeur diminue avec le temps

Jusqu'ici, seule les durées d'exécution des tâches sont variables. Un autre cas intéressant est considéré dans [37]. Les tâches ont des valeurs positives qui se détériorent (diminuent) au fil du temps. Le critère pertinente à maximiser est la somme des valeurs résiduelles des tâches après l'achèvement de leur exécution. Deux propriétés, l'une reliant ce critère à celui de minimisation de la somme des dates de fin des tâches et l'autre à la somme pondérée des dates de fin d'exécution ont été déduites dans [38] :

$$\max \sum_{i=1}^n V_i = \sum_{i=1}^n V_{0,i} - a \cdot \min \sum_{i=1}^n C_i \quad \text{et} \quad \max \sum_{i=1}^n V_i = \sum_{i=1}^n V_{0,i} - \min \sum_{i=1}^n a_i C_i ,$$

où  $V_{0,i}$  est la valeur attribuée à la tâche  $J_i$ ,

$V_i$  est la valeur résiduelle après la fin d'exécution de  $J_i$ ,

$a_i$  est le poids associé à  $J_i$ . S'il est identique pour toutes les tâches, il est noté par  $a$ .

## 2. LE CONCEPT DE TÂCHES DÉTÉRIORANTES DANS LA LUTTE CONTRE LES INCENDIES DE FORÊTS

### 2.1. Introduction

Le thème développé ci-dessous est la lutte contre les incendies de forêts. Après un rappel sur les notions de «pyrologie forestière», les différents modèles de propagation d'un incendie sont présentés. Il est également question des différentes mesures prises dans la gestion des incendies de forêts. La théorie d'ordonnement est d'un apport dans la construction d'une stratégie optimale d'extinction des incendies. Une attention est portée sur la construction du modèle de RACHANIOTIS et PAPPIS [4]. Ce modèle détermine les durées des interventions de la ressource de lutte qui augmentent tant que les incendies ne sont pas maîtrisés. L'objectif est de réduire au minimum la perte de valeur, monétaire ou d'une autre nature, des zones brûlées par les incendies. Cette fonction augmente quadratiquement avec la date d'achèvement des opérations de confinement. Un algorithme de type par séparation et évaluation proposé dans [5] pour le résoudre exactement et optimalement est présenté. Le problème à ressources parallèles étudié dans [6] et l'heuristique développée sont aussi exposés.

### 2.2. Notions de Pyrologie Forestière

Les incendies exercent une importante fonction dans la préservation de certains écosystèmes forestiers formés de réseaux d'arbres, de végétaux, d'animaux, de champignons et de bactéries. Ils constituent aussi une grave menace de dévastation pour de nombreuses forêts et pour la biodiversité qu'elles renferment.

La pyrologie forestière est l'étude scientifique des incendies de forêts et de leurs propriétés. Elle décrit le processus de la combustion, les caractéristiques des incendies forestiers ainsi que les facteurs qui influencent leurs origines et leurs développements.

Par forêt, il faut entendre les arbres, les milieux naturels, c'est l'ensemble des formations végétales sur des terrains dégradés ou non, dites de substitution.

Un incendie de forêt est la destruction d'une surface d'au moins un hectare d'un seul tenant. Dans la suite, nous ne ferons pas de restriction sur la taille des zones brûlées.

Feu et incendie ne sont pas synonymes. Un feu est la combustion contrôlée par des pompiers ou forestiers. Un incendie est une combustion qui se développe sans contrôle, dans le temps et dans l'espace.

Le comportement d'un incendie forestier est régi par un certain nombre de facteurs dont les influences s'opposent ou s'additionnent. Les facteurs déterminants sont les facteurs météorologiques et climatiques, la topographie ainsi que les combustibles. Ces diverses variables constituent ce qu'on appellera l'environnement de l'incendie.

Certains facteurs météorologiques influencent grandement le comportement de l'incendie. Ces facteurs sont la température, le vent, l'humidité relative, les précipitations et la stabilité atmosphérique.

La topographie joue également un rôle de premier plan dans le comportement des incendies de forêts par son influence sur la forme, la direction et la vitesse de propagation. En général, l'influence de la topographie s'exerce selon l'inclinaison des pentes, leur exposition au soleil, l'altitude, la forme du terrain et la structure du sol.

Les combustibles se définissent comme étant toute matière organique, vivante ou non, qui se trouve en profondeur, en surface ou dans les airs et qui peut prendre feu et brûler. On trouve des combinaisons quasi infinies de type, de quantité, de grosseur, de forme et de disposition de combustibles.

Si un incendie de forêt a de fortes probabilités de se déclarer ou de se propager en un lieu donné, on dit alors que «l'aléa» incendie de forêt est élevé. S'il peut causer des dommages sur certains biens du territoire concerné on parle alors de «vulnérabilité» de ces enjeux.

On appelle «risque», la probabilité d'apparition du phénomène aléa. Le risque d'incendie varie dans le temps en fonction des facteurs météorologiques, de la topographie du sol forestier et des caractéristiques des combustibles.

L'évolution d'un incendie forestier est une fonction de nombreux paramètres. Ils sont récapitulés dans le tableau suivant :



Paramètre	Unité de mesure
Taux de propagation <i>ROS</i> (Rate Of Spread)	mètre / minute
Direction de propagation	0-360 degrés
Intensité frontale de l'incendie	Kilo Joule / minute
Hauteur de la flamme	Mètre
Intensité de réaction	Kilo Joule / mètre <sup>2</sup>
Temps de tir à un point précis	Minute
Quantité totale d'énergie libérée par unité de surface	Kilo Joule / mètre <sup>2</sup>
Périmètre de l'incendie	Mètre

Tableau 2.1: Paramètres utilisés pour décrire le comportement d'un incendie.

### 2.3. Quatre modèles de propagation d'un incendie de forêt

Certaines méthodes d'évaluation du risque d'incendie utilisent des modèles du taux de propagation de l'incendie. Ils permettent de prévoir le comportement de l'incendie pour positionner des moyens de lutte à des endroits stratégiques. Ils appartiennent à trois catégories, empirique, semi-empirique, et théorique [39,40,41]. Ces approches se différencient entre elles par la façon dont le taux de propagation de l'incendie *ROS* est déterminé et calculé.

- Les modèles empiriques sont établis sur la base de données expérimentales, souvent de terrain, parfois de laboratoire, relie statistiquement le taux de propagation de l'incendie aux conditions environnementales, vent et pente, et au type de combustible pour allumer l'incendie de forêt.
- Les semi-empirique utilisent des outils théoriques ainsi que des résultats expérimentaux pour évaluer le taux de propagation de l'incendie. Ils se basent sur la loi de conservation d'énergie et sur des mesures prises en laboratoire.
- Quand aux modèles théoriques, ils utilisent les équations de la mécanique des fluides, les lois de la thermodynamique des processus irréversibles et des bilans énergétiques d'équations de réactions chimiques.

Les gestionnaires des incendies utilisent des logiciels pour prédire la progression des incendies dans la forêt. Plusieurs produits ont été développés. Les plus connus sont :

- Le système canadien de prévision du comportement des incendies de forêt FBP, de l'anglais Fire Behavior Prediction [42], basé sur une approche empirique.

- Le système américain dénommé BEHAVE [43] et le système FARSITE, Fire Area Simulator [44] utilisent un taux de propagation fourni par des modèles semi-empiriques.
- Les simulateurs FIRETEC [45] et WFDS, Wildland Fire Dynamics Simulator [46]. Ils sont établis sur la base de modèles physiques.

#### 2.4. Les mesures prises dans la gestion des incendies de forêts

La gestion des incendies de forêts repose sur une stratégie globale de prévention, de prévision et de lutte. La déclaration d'un incendie marque l'échec des dispositifs de prévention. La qualité du système de prévision est déterminante pour le succès des opérations de lutte. La mobilisation préventive des moyens d'intervention est d'une extrême importance pour réduire le plus possible les délais d'action. La réponse dépend de l'importance prise par l'incendie. Les moyens engagés augmentent avec la taille et la virulence de l'incendie. L'analyse des équipements existants et l'indication de leur degré d'efficacité peuvent orienter le gestionnaire dans l'élaboration d'un plan plus efficace de lutte.

Dans certains cas, les problèmes de la gestion des incendies de forêt relèvent des techniques de la recherche opérationnelle [47,48, 49, 50]. Les recherches réalisées dans ce domaine sont de type planification à long terme. La prise de décision à court terme concernent l'attribution des ressources à la construction de lignes pare-feu pour qu'un incendie de forêt soit pris en charge à l'intérieur d'une limite de temps déterminée. Nous nous sommes attachés à présenter l'essentiels des études réalisées dans [4,5,6]. Ils constituent la motivation de faire ce travail.

#### 2.5. Un modèle par tâches détériorantes pour la lutte contre les incendies forestiers

Dans cette section, nous présentons le problème étudié dans [4], ses notations, ses hypothèses théoriques et terminologies, et nous donnons des détails sur la construction du modèle établi.

##### 2.5.1. Présentation du problème

Un espace forestier est généralement découpé en plusieurs zones suivant deux critères, les caractéristiques du milieu et les moyens de lutte disponibles. A chaque zone une valeur initiale ou un poids est attribuée reflétant son urgence dans l'intervention par rapport aux autres zones. L'importance du type de combustible, la proximité de la zone forestière de la population, zone militaire, importance historique du site, l'impacte de l'incendie sur la formation animale et végétale, les conséquences

économiques, etc, constituent des exemples de cette urgence. Cette valeur peut être monétaire ou écologique, par exemple, et peut être calculée par des experts forestiers [47].

L'efficacité de la lutte contre un incendie de forêt dépend, en grande partie, de la rapidité de l'engagement des dispositifs d'intervention. La théorie d'ordonnement est d'un apport dans la construction d'une stratégie optimale d'extinction des incendies.

Respectivement à l'organisme de gestion des incendies forestiers et l'ensemble d'incendies sont associés l'atelier et les tâches. L'atelier est décrit par le nombre et les types d'équipements de lutte, correspondant aux machines, préalablement organisés par la prévision. Les sapeurs-pompiers, les camions citernes, les hélicoptères et les bombardiers aériens sont des exemples de ces moyens. Les ressources de lutte seront appelées "processeurs".

Les incendies de forêts peuvent se déclarer simultanément ou à des instants différents. Nous nous limitons au cas des incendies déclarés simultanément, phénomène qui apparaît assez souvent dans les pays méditerranéens [51].

Le problème d'ordonnement considéré est à une seule machine. Certains processeurs peuvent opérer séparément ou ils peuvent être considérés comme un seul processeur quand ils doivent fonctionner collectivement.

Les caractéristiques d'un incendie de forêt sont la «durée opératoire», la période de temps pour que le processeur puisse contrôler un incendie, la «date de début d'exécution», l'instant auquel le processeur débute l'exécution des opérations pour contenir un incendie, et une «date d'échéance». Cette dernière date traduit la fin de la phase d'accélération de l'incendie. Dépasser cet instant, l'incendie peut s'échapper aux efforts de la première intervention.

La durée opératoire d'un incendie s'accroît (se détériore) s'il y a un retard dans le début des opérations de confinement, tandis que la valeur de la zone brûlée par l'incendie, diminue dans le temps avec un taux qui, à son tour, dépend de l'importance relative de la zone.

### 2.5.2. Notations et terminologies

Supposons que  $n$  incendies de forêts sont déclarés simultanément, détectés et signalés, et un processeur est attribué à leur confinement. Notons par :

- $J_{[i]}$  est l'incendie placé dans la  $i^{\text{ème}}$  position d'une séquence de confinement.

- $T_{0,[1]}$  (en  $h$  (heure)) est le temps nécessaire pour le processeur de se déplacer de la base où il est stationné à l'endroit du premier incendie choisi pour être confiné.
- $T_{[i],[i+1]}$  (en  $h$ ),  $i = 1, 2, \dots, n - 1$ , est la période de temps pour que le processeur puisse se rendre au point d'eau le plus proche pour recharger après l'achèvement de ses opérations pour l'incendie  $J_{[i]}$  et passer au front de l'incendie  $J_{[i+1]}$ .
- $p_{[i]}(t)$  (en  $h$ ) est la durée opératoire de l'incendie  $J_{[i]}$  si le confinement de  $J_{[i]}$  est débutée à l'instant  $t$ .
- $C_{[i]}$  (en  $h$ ) est la date de fin d'exécution de  $J_{[i]}$ . Elle représente la date de fin des efforts de confinement pour  $J_{[i]}$ , c'est-à-dire l'incendie s'éteint.

Elle est donnée par la relation :

$$C_{[i]} = \sum_{j=1}^i p_{[j]}(s_{[j]}) + \sum_{j=1}^i T_{[j-1],[j]} \quad (2.1)$$

où  $s_{[j]}$  est la date de début d'exécution de  $J_{[j]}$ , et  $T_{[0],[1]} = T_{0,[1]}$ .

- $d_{[i]}$  (en  $h$ ) est la date d'échéance de  $J_{[i]}$ .
- $V_{[i]0}$  (monétaire ou d'une autre nature) est la valeur initiale attribuée à une zone d'indice  $i$  avant qu'elle sera touchée par un incendie  $J_{[i]}$ .
- $V_{[i]}$  est la valeur résiduelle d'une zone brûlée par un incendie  $J_{[i]}$ . Dans [4], cette valeur diminue de façon quadratique avec la date de fin d'exécution  $C_{[i]}$ . Elle est donnée par la relation :

$$V_{[i]} = V_{[i]0} - a_{[i]} C_{[i]}^2$$

et elle doit être positive.  $a_{[i]}$  (en unité de  $V_{[i]}$  par  $t^2$ ) est une constante à estimer empiriquement ou des techniques de la recherche opérationnelle peuvent être employées [52].

- $Q$  (en  $L$  (Litre)) est la capacité de transport du processeur.
- $\rho$  (en  $L \times h^{-1}$ ) est le taux d'application du processeur.

- Au cours de son travail pour contenir  $J_{[i]}$ , le processeur peut interrompre ses opérations autant de fois que nécessaire.  $\tau_{[i]}$  (en  $h$ ) est la durée nécessaire au processeur pour aller au point d'eau le plus proche pour recharger.
- $E_{[i]}(t)$  (en  $m^2$  ( $m$  est le mètre)) est la superficie brûlée par l'incendie  $J_{[i]}$  si le processeur commence le confinement de  $J_{[i]}$  à l'instant  $t$ .
- $\omega_{[i]}(t)$  (en  $L \times m^2$ ) est la quantité relative d'eau nécessaire pour éteindre  $J_{[i]}$ .

Dans [4], le problème est de déterminer un ordonnancement déterministe optimal  $J_{optimal} = \langle J_{[1]}, J_{[2]}, \dots, J_{[n]} \rangle$  qui maximiser la valeur totale résiduelle des zones brûlées après l'achèvement des opération de confinement,  $\max \sum_{i=1}^n V_{[i]}$ , ou d'une manière équivalente, minimise la perte de valeur totale,  $\min \sum_{i=1}^n a_i C_{[i]}^2$ .

Adopter la notation des trois champs [53], le problème est dénoté : par  $1 | V_i | \sum_{i=1}^n V_i$ .

Si les incendies  $J_{[i]}$  sont déclarés à des moments différents  $r_{[i]}$ , la valeur résiduelle  $V_{[i]}$  a la forme :

$$V_{[i]} = V_{[i]0} - a_{[i]} \left( C_{[i]} - r_{[i]} \right)^2 \geq 0.$$

### 2.5.3. Les hypothèses du modèle de RACHANIOTIS et PAPPIS [4]

Dans la construction du modèle de RACHANIOTIS et PAPPIS [4], les hypothèses suivantes sont faites:

- Les  $n$  incendies sont repérés instantanément au moment de leur déclaration, et ils présentent tous un danger potentiel.
- Les  $n$  incendies évoluent indépendamment et aucune contrainte n'est supposée qui impose l'ordre dans lequel les incendies doivent être supprimés.
- Les  $n$  incendies se propagent s'il ya un retard dans les efforts de lutte entrepris.
- Au moment où le processeur commence le confinement d'un incendie, celui-ci s'arrête de s'étendre.

- Les  $n$  incendies sont déclarés dans des zones différentes, étant donné qu'elles ne sont pas vastes.
- Elles sont accessibles à tout moment et il n'y a pas de contraintes, par exemple, le processeur ne peut pas opérer pendant la nuit.
- Chaque zone est supposée contenir un combustible homogène.
- La vitesse et la direction du vent sont uniformes.
- Le processeur a une capacité de transport limitée et un taux d'application constant, utilise l'eau comme un moyen de lutte.
- Le processeur n'est pas exigé d'éteindre un incendie lorsqu'il est occupé à en contenir un autre, qui s'interprète que la préemption n'est pas autorisée.
- La durée opératoire d'un incendie, la superficie brûlée et la quantité d'eau utilisée se détériorent (augmentent) à un taux qui dépend du temps écoulé depuis la déclaration de l'incendie. Elles dépendent aussi de la topographie des terres forestières, la végétation, les conditions météorologiques et l'efficacité du processeur (capacité de transport et taux d'application).
- Les périodes de déplacement du processeur pour contenir un incendie prescrit et les périodes d'interruption pour recharger sont considérées comme des périodes de configuration, comme ils le seraient dans un environnement de fabrication typique. Ces durées sont considérées comme indépendantes du temps.

#### 2.5.4. Construction du modèle de RACHANIOTIS et PAPPIS [4]

Généralement, un incendie forestier passe respectivement par quatre phases, à savoir, la phase d'allumage, la phase d'accélération, la phase d'équilibre et la phase d'extinction.

Pour l'organisme de lutte contre les incendies de forêt, la phase d'accélération est la plus importante car dans de nombreux cas, elle représente la seule période pendant laquelle les efforts de lutte pourraient être efficaces [54].

Pour évaluer cette phase, MCALPINE et WAKIMOTO [55] ont mené une série d'expérimentation en laboratoire, sous conditions contrôlées de combustion et de vitesse du vent. Le résultat analysé est la distance  $D$  parcourue par l'incendie en fonction du temps  $t$  calculé depuis le départ de l'incendie. Une équation prédictive produite a la forme :

$$D(t) = A t^B . \quad (2.2)$$

Les constantes  $A$  et  $B$  sont estimés d'avoir les expressions suivantes :

$$A = 0.628 \times ROS_{eq}^{1.188} \quad \text{et} \quad B = 1.204 + 0.740 \times \left(1 - e^{(-0.319 \times U)}\right) \quad (2.3)$$

où  $ROS_{eq}$  est le taux de propagation d'équilibre et  $U$  est la vitesse du vent.

La dérivée de la fonction  $D$ , donnée par l'équation (2.2), par rapport à  $t$  quantifie la variation de  $ROS$  pendant la phase d'accélération de l'incendie. Cette dérivée notée  $ROS_{ac}$  prend la forme :

$$ROS_{ac}(t) = A B t^{B-1} . \quad (2.4)$$

La figure ci-après illustre la convergence de la courbe représentative de la fonction  $ROS_{ac}$  vers la droite d'équilibre de  $ROS$ .

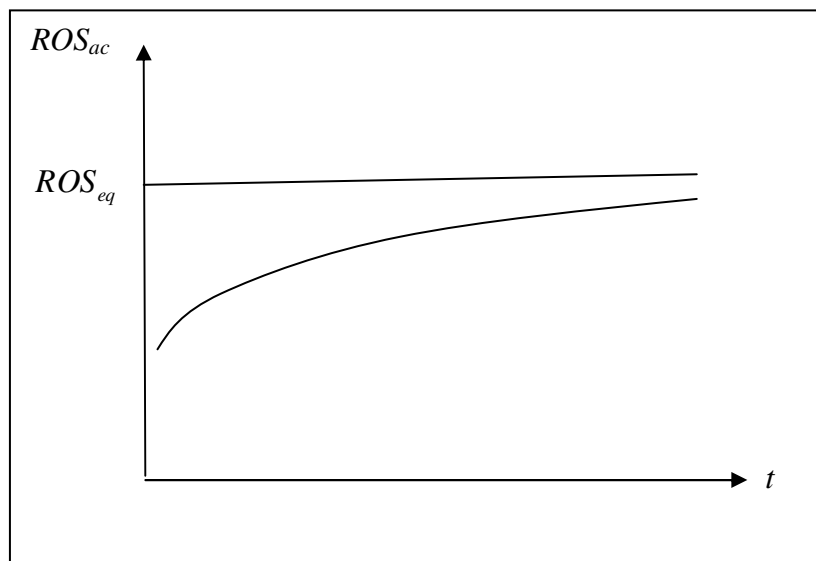


Figure 2.1 : Allure exponentielle du taux de propagation  $ROS_{ac}$

Les deux méthodes de prévision du comportement des incendies, FBP et BEHAVE, ont été présentées dans la deuxième section de ce chapitre. Au risque de se répéter, dans l'élaboration de  $ROS_{eq}$ , les systèmes FBP et BEHAVE utilisent des approches différentes, respectivement empirique et semi-empirique. Des expérimentations ont été menées dans [56] pour comparer le  $ROS$  d'équilibre observé, noté  $ROS_{eq}^{observé}$ , aux deux estimations  $ROS_{eq}^{BEHAVE}$  et  $ROS_{eq}^{FBP}$ , calculées respectivement par BEHAVE et FBP. Le système BEHAVE avait tendance à sous-prédire le  $ROS_{eq}^{observé}$  tandis que FBP avait tendance à sur-prédire le  $ROS_{eq}^{observé}$ .

On peut alors déduire qu'à l'instant  $t > 0$ , on a

$$ROS_{ac}^{FBP}(t) = ROS_{ac}^{BEHAVE}(t) + c(t) \quad (2.5)$$

où  $ROS_{ac}^{FBP}(t)$  est le  $ROS$  basé sur le système FBP et  $ROS_{ac}^{BEHAVE}(t)$  est celui basé sur BEHAVE.  $c(t) \geq 0$  est la différence entre les deux  $ROS$ .

Si l'aléa d'un incendie est faible, il nous faut environ 0.5 L d'eau pour éteindre une surface d'incendie de 1  $m^2$ . La quantité est estimée de 2 à 3.5 L /  $m^2$  pour les incendies enflammés dans les zones de végétation basse, et d'environ 15 L/ $m^2$  dans des conditions de fort vent [57]. Tous dépend donc des caractéristiques du milieu, les conditions météorologiques, le temps de début des efforts d'extinction et de l'intensité de l'incendie.

KATSANOS [58] a formulé une équation empirique en fonction de  $ROS_{ac}^{FBP}(t)$  pour calculer la quantité d'eau  $\omega_{[i]}(t)$  (en  $L \times m^2$ ) nécessaire pour contrôler un incendie  $J_{[i]}$ . Elle a la forme suivante :

$$\omega_{[i]}(t) = 1.2 \sqrt{ROS_{ac}^{FBP}(t)} \quad (2.6)$$

Utiliser les relations (2.5) et (2.6),  $\omega_{[i]}(t)$  peut s'écrire en fonction de  $ROS_{ac}^{BEHAVE}(t)$  comme suit :

$$\omega_{[i]}(t) = 1.2 \sqrt{ROS_{ac}^{BEHAVE}(t) + c(t)} \quad (2.7)$$

substituant (2.4) dans (2.7), nous obtenons :

$$\omega_{[i]}(t) = 1.2 \sqrt{A_{[i]} \times B_{[i]} \times t^{B_{[i]}-1} + c_{[i]}(t)} \quad (2.8)$$

Il est à noter que dans (2.8),  $ROS_{eq}^{BEHAVE}$  est utilisé pour déduire la valeur de  $A_{[i]}$  exprimée dans (2.3).

Le temps nécessaire,  $q_{[i]}(t)$ , au processeur pour contenir à l'instant  $t$  une surface de 1  $m^2$  enflammée par l'incendie  $J_{[i]}$  en tenant compte du taux d'application  $\rho$  du processeur détiert de l'équation :

$$q_{[i]}(t) = \frac{\omega_{[i]}(t)}{\rho} \quad (2.9)$$

Remplacer (2.8) dans (2.9), il résulte :



$$q_{[i]}(t) = \frac{1.2}{\rho} \sqrt{A_{[i]} \times B_{[i]} \times t^{B_{[i]}-1} + c_{[i]}(t)} \quad (2.10)$$

Pour la superficie totale brûlée par  $J_{[i]}$ , il est établi que

$$p_{[i]}(t) = E_{[i]}(t) \times q_{[i]}(t) + \text{int} \left( \frac{E_{[i]}(t) \times \omega_{[i]}(t)}{Q} \right) \times \tau_{[i]} \quad (2.11)$$

où "int" est la partie entière d'un nombre.

RACHANIOTIS et PAPPIS [4], montrent que la superficie brûlée par l'incendie  $J_{[i]}$  à l'instant  $t$  obéit à l'équation :

$$E_{[i]}(t) = \frac{\pi}{\psi_{[i]}} \left( \frac{A_{[i]}}{B_{[i]}+1} \times t^{B_{[i]}+1} + c_{[i]}(t) \times t \right)^2 \quad (2.12)$$

où  $\psi_{[i]}$  est une fonction de la vitesse du vent  $U_{[i]}$  (en  $Km \times h^{-1}$  ( $Km$  est le Kilomètre)).

Elle est donnée dans [56] par la formule:

$$\psi_{[i]} = 0.936 \times e^{0.07127 \times U_{[i]}} + 0.461 \times e^{-0.043 \times U_{[i]}} \quad (2.13)$$

Pour une preuve pour (2.12) voir Annexe C.

Substituons les équations (2.10) et (2.12) dans l'équation (2.11). Il découle que:

$$p_{[i]}(t) = \frac{1.2 \times \pi}{\psi_{[i]} \times \rho} \times \left( \frac{A_{[i]}}{B_{[i]}+1} \times t^{B_{[i]}+1} + c_{[i]}(t) \times t \right)^2 \times \sqrt{A_{[i]} \times B_{[i]} \times t^{B_{[i]}-1} + c_{[i]}(t)} + \text{int} \left( \frac{1.2 \times \pi}{\psi_{[i]} \times Q} \times \left( \frac{A_{[i]}}{B_{[i]}+1} \times t^{B_{[i]}+1} + c_{[i]}(t) \times t \right)^2 \times \sqrt{A_{[i]} \times B_{[i]} \times t^{B_{[i]}-1} + c_{[i]}(t)} \right) \tau_{[i]} \quad (2.14)$$

Dans le cas des incendies déclarés à des instants différents, il est déduit que

$$p_{[i]}(t) = \frac{1.2 \times \pi}{\psi_{[i]} \times \rho} \times \left( \frac{A_{[i]}}{B_{[i]}+1} \times (t - r_{[i]})^{B_{[i]}+1} + c_{[i]}(t) \times (t - r_{[i]}) \right)^2 \times \sqrt{A_{[i]} \times B_{[i]} \times (t - r_{[i]})^{B_{[i]}-1} + c_{[i]}(t)} + \text{int} \left( \frac{1.2 \times \pi}{\psi_{[i]} \times Q} \times \left( \frac{A_{[i]}}{B_{[i]}+1} \times (t - r_{[i]})^{B_{[i]}+1} + c_{[i]}(t) \times (t - r_{[i]}) \right)^2 \times \sqrt{A_{[i]} \times B_{[i]} \times (t - r_{[i]})^{B_{[i]}-1} + c_{[i]}(t)} \right) \tau_{[i]}$$

## 2.6. Approche de solution [5]

En présence de cette forme complexe de détérioration, équation (2.14), le problème d'ordonnancement devient généralement NP-difficile. Un seul processeur de lutte ne peut contenir qu'un nombre relativement petit d'incendies. La principale méthode de résolution est la procédure par séparation et évaluation.

### 2.6.1. Rappels sur les procédures par séparation et évaluation

Le principe d'une procédure par séparation et évaluation (PSE) est de construire progressivement l'ordonnancement. Classiquement, une nouvelle tâche est ajoutée à la solution partielle connue, décomposant ainsi le problème initial en sous problèmes de taille plus petite. L'espace de recherche est représenté par un arbre où les nœuds symbolisent les ordonnancements partiels, et les nœuds fils sont les sous problèmes à résoudre. Les feuilles de l'arbre sont les solutions admissibles. L'évaluation d'un nœud permet de connaître s'il peut conduire à une solution optimale. Une borne supérieure est calculée (cas de la maximisation d'un critère) et est comparée à la meilleure solution connue. Si cette borne est inférieure ou meilleure solution, la séparation du nœud est inutile (construction des nœuds fils) car le critère ne pourra pas être amélioré. il existe plusieurs stratégies pour la sélection des nœuds à séparer. Les plus connues sont «la profondeur d'abord», «la largeur d'abord» et «la manière progressive».

### 2.6.2. Borne supérieure

Un nœud représente un ordonnancement partiel  $\langle J_{[1]}, J_{[2]}, \dots, J_{[r]} \rangle$ . Pour chaque nœud,  $y$  désigne l'ensemble des incendies non élus pour être confiné. Une borne supérieure est obtenue comme :

$$\begin{aligned}
 UB = & \sum_{i=1}^r V_{[i]0} - \sum_{i=1}^r a_{[i]} C_{[i]}^2 + \sum_{j=1}^{|y|} V_{j0} \\
 & - \sum_{j=1}^{|y|} a_j \left[ C_{[r]} + p_j \left( C_{[r]} + \min_{j=1, \dots, |y|} T_{[r],j} \right) + \min_{j=1, \dots, |y|} T_{[r],j} \right]^2 \\
 & - \min_{j=1, \dots, |y|} a_j \frac{(|y|-1)|y|}{2} \left( \min_{j=1, \dots, |y|}^2 p_j \left( C_{[r]} + \min_{j=1, \dots, |y|} T_{[r],j} \right) + \min_{j=1, \dots, |y|}^2 T_{j-1,j} \right).
 \end{aligned}$$

### 2.6.3. Règle de séparation

L'exploration se fait au profondeur. Un nœud séparé est celui qui a le plus grand borne supérieur.

#### 2.6.4. Borne inférieure

Une heuristique est confectionnée pour extraire la meilleure solution connue. Elle est exposée ci-après :

Algorithme 2.1: Calcul de la borne inférieure initiale

Etape 1:

- Arranger les incendies dans un ordre croissant des ratios  $\frac{p_i + T_{0,i}}{a_i}$ .
- Calculer la valeur totale  $LB_1$  de cette séquence.

Etape 2: Elle comporte trois sous-étapes.

Sous étape 1: Elle construit une séquence initiale.

- Placer l'incendie ayant le plus grand ratio  $\frac{V_i}{C_i}$  à la première position de la séquence.
- Retirer cet incendie de l'ensemble des incendies.
- Répéter la procédure à partir de l'instant  $t = C_{[i]}$ .
- La procédure se termine lorsque tous les incendies sont ordonnancés.

Sous-étapes 2: Elle améliore la séquence générée.

échanger deux incendies consécutifs

- D'abord en avant, de la première à la dernière position.
- Puis en arrière, de la dernière à la première position.
- Calculer la valeur totale  $LB_2$  de cette séquence.

La borne inférieure initiale utilisée dans l'algorithme PSE est  $LB_{in} = \max \{LB_1, LB_2\}$ .

#### 2.6.5. Règle de priorité

La propriété suivante donne les conditions sous lesquelles la séparation d'un nœud actif est inutile.

Soient  $\langle J_{S1} \rangle$  et  $\langle J_{S2} \rangle$  deux ordonnancement partiels. Si

- a)  $\langle J_{S1} \rangle$  et  $\langle J_{S2} \rangle$  ont les mêmes ensembles d'indices d'incendies avec  $|J_{S1}| = |J_{S2}| = r \geq 3$ ,
- b) L'incendie  $J_{[r]}$  ordonnancé dans la dernière position de  $\langle J_{S1} \rangle$  et  $\langle J_{S2} \rangle$  est le même,
- c)  $\sum_{J_i \in J_{S1}} a_i C_i^2 \leq \sum_{J_i \in J_{S2}} a_i C_i^2$ ,
- d)  $C_{S1,[r]} \leq C_{S2,[r]}$  (la date de fin d'exécution de  $J_{[r]}$  dans  $\langle J_{S1} \rangle$  est inférieur à celle de  $J_{[r]}$  dans  $\langle J_{S2} \rangle$ )

alors  $\langle J_{S1} \rangle$  domine  $\langle J_{S2} \rangle$  et le nœud qui symbolise  $\langle J_{S2} \rangle$  est éliminé de l'ensemble des nœuds actifs.

### 2.7. Problème à processeurs parallèles

Le problème dans lequel chaque incendie forestier n'est constitué que d'une opération qui peut être réalisée par une ou plusieurs processeurs de lutte est étudié par PAPPIS et RACHANIOTIS [6]. Les processeurs sont de  $K$  différentes catégories. Chaque catégorie  $j$  contient  $M_j$  processeurs identiques. Les  $n$  incendies,  $n \geq 2$ , sont déclarés en  $L$ ,  $L \leq n$ , différents instants  $0 = r_1 < r_2 < \dots < r_L$ . Un modèle de tâches détériorantes est établi. Il a pour équation :

$$p(\underline{s}_i(t), t) = \frac{Area_i(t)}{\sum_{j=1}^K s_{ij}(t) \rho_{ij}(t)}.$$

Dans cette expression :

- $\underline{s}_i(t) = (s_{i1}(t), s_{i2}(t), \dots, s_{iK}(t))$  est le vecteur d'affectation des processeurs de chaque catégorie pour qu'un incendie  $J_i$  soit maîtrisé à l'instant  $t$  (en *min*). C'est la variable de décision du problème.
- $p(\underline{s}_i(t), t)$  est la durée opératoire de  $J_i$ , si l'affectation  $\underline{s}_i(t)$  est utilisée pour contenir  $J_i$  à l'instant  $t$ .
- $Area_i(t)$  (en  $m^2$ ) est l'aire de la bande elliptique de  $J_i$  qui doit être créée afin de contenir  $J_i$  à l'instant  $t$ .  $Area_i(t)$  est exprimée dans [6] par l'équation

$$Area_i(t) = \alpha_i t^{\beta_i} + \gamma_i t + \varepsilon_i.$$

Les constantes  $\alpha_i, \beta_i, \gamma_i, \varepsilon_i$  dépendent des caractéristiques de la zone touchée par l'incendie  $J_i$  et des conditions météorologiques.

- $\rho_{ij}(t)$  (en  $m^2 / \text{min}$ ) est le taux d'application du processeur de catégorie  $j$  employée pour contenir à l'instant  $t$  l'incendie  $J_i$ .
- La hauteur de l'incendie  $J_i$  à l'instant  $t$ , notée  $h_i(t)$ , est définie comme étant l'aire d'une bande elliptique de  $J_i$ , qui n'est pas encore contrôlée à l'instant  $t$ , divisée par le taux maximum de confinement. Formellement

$$h_i(t) = \frac{Area_i(t_{0i}) - \int_{t_{0i}}^t P(s_i(\tau), \tau) \sum_{j=1}^K s_{ij}(\tau) \rho_{ij}(\tau) d\tau}{\sum_{j=1}^K M_j \rho_{ij}(t)},$$

où  $t_{0i}$  (en min),  $t_{0i} > 0$ , est l'instance de début des efforts de suppression pour  $J_i$ .

Un algorithme heuristique synchrone en temps réel est proposé. Il est exposé ci-après.

Algorithme 2.2: Heuristique pour une affectation parallèle des processeurs

Étape 1: Arranger les incendies dans l'ordre décroissant des ratios  $\frac{a_i}{h_i(t)}$ .

Les intervalles  $[r_k, r_{k+1}]$ ,  $k = 1, \dots, L$ , sont considérés consécutivement.

Les sous-intervalles sont considérés là où les ratios  $\frac{a_i}{h_i(t)}$  sont constants.

Étape 2: Affectation des ressources pour confiner les incendies déjà déclarés.

Les incendies dont les ratios sont les plus élevés sont privilégiés. Les ressources sont partagées selon la règle : le plus grand  $\frac{\text{tau d'application}}{\text{temps de déplacement}}$ . S'il y a plus de

ressources de même types que celles qui sont requises par ces incendies, un nombre maximum possible de ressources est assigné. Les ressources restantes sont partagées pour diminuer, avec le même rythme, la hauteur des incendies qui ont le second plus grands ratios.

Étape 3: Vérifiez si nous avons besoin de modifier l'affectation des ressources.

L'affectation des ressources change dans trois cas:

(a) La hauteur d'un incendie avec le plus grand ratio devient égale à la hauteur d'un incendie avec un ratio initialement inférieur.

(b) La suppression d'un incendie est achevé avant la fin de l'intervalle  $[r_k, r_{k+1}]$ .

(c) Atteint la fin de l'intervalle  $[r_k, r_{k+1}]$ .

Étape 4: Utilisez la règle de McNaughton [59] pour ordonnancer les tâches de confinement des incendies.

Cette règle utilise des ressources supplémentaires pour contenir un incendie uniquement si les précédentes sont entièrement utilisées.

Étape 5: Calculer à nouveau les hauteurs des incendies non supprimés et retourner à l'étape 1.

### 3. UN ORDONNANCEMENT STOCHASTIQUE À UNE SEULE MACHINE DE TÂCHES DÉTÉRIORANTES : CAS DES INCENDIES FORESTIERS

#### 3.1. Introduction

Dans le chapitre 2, une attention est portée sur le modèle de RACHANIOTIS et PAPPIS [4]. Afin d'appréhender certains aspects de cette modélisation, nous avons expliqué ses hypothèses théoriques et ses paramètres. Ensuite, nous avons décrit la manière dont le modèle est construit. Enfin, nous avons présenté la méthodologie par séparation et évaluation employée pour une résolution optimale du problème. Cela nous a permis, par la suite, de bien incorporer le facteur «incertitude» souvent présent dans la déclaration, le développement et la propagation d'un incendie de forêt. Une description stochastique plus précise du problème est présentée et un modèle décisionnel semi-markovien à deux décisions est établi. Le critère est l'espérance des coûts linéaires et prévisionnels. Le problème stochastique est résolu par la technique des «indice d'allocation dynamique». Une caractérisation de cette priorité et sa détermination sont expliquées. Des tests numériques sont effectués. Une conclusion reprenant les points essentiels de cette expérimentation est fournie. Les résultats développés dans ce chapitre ont été présentés dans l'article [7], il est donné dans l'Annexe A.

#### 3.2. Restrictions du modèle de RACHANIOTIS et PAPPIS [4]

En examinant attentivement le modèle de RACHANIOTIS et PAPPIS [4] indiqué par l'équation (2.14), on aperçoit quelques faiblesses :

- Le modèle est valide seulement pour les terrains forestier horizontaux ou inclinés, ont des pentes uniformes et que les conditions météorologiques et le combustible sont homogènes.
- Dans l'expression de  $E_i(t)$  donnée par l'équation (2.12), les fronts de l'incendie  $J_i$  à des instants donnés sont approximés par des ellipses de différents rapports longueur/largeur (Voir Annexe C).

Regardons ici l'effet que produit la pente sur un front initialement circulaire. Nous supposons que la végétation est uniforme et qu'il n'y a pas de vent. Le front de l'incendie se propage et prend une forme oblongue lors de la présence d'une pente. Plus l'angle d'élévation est grand, plus l'ellipse s'allonge dans la direction

de la plus grande pente et moins il se déplace dans la direction opposée. S'il n'y a pas de pente, il reste circulaire. Il est à noter que lorsque l'angle d'élévation devient plus grand, les fronts ne peuvent pas être tout à fait des ellipses. La vitesse est quasi nulle dans la direction opposée à la plus grande pente.

Les figures 3.1, 3.2 et 3.3 illustrent la position du front à chaque 15 minutes [60]. Les résultats obtenus avec trois angles d'élévation : 0 (correspondant à un terrain horizontal),  $\pi/16$  et  $\pi/4$ .

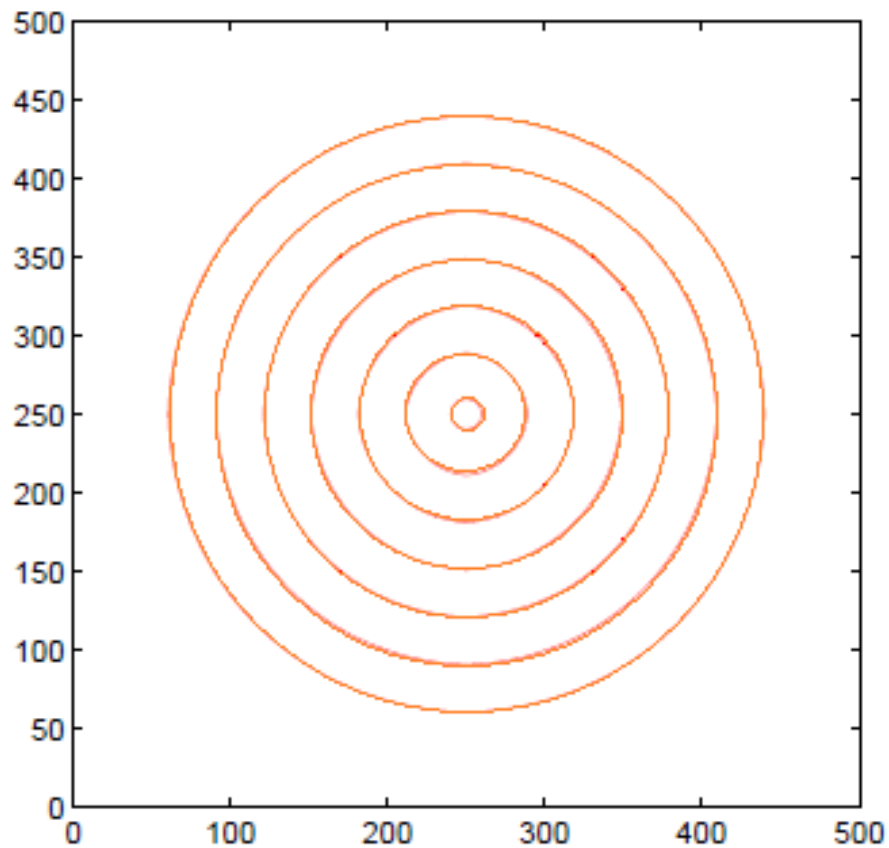


Figure 3.1 : Évolution du front sur un terrain plat avec une végétation uniforme. Le front est illustré à toutes les 15 minutes.

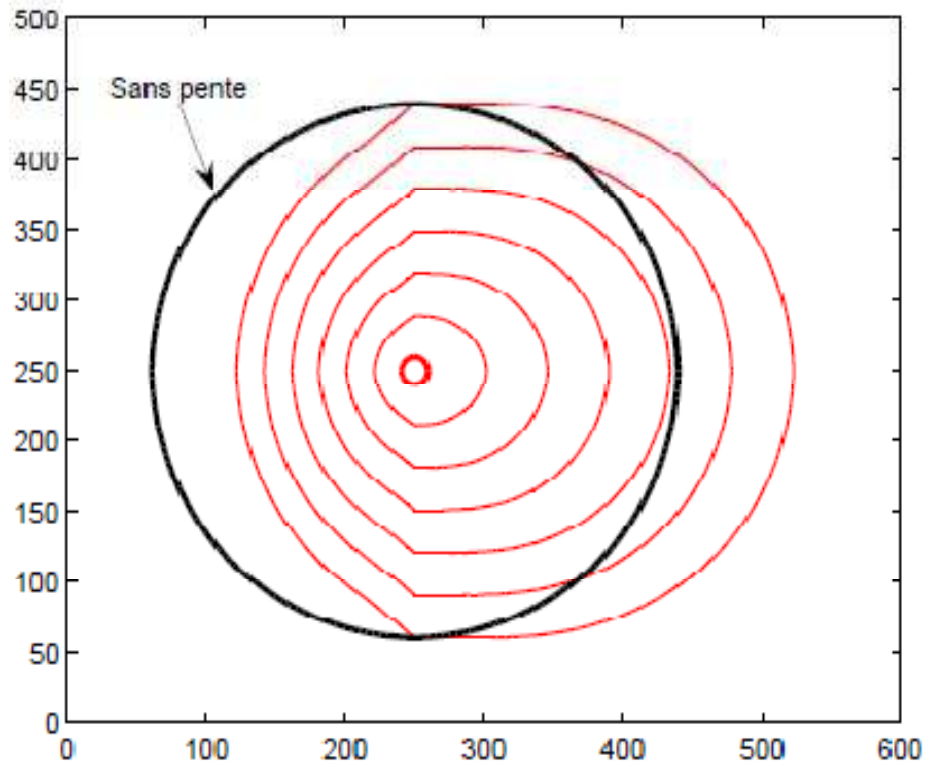


Figure 3.2 : Évolution du front sur une pente ayant une élévation de  $\pi/16$ .  
Le front est illustré à toutes les 15 minutes.

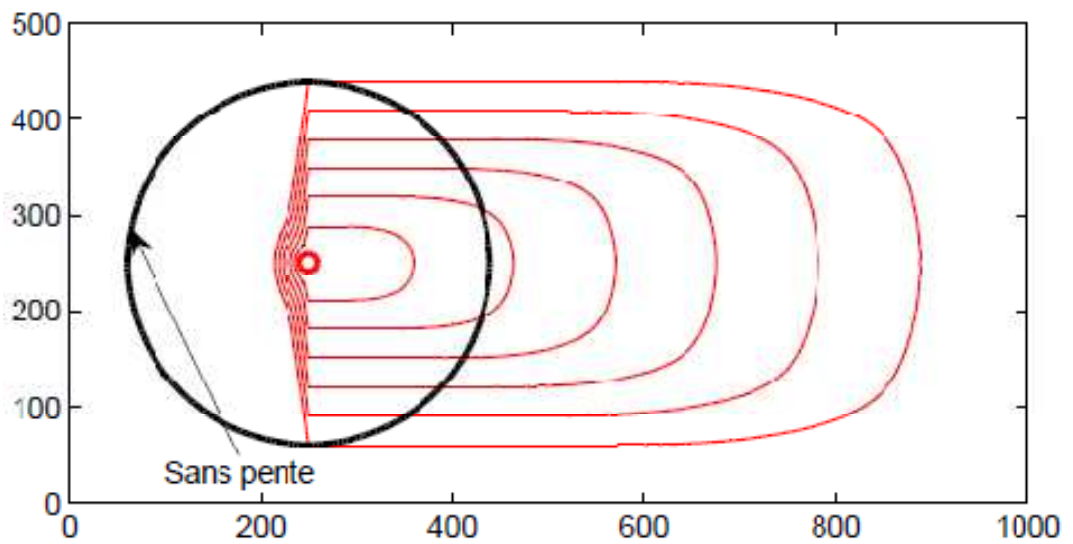


Figure 3.3 : Évolution du front sur une pente ayant une élévation de  $\pi/4$ .  
Le front est illustré à toutes les 15 minutes.



- Le modèle n'est pas aussi valide pour le facteur vent. Dans la relation (2.2), la vitesse du vent  $U$  n'a qu'une contribution positive sur la vitesse du front. Cela ne correspond pas à la réalité, car le front peut se propager dans la direction contraire aux vents. Il avance moins vite, mais il avance tout de même.
- Dans l'équation (2.6), le calcul de  $\omega_i(t)$  est fondée sur le système BEHAVE. Celui ci prend en considération le facteur vent. Par contre, il a des restriction au niveau de la vitesse de vents [60].
- L'hypothèse faite est que la végétation est homogène. Cela n'est pas le cas pour une forêt.
- Le modèle décrit la phase d'accélération d'un incendie. Il faut que l'intensité des flammes soit faible, sinon nous sortons complètement du domaine de sa validité.
- Il n'est valide que pour un domaine restreint de paramètres. Il existe de nombreuses et complexes interactions entre les facteurs physiques du milieu naturel et les caractéristiques biologiques des combustibles. Il est très difficile de prévoir tout les paramètres du problème et de faire la part de chacun d'entre eux dans la déclaration, le développement et la propagation d'un incendie.
- Dans le modèle, l'exposant de  $t$ ,  $B_i$  exprimé dans (2.3), est fonction de la vitesse du vent  $U$ .  $U$  change de valeur d'une région à l'autre.  $B_i$  n'est pas identique pour chaque cas d'incendie. Cette propriété rend le problème d'ordonnancement difficile à résoudre. L'algorithme de type par séparation et évaluation établi dans [5] produit des solutions optimales pour une taille réduite de l'ensemble des incendies. La formulation complexe du modèle est la base de sa défaillance.
- Le modèle est déterministe, cependant, dans la pratique, il s'agit d'une variable aléatoire.
- Quel que soit le niveau de la fiabilité du modèle, il est important de prendre en considération la probabilité, bien qu'elle soit très petite, que des événements indésirables et inattendus peuvent se produire. Cela est justifiée par les risques liés à une défaillance éventuelle dans le confinement de l'incendie.

### 3.3. Un modèle stochastique pour le problème de la lutte contre les incendies de forêts

Dans le cas déterministe, le modèle de RACHANIOTIS et PAPPIS [4] a fait ses preuves. Il est unanimement reconnu. Mais comme nous avons pu le constater, le modèle possède quelques faiblesses. Sa contribution a pu être corrigée grâce à l'ajout du facteur «incertitude».

Malgré des multiples recherches, les propriétés statistiques d'un incendie de forêt ne sont pas entièrement établies. La conception de la prévention vise à fournir des solutions avec des niveaux de risque tolérables. Les distributions normales sont extrêmement importantes. Elles sont utilisées pour maximiser la chance de succès avec un niveau de risque acceptable, ou pour minimiser le risque pour un niveau de succès désiré.

La densité normale  $f_{[i]}(t)$  associée à l'incendie  $J_{[i]}$  est exprimée par :

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right), \quad (3.1)$$

où  $\mu$  et  $\sigma$  sont respectivement l'espérance mathématique et l'écart-type.  $t = C_{[i]}$ , la date de fin d'exécution de  $J_{[i]}$ . Elle est donnée par l'équation (2.1).

L'événement: "L'incendie  $J_{[i]}$  est contrôlé à l'instant  $d_{[i]}$ " doit avoir une probabilité d'occurrence la plus élevée. La moyenne  $\mu$  doit être égale à  $d_{[i]}$ .

Comme la lutte contre un incendie de forêt dépasse souvent  $d_{[i]}$ , cette date ne peut pas être considérée comme une date de fin impérative.

Une période de sécurité  $[d_{[i]}; d_{[i]} + \varepsilon]$  est à considérer dans laquelle la probabilité de contenir  $J_{[i]}$  a une valeur plus élevée. Le réel  $\varepsilon > 0$  dépend de  $s_{[i]}$ , la date de début d'exécution de  $J_{[i]}$ .

Une propriété de la distribution normale, la valeur de l'écart-type  $\sigma$  est liée au niveau de risque. Si le risque toléré est de 5%, on obtient que 2-écarts-types représentent  $d_{[i]} - r_{[i]}$ . Lorsque le confinement de  $J_{[i]}$  n'est pas terminé à la date  $d_{[i]} + \varepsilon$ , la probabilité que le processeur peut contenir l'incendie décroît. Il s'interprète que l'incendie prend rage et mérite par conséquent des efforts supplémentaires de lutte (voir la figure 3.4).

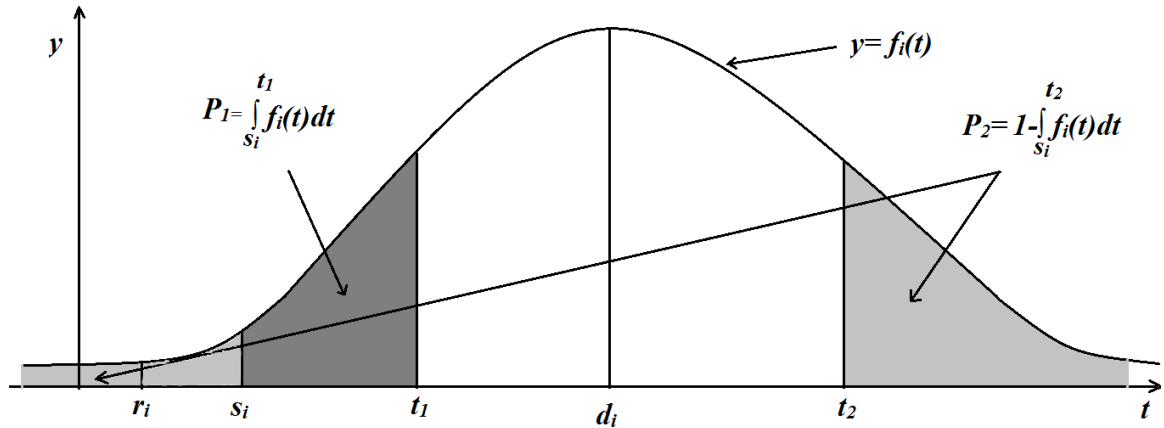


Figure 3.4 : Calcul de probabilité:  $P_1$  et  $P_2$ .

La densité normale  $f_i(t)$  est utilisée pour estimer la probabilité de contrôler un incendie  $J_i$  à tout instant  $t \geq s_i$ .

Dans la figure 3.4,  $P_1$  est l'aire de la surface comprise entre la courbe de la fonction  $f_i$ , l'axe du temps et les droites d'équations  $t = s_i$  et  $t = t_1$ .

$P_1$  représente la probabilité que  $J_i$  soit contrôlé à l'instant  $t_1$  avant la date  $d_i$ .  $P_1$  augmente en fonction de  $t_1$ , et lorsque  $s_i$  s'approche de  $d_i$ , la probabilité de contenir  $J_i$  à l'instant  $d_i$  diminue et la période de sécurité  $[d_{[i]}; d_{[i]} + \varepsilon]$  devienne large.

La probabilité  $P_2$  que l'incendie  $J_i$  soit maîtrisé dans  $[d_{[i]}; d_{[i]} + \varepsilon]$  doit avoir des valeurs plus élevées que celle pour  $t_1 = d_i$ . Cette probabilité diminue au fil du temps et coïncide à l'instant  $t_2 = d_i + \varepsilon$  avec celle à l'instant  $t_1 = d_i$ . Dépassant  $d_i + \varepsilon$ , l'incendie est incontrôlable et la probabilité de contenir l'incendie diminue rapidement. Ces aspects naturelles précitées dans la lutte contre les incendies forestier sont justifiées par l'expression prise pour  $P_2$ . La valeur de  $\varepsilon$  est la solution de l'équation :

$$\int_{d_i}^{d_i + \varepsilon} f_i(t) dt = 1 - 2 \int_{s_i}^{d_i} f_i(t) dt .$$

### 3.4. Un modèle semi-markovien pour le problème de confinement optimal

$J_i$  représente un incendie forestier appelé aussi une tâche.

Les problèmes où les temps d'exécution des tâches sont incertains sont dit stochastiques. Une approche théorique pour l'ordonnancement stochastique est la théorie "des processus bandits" [8]. En leur octroyant des coûts positifs, le critère à maximisé est l'espérance des coûts linéaires et prévisionnels. Ces processus modélisent les tâches à exécuter par le processeur. Une façon d'ordonner les tâches est de leur attribuer une priorité dynamique appelée "indices d'allocation dynamiques" ou "indices de Gittins". Ces indices sont calculés à tout instant de décision, permettant à l'ordonnanceur d'adapter dans son ordre les arrivées et les exécutions au fur et à mesure que l'information sur la tâche est recueillie.

#### 3.4.1. Les processus bandits

Le processus bandit  $x_i$  associé à une tâche  $J_i$  est un processus semi-markovien à deux décisions. Il est défini par la donnée d'un sextuplet  $(\Omega_i, \Gamma, P_i, F_i, R_i, \alpha_i)$  où

- $\Omega_i$  est l'espace d'états du processus. Il est supposé fini.
- $\Gamma = \{e, g\}$  est l'ensemble des décisions à prendre. « $e$ » représente la décision d'exécuter le processus bandit  $x_i$  et « $g$ » est la décision de geler le processus  $x_i$ .
- $P_i$  est une matrice stochastique de transition entre les états du processus  $x_i$ . Sa construction est définie dans la sous-section (3.4.3) ci-dessous.
- Le temps de séjour dans un état est une variable aléatoire obéissant à une loi de probabilité  $F_i$  quelconque.
- $R_i(x_i(t))$  est le coût de transition du processus  $x_i$  de l'état  $x_i(t)$  à l'état  $x_i(t+1)$  quand la décision  $u \in \Gamma$  est appliquée. L'application de la décision  $e$ , entraîne un coût actualisé  $\alpha_i^t V_i$ , où  $\alpha_i \in ]0, 1[$  un nombre réel appelé «taux d'actualisation» et  $V_i$  est la valeur attribuée à une zone forestière  $i$ . On impose la condition suivante. Si on n'a pas changé d'état, on est presque sûr d'avoir utilisé la décision  $g$ . Utilisé la décision  $g$ , n'entraîne aucun coût.

Le problème stochastique de la lutte contre  $n$  incendies de forêts est modélisé par ce qu'on appelle «une famille de processus bandits alternatifs simple»  $S = \{(\Omega_i, \Gamma, P_i, F_i, R_i, \alpha) : i = 1, 2, \dots, n\}$ .

Les  $n$  processus bandits sont dits «alternatifs» si

- Ils ont les mêmes instants de décision.
- En tout instant  $t$ , la décision  $e$  est appliquée à un processus bandit et  $g$  est appliquée à tous les autres.
- Le taux d'actualisation  $\alpha$  est le même pour tous les processus.
- En tout instant  $t$ , l'état du système  $S$  est donné par le vecteur  $X(t) = (x_1(t), x_2(t), \dots, x_n(t))$  des états  $x_i(t)$  des processus bandits  $x_i$ .

Le coût de  $S$  encourut,  $R(X(t))$ , est celui contracté par le processus bandit auquel la décision  $e$  est appliquée. Si la décision  $e$  est prise pour le processus bandit  $x_i$ , on aura  $R(X(t)) = R_i(x_i(t)) = \alpha^t V_i$ .

La famille de processus bandits alternatifs  $S$  est dite «simple» si la décision  $e$  est appliquée à tous les processus bandits au moins une fois.

Une politique  $\pi$  est une suite de décisions à entreprendre. Nous considérons l'objectif celui de maximiser l'espérance des coûts linéaires et prévisionnels :

$$\mathbb{E}_\pi \left\{ \sum_{t=0}^{+\infty} R(X(t)) \mid X(0) = \mathbf{0}_{\mathbb{R}^n} \right\}.$$

$\mathbb{E}_\pi$  désigne l'espérance conditionnelle sachant que la politique  $\pi$  est employée et que l'état du système  $S$  à l'instant zéro est  $(0, 0, \dots, 0) \in \mathbb{R}^n$ .

Cette fonction coût peut s'écrire comme :

$$\mathbb{E}_\pi \left\{ \sum_{i=1}^n \alpha^{s_i} V_i \right\},$$

où  $s_i$  est la date de début d'exécution de la tâche  $J_i$ .

Les valeurs  $V_i$  sont supposées bornées, la fonction objective sera bien définie.

### 3.4.2. Existence des indices d'allocation dynamiques

GITTINS et JONES [9] ont énoncé le premier théorème d'existence d'une politique maximisant la fonction coût  $\mathbb{E}_\pi \left\{ \sum_{t=0}^{+\infty} R(X(t)) \mid X(0) = 0_{\mathbb{R}^n} \right\}$ .

Ce théorème se formule comme suit:

#### Théorème 3.1

Il existe  $n$  fonctions  $\gamma_j : \Omega_j \rightarrow \mathbb{R}$ ,  $j = 1, 2, \dots, n$ , tel que à l'instant  $t$  l'application de la décision  $e$  au processus bandit  $x_i$  est optimal si et seulement si :

$$\gamma_i \{x_i(t)\} = \text{Max}_{1 \leq j \leq n} \gamma_j \{x_j(t)\}.$$

La fonction  $\gamma_i$  est appelée Indice d'Allocation Dynamique, notée en abrégé I.A.D. Elle est aussi appelée par WHITTLE [61] Indice de GITTINS. La politique  $\pi^*$  qui en tout instant  $t$ , exécute le processus bandit  $x_i$  tel que  $\gamma_i \{x_i(t)\} = \text{Max}_{1 \leq j \leq n} \gamma_j \{x_j(t)\}$  est appelée "règle d'indices".

Beaucoup d'auteurs ont contribué à la caractérisation et à la détermination de ces I.A.D. Des exemples sont donnés dans [8,62,63].

### 3.4.3. Matrice stochastique de transition $P$

La matrice  $P$  a les caractéristiques suivantes:

- Une tâche  $J_i$  transite à un état désigné par le symbole  $*_i \in \Omega_i$  seulement si elle est terminée. Si elle est à l'état  $*_i$  alors aucune transition vers un autre état ne peut être faite. On définit l'état de  $J_i$  à l'instant  $t$  par :

$$x_i(t) = \begin{cases} *_i & \text{si l'exécution de la tâche } J_i \text{ est terminée à l'instant } t. \\ \theta & \text{ayant reçu } \theta \text{ unités d'exécution jusqu'à l'instant } t, \\ & J_i \text{ n'a pas terminée son exécution} \end{cases}$$

- Si à l'instant  $t$  la décision  $e$  est appliquée à  $J_i$ , l'exécution de  $J_i$  progresse, ou bien elle se termine dans l'intervalle du temps  $[t, t + 1[$ . Les états des tâches non élues pour s'exécuter à l'instant  $t$  restent fixés à zéro. Ce qui se récapitule par :

$$x_i(t+1) = *_{i} \quad \text{ou bien} \quad x_i(t+1) \geq x_i(t)$$

$$\text{et } \forall j, j \neq i, x_j(t+1) = *_{j} \quad \text{ou bien} \quad x_j(t+1) = x_j(t) = 0.$$

- On appelle matrice stochastique ou markovienne, une matrice carrée dont tous les coefficients sont positifs et dont la somme de chaque ligne est égale à 1. Ainsi, la matrice des probabilités de transition d'un processus de Markov est markovienne. La transition de l'état  $x_i(t)$  à l'état  $x_i(t+1)$  est définie par les expressions probabiliste suivantes:

$$P[x_i(t+1) = *_{i} | x_i(t) = *_{i}] = 1, \quad (3.2)$$

$$P[x_i(t+1) = *_{i} | x_i(t) = \theta] = \begin{cases} \int_{s_i}^{\theta+1} f_i(t) dt & \text{si } s_i \leq \theta \leq d_i \\ 1 - \int_{s_i}^{\theta+1} f_i(t) dt & \text{si } \theta > d_i \end{cases}. \quad (3.3)$$

Elle sont indiquées dans la figure 3.4.

Et

$$P[x_i(t+1) = \theta+1 | x_i(t) = \theta] = 1 - P[x_i(t+1) = *_{i} | x_i(t) = \theta], \quad (3.4)$$

avec  $x_i(s_i) = 0$  et  $f_i$  est la distribution normale donnée par l'équation (3.1).

### 3.5. Règle d'I.A.D pour la politique optimale de confinement

Rappelons d'abord qu'on cherche une politique de confinement optimale non-préemptive. Nous supposons qu'une tâche  $J_k$  a terminée son exécution à l'instant  $t$ , et qu'une tâche  $J_i$  est élue pour s'exécuter. La prochaine date de décision est  $t + T_{k,i} + p_i(t)$ , la date de fin d'exécution de  $J_i$ .

Si aucune tâche n'a été performée, nous posons  $t=0$  et  $k=0$ .

Si une tâche  $J_j$  est choisie immédiatement après  $J_i$ , la somme des coûts actualisés obtenue de l'exécution successive de  $J_i$  puis  $J_j$  est :

$$\alpha^t V_i + \alpha^{t+T_{k,i}+p_i(t)} V_j.$$

Si l'ordre d'exécution de  $J_i$  et  $J_j$  est inter-changé, nous obtenons :

$$\alpha^t V_j + \alpha^{t+T_{k,j}+p_j(t)} V_i.$$

Le premier ordre a le plus grand valeur si, et seulement si :

$$\frac{V_i}{1 - \alpha^{T_{k,i}+p_i(t)}} > \frac{V_j}{1 - \alpha^{T_{k,j}+p_j(t)}}.$$

L'espérance des coûts linéaires et prévisionnels  $\mathbb{E}_\pi \left\{ \sum_{i=1}^n \alpha^{s_i} V_i \right\}$  est maximisée en allouant

à chaque date de décision  $t$  la tâche  $J_i$  pour s'exécuter qui a le plus grand I.A.D  $\gamma_i(t)$ , donné par :

$$\frac{V_i}{1 - \alpha^{T_{k,i}} \mathbb{E}_\pi \left\{ \alpha^{p_i(t)} \right\}}.$$

Notons par  $K(t)$  l'ensemble des tâches non choisies pour s'exécuter jusqu'à l'instant  $t$ .

Le théorème ci-après est le résultat principal de ce chapitre :

### Théorème 3.2.

À chaque date de décision  $t$ , il est optimal de choisir la tâche  $J_i$  pour s'exécuter si et seulement si,

$$\gamma_i(t) = \max_{j \in K(t)} \gamma_j(t) = \max_{j \in K(t)} \frac{V_j}{1 - \alpha^{T_{k,j}} \mathbb{E}_\pi \left\{ \alpha^{p_j(t)} \right\}}.$$

### 3.6. Détermination de la priorité de confinement optimale

Nous supposons que la date de début d'exécution de la tâche  $J_{[i]}$  est l'instant  $t$ .

Si nous utilisons la somme géométrique :

$$\sum_{h=0}^{p_{[i]}(t)-1} \alpha^h = \frac{1 - \alpha^{p_{[i]}(t)}}{1 - \alpha},$$

nous pouvons écrire l'espérance conditionnelle :

$$\mathbb{E}_\pi \left\{ \alpha^{p_{[i]}(t)} \mid x_{[i]}(t) = 0 \right\}$$

sous la forme :

$$1 - (1 - \alpha) \mathbb{E}_\pi \left\{ \sum_{h=0}^{p_{[i]}(t)-1} \alpha^h \mid x_{[i]}(t) = 0 \right\}.$$



Nous notons  $H_{[i]}^{(0)}$  comme :

$$\mathbb{E}_{\pi} \left\{ \sum_{h=0}^{P_{[i]}(t)-1} \alpha^h \mid x_{[i]}(t) = 0 \right\}.$$

Nous obtenons la valeur de  $H_{[i]}^{(0)}$  en résolvant l'équation de Bellman suivante :

$$\begin{aligned} H_{[i]}^{(0)} &= 1 + \alpha P_{[i]}^{(0,1)} H_{[i]}^{(1)} + \alpha P_{[i]}^{(0,*_{[i]})} H_{[i]}^{(*_{[i]})} \\ H_{[i]}^{(1)} &= 1 + \alpha P_{[i]}^{(1,2)} H_{[i]}^{(2)} + \alpha P_{[i]}^{(1,*_{[i]})} H_{[i]}^{(*_{[i]})} \\ &\dots \\ H_{[i]}^{(q-1)} &= 1 + \alpha P_{[i]}^{(q-1,q)} H_{[i]}^{(q)} + \alpha P_{[i]}^{(q-1,*_{[i]})} H_{[i]}^{(*_{[i]})} \\ H_{[i]}^{(*_{[i]})} &= 1 + \alpha P_{[i]}^{(*_{[i]},*_{[i]})} H_{[i]}^{(*_{[i]})} \end{aligned} \tag{3.5}$$

Dans le système d'équations (3.5) :

$$\begin{aligned} q &= P_{[i]}(t), \\ H_{[i]}^{(y)} &= \mathbb{E} \left\{ \sum_{h=y}^{q-1} \alpha^h \mid X_{[i]}(t+y) = y \right\}, \quad y = 1, 2, \dots, q-1, \\ H_{[i]}^{(q)} &= 0, \\ P_{[i]}^{(l,l+1)} &= P \left[ X_{[i]}(t+l+1) = l+1 \mid X_{[i]}(t+l) = l \right], \quad l = 0, 1, \dots, q-1, \\ P_{[i]}^{(l,*_{[i]})} &= P \left[ X_{[i]}(t+l+1) = *_{[i]} \mid X_{[i]}(t+l) = l \right], \quad l = 0, 1, \dots, q-1, \end{aligned}$$

et  $P$  est la probabilité définie par les équations (3.2), (3.3) et (3.4).

Le système (3.5) a une structure triangulaire supérieure. Sa résolution se fait directement de manière ascendante :

$$\begin{aligned}
H_{[i]}^{(*[i])} &= \frac{1}{1-\alpha}, \\
H_{[i]}^{(q-1)} &= 1 + \frac{\alpha}{1-\alpha} P_{[i]}^{(q-1, *[i])}, \\
&\dots \\
H_{[i]}^{(0)} &= 1 + \frac{\alpha}{1-\alpha} P_{[i]}^{(0, *[i])} + \sum_{j=1}^{q-1} \alpha^j \left( \prod_{l=0}^{j-1} P_{[i]}^{(l, l+1)} \right) \left( 1 + \frac{\alpha}{1-\alpha} P_{[i]}^{(j, *[i])} \right).
\end{aligned} \tag{3.6}$$

La forme finale de la priorité  $\gamma_{[i]}(t)$  est alors :

$$\gamma_{[i]}(t) = \frac{V_{[i]}}{1 - \alpha^{T_{[i-1],[i]}} \left( 1 - (1-\alpha) H_{[i]}^{(0)} \right)}, \tag{3.7}$$

où  $H_{[i]}^{(0)}$  est donnée dans l'équation (3.6) et  $T_{[0],[1]} = T_{0,[1]}$ .

### 3.7. Implémentation et expérimentation

La priorité  $\gamma_{[i]}(t)$ , exprimée par l'équation (3.7), a été codé et exécutée dans Scilab version 5.5.2. Le programme est annexé à la fin de ce manuscrit. Les tests ont été effectués sur un processeur Intel Core i5-2450M cadencé à 2.50 GHz et 4 Go de RAM sous Windows 7 (64 bits).

Les données du problème sont classées en trois catégories comme suit :

- Les paramètres liés aux incendies :

$$A_{[i]} \in ]0, 14[ , \quad B_{[i]} \in ]1, 2[ \quad \text{et} \quad c_{[i]}(t) \in ]0, 11[ \quad \text{pour} \quad U_{[i]} \in \{0, 1, \dots, 8\}.$$

Pour une description plus détaillée de ces estimations, le lecteur est prié de se référer à l'article [55].

- Les paramètres liés aux zones brûlées par les incendies et/ou au processeur de lutte sont donnés dans [4] par :

$$V_{[i]0} \in ]10^5, 2 \times 10^5[ , \quad \alpha \in ]10^{-6}, 10^{-4}[ , \quad T_{0,[1]} \text{ et } T_{[i],[i+1]} \in ]0, 1[ , \quad \tau_i = 0.2, \\ \rho = 6.48 \times 10^6 \text{ et } Q = 3 \times 10^4.$$

- Pour la distribution normale, la surface délimitée par la courbe en cloche et l'axe du temps se trouve presque à quatre-écarts-types de la moyenne. Les points de cette surface qui se trouvent à des distances supérieures à quatre-écarts-types par rapport à la moyenne sont extrêmement rares d'apparaître dans la pratique. L'écart-type  $\sigma_{[i]}$  est généré selon la loi uniforme, dans l'ensemble

$$\left\{ \frac{d_{[i]} - r_{[i]}}{4}, \frac{d_{[i]} - r_{[i]}}{3}, \frac{d_{[i]} - r_{[i]}}{2}, d_{[i]} - r_{[i]} \right\},$$

comme envisagé pour les autres paramètres du problème.

Les étapes suivies pour calculer un IAD  $\gamma_i(t)$  sont exposées ci-après.

**Algorithme 3.1:** Calcul de l'IAD  $\gamma_i(t)$

Les données:

$r_i$  : date de disponibilité de l'incendie  $i$ .

$s_i$  : la date de début d'exécution de l'incendie  $i$ .

$d_i$  : la date d'échéance de l'incendie  $i$ .

$V_i$  : la valeur attribuée à une zone de forêt  $i$ .

$\alpha$  : le taux d'actualisation.

$\mu_i$  : l'espérance mathématique de la densité normale  $f_i$ . Dans notre cas  $\mu_i = d_i$ .

$\sigma_i$  : l'écart-type de  $f_i$ . Dans notre cas  $\sigma_i \in \left\{ \frac{d_i - r_i}{4}, \frac{d_i - r_i}{3}, \frac{d_i - r_i}{2}, d_i - r_i \right\}$ .

$q$  : nombre d'états de l'incendie  $i$ .

Etape 1: Calcul de probabilité

Le symbole  $*_i$  désigne l'état terminal de l'incendie  $i$ , i.e., quand il s'éteint.

- $P_i^{(*_i, *_i)} = 1$  la probabilité de rester à l'état  $*_i$ .
- pour  $l=1, \dots, q-1$ , la probabilité de transition de l'état  $l$  à l'état  $*_i$  :

$$P_i^{(l, *_i)} = \begin{cases} \int_{s_i}^{l+1} f_i(t) dt & \text{si } s_i \leq l \leq d_i \\ 1 - \int_{s_i}^{l+1} f_i(t) dt & \text{si } l > d_i \end{cases}.$$

Le calcul de l'intégrale  $\int_{s_i}^{l+1} f_i(t) dt$  est expliqué par la figure 3.4.

- pour  $l=1, \dots, q-2$ , la probabilité de transition de l'état  $l$  à l'état  $l+1$ :

$$P_i^{(l, l+1)} = 1 - P_i^{(l, *_i)}.$$

Etape 2: Calculer la quantité  $H_i$  définies par :

$$H_i = 1 + \frac{\alpha}{1 - \alpha} P_{[i]}^{(0, *_i)} + \sum_{j=1}^{q-1} \alpha^j \left( \prod_{l=0}^{j-1} P_{[i]}^{(l, l+1)} \right) \left( 1 + \frac{\alpha}{1 - \alpha} P_{[i]}^{(j, *_i)} \right).$$

Etape 3: Calculer  $\gamma_i(t)$  :  $\gamma_i(t) = \frac{V_i}{1 - \alpha^{T_{i-1,i}} (1 - (1 - \alpha) H_i)}$ .

Une configuration d'un problème est appelée «instance». Nous avons mené une série d'expérimentation sur un grand nombre d'exemples. Le but est d'estimer la date d'échéance  $d$  des incendies, et d'autre part, d'indiquer le degré d'efficacité du processeur.

A cet effet, la valeur de  $d$  est augmentée par pas de "un", de 1 à 50 (en heures). Pour chaque cas de  $d$ , l'algorithme a été appliqué à une variété d'instances, de l'ordre de 1000. Chacune d'entre elles contient 100 exemples d'incendie.

Pour obtenir la précision dans le calcul des I.A.D  $\gamma_{[i]}(t)$ , le nombre d'états du processus bandit de chaque incendie est fixé à 100 états.

Pour chaque cas de  $d$  et pour les 1000 instances générées aléatoirement, les résultats analysés sont :

- $\bar{N}$  est la moyenne des nombres d'incendies maîtrisés dans l'intervalle du temps  $[1, d]$ .
- $\bar{C}$  est la moyenne des dates de fin d'exécution des premiers incendies sélectionnés par la règle d'indice et qui vont échapper aux efforts de lutte de la première intervention.
- $\bar{T}$  est la moyenne des temps d'exécution de l'algorithme.

Ces résultats sont ci-dessous récapitulés :

$d$	$\bar{N}$	$\bar{C}$	$\bar{T}$	$d$	$\bar{N}$	$\bar{C}$	$\bar{T}$	$d$	$\bar{N}$	$\bar{C}$	$\bar{T}$
1	1.22	1.37	1.03	18	7.14	537.2	1.56	35	7.76	30 272.0	1.61
2	2.54	2.37	1.18	19	7.06	839.6	1.57	36	7.6	12 106.0	1.60
3	3.86	3.69	1.31	20	7.34	4 260.0	1.58	37	7.14	41 890.0	1.59
4	5.18	6.47	1.42	21	7.08	12 791.0	1.57	38	7.36	12 334.0	1.58
5	5.86	11.40	1.48	22	7.96	2 397.0	1.6	39	7.8	45 222.0	1.60
6	6.24	22.95	1.49	23	7.24	5620.0	1.6	40	7.62	108 125.0	1.59
7	6.3	20.96	1.52	24	7.1	5 000.0	1.59	41	7.18	564 072.0	1.59
8	6.2	66.35	1.51	25	7.24	2 748.0	1.58	42	7.36	12 334.0	1.58
9	6.8	120.53	1.53	26	7.7	14 101.0	1.59	43	7.82	173 442.0	1.59
10	6.72	196.28	1.53	27	7.5	10 661.0	1.60	44	7.66	108 284.0	1.62
11	7.08	98.76	1.54	28	7.1	10 352.0	1.59	45	7.18	564 072.0	1.61
12	7.14	156.5	1.54	29	7.24	2 801.0	1.60	46	7.88	186 788.0	1.62
13	7.08	164.48	1.54	30	7.7	14 101.0	1.59	47	7.68	108 302.0	1.62
14	7.06	403.24	1.55	31	7.56	11 212.0	1.61	48	7.2	582 567.0	1.66
15	7.18	1721.6	1.56	32	7.12	10 577.0	1.60	49	7.38	13 026.0	1.62
16	7.18	486.37	1.56	33	7.32	10 197.0	1.59	50	7.92	332 857.0	1.63
17	7.2	449.96	1.57	34	7.66	18 542.0	1.58				

Tableau 3.1 : Résultats des expérimentations numériques

A la lecture du tableau ci-dessus, il apparaît que dans la période du temps allant de 11 à 50 premières heures, le processeur a le pouvoir de contenir en moyenne 07 incendies. Le temps nécessaire pour contrôler le 08<sup>ième</sup> incendie se détériore rapidement et dépasse la capacité du processeur. En tenant compte de la machine de calcul utilisée, le temps d'exécution de l'algorithme pour chaque instance n'est pas significative. Il est d'environ 1,6 s.

## CONCLUSION

Dans des conditions assez particulières, il arrive que les durées opératoires des tâches dépendent du temps passé dans l'atelier à attendre que la machine se libère. De telle situation se pose par exemple pour les incendies de forêts, qui se propageant, requièrent un traitement d'extinction de plus en plus long. En théorie d'ordonnancement, ce problème est dit «à tâches détériorantes». Il a constitué le thème principale de notre travail. Dans la littérature, plusieurs modèles de propagation des incendies de forêts sont établis. Le gros du travail a été d'analyser le modèle de RACHANIOTIS et PAPPIS [4] qui semblait être en mesure de bien répondre au problème étudié. En résumé, ce modèle détermine les durées des interventions du processeur de lutte utilisé, qui augmentent tant que les incendies ne sont pas contrôlés. Il prenait en considération les différents phénomènes physiques liés aux incendies ainsi que les caractéristiques du processeur. En regardant les différents aspects de sa constriction, nous avons pu déterminer ses forces et ses faiblesses. Sa contribution à la résolution du problème étudié a pu être corrigée grâce à l'ajout du facteur «incertitude». Un processus de décision semi-markovien est développé et la technique des «Indices d'Allocation Dynamiques» est utilisée pour résoudre le problème. L'optimalité de cette règle est prouvée, et sa détermination est également expliquée. Par des tests numériques, les dates d'échéance des incendies sont d'au plus 11 heures, et seulement 7 incendies peuvent être surmontés par le processeur.

Dans notre processus de décision semi-markovien, les coûts sont fixés de sorte que les seuls paramètres qui varient sont les probabilités de transition. A notre connaissance, les propriétés statistiques des incendies de forêts ne sont pas entièrement établies. Nous utilisons la distribution normale pour représenter l'incertitude dans les probabilités de transition. La distribution normale décrit la variation aléatoire des données dont les distributions sont inconnues. Le choix de l'espérance mathématique d'être la date d'échéance de l'incendie et l'écart-type gouverné par le risque de défaillance dans l'extinction de l'incendie semblent naturels dans la lutte contre les incendies de forêts.

Des extensions et des généralisations peuvent être envisagées et peuvent faire l'objet des études ultérieures. Un cas notable est d'examiner le problème quand l'information sur l'espérance mathématique de la loi normale ou quand l'information sur les durées de confinement des incendies sont incomplètes [64,65]. Un autre cas intéressant,

est d'utiliser des estimations cohérentes de la loi et du coût de transition, telles qu'ils sont définies dans [66].

Dans notre étude, les incendies sont déclarés simultanément. La généralisation au cas des incendies déclarés à des moments différents, comme a été décrit dans [4], est assez simple. Le cas où les arrivées des incendies sont poissonniennes [67] semble d'intérêt. Il est également intéressant de discuter le cas où les incendies sont désavantagés pendant leur confinement. Dans cette situation, et contrairement aux modèles à tâches détériorantes, la suppression d'un incendie génère un temps d'exécution supplémentaire pour terminer l'extinction de l'incendie.

L'ordonnancement parallèle de processeurs de lutte identiques ou non-identiques a été abordé dans [6] à l'aide du concept des tâches détériorantes. Dans ce problème, la préemption est autorisée avec un nombre variable de processeurs utilisés pour chaque incendie à un instant donné. Un modèle des durées opératoires de l'incendie est formulé en fonction du nombre et du type de processeurs utilisés dans son confinement. Une généralisation de ce modèle au cas stochastique est presque similaire à celle développée dans la section 3.4. L'indice d'allocation dynamique  $\gamma_i(t)$  peut être obtenu sans compromettre l'optimalité de la règle d'index. Chaque incendie  $J_i$ , qui est à l'instant  $t$

dans l'état  $x_i(t)$ , est associé une priorité  $\gamma_i(t) = \sup_{\xi > t} \frac{V_i}{1 - \alpha^{\xi} \mathbb{E}_{\pi} \{ \alpha^{\xi} \}}$ , où  $t_i$  est le temps

nécessaire pour déplacer les processeurs intervenant dans les autres sites à l'endroit où l'incendie  $J_i$  est allumé, et  $\xi$  est l'instant d'arrêt de  $J_i$ .  $\xi$  est une variable de décision qui dépend du temps d'exécution de  $J_i$  calculé à l'instant  $t$ . Cette priorité permet au décideur d'affecter un ensemble minimal de processeurs pour contenir un incendie qui a le plus grand indice  $\gamma_i$ . En cas de conflit ou d'égalité entre les plus grands indices, on arbitrera en choisissant un incendie selon une règle d'ordonnancement connue. Une propriété de l'indice d'allocation dynamique  $\gamma_i$ , le supremum est atteint par  $\xi = \min_{h > t} \{h, \gamma_i(h) < \gamma_i(t)\}$  et par tous les instants  $\sigma$  vérifiant  $\sigma \leq \xi$  et  $\gamma_i(\sigma) \leq \gamma_i(t)$ .

A chaque instant de décision, qui correspond à un instant d'arrêt optimal ou une date de fin d'exécution ou une nouvelle arrivée d'un incendie, les durées opératoires des incendies non encore contrôlés sont recalculés pour une nouvelle indexation et de nouvelles

affectations de processeurs sont effectuées. Les deux articles [63,68] discutent des méthodologies appropriées pour calculer  $\gamma_i$ .

Dans [6] : «lorsqu'il y a plus de processeurs disponibles que ceux déjà assignés et qui sont requis par les incendies déjà allumés, un nombre maximal possible de processeurs est assigné aux incendies dont le ratio  $\frac{\text{tau d'application}}{\text{temps de déplacement}}$  est le plus élevé.

Sinon, ces processeurs sont partagés pour diminuer au même rythme les hauteurs des incendies non encore élus pour la suppression". Veuillez noter que la décision prise à chaque instant de décision  $t$  est toujours optimale par rapport à l'état résultant de l'allocation des processeurs résiduels aux incendies déjà allumés jusqu'au l'instant  $t$ .



## ANNEXE A

## ARTICLE PARU DANS UN JOURNAL SCIENTIFIQUE

Stochastic scheduling of single forest firefighting processor, *Canadian Journal of Forest Research*, **2016**, 46(3): 370-375, 10.1139/ cjfr-2015-0294.

Url de la revue/article : <https://doi.org/10.1139/cjfr-2015-0294>

Lien pour télécharger l'article :

<http://www.nrcresearchpress.com/eprint/pRWcDa59xwTDf7GrBd56/full>

Canadian Journal of Forest Research

Publisher: Canadian Science Publishing, NRC Research Press

ISSN : 0045-5067

E-ISSN : 1208-6037

Country : CANADA

Language: Multi-Language

page 25 of 142 of publist\_sciex2017, Updated May 2017, Web of Science, Science Citation Index Expanded

Facteur d'impact (2017/2018) : 1.887

# Stochastic scheduling of single forest firefighting processor

Abdesselem Kali

**Abstract:** In forest firefighting, the longer the fires wait, the larger they grow and the longer they take to control. This study concerns the optimal deployment of single forest suppression processor of initial attack in the case of fires ignited simultaneously. The aim is to minimize the total damage caused by the fires to the burnt areas when all fires are suppressed. In Rachaniotis and Pappis (Can. J. For. Res. 36: 652–658, 2006), they use the concept of start-time dependent job processing times for modelling the time needed for fire suppression. The model is intricate but interesting in the sense that is based on theoretical and empirical research in the field of forest firefighting. As a continuation, a stochastic formulation that includes unpredicted parameters in the proposed model is considered and the “dynamic allocation index” rule is used to solve the problem. The optimality of this rule and its effectiveness are proven. Experimental results depict the framework, and inside it, the forest suppression processor achieves greater efficiency.

**Key words:** scheduling, single forest firefighting processor, Markovian model, dynamic allocation index rule.

**Résumé:** Dans la lutte contre les feux de forêt, plus en tard plus les feux propagent et prennent du temps pour leur confinement. Cette étude concerne le déploiement optimal d'un processeur de suppression de la première intervention, dans le cas des incendies enflammés simultanément. L'objectif est de minimiser les dommages dans les zones brûlées lorsque tous les feux sont supprimés. Dans Rachaniotis et Pappis (Can. J. For. Res. 36: 652–658, 2006), ils montrent que la durée opératoire du processeur dans l'extinction des incendies de forêt est dépendante du temps (détériorante). Le modèle est difficile à utiliser mais intéressant dans le sens qu'il est basé sur la recherche théorique et empirique dans le domaine de la lutte contre les feux de forêt. Dans cet article, une formulation stochastique qui inclut les paramètres imprévus dans le modèle est considérée, et la règle “d'indice d'allocation dynamique” est utilisée pour résoudre le problème. L'optimalité de cette règle et son efficacité ont fait leurs preuves. Les résultats expérimentaux déterminent le cadre, à l'intérieur le processeur atteint une plus grande efficacité. [Traduit par la Rédaction]

**Mots-clés :** ordonnancement, un seul processeur de la lutte contre les feux de forêt, modèle Markovien, règle d'indice d'allocation dynamique.

## 1. Introduction

Fire is a natural component of many forest ecosystems, but it also poses significant threats to people and forest resources. The effectiveness of forest firefighting depends, in large part, on the early detection of the fire's ignition, on the accurate prediction of fire behaviour, and on the rapid commitment of resources for intervention. So far, the approaches to modelling forest fire have involved mathematical analysis of physicochemical phenomena, laboratory tests, and observation of real cases of forest fire (Sullivan 2009a, 2009b, 2009c; Taylor et al. 2013).

Scheduling, a field of operational research, supports decisions for preventive mobilization of resources to contain fires in real time. A review of operation research contributions made in forest fire management can be found in Minas et al. (2012) and Weintraub et al. (2007).

The problem of scheduling exists whenever there is a choice as to the order in which a number of tasks can be performed (Conway et al. 1967). For our particular interest, the problem description starts with a forest fire agency (a shop) and a set of fires (jobs). The forest fire agency is completely described by giving the number and deferent types of firefighting resources of initial attack (machines), e.g., fire duty officers, trucks, fixed-wing air tankers, helicopters, etc. Fires can be ignited either simultaneously or at different times. We limit our attention to the problem of fires

that simultaneously ignite, which appears quite often in Mediterranean countries (Vorisis 1999).

The relevant attributes of a fire that are given as part of the problem description are processing time (the amount of time required to control the fire), release time (the time at which fire suppression can be started), and deadline (the time at which fire takes potentially destructive behaviour and merits aggressive attack).

There are scheduling models in which a single machine is considered. From a practical point of view, direct applications of this exception in forest firefighting are more frequent. Some firefighting processors must operate separately, or they can be viewed as a single processor if they must operate commonly.

Undeniably, the variability of the fire processing time with the other attributes are inextricably involved. There are various ways of regarding this variability. One can consider the processing time of a fire as a random variable or as a fuzzy number that is unidentified in advance.

Based on the characteristics of the area where fire is ignited and on the characteristics of the processor of initial attack, Rachaniotis and Pappis (2006) propose a model in which the processing time of a fire increases if the beginning of the fire containment effort is delayed. This is known in the scheduling field as “deteriorating jobs”.

Received 28 July 2015. Accepted 12 December 2015.

A. Kali, University Djilali Bounaama, Khemis-Miliana, Ain-Defla, Algeria.

**Email for correspondence:** abdesselem.kali@gmail.com.

With the objective to maximize the total value of the burnt areas remaining after the completion of the containment operation, a branch and bound algorithm was presented in Pappis and Rachaniotis (2009). In the case where the objective is to minimize the total weighted tardiness, a lower bound for the problem is given and a heuristic algorithm is used for the extraction of an upper bound of the solution (Rachaniotis and Pappis 2011). Unfortunately, the proposed branch and bound algorithms can only yield optimal results for moderately sized fires. The intricate formulation of the model is among their basic deficiencies. In contrast, the deteriorating model may help to give an insight and stimulation for future, more realistic approximations of real forest management problems that are expected to be encountered (Rachaniotis and Pappis 2011).

A stochastic approach that captures uncertainty and enables a more accurate description of the decisionmaking process may be adopted. Markov decision processes are a widely used tool to model decisionmaking under uncertainty. In the paper, a Markovian decision process for fire processing time evaluation based on the deteriorating model given in Rachaniotis and Pappis (2006) is considered. The objective is the maximization of the total expected discounted value of the burnt areas when all fires are suppressed. In a Markov decision process, the uncertainty in transition probabilities may not be known precisely. In our case, this can happen because quantities in the model of Rachaniotis and Pappis (2006) may have inherent variability. The probabilities may have been obtained via an estimation process. From the well-known "central limit theorem" (Ross 1973), it is natural to consider the normal distribution for representing the uncertainty in the transition probabilities. The dynamic allocation index rule (Gittins and Jones 1974) plays an important role in finding an optimal strategy for the Markovian process. Bellman equations appear naturally in computing this priority rule. There are a number of methods to solve the Bellman equation (e.g., see Sonin (2006)). In our Markovian model, the particular form of the transition matrix allows us to solve the corresponding Bellman equation efficiently and analytically.

The remainder of this paper is composed of seven sections. We begin by giving a description of the deterministic forest firefighting problem, which can be modelled by deteriorating jobs, in section 2. Section 3 discusses considerations for stochastic approach. In section 4, a Markovian model is given for the stochastic scheduling problem when time proceeds in discrete steps. The dynamic allocation index technique is used for determining the optimal non pre-emptive containment policy. This result is given in section 5. We compute the dynamic allocation index in section 6. Section 7 reports on our numerical tests. In section 8, we conclude our work and provide some recommendations for future development.

**2. A deteriorating model for scheduling some forest firefighting tasks**

In the scenario under consideration,  $N$  forest fires are ignited simultaneously, detected, and reported, and one processor of initial attack is assigned for their containment.

All of the  $N$  fires evolve independently, and no constraint is assumed, which imposes the order in which fires must be suppressed.

The affected areas may require greater protection because of their characteristics, e.g., proximity of the population, as well as proximity to other socially, economically, and historically important sites and military sites.

As time elapses, the fire expands if there is a delay in the firefighting efforts.

The surviving value of the area at time  $t$  where fire  $F_i$  is ignited can be estimated by assigning a discounted value  $\alpha^t V_i$ , where  $V_i$  is

a positive-bounded constant and  $\alpha$  lies strictly between 0 and 1. This estimate is done by taking into account all characteristics of concern in the area.

The objective is to minimize the total damage caused by the fires to the burnt areas when all fires are suppressed, i.e., the total discounted value of the surviving areas is maximized.

The processor is of limited transportation capacity  $Q$  (in liters ( $L$ )) and application rate  $\rho$  (in  $L \cdot h^{-1}$ ) and uses water as the means of fire suppression.

The processor is not required to extinguish a fire while busy suppressing another fire, i.e., no pre-emption is allowed.

During the containment of selected fire  $F_i$ , the processor can interrupt its operation as many times as necessary,  $\tau_i$  (in hours) in total, for going to the nearest depository to refill and to come back.

In general, square brackets will be used to denote position in sequence in a permutation schedule. Therefore,  $F_{[i]}$  is the fire that is in the  $i$ th position in suppression sequence.

Let  $T_{0,[i]}$  (in hours) be the time needed for the processor to travel from the base where it is stationed to the spot where the first prioritized fire might be attacked.

Let  $T_{[i],[i+1]}$  (in hours), where  $i = 1, 2, \dots, N-1$ , be the time required for the processor to travel from the front of the suppressed fire  $F_{[i]}$  to the depository to refill and move to the front of the fire  $F_{[i+1]}$ .

The time needed for the processor to travel to suppress a prescribed fire or the time needed for the processor to refill were considered as setup times (as they would be in a typical manufacturing environment).

For various forest firefighting considerations, the suppression of fire  $F_i$ , where  $0 \leq i \leq N$ , might be started at time  $r_i$ . The release time  $r_i$ , where  $r_i > 0$ , can simply represent the fire's initial inaccessibility period. It represents, for example, the preparation time that precedes the containment process.

The total forest area being burnt by fire  $F_{[i]}$  at time  $t$  at which the fire suppression effort starts,  $E_{[i]}(t)$  (in  $m^2$ ), as well as the relative amount of water needed to suppress the fire  $F_{[i]}$ ,  $\omega_{[i]}(t)$  (in  $L \cdot m^{-2}$ ), deteriorates (i.e., increases) at a rate that depends on the forest land that the fire burns, the forest vegetation, and the weather conditions.

$p_{[i]}(t)$  (in hours) is the processing time of fire  $F_{[i]}$  if the "decision" of fire containment is taken at time  $t$ .  $p_{[i]}(t)$  deteriorates (i.e., increases) with  $t$ . The following expression holds (Rachaniotis and Pappis 2006)

$$(1) \quad p_{[i]}(t) = \frac{E_{[i]}(t)\omega_{[i]}(t)}{\rho} + \text{int} \left[ \frac{E_{[i]}(t)\omega_{[i]}(t)}{Q} \right] \tau_{[i]}$$

where "int" is the integer part.

$E_{[i]}(t)$  is given in Rachaniotis and Pappis (2006) by

$$(2) \quad E_{[i]}(t) = \frac{\pi}{\psi_{[i]}} \left[ \frac{A_{[i]}}{B_{[i]} + 1} t^{B_{[i]}+1} + c_{[i]}(t)t \right]^2$$

And  $\omega_{[i]}(t)$  in Katsanos (1978) is of the form

$$(3) \quad \omega_{[i]}(t) = 1.2 \sqrt{A_{[i]} B_{[i]} t^{B_{[i]}-1} + c_{[i]}(t)}$$

where  $A_{[i]}$  and  $B_{[i]}$  are constant coefficients depending on the varying environmental factors of the fire in each area (wind speed, type of fuel, and fuel loading) and (or) the equilibrium rate of spread (McAlpine and Wakimoto 1991).  $c_{[i]}(t)$  is a nonnegative coefficient at time  $t$ , defined as the difference between the forest fire's rate of spread based on the fire behaviour prediction system

Can. J. For. Res. Downloaded from www.nrcresearchpress.com by 105.235.133.123 on 02/04/16 For personal use only.



and the forest fire's rate of spread during the fire's acceleration phase (McAlpine and Xanthopoulos 1998).  $\psi_{[ij]}$  is a constant that depends on the wind speed  $U_{[ij]}$  (in  $\text{km}\cdot\text{h}^{-1}$ ).  $\psi_{[ij]}$  is given in Rachaniotis and Pappis (2006) by the formula

$$\psi_{[ij]} = 0.936e^{0.07127U_{[ij]}} + 0.461e^{-0.043U_{[ij]}}$$

Substituting eqs. 2 and 3 into eq. 1, it follows that

$$(4) \quad p_{[ij]}(t) = \frac{1.2\pi}{\psi_{[ij]}^2} \left[ \frac{A_{[ij]}}{B_{[ij]} + 1} t^{B_{[ij]}+1} + c_{[ij]}(t) \right]^2 \sqrt{A_{[ij]}B_{[ij]}t^{B_{[ij]}-1} + c_{[ij]}(t)} + \text{int} \left\{ \frac{1.2\pi}{\psi_{[ij]}^2} \left[ \frac{A_{[ij]}}{B_{[ij]} + 1} t^{B_{[ij]}+1} + c_{[ij]}(t) \right]^2 \sqrt{A_{[ij]}B_{[ij]}t^{B_{[ij]}-1} + c_{[ij]}(t)} \right\} \tau_{[ij]}$$

Let  $C_{[ij]}$  (in hours) be the completion time of  $F_{[ij]}$  that is the time at which the suppression effort of  $F_{[ij]}$  stops. It is

$$(5) \quad C_{[ij]} = \sum_{j=1}^i p_{[ij]}(t_{[ij]}) + \sum_{j=1}^i (\tau_{[ij-1,j]} + r_{[ij]})$$

where  $t_{[ij]}$  is the decision date for the containment of  $F_{[ij]}$ , with  $t_{[i1]} = 0$  and  $t_{[ij]} = C_{[i,j-1]}$  for  $j = 2, 3, \dots, N$  and  $T_{[ij-1,i]} = T_{0,i}$ .

A fire  $F_{[ij]}$  has escaped initial attack if its containment time  $C_{[ij]}$  exceeds a specific "time limit"  $d_{[ij]}$  (in hours), which can be considered as the fire deadline.

**3. Considerations for stochastic approach**

In the expression of  $p_{[ij]}(t)$  given by eq. 4, the exponent  $B_{[ij]}$  of  $t$  is different in each fire case. This fact makes the problem more complex and difficult to solve. In addition, whatever the reliability of this model, it is important to realize that there is a probability, albeit very small, that the unwanted and unexpected event will happen. This consideration is justified by the risks involved in a potential fire containment failure. Despite the research undertaken, the statistical properties of a forest fire are not fully known.

Fire safety design aims to provide solutions with risk levels that our society can tolerate. For a diverse variety of types of insurance, normal distributions are extremely important in natural sciences for real-valued random variables whose distributions are not known. It can be used to maximize the chance of success for an acceptable level of risk or to minimize the risk for a desired level of success.

Formally, the associated normal density  $f_{[ij]}(t)$  to fire  $F_{[ij]}$  is expressed as

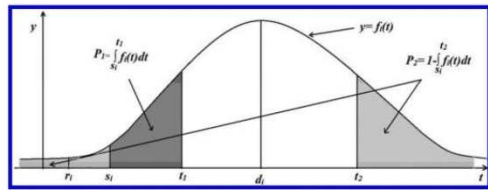
$$(6) \quad f_{[ij]}(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[ -\frac{(t - \mu)^2}{2\sigma^2} \right]$$

where  $\mu$  and  $\sigma$  are the mean and standard deviation, respectively, and  $t$  is the completion time of  $F_{[ij]}$  given by eq. 5.

The event "the fire  $F_{[ij]}$  is controlled by time  $d_{[ij]}$ " must have a considerable probability of occurrence. Consequently, the mean  $\mu$  might be  $d_{[ij]}$ . The deadline  $d_{[ij]}$  cannot be considered as a "hard deadline", as wildfires often exceed this limit. A safety zone ( $d_{[ij]}, d_{[ij]} + \varepsilon$ ) must be considered in which the probability of success in the suppression of  $F_{[ij]}$  has a higher value.  $\varepsilon$  depends naturally on  $s_{[ij]}$ , which is the starting time of fire  $F_{[ij]}$  containment.

The value of the standard deviation  $\sigma$  is related to the level of risk. If we take, for example, 5% as the accepted risk, then two standard deviations represent  $d_{[ij]} - r_{[ij]}$ . The fire-containment fail-

Fig. 1.  $P_1$  and  $P_2$  are the probabilities that fire  $F_i$  is controlled at time  $t_1$  before time  $d_i$  and at time  $t_2$  after time  $d_i$ , respectively.



ure by time  $d_{[ij]} + \varepsilon$ , which is the probability that the intended processor can suppress the fire, is interpreted to be when the fire takes rage and merits additional firefighting efforts (see Fig. 1). The normal density  $f_i(t)$  is used to estimate the probability of controlling the fire  $F_i$  at any time  $t \geq s_i$ . In Fig. 1, the area  $P_1$  under  $f_i(t)$  between  $s_i$  and  $t_1$  represents the probability that  $F_i$  is controlled by time  $t_1$  before  $d_i$ .  $P_1$  increases with  $t_1$ , and when  $s_i$  approaches  $d_i$ , the probability to contain  $F_i$  in  $d_i$  decreases and the safety zone ( $d_i, d_i + \varepsilon$ ) becomes wide. The probability that fire  $F_i$  is controlled in the safety zone must have higher values to that for  $t_1 = d_i$ . This probability decreases over time and coincides at time  $t_2 = d_i + \varepsilon$  with that at  $t_1 = d_i$ . Exceeding  $d_i + \varepsilon$ , the fire is uncontrollable and the probability to contain the fire decreases rapidly. These aforementioned natural data are justified by the expression taken for  $P_2$ . The value of  $\varepsilon$  is then the solution of  $\int_{d_i}^{d_i+\varepsilon} f_i(t) dt = 1 - 2 \int_{s_i}^{d_i} f_i(t) dt$ .

Under the assumption of a stationary probability transition matrix and total expected discounted reward maximization criteria, the problem is a Markov decision problem (Bellman 1956). In theory, the problem can be solved by dynamic programming technique. However, the combinatorial explosion of states puts severe limitations on finding optimal solutions. We solved this problem optimally and efficiently by computing at each decision epoch a "priority" for each unsuppressed fire and selecting the fire that has the larger priority.

**4. A Markovian model for the stochastic problem**

We denote the associated cost-discounted Markov decision process of the stochastic forest firefighting problem by  $S = (\Omega, \Gamma, P, V, \alpha)$ ,  $1 \leq i \leq N$ , with the following special features:

$\Omega_i$  is the state space of  $F_i$ , which may be discrete in our case. Let  $X_i(t)$  be the state of  $F_i$  at time  $t$ ; then, the state of  $S$  at time  $t$  is  $X(t) = (X_1(t), X_2(t), \dots, X_N(t))$ .

Fire  $F_i$  enters some state of  $\Omega_i$  denoted by the symbol  $^*i$  if  $F_i$  is completed.  $^*i$  is the stationary state of  $F_i$  in which no transition to another state can be made.

At time  $t$ , the state  $X_i(t)$  is then either  $^*i$  or it takes a nonnegative integer  $\theta$ , interpreted as the number of time units already spent on  $F_i$ .

The action space  $\Gamma_i$  consists of just two elements, namely  $a_i$  and  $\bar{a}_i$ . Action  $a_i$  will be referred to as "select  $F_i$  for containment" or "maintain the suppression of  $F_i$  if its treatment is started and is not completed". In contrast,  $\bar{a}_i$  will be referred to as "stop the suppression process of  $F_i$ ", i.e., if  $F_i$  is extinguished.

If action  $a_i$  is taken at time  $t$ , then the suppression of fire  $F_i$  makes progress or completes during the interval  $[t, t + 1]$ , and the state of all fires not in receipt of processor remain fixed to zero, i.e., either  $X_i(t + 1) = ^*i$  or  $X_i(t + 1) \geq X_i(t)$  and  $\forall j, j \neq i$  either  $X_j(t + 1) = ^*j$  or  $X_j(t + 1) = X_j(t) = 0$ .

Henceforth, we shall only consider Markovian transitions. The transition probability from state  $X_i(t)$  to state  $X_i(t + 1)$  is

$$(7) \quad P[X_i(t + 1) = ^*i | X_i(t) = ^*i] = 1$$

Can. J. For. Res. Downloaded from www.nrcresearchpress.com by 105.235.133.123 on 02/04/16 For personal use only.

$$(8) \quad P[X_i(t+1) = * | X_i(t) = \theta, a_i] = \begin{cases} \int_{s_i}^{\theta+1} f_i(t) dt & \text{if } s_i \leq \theta \leq d_i \\ 1 - \int_{s_i}^{\theta+1} f_i(t) dt & \text{if } \theta > d_i \end{cases}$$

See Fig. 1 and

$$(9) \quad P[X_i(t+1) = \theta + 1 | X_i(t) = \theta, a_i] = 1 - P[X_i(t+1) = * | X_i(t) = \theta, a_i]$$

where  $X_i(s_i) = 0$  and  $f_i$  is the normal distribution given beforehand in eq. 6.

In decision process theory, a policy is defined as any rule for choosing at each decision epoch that fire to be suppressed.

Remember that in section 2, the optimal policy (if it exists) must not be pre-emptive. Decision times are then the initial time and the fire completion times.

If action  $a_i$  is taken at time  $t$ , an incremental reward  $\alpha^t V_i$  is earned. The current reward  $R(X(t))$  of the process  $S$  is then  $R(X(t)) = \alpha^t V_i$ .

The objective function to be maximized is the total expected discounted reward function over an infinite horizon

$$E \left\{ \sum_{t=0}^{+\infty} R(X(t)) \mid X(0) = 0 \right\}$$

which can be written as

$$(10) \quad E \left\{ \sum_{i=1}^N \alpha^{t_i} V_i \right\}$$

where  $t_i$  is the first decision time at which  $F_i$  is selected for containment.

Note that eq. 10 is well defined because rewards  $V_i$  are bounded and  $\alpha \in ]0, 1[$ .

The general theory of the discounted Markov decision process (Ross 1973) asserts the existence of an optimal policy for the process  $S$  that maximizes the total expected reward (eq. 10), which is deterministic, stationary, Markovian, and satisfies the optimality equations of dynamic programming.

**5. Dynamic allocation index rule for optimal processing policy**

In this section, we denote by  $k$  the last suppressed fire and by  $t$  its completion time  $C_k$ . If none exist, then  $k = 0$  and  $t = 0$ .

If action  $a_i$  is taken first at time  $t$  and action  $a_j$  is taken immediately after it, namely at time  $t + T_{k,t} + r_i + p_i(t)$ , then the sum of discounted rewards obtained from the application of these two actions is

$$\alpha^t V_i + \alpha^{t+T_{k,t}+r_i+p_i(t)} V_j$$

If the action order is interchanged, we obtain

$$\alpha^t V_j + \alpha^{t+T_{k,t}+r_j+p_j(t)} V_i$$

A little algebra shows that the first ordering has the greater reward if

$$\frac{V_i}{1 - \alpha^{T_{k,t}+r_i+p_i(t)}} > \frac{V_j}{1 - \alpha^{T_{k,t}+r_j+p_j(t)}}$$

Using this idea, it is not hard to see that the total expected discounted reward obtained from the  $N$  fires is maximized by allocating at each decision epoch  $t$  the processor to whichever unsuppressed fire has the largest dynamic allocation index  $\gamma_i(t)$ , which is computed as

$$\gamma_i(t) = \frac{V_i}{1 - \alpha^{T_{k,t}+r_i} E\{\alpha^{p_i(t)}\}}$$

Let  $D(t)$  denotes the set of delayed fires up to decision time  $t$ . The following is the main result of this paper.

Theorem 1. At each decision epoch  $t$ , it is optimal to choose action  $a_i$  if and only if

$$\gamma_i[X_i(t)] = \max_{j \in D(t)} \gamma_j[X_j(t)] = \max_{j \in D(t)} \frac{V_j}{1 - \alpha^{T_{k,t}+r_j} E\{\alpha^{p_j(t)}\}}$$

**6. Computing the dynamic allocation index**

In this study, determining  $\gamma_i[X_i(t)]$  passes by the calculation of

$$(11) \quad E\{\alpha^{p_j(t)}\}, \quad j \in D(t)$$

The simple form of the transition matrix of probabilities is the main virtue in computing eq. 11.

Suppose that  $(F_{[1]}, F_{[2]}, \dots, F_{[i-1]})$  is the current permutation of suppressed fires.

Then the current decision time is  $t = C_{[i-1]}$ .

Denoted by  $M_{[i]}^{(0)}$ , the probability generating function of  $p_{[i]}(t)$  is evaluated at  $\alpha$  and conditional on initial state " $X_{[i]}(s_{[i]}) = 0$ ", where  $s_{[i]} = C_{[i-1]} + T_{[i-1],[i]} + r_{[i]}$ , i.e.,

$$(12) \quad M_{[i]}^{(0)} = E\{\alpha^{p_{[i]}(t)} \mid X_{[i]}(s_{[i]}) = 0\}$$

Using the geometric sum, it is easy to show that eq. 12 can be expressed as

$$(13) \quad M_{[i]}^{(0)} = 1 - (1 - \alpha) E \left\{ \sum_{h=0}^{p_{[i]}(t)-1} \alpha^h \mid X_{[i]}(s_{[i]}) = 0 \right\}$$

Let  $H_{[i]}^{(0)}$  denote in eq. 13 the conditional expectation

$$E \left\{ \sum_{h=0}^{p_{[i]}(t)-1} \alpha^h \mid X_{[i]}(s_{[i]}) = 0 \right\}$$

$H_{[i]}^{(0)}$  is obtained by solving the following Bellman equation (Ross 1973):

$$(14) \quad H_{[i]}^{(y)} = 1 + \alpha \sum_{h=y+1, \dots, q-1, *_{[i]}} P_{[i]}^{(y,h)} H_{[i]}^{(h)} \quad y = 0, 1, \dots, q-1$$

where the variables in eq. 14 are defined as

$$\begin{aligned} q &= p_{[i]}(t) \\ H_{[i]}^{(0)} &= 0 \\ P_{[i]}^{(y,h)} &= 0 \quad h > y + 1, h \neq *_{[i]} \\ P_{[i]}^{(y,y+1)} &= P[X_{[i]}(s_{[i]} + y + 1) = y + 1 \mid X_{[i]}(s_{[i]} + y) = y, a_{[i]}] \\ & \quad y = 0, 1, \dots, q-1 \\ P_{[i]}^{(y,*_{[i]})} &= P[X_{[i]}(s_{[i]} + y + 1) = *_{[i]} \mid X_{[i]}(s_{[i]} + y) = y, a_{[i]}] \\ & \quad y = 0, 1, \dots, q-1 \end{aligned}$$

and  $P$  is the transition probability defined in eqs. 7–9. Consequently,

$$\begin{aligned}
 H_{[i]}^{(0)} &= 1 + \alpha P_{[i]}^{(0,1)} H_{[i]}^{(1)} + \alpha P_{[i]}^{(0,*)} H_{[i]}^{(0*)} \\
 H_{[i]}^{(1)} &= 1 + \alpha P_{[i]}^{(1,2)} H_{[i]}^{(2)} + \alpha P_{[i]}^{(1,*)} H_{[i]}^{(0*)} \\
 \dots \\
 H_{[i]}^{(q-1)} &= 1 + \alpha P_{[i]}^{(q-1,q)} H_{[i]}^{(q)} + \alpha P_{[i]}^{(q-1,*)} H_{[i]}^{(0*)} \\
 H_{[i]}^{(q)} &= 1 + \alpha P_{[i]}^{(q,*)} H_{[i]}^{(0*)}
 \end{aligned}
 \tag{15}$$

The structure of eq. 15 allowed us to solve it directly in the following ascending fashion:

$$\begin{aligned}
 H_{[i]}^{(0*)} &= \frac{1}{1 - \alpha} \\
 H_{[i]}^{(q-1)} &= 1 + \frac{\alpha}{1 - \alpha} P_{[i]}^{(q-1,*)} \\
 \dots \\
 H_{[i]}^{(0)} &= 1 + \frac{\alpha}{1 - \alpha} P_{[i]}^{(0,*)} + \sum_{j=1}^{q-1} \alpha^j \left[ \prod_{l=0}^{j-1} P_{[i]}^{(l,l+1)} \right] \\
 &\quad \times \left[ 1 + \frac{\alpha}{1 - \alpha} P_{[i]}^{(j,*)} \right]
 \end{aligned}
 \tag{16}$$

Consequently, the final form of the dynamic allocation index of fire  $F_{[i]}$  is then

$$\gamma_{[i]}(t) = \frac{V_{[i]}}{1 - \alpha^{T_{[i],[i]} + \tau_{[i]}} [1 - (1 - \alpha)] H_{[i]}^{(0)}}
 \tag{17}$$

where  $H_{[i]}^{(0)}$  is given in eq. 16 and  $T_{[i],[i]} = T_{0,[i]}$ .

**7. Numerical tests**

The reported dynamic allocation index, given by eq. 17, was coded and run in Scilab version 5.5.2. All tests were made on a personal computer with an Intel Core i5-2450M CPU at 2.50 GHz with 4 GB of RAM running Windows 7 (64 bits) operating system. All parameters are randomly generated according to the uniform law in specific intervals esteemed using experimental data from McAlpine and Wakimoto (1991) and Rachaniotis and Pappis (2006).

$$A_i \in ]0, 14[ \text{ and } B_i \in ]1, 2[$$

In general,  $c_i(t) \in ]0, 1[$  for  $U_i \in \{0, 1, \dots, 8\}$

$$\begin{aligned}
 \alpha &\in ]10^{-6}, 10^{-4}[ \text{ and } V_i \in ]10^5, 2 \times 10^5[ \\
 \rho &= 6.48 \times 10^6 (\text{in } L \cdot h^{-1}) \text{ and } Q = 3 \times 10^4 (L) (\text{estimations})
 \end{aligned}$$

In this experiment, it is assumed that  $\tau_i = 0.2$  (in hours) and  $T_{0,[i]}$  and  $T_{[i],[i+1]}$  are in  $]0, 1[$  (in hours).

The standard deviation  $\sigma_i$  is taken randomly according to the uniform law, as envisaged for the other parameters in this experimentation, in the set  $\left\{ \frac{d_i - r_i}{4}, \frac{d_i - r_i}{3}, \frac{d_i - r_i}{2}, d_i - r_i \right\}$ , as almost all of the area under the bell curve lies within four standard deviations from the mean. Thus, outliers greater than four standard deviations from the mean will be extremely rare.

In our tests, we were mainly interested in estimating the deadline  $d$  of fire containment and, in respect of this delay, the number of fires that can be surmounted by the processor. For this purpose,  $d$  was raised by "one" from 1 to 50 (in hours). For each  $d$  value, the algorithm has been applied on a variety of instances (1000 instances in total, each of which contains 100 cases of fires).

To get accuracy in the calculation of the dynamic allocation indices, we set the number of fire states to 100.

**Table 1.** Experiment results.

$d$	$\bar{N}$	$\hat{C}$	$\bar{T}$
1	1.22	1.37	1.03
2	2.54	2.37	1.18
3	3.86	3.69	1.31
4	5.18	6.47	1.42
5	5.86	11.40	1.48
6	6.24	22.95	1.49
7	6.3	20.96	1.52
8	6.2	66.35	1.51
9	6.8	120.53	1.53
10	6.72	196.28	1.53
11	7.08	98.76	1.54
12	7.14	156.5	1.54
13	7.08	164.48	1.54
14	7.06	403.24	1.55
15	7.18	1721.6	1.56
16	7.18	486.37	1.56
17	7.2	449.96	1.57
18	7.14	537.2	1.56
19	7.06	839.6	1.57
20	7.34	4260.0	1.58
21	7.08	12791.0	1.57
22	7.96	2397.0	1.6
23	7.24	5620.0	1.6
24	7.1	5000.0	1.59
25	7.24	2748.0	1.58
26	7.7	14101.0	1.59
27	7.5	10661.0	1.6
28	7.1	10352.0	1.59
29	7.24	2801.0	1.6
30	7.7	14101.0	1.59
31	7.56	11212.0	1.61
32	7.12	10577.0	1.6
33	7.32	10197.0	1.589
34	7.66	18542.0	1.58
35	7.76	30272.0	1.61
36	7.6	12106.0	1.60
37	7.14	41890.0	1.59
38	7.36	12334.0	1.58
39	7.8	45222.0	1.60
40	7.62	108125.0	1.59
41	7.18	564072.0	1.59
42	7.36	12334.0	1.58
43	7.82	173442.0	1.59
44	7.66	108284.0	1.62
45	7.18	564072.0	1.61
46	7.88	186788.0	1.62
47	7.68	108302.0	1.62
48	7.2	582567.0	1.662
49	7.38	13026.0	1.62
50	7.92	332857.0	1.634

**Note:**  $d$ , deadline;  $\bar{N}$ , average number of fires suppressed by the processor;  $\hat{C}$ , the average of the containment times for the first fires that were selected by the index rule and escaped the efforts of initial attack;  $\bar{T}$ , the average CPU time consumption.

For each value of  $d$  and for the 1000 instances generated for this delay, we tabulate the average number of fires suppressed by the processor (denoted  $\bar{N}$ ), the average of the containment times for the first fires that were selected by the index rule and escaped the efforts of initial attack (denoted  $\hat{C}$ ), and the average CPU time consumption (in seconds; denoted  $\bar{T}$ ). These results are reported in Table 1. It appears that only seven fires, on average, can be confined within the period of the first 11–50 h. With  $d$  between 11 and 50, the time required to control the eighth fire deteriorates and exceeds all barriers feasible in practice. The calculation time for each instance is about 1.6 s.



## 8. Conclusion

Regardless of the complex structure of the model by Rachaniotis and Pappis (2006) and the inability to solve the problem perfectly, it may be also argued that uncertainty can enter the model because quantities in the model have an inherent variability. We have addressed a novel approach for the optimal solution when uncertainty occurs. A Markovian decision process is developed and the dynamic allocation index technique is used to solve the problem. The optimality of this rule is proved, and its determination is also explained. By field tests, the deadlines of the fires are at most 11 h, and only seven fires can be surmounted by the processor.

In our Markovian process, the rewards are fixed so that the only parameters that vary are the transition probabilities. To our knowledge, the statistical properties of forest fires are not fully known. We use the normal distribution for representing the uncertainty in the transition probabilities. The normal distribution is most often assumed to describe the random variation that occurs in the data from many scientific disciplines with the condition that their distributions are unknown. The choice of the mean to be the deadline of the fire and standard deviation to be governed by the risk of fire suppression failure seem natural in forest firefighting.

A notable direction of study is to consider the problem of scheduling a single firefighting processor under incomplete information about the normal distribution means, as in Katehakis and Robbins (1995), or incomplete information about the fire containment lengths, as in Burnetas and Katehakis (1993), or use the consistent estimates of the transition law and reward, as defined in Burnetas and Katehakis (1997).

In our study, fires are ignited at the same time. The generalization to the case of fires ignited at different times, as depicted in Rachaniotis and Pappis (2006), is quite simple. Thereby, disputing the difficulty of Poissonian arrivals (Blake 1979) seems worthwhile. It is also interesting to discuss the case in which fires are disadvantaged during their containment. In this situation, and unlike the deteriorating models, the suppression of a fire generates a supplementary processing time to complete the fire.

The problem of scheduling parallel identical or nonidentical firefighting processors is treated by Pappis and Rachaniotis (2010) as job scheduling with deteriorating jobs. In this problem, the pre-emption is allowed with a variable number of processors used by a fire over time. An interesting model for fire processing times is formulated depending on the number and type of each processor employed to contain a fire. A stochastic continuation of this model is almost similar to that developed in section 4, and the dynamic allocation index  $\gamma_i$  can be generalized without jeopardizing the optimality of the index rule. Each fire  $F_i$ , which is at time  $t$  in state  $X_i(t)$ , is associated a priority  $\gamma_i[X_i(t)] = \sup_{\sigma > t} (V_j/1 - \alpha^{T_i + \tau} E\{\alpha^\xi\})$ , where  $T_i$  is the time required to move the needed processors from other sites to the spot where fire  $F_i$  is ignited, and  $\xi$  is the stopping time of  $F_i$  and is a decision variable that depends on the fire processing time calculated at time  $t$ . This prioritization allows the decisionmaker to allocate a minimal set of required processors to the fire with the highest index. In a case of a tie, any well-known rule is employed to break it. An important property of dynamic allocation index  $\gamma_i$  is that the supremum is achieved by  $\xi = \min_{\sigma > t} \{t, \gamma_i[X_i(t)] < \gamma_i[X_i(\sigma)]\}$  and by any stopping time  $\sigma > t$  that satisfies  $\sigma \leq \xi$  and  $\gamma_i[X_i(\sigma)] \leq \gamma_i[X_i(t)]$ . At each decision epoch, which corresponds to an optimal stopping time or a completion time or new arrival of a fire, the processing times of uncompleted fires are recalculated for new indexation and new assignments of resources are done. Suitable methodologies for computing  $\gamma_i$  are discussed in Katehakis and Veinott (1987).

In Pappis and Rachaniotis (2010), "when there are more processors available than those already assigned, which are simultane-

ously required by the fires already ignited, a maximal possible number of processors are assigned. Otherwise, processors are shared by fires with equal ratios so that their heights decrease at the same rate. The processors are assigned according to non ascending order of the ratios containment rate/travelling time, which depends directly on the distance from their location to the fire". Please note that the action taken at each decision epoch  $t$  is always optimal with regard to the state resulting from the allocation of residual processors to fires already ignited up to time  $t$ .

## References

- Bellmann, R.E. 1956. A problem in the sequential design of experiments. *Sankhya*, **16**: 221–229.
- Blake, I.F. 1979. An introduction to applied probability. John Wiley and Sons, New York.
- Burnetas, A.N., and Katehakis, M.N. 1993. On sequencing two types of tasks on a single processor under incomplete information. *Prob. Eng. Inf. Sci.* **7**(1): 85–119. doi:10.1017/S0269964800002801.
- Burnetas, A.N., and Katehakis, M.N. 1997. Optimal adaptive policies for Markov decision processes. *Math. Oper. Res.* **22**(1): 222–255. doi:10.1287/moor.22.1.222.
- Conway, R.W., Maxwell, W.L., and Miller, L.W. 1967. *Theory of scheduling*. Addison-Wesley, Reading, Massachusetts.
- Gittins, J.C., and Jones, D.M. 1974. A dynamic allocation index for the sequential design of experiments. In *Proceedings of European Meeting of Statisticians 1972 and in Progress in Statistics*, Edited by J. Gani. North Holland, pp. 241–266.
- Katehakis, M.N., and Robbins, H.E. 1995. Sequential choice from several populations. *In Proc. Natl. Acad. Sci.* **92**: 8584–8585. doi:10.1073/pnas.92.19.8584.
- Katehakis, M.N., and Veinott, A.F., Jr. 1987. The multi-armed bandit problem: decomposition and computation. *Math. Oper. Res.* **12**(2): 262–268. doi:10.1287/moor.12.2.262.
- Katsanos, A. 1978. *Suppression of forest fires by air*. Ministry of Agriculture, Department of Forest Research, Athens, Greece. **40**: 13–73.
- McAlpine, R.S., and Wakimoto, R.H. 1991. The acceleration of fire from point source to equilibrium spread. *For. Sci.* **37**(5): 1314–1317.
- McAlpine, R.S., and Xanthopoulos, G. 1998. Predicted vs. observed fire spread rates in ponderosa pine fuel beds: a test of American and Canadian systems. *In Proceedings of the 10th Conference on Fire and Forest Meteorology*, Ottawa, Ontario, 17–21 April 1989. Forestry Canada, Ottawa, Ontario. pp. 287–294.
- Minas, J.P., Hearne, J.W., and Handmer, J.W. 2012. A review of operations research (OR) methods applicable to wildfire management. *Int. J. Wildland Fire*, **21**(3): 189–196. doi:10.1071/WF10129.
- Pappis, C.P., and Rachaniotis, N.P. 2009. Scheduling a single fire fighting resource with deteriorating fire suppression times and set-up times. *Oper. Res. Int. J.* doi:10.1007/s12351-009-0038-6.
- Pappis, C.P., and Rachaniotis, N.P. 2010. Scheduling in a multi-processor environment with deteriorating job processing times and decreasing values: the case of forest fires. *J. Heuristics*, **16**(4): 617–632. doi:10.1007/s10732-009-9110-x.
- Rachaniotis, N.P., and Pappis, C.P. 2006. Scheduling fire-fighting tasks using the concept of "deteriorating jobs". *Can. J. For. Res.* **36**(3): 652–658. doi:10.1139/x05-267.
- Rachaniotis, N.P., and Pappis, C.P. 2011. Minimizing the total weighted tardiness in wildfire suppression. *Oper. Res.* **11**(1): 113–120. doi:10.1007/s12351-010-0079-x.
- Ross, S.M. 1973. *Applied probability models with optimization applications*. Holden-Day, San Francisco.
- Sonin, I. 2006. The optimal stopping of Markov chain and recursive solution of Poisson and Bellman equations. *In Stochastic Calculus to Mathematical Finance*, pp. 609–621.
- Sullivan, A.L. 2009a. Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. *Int. J. Wildland Fire*, **18**: 349–368. doi:10.1071/WF06143.
- Sullivan, A.L. 2009b. Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. *Int. J. Wildland Fire*, **18**: 369–386. doi:10.1071/WF06142.
- Sullivan, A.L. 2009c. Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. *Int. J. Wildland Fire*, **18**: 387–403. doi:10.1071/WF06144.
- Taylor, S.W., Woolford, D.G., Dean, C.B., and Martell, D.L. 2013. Wildfire prediction to inform fire management: statistical science challenges. *Stat. Sci.* **28**(4): 586–615. doi:10.1214/13-STS451.
- Vorisis, D. 1999. Definition of the needs of suppress groups as concern of the prediction of the behaviour of the wild land fires. *International Symposium, Forest Fires: Needs and Innovations*, EC, DG XII, Athens, Greece. pp. 18–19.
- Weintraub, A., Romero, C., Björndal, T., Epstein, R., and Miranda, J. 2007. *Handbook of operations research in natural resources*. International series in operations research and management science. Springer.

## ANNEXE B

ARTICLES PARUS DANS DES COLLOQUES, SYMPOSIUM  
ET CONGRÈS

- Derbala Ali et Abdesselem Kali. Un Algorithme de détermination des indices d'allocation dans les problèmes d'ordonnements stochastiques. Colloque sur l'Optimisation et les Systèmes d'Information, Session Posters, COSI2013, 9-11 Juin 2013, Alger-Algérie, pp. 1-2.  
<http://www.isima.fr/cosi/cosi2013/acceptedPosters.php>
- Kali Abdesselem et Derbala Ali. Complexity Analysis of Scheduling Linear Deteriorating Jobs in a Single-Machine for Minimum Sum of Completion Times. Actes de la huitième édition du Colloque sur l'Optimisation et les Systèmes d'Information, COSI2011, Session 5C- Ordonnement, Guelma, Algérie, le 24-28 Avril 2011, pp 402-408.  
<http://www.isima.fr/cosi/cosi2011/accepted.pdf>
- Kali Abdesselem et Derbala Ali. A Dynamic Programming Algorithm for Single-Machine JIT Production Scheduling with Random Task Shortening-Rates and a Common Due-Date Assignment. SSCI 2011, IEEE Symposium Series on Computational Intelligence 2011 - CISched - 2011 IEEE Symposium on Computational Intelligence in Scheduling. Paris, France, Avril du 11-15, 2011. Article accepté comme poster.  
<http://ieee-ssci.org/>
- Derbala Ali et Kali Abdesselem. Scheduling deteriorating jobs : Ten open problems. HISAROB2010, Journées Hispano-Algériennes de l'enseignement en Robotique, 30/11/2010 au 02/12/2010, ELCHE, Espagne.



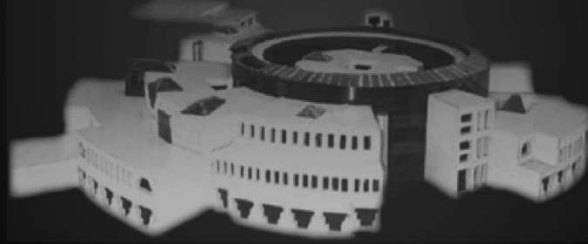


10<sup>e</sup> Colloque sur l'Optimisation et les Systèmes d'Information

**COSI 2013**

09-11 juin 2013, CDTA, Alger, Algérie

Organisé par :  
le Centre de Développement des Technologies Avancées



**Posters**



## Table des matières

Organisation .....	iii
Comité de Pilotage .....	iv
Comité de Programme .....	v
<b>Posters</b>	
Un algorithme de détermination des indices d'allocation dynamiques dans les problèmes d'ordonnement stochastiques .....	1
<i>Ali Derbala and Abdeselem Kali</i>	
Etude et Application d'une Méthode d'Accélération des Contours Actifs Implicites .....	3
<i>Belahcene Kamel and Chilali Ouardia</i>	
Le Nombre broadcast domination dans le graphe 2 section de l'hypergraphe des intervalles, d'un ordre d'intervalles, d'un treillis booléen et d'un poset sans N .....	5
<i>Aouane Mohouhand and Sadi Bachir</i>	
Algorithmes biologiques pour la cryptanalyse du chiffrement Feistel .....	7
<i>Mekhaznia Tahar and M.B. Menai and A. Zidani</i>	
A Fully Dynamic Algorithm for Weakly Chordal Graphs .....	9
<i>Mohamed Amine Boutiche</i>	
On the use of diversity for handling the plethoric answers problem .....	11
<i>Abid Boualem and Allel Hadjali</i>	
Détection collective d'intrusions dans les réseaux étendus .....	13
<i>Abdenacer Nafir, Smaïne Mazouzi and Salim Chikhi</i>	
Towards Semantic Web Service Composition using Immune Algorithm ...	15
<i>Bekkouche Amina, Benslimane Sidi Mohamed, Hassaine Chafik and Merzoug Mohammed</i>	
Multi-criteria optimization for sensor placement problem: Tripartite graph approach .....	17
<i>Abid Sabrina and Haffaf Hafid</i>	
Exploitation des indicateurs spectraux pour la caractérisation de l'état physio hydrique de la végétation en utilisant des données AVHRR .....	19
<i>Yagoub Houria, Belbachir Ahmed Hafid Ahmed Hafid and Benabadji Noureddine</i>	
L'indexation des images Mammaires par les Lois de Zipf .....	21
<i>Asma Saadi and Hayet Farida Merouani</i>	

## Un algorithme de détermination des indices d'allocation dynamiques dans les problèmes d'ordonnancement stochastiques

Ali Derbala<sup>1)</sup>, Abdesselem Kali<sup>2)</sup>

<sup>1)</sup>Laboratoire LAMDA-RO, Université Saad Dahlab de Blida  
BP 270, Route de Soumaa, BLIDA, ALGERIE. Tel et Fax : (+) 213-25-43-36-42,  
[aliderbala@univ-blida.dz](mailto:aliderbala@univ-blida.dz)

<sup>2)</sup> Centre universitaire de Khemis-Miliana, Algérie.

**Résumé :** *Dans un atelier, " N " tâches sont à exécuter sur une machine afin de maximiser l'espérance de la somme des profits prévisionnels. Les temps d'exécution des tâches sont incertains et sont supposés aléatoires de lois connues. Ces problèmes d'ordonnancement sont dits stochastiques. On associe à chaque tâche une priorité dynamique appelée indice d'allocation dynamique et notée I.A.D. A tout instant, on exécute la tâche qui a le plus grand indice. En cas de conflit ou d'égalité entre les plus grands indices, on arbitre entre eux en choisissant une tâche pour exécution selon une règle connue. Ils sont calculés à tout instant et durant l'exécution de la tâche. Cette politique est appelée d'indices. Si l'objectif du problème est une fonction à coûts séparables, la politique d'indices est optimale. Un algorithme de détermination des I.A.D est proposé et est exposé en détail. Dans la bibliographie, au moins trois autres algorithmes existent. Ces quatre algorithmes ont été implémentés et des expérimentations numériques ont été conduites sur un grand nombre d'exemples de problèmes d'ordonnancement. Une étude comparative entre ces algorithmes est fournie. Si le facteur d'actualisation dans la fonction objectif est proche de zéro, notre nouvel algorithme peut prendre en charge des problèmes à cent soixante états. Le temps de calculs est négligeable.*

**Mots-clés :** *Processus Bandit, indice de Gittins, règle d'indice, Ordonnancement Stochastique, Stratégies préemptive et non préemptive.*

### 1 Introduction

"N" tâches sont à exécuter sur une machine afin de maximiser l'espérance de la somme des profits actualisés. Les temps d'exécution des tâches sont supposés aléatoires de lois connues. L'aléatoire ou le stochastique peut représenter, l'incertitude de l'Ordonnanceur sur les temps d'exécution, des erreurs sur les mesures des temps d'exécution, les fluctuations aléatoires dans la fonction objectif ou la vitesse de l'opérateur et / ou de la machine, la non homogénéité des tâches nécessitant de différents temps d'exécution et / ou l'aléatoire en temps requis pour le réglage de la machine pour les différentes tâches à exécuter. On associe à chaque tâche une priorité dynamique appelée indice d'allocation dynamique notée I.A.D. Ces indices sont calculés à tout instant et durant l'exécution des tâches, permettant à l'Ordonnanceur de prendre en considération leurs exécutions et les instants de préemption au fur et à mesure que l'information sur les tâches est recueillie. A tout instant, on exécute la tâche qui a le plus grand indice. En cas de conflit ou d'égalité entre les plus grands indices, on arbitre en choisissant une tâche selon généralement une règle connue de type SPT, LPT, FIFO...etc. Une règle d'indice est une politique qui exécute à tout instant une tâche de plus grand IAD. L'indice d'allocation dynamique s'interprète comme un coût d'arrêt d'exécution de la tâche. Dès qu'une tâche ne rapporte plus le maximum de profit, on l'interrompt et on exécute une autre qui rapporte beaucoup plus. Si l'objectif est une fonction à coûts séparables, une somme de coûts, Gittins

et Jones [1] ont montré que la règle d'indices est optimale. De notre recherche bibliographique, au moins trois algorithmes de détermination des I.A.D sont en notre possession. Il s'agit de celui de Robinson [2], de Varaiya et al.[3] et de Sonin [4]. Ils ont été implémentés par nos soins. Le nombre maximum d'états que ces algorithmes peuvent prendre en charge ne dépassait pas respectivement 45, 143 et 113 états. Ce qui nous a motivé à fournir un nouvel algorithme pouvant prendre en charge des processus à un nombre d'états beaucoup plus élevés. Dans la fonction objectif, maximiser l'espérance de la somme des gains prévisionnels, si le facteur prévisionnel ne dépasse pas la valeur 0.5 une méthode itérative est utilisée pour la résolution des systèmes linéaires qui caractérisent les IAD. Le temps de calculs est négligeable.

Dans la section deux, nous donnons des concepts préliminaires et résultats connus sur les processus bandits. L'existence et la caractérisation des I.A.D sont données respectivement par Gittins [5,6] et Nash [7] et sont énoncés sans preuves. La détermination des I.A.D s'avère d'une importance pratique pour le développement d'une politique optimale d'exécution de tâches. Le nouvel algorithme de détermination des I.A.D est proposé et est exposé en détail à la section trois. Dans la section quatre, une étude comparative entre les quatre algorithmes est élaborée. Les critères de comparaison sont le temps de calcul et le nombre maximum d'états que ces algorithmes peuvent prendre en charge.

#### Références

- [1]. Gittins, J.C. et Jones, D.M., (1972), "A dynamic index for the sequential Design of experiments, *Colloquia Mathematica Janes Bolai*, European meeting of statisticians, Budapest, Hungary, 241-266.
- [2]. Robinson, D.R. (1982), "Algorithms for evaluating the dynamic allocation index", *Operations Research Letters*, volume 1, n°2, 72-74.
- [3]. Varaiya, P., Walrand, J., Buyukkoc, C., (1985), "Extensions of the Multiarmed Bandit Problem: The Discounted Case". *IEEE Trans.Autom. Control*, AC-30, 5, 426-439.
- [4]. Sonin, I. M., (2005), "The optimal Stopping of a Markov Chain, the Generalized Gittins Index, and Recursive Solution of Poisson and Bellman Equations", <http://www.math.uncc.edu/~imsonin>.
- [5]. Gittins, J. C. (1979), "Bandit processes and dynamic allocation indices (with discussion)", *J. Roy. Statist. Soc. Ser. B* 41, 2, 148-177.
- [6]. Gittins, J.C., (1989), "Multi-armed Bandit Allocation Indices", John Wiley & Sons.
- [7]. Nash, P., (1973), "Optimal allocation of resources between research projects", *Ph.D. Thesis*, Cambridge University.

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
 UNIVERSITÉ 8 MAI 1945 – GUELMA, ALGÉRIE  
 Faculté des Mathématiques, Informatique et Sciences de la Matière  
 Département d'informatique



**LIRIS**  
France

**LMAM**

*LabSTIC*  
Université 8 Mai 1945

**PIMIS**



# COSI 2011

Proceeding de la 8ème édition du Colloque sur  
 L'Optimisation et les Systèmes d'Information

24 -27 Avril, Guelma, Algérie

## Sponsors

<b>SARL AGROSATI</b>	<b>SARL BOUKABOU</b>	<b>SARL FENDJEL</b>	<b>DIWAN INFORMATIQUE</b>	<b>PROMO. IMMOB. ZENACHE</b>
<b>SARL EL-BARAKA</b>	<b>GROUPE BENAMOR</b>	<b>ETB MOKHNACHE</b>	<b>TRANSPORT HAZEM(TVH)</b>	<b>COMPLEXE BOUCHAHRINE</b>
		<b>SARL BORDJIBA</b>	<b>CONSTANTINE MEUBLE</b>	



Optimization of problem Min-Max, <i>Ticherfatine Samira, Aidene Mohamed</i>	242
Une note sur la coloration dominante, <i>Hocine Boumediene Merouane, Mustapha Chellali</i>	254
La b-coloration des graphes de Spider complets, <i>Zoham Zémir, Mostafa Blidia</i>	260
<b>Session 4B- Traitement d'images</b>	280
Une méthode rapide et efficace pour le cryptage évolutionnaire d'images, <i>Ismahane Souici, Hamid Seridi</i>	281
Compression des Images basée sur les essaims particulaires et la recherche taboue, <i>Mansouri Douelkefel, Benamrane Nacéra</i>	292
<b>Session 5A - Optimisation multicritères</b>	302
Weak pseudo-invexity in multiobjective programming, <i>Hachem Slimani, Mohammed Saïd Radjef</i>	303
A leximin linear approach for solving multicriteria package upgradability problem, <i>Noureddine Aribi, Yahia Lebbah</i>	317
Développement d'une Métaheuristique Hybride pour la Résolution d'un Problème de Job Shop Flexible Multicritère au niveau du Complexe Cevital de Béjaïa., <i>Mohammed Saïd Radjef, Naouel Halimi, Kahina Bouchama, Assia Amer</i>	329
A Novel Method for Integer Chance Constrained Problems with Multiple Objective, <i>Fatima Bellahcene</i>	342
<b>Session 5B - Représentation des connaissances et applications</b>	351
Un Système de Classification basé sur la Logique Floue et la Recherche Locale pour la Détection d'Intrusions, <i>Dalila Boughaci, Samia Bouhali, Selma Ordeche</i>	352
Service de tolérance aux fautes dans les réseaux Ad hoc à base de système multi agents, <i>Esmâ Insaïf Djebbar, Ghalem Belalem</i>	364
systèmes immunitaires artificiels pour la détection d'intrusions, <i>Meriem Zekri, Labiba Souici-Meslati</i>	377
Génération automatique des modèles de services à partir des modèles de processus métiers : Approche dirigée par les ontologies, <i>Mokhtar Soltani, Sidi Mohamed Benslimane</i>	389
<b>Session 5C - Ordonnancement</b>	401
Complexity Analysis of Scheduling Linear Deteriorating Jobs in a Single-Machine for Minimum Sum of Completion Times, <i>Abdesselem Kali, Ali Derbala</i>	402

## Complexity Analysis of Scheduling Linear Deteriorating Jobs in a Single-Machine for Minimum Sum of Completion Times

Abdesselem Kali <sup>1,1</sup> and Ali Derbala <sup>1,2</sup>

<sup>1</sup>LAMDA-RO laboratory, University Saad Dahlab of Blida. BP 270, Route de  
Soumaa, Blida -Algeria.

<sup>1</sup>[aeskali@gmail.com](mailto:aeskali@gmail.com)

<sup>2</sup>[aliderbala@univ-blida.dz](mailto:aliderbala@univ-blida.dz)

**Abstract:** We consider the single-machine scheduling problem which has the natural character that the processing-time of the job is an increasing linear function dynamically determined by the starting time of its processing. This phenomenon is known as linear deteriorating jobs. The scheduling objective is the total completion time minimization. The computationally of the problem is still intractable and open question. We prove that the problem is NP-Complete in the ordinary sense.

**Keywords:** Deterministic Single-Machine Scheduling; Linear Deteriorating Jobs; Total Completion Time Minimization; Complexity.

### 1 Introduction

In this paper, a single-machine is used to process a set of  $n$  independent jobs. All jobs are simultaneously available with no ready times or deadlines at time 0 and can be processed without interruption. Each job  $i$  is characterized by a normal processing-time  $a_i > 0$  interpreted as the length of time required to complete the job if it is scheduled first, and a parameter  $b_i > 0$  depending on the job starting time  $S_i$ , interpreted as the growth rate of its processing-time. The actual processing-time  $p_i$  increases linearly with its starting times  $S_i \geq 0$  and is given by  $p_i(S_i) = a_i + b_i S_i$ . Similar to the classical scheduling problems,  $a_i$  is assumed to be positive integers. The processing rates  $b_i$  are not integers in many practical cases,  $b_i$  is allowed to be a positive rational number. Let  $y \geq 0$  denote a given threshold, the flow time problem is to decide whether there is a feasible schedule with  $\sum_{i=1}^n C_i \leq y$ . Adapting the three-field notation  $\alpha|\beta|\gamma$  introduced by Graham et al. [1] used to describe the machine characteristics, job characteristics, and performance measure of interest, the above problem is denote as  $1|p_i(S_i) = a_i + b_i S_i|\sum C_i$ . The problem was formulated

for the first time in [2] and its time complexity is yet an open question even if  $a_i = 1$  for  $1 \leq i \leq n$  [3, 4].

This paper solves the complexity of the problem  $1|p_i(S_i) = a_i + b_i S_i| \sum C_i$ . We show that the problem is NP-complete in ordinary sense by a reduction from Subset Product Problem.

## 2 Computational Complexity

We prove that the problem  $1|p_i(S_i) = a_i + b_i S_i| \sum C_i$ , is NP-hard. The associated decision problem is the following:

**SDC:** Given a single-machine,  $n$  jobs,  $a_i \in \mathbb{N}^+$  ( $i = 1, 2, \dots, n$ ),  $b_i \in \mathbb{Q}_+^*$  ( $i = 1, 2, \dots, n$ ) and  $y \in \mathbb{R}_+$ . Is there a schedule such that  $\sum_{i=1}^n C_i \leq y$ ?

Our proof of NP-completeness proceeds by a reduction in polynomial time from "Subset Product" which was shown to be NP-complete in the ordinary sense (see [5] and the correction of the complexity status in Johnson [6]) into problem **SDC**.

The Subset Product Problem **SP** can be stated as follows:

**SP:** Given  $m+1$  positive integers  $x_1, x_2, \dots, x_m, B$  such that  $\prod_{i=1}^m x_i = B^2$ . Is there a subset  $Y$  of the set  $X = \{x_1, x_2, \dots, x_m\}$  such that

$$\prod_{i \in Y} x_i = \prod_{i \in X \setminus Y} x_i = \sqrt{\prod_{i \in X} x_i} \equiv B?$$

In the instances of **SP**, we can omit the element  $x_i = 1$  because it will not affect the product of any subset. Therefore, without loss of generality, we can assume that  $x_i \geq 2$  for every  $x_i \in X$ .

Our first task in proving the NP-Completeness of **SDC** is to show it lies in NP. But this is obviously true. To check if a guessed schedule has  $\sum_{i=1}^n C_i \leq y$  is very easy and can clearly be accomplished in a number of operations bounded by a polynomial in  $n$ . So **SDC** lies in NP.

Our next aim is to show that in polynomial time we can turn any **SP** into an equivalent **SDC** such the former has a yes answer if and only if the latter does.

In order to show that **SP** can be reduced to **SDC**, we construct the corresponding instance of **SDC**, as follows:

- We sort respectively  $Y$  and  $X \setminus Y$  in the non-decreasing order and let  $(\beta_1, \beta_2, \dots, \beta_l)$  and  $(\beta_{l+1}, \beta_{l+2}, \dots, \beta_m)$  denote respectively the sequences obtained for  $Y$  and  $X \setminus Y$ . These transformations can be done in  $O(\log l(m-l))$  where  $l = |Y|$ .
- Given a single machine and  $n = m+1$  jobs available for processing at time zero. The parameters of  $l$  jobs constructed on the basis of the elements from the subset  $Y$  are given as:  
 $a_i = \omega_i \prod_{k=1}^i \beta_k$  and  $b_i = \beta_i - 1$  ( $i = 1, 2, \dots, l$ ), where the sequence  $(\omega_i)$  is given as:  $\omega_1 = 1 - \frac{1}{\beta_1}$  and  $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$ ;  $i = 2, \dots, l$ .
- The parameters of  $(m-l)$  jobs constructed on the basis of the elements from the set  $X \setminus Y$ , are given as:  
 $a_i = \omega'_i \prod_{k=l+1}^i \beta_k$  and  $b_i = \beta_i - 1$  ( $i = l+1, \dots, m$ ), where the sequence  $(\omega'_i)$  is set as  $\omega'_{l+1} = 1 - \frac{1}{\beta_{l+1}}$  and  $\omega'_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$ ;  $i = l+2, \dots, m$ .



- And the parameters of the  $(m + 1)$  job, called further the extra job, are given as  $a_e = \prod_{k=1}^{l-1} \beta_k$  and  $b_e = \frac{1}{\prod_{k=1}^{l-1} \beta_k}$ .
- Finally, we define a threshold  $y = (3 + \lambda)B + \left(1 - \frac{1}{\beta_l}\right)\lambda - 1 - \frac{1}{\beta_l}$  where  $\lambda = \sum_{i=l+1}^m \prod_{k=l+1}^i \beta_k$ .

The above construction can be accomplished in polynomial time, i.e., in  $O(m)$ . Under the assumption that the normal processing time  $a_i$  of job  $i$  is a positive integer, jobs with indices respectively in  $Y$  and  $X \setminus Y$  must be scheduled in the non-decreasing order of the deterioration rates  $b_i$ . More precisely, we have the following result:

**Preliminary 1.** For  $i = 2, \dots, l$  (respectively, for  $i = l + 2, \dots, m$ )  $a_i$  is a positive integer if, and only if,  $b_i \geq b_{i-1}$ .

**Proof:**

Since, for  $i = 2, \dots, l$ ,  $a_i = \omega_i \prod_{k=1}^i \beta_k$  where  $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$  and  $b_i = \beta_i - 1$ , it is easy matter to see that:

$$a_i = (b_i - b_{i-1}) \prod_{k=1}^{i-2} \beta_k.$$

With the fact that  $\prod_{k=1}^{i-2} \beta_k > 0$ , we conclude that  $a_i \geq 0$  if, and only if,  $b_i \geq b_{i-1}$ . For  $i = l + 2, \dots, m$  the proof is similar to that for  $i = 2, \dots, l$  and hence is omitted.

*Remark.* In view of the condition that,  $b_{l+1}, b_{l+2}, \dots, b_m$  is in non-decreasing order, the sum  $\lambda = \sum_{i=l+1}^m \prod_{k=l+1}^i (1 + b_k) = \sum_{i=l+1}^m \prod_{k=l+1}^i \beta_k$  is minimized [7].

In the instance of **SDC**, if  $\prod_{i \in Y} \beta_i \geq \prod_{i \in X \setminus Y} \beta_i$  and accordingly to preliminary 1, the jobs with indices in  $Y$  are processed first in the non-decreasing order of  $b_i$ , then the extra job, and then the remaining jobs in the non-decreasing order of  $b_i$  as well. Else, the jobs constructed on the basis of the elements from the set  $X \setminus Y$  are performed first. Without loss of generality, we assume in what follows that  $\prod_{i \in Y} \beta_i \geq \prod_{i \in X \setminus Y} \beta_i$  and hence the growth rate of the extra job is smallest as possible.

Notice that for the instance of **SDC**, the completion time of the job with index respectively in  $Y$  and  $X \setminus Y$  can be, by induction with respect to  $i$ , expressed as follows:

$$C_i = \sum_{j=1}^i a_j \prod_{k=j+1}^i (1 + b_k), \text{ for } 1 \leq i \leq l,$$

$$C_i = \sum_{j=l+1}^i a_j \prod_{k=j+1}^i (1 + b_k) + C_e \prod_{k=l+1}^i (1 + b_k), \text{ for } l + 1 \leq i \leq n.$$

Since the job processed before the extra job is  $l$ , the completion time of the extra job can be written as:

$$C_e = a_e + (1 + b_e)C_l.$$

Using the above established expressions, the completion time of each job in the constructed schedule of **SDC** is given in the next preliminary, easily proven by simple replacing  $a_i$  and  $b_i$  by their definitions in  $Y$  and  $X \setminus Y$  respectively:

**Preliminary 2.** For the jobs constructed on the basis of the elements from the subset  $Y$ , the completion time is  $C_i = (\prod_{k=1}^i \beta_k) \left(1 - \frac{1}{\beta_i}\right)$ .

For the extra job, we obtain  $C_e = (\prod_{k=1}^l \beta_k) + 1 - \frac{1}{\beta_l}$ .

And for the jobs constructed on the basis of the elements from the set  $X \setminus Y$ , we obtain  $C_i = \left(1 - \frac{1}{\beta_i}\right) (\prod_{k=i+1}^l \beta_k) + \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=i+1}^l \beta_k) + (\prod_{k=1}^i \beta_k)$ .

**Proof:**

Recall that the parameters of  $l$  jobs with indices in  $Y$  are the following:

for  $i = 1, \dots, l$ ,  $a_i = \omega_i \prod_{k=1}^i \beta_k$ ,

where  $b_i = \beta_i - 1$ ,  $\omega_1 = 1 - \frac{1}{\beta_1}$  and  $\omega_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$  for  $i = 2, \dots, l$ .

And using the equality  $C_i = \sum_{j=1}^i a_j \prod_{k=j+1}^i (1 + b_k)$ , we get:

$$\begin{aligned} C_i &= \sum_{j=1}^i (\omega_j \prod_{k=1}^j \beta_k) \prod_{k=j+1}^i \beta_k \\ &= \sum_{j=1}^i \omega_j \prod_{k=1}^i \beta_k \\ &= (\prod_{k=1}^i \beta_k) \sum_{j=1}^i \omega_j. \end{aligned}$$

It is simple to show that  $\sum_{j=1}^i \omega_j = 1 - \frac{1}{\beta_i}$ , thus:

$$C_i = \left(1 - \frac{1}{\beta_i}\right) (\prod_{k=1}^i \beta_k).$$

For the extra job, we have:

$$C_e = a_e + (1 + b_e) C_l.$$

Since  $a_e = \prod_{k=1}^{l-1} \beta_k$  and  $b_e = \frac{1}{\prod_{k=1}^l \beta_k}$ , then:

$$\begin{aligned} C_e &= (\prod_{k=1}^{l-1} \beta_k) + \left(1 + \frac{1}{\prod_{k=1}^l \beta_k}\right) \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=1}^l \beta_k) \\ &= (\prod_{k=1}^l \beta_k) + 1 - \frac{1}{\beta_l}. \end{aligned}$$

Remember that the parameters of remaining  $(m - l)$  with indices in  $X \setminus Y$ , are set as:

$a_i = \omega'_i \prod_{k=i+1}^m \beta_k$  and  $b_i = \beta_i - 1$  ( $i = l + 1, \dots, m$ ) where  $\omega'_{l+1} = 1 - \frac{1}{\beta_{l+1}}$  and  $\omega'_i = \frac{1}{\beta_{i-1}} - \frac{1}{\beta_i}$ ;  $i = l + 2, \dots, m$ .

then

$$\begin{aligned} C_i &= \sum_{j=i+1}^m a_j \prod_{k=j+1}^m (1 + b_k) + C_e \prod_{k=i+1}^m (1 + b_k) \\ &= \sum_{j=i+1}^m \omega'_j (\prod_{k=i+1}^j \beta_k) (\prod_{k=j+1}^m \beta_k) + \left[ (\prod_{k=1}^l \beta_k) + 1 - \frac{1}{\beta_l} \right] (\prod_{k=i+1}^m \beta_k) \\ &= (\prod_{k=i+1}^m \beta_k) \sum_{j=i+1}^m \omega'_j + (\prod_{k=1}^l \beta_k) + (\prod_{k=i+1}^m \beta_k) \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=i+1}^m \beta_k) \end{aligned}$$

Recognizing that  $\sum_{j=i+1}^m \omega'_j = 1 - \frac{1}{\beta_i}$  hence:

$$C_i = \left(1 - \frac{1}{\beta_i}\right) (\prod_{k=i+1}^m \beta_k) + \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=i+1}^m \beta_k) + (\prod_{k=1}^l \beta_k).$$

Using the results established in preliminary 2, the next preliminary gives the expression of the total completion time for the constructed schedule.

**Preliminary 3.** The total completion time for the constructed schedule is given as:  
 $\sum C_i = 2(\prod_{k=1}^l \beta_k) + (\prod_{k=l+1}^m \beta_k) - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + (\prod_{k=1}^l \beta_k) \right] \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k)$ .

**Proof:**

The sum of completion times for constructed schedule can be expressed as follows:

$$\sum C_i = \sum_{i=1}^l C_i + C_e + \sum_{i=l+1}^m C_i.$$

Applying preliminary 2, we thus obtain:

$$\begin{aligned} \sum C_i &= \\ \sum_{i=1}^l \left(1 - \frac{1}{\beta_i}\right) (\prod_{k=1}^i \beta_k) &+ (\prod_{k=1}^l \beta_k) + 1 - \frac{1}{\beta_l} + \sum_{i=l+1}^m \left[ \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=l+1}^i \beta_k) + \right. \\ &\left. \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=l+1}^i \beta_k) + (\prod_{k=1}^l \beta_k) \right]. \end{aligned}$$

Since  $\sum_{i=1}^l (\prod_{k=1}^i \beta_k) \left(1 - \frac{1}{\beta_i}\right) = (\prod_{k=1}^l \beta_k) - 1$  and  $\sum_{i=l+1}^m \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=l+1}^i \beta_k) = (\prod_{k=l+1}^m \beta_k) - 1$ , it follows:

$$\sum C_i = (\prod_{k=1}^l \beta_k) - 1 + (\prod_{k=1}^l \beta_k) + 1 - \frac{1}{\beta_l} + (\prod_{k=l+1}^m \beta_k) - 1 + \sum_{i=l+1}^m \left[ \left(1 - \frac{1}{\beta_l}\right) (\prod_{k=l+1}^i \beta_k) + (\prod_{k=1}^l \beta_k) \right]$$

That is :

$$\begin{aligned} \sum C_i &= \\ 2(\prod_{k=1}^l \beta_k) &+ (\prod_{k=l+1}^m \beta_k) - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + (\prod_{k=1}^l \beta_k) \right] \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k). \end{aligned}$$

Now we shall show that answering **SP** is equivalent to answering **SDC**. Before we can do so, we show that the following two statements hold:

**Lemma 1.** If the given instance of **SP** has a solution then the constructed instance of **SDC** also has one.

**Proof:**

Let a subset  $Y$  such that  $\prod_{i \in Y} \beta_i = \prod_{i \in X \setminus Y} \beta_i = \sqrt{\prod_{i \in X} \beta_i} = B$ .

The schedule **SDC** is defined as follows: first we execute the jobs with indices in the subset  $Y$  in the non-decreasing order of  $b_i$ , then the extra job, and then the remaining jobs in the non-decreasing order of  $b_i$  as well, i.e., jobs with indices in the set  $X \setminus Y$ .

The sum of completion times has then the following value:

$$\begin{aligned} \sum C_i &= 2(\prod_{k=1}^l \beta_k) + (\prod_{k=l+1}^m \beta_k) - 1 - \frac{1}{\beta_l} + \\ &\left[ \left(1 - \frac{1}{\beta_l}\right) + (\prod_{k=1}^l \beta_k) \right] \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k) \\ &= 3B - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + B \right] \lambda, \end{aligned}$$

where  $\lambda = \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k)$ .

It follows that:

$$\begin{aligned} \sum C_i &= (3 + \lambda)B + \left(1 - \frac{1}{\beta_l}\right) \lambda - 1 - \frac{1}{\beta_l} \\ &= y. \end{aligned}$$

Therefore, for the schedule considered we have  $\sum C_i \leq y$ , as required.

It has been shown that the sum of completion times for **SDC** is not greater than  $y$ . Thus, we proved that if **SP** has a solution **SDC** also has one.

**Lemma 2.** If the given instance of **SP** has no solution then the constructed instance of **SDC** has no solution either.

**Proof:**

If the instance of **SP** has no solution then the product of elements in both subsets  $Y$  and  $X \setminus Y$  is not equal.

Then in view of the condition  $\prod_{i \in Y} \beta_i \geq \prod_{i \in X \setminus Y} \beta_i$  made in the construction of **SDC**, we get:

$$\prod_{i \in Y} \beta_i > \prod_{i \in X \setminus Y} \beta_i.$$

Let  $\alpha$  denote the product  $\prod_{i \in Y} \beta_i$ , it follows that:

$$\alpha > B,$$

and then:

$$\prod_{i \in X \setminus Y} \beta_i = \frac{B^2}{\alpha}.$$

Notice that  $\alpha$  and  $\frac{B^2}{\alpha}$  are positive integers.

We shall show now that, the sum of completion times is greater than  $y$ , i.e.,  $\sum C_i > y$ .

We have from preliminary 3:

$$\sum C_i = 2(\prod_{k=1}^l \beta_k) + (\prod_{k=l+1}^m \beta_k) - 1 - \frac{1}{\beta_l} +$$

$$\left[ \left(1 - \frac{1}{\beta_l}\right) + (\prod_{k=1}^l \beta_k) \right] \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k),$$

then:

$$\sum C_i = 2\alpha + \frac{B^2}{\alpha} - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + \alpha \right] \lambda > 3B - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + B \right] \lambda = y,$$

where  $\lambda = \sum_{i=l+1}^m (\prod_{k=l+1}^i \beta_k)$ .

The last inequality is true, since the function  $f(x) = (2 + \lambda)x + \frac{1}{x} - (3 + \lambda)$  is strictly positive over the interval  $]1; +\infty[$ :

Indeed, the inequality:

$$2\alpha + \frac{B^2}{\alpha} - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + \alpha \right] \lambda > 3B - 1 - \frac{1}{\beta_l} + \left[ \left(1 - \frac{1}{\beta_l}\right) + B \right] \lambda$$

can be written as:

$$(2 + \lambda) \frac{\alpha}{B} + \frac{B}{\alpha} - (3 + \lambda) > 0.$$

With a change of variable  $x = \frac{\alpha}{B}$  (where  $\alpha > B$ ), we obtain the expression and the domain of definition of the above function.

Thus, it has been shown, that if there is no solution for **SP**, then there is not one for **SDC** either.

Lemmas 1 and 2 immediately lead to the following theorem:

**Theorem.** The problem  $1|p_i(S_i) = a_i + b_i S_i | \sum C_i$  is NP-complete in the ordinary sense.

### Conclusion

The NP-completeness of the single machine scheduling problem with deteriorating jobs for sum of completion times minimization has been shown NP-Complete in the ordinary sense. In this problem, the processing time of a job is a non-decreasing linear function dependent on the starting time of the job processing. For further research, it is suggested to investigate the model  $1|p_i(S_i) = 1 + b_i S_i|\sum C_i$  and for the same objective, the non-increasing linear function model.

### References

1. Graham R.L., Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, N° 5, 287–326 (1976).
2. Mosheiov G.: V-shaped policies for scheduling deteriorating jobs, *Oper. Res.* 39 979–991 (1991).
3. Cheng T.C.E., Ding Q., Lin B.M.T.: A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*. N° 152 1–13 (2004).
4. Gawiejnowicz S.: *Time-dependent Scheduling*. Springer-Verlag Inc, New York, (2008).
5. Garey M.R., Johnson D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman: New York, (1979).
6. Johnson D.S.: The NP-completeness column: an ongoing guide, *J. Algorithms* N° 4 393–405 (1981).
7. Mosheiov G.: Scheduling jobs under simple linear deterioration. *Computers and Operations Research*. 21 (6), 653–659 (1994).

# A Dynamic Programming Algorithm for Single-Machine JIT Production Scheduling with Random Task Shortening-Rates and a Common Due-Date Assignment

Abdesselem Kali  
LAMDA-RO laboratory,  
University Saad Dahlab of Blida.  
BP 270, Route de Soumaa, Blida -Algeria.  
[aeskali@gmail.com](mailto:aeskali@gmail.com)

Ali Derbala  
LAMDA-RO laboratory,  
University Saad Dahlab of Blida.  
BP 270, Route de Soumaa, Blida -Algeria.  
[aliderbala@univ-blida.dz](mailto:aliderbala@univ-blida.dz)

*Abstract—This work extends, in (JIT) production, the previous research on static shortening-rates and stochastic compressions, to include uncertainty of shortening-rates and combine for compressions, stochastic with dynamic aspect. The analysis is carried out under a common due-date which is decision variable. The objective is to find jointly the optimal common due-date, the optimal task sequence and the optimal processing-time compressions so as to minimize the total penalty based on the expected task earliness/tardiness, compression and common due-date. It is shown that expected of the penalty function for each task is “monotone about” and hence any optimal schedule which minimizes the sum of expected penalties must be V-shaped. To search for exact solution, some properties of the problem have been used to develop an efficient dynamic programming algorithm with  $O(n \log n)$  time complexity.*

*Index Terms— Single-Machine Scheduling; Random Shortening-Rates; Common Due-Date Assignment; Stochastic Just-In-time Production Scheduling; Sum of Expected Earliness/Tardiness, Compression and Common Due-Date Penalties; Monotony About; V-Shaped Sequence; Dynamic Programming Algorithm.*

## I. INTRODUCTION

The Just-In-Time (JIT) production philosophy basically means to produce the necessary units in the necessary quantities at the necessary time. Very briefly, while early tasks result in inventory holding costs, late tasks results in penalties, such as loss customer's goodwill and loss of orders. The main purpose of JIT system is to reduce the earliness and tardiness penalties associated with production processes thus improving the total of a company. Prescribing a common due-date might represent a situation where several items constitute a single customer's order, or it might reflect an assembly environment in which the components should all be ready at the same time in order to avoid staging delays. Treating the common due-date as decision variable reflects the practice in some shops of setting due-date internally, as target to guide the progress of shop floor activities. Add the due-date penalty to the objective function makes

practical sense suppose that there is a given parameter  $d$  that represents a maximally acceptable due-date.

In many manufacturing systems, tasks can be accomplished in shorter durations by increasing additional resources. Then task may also incur a processing-time compression penalty. In the paper, the time required to process the task is dynamically determined by the starting time of its processing and its shortening-rate. The task shortening-rate, which determine the task's (actual) processing-time, depend in practice on task's characteristics and additional efforts introduced. Note only the use of time dependent processing-times is justified, but also it provides more realistic models (see, e.g., [1] and references therein). So far as we know, all research to date has been concerned with deterministic shortening-rate (see, e.g., [2], [3], [4], [5], [6], [7]). However, more often than not, the deterministic formulation alone may correspond to a somewhat oversimplified picture of reality, though they can take great advantage of computational convenience. Task shortening-rate is usually unknown in advance, is unavoidably to include uncertainty, and cannot be predicted before hand. It should, however, be noted that studies on problems with random processing-time compressions was initiated by [8]. To the best of our knowledge, the case when the task processing-time compression is stochastic and dynamically determined by the starting time of the task has not studied.

In this study, the problem of “scheduling a set of independent tasks on a single-machine with random shortening-rates and a common due-date which is decision variable” is considered. The aim is to determine a combination of the optimal common due-date, the optimal schedule and the optimal processing-time compressions so as to minimize the sum of expected earliness/tardiness, compression and common due-date penalties.

The remaining parts of this paper are organized as follows. In the next section, we formulate in (JIT) production, the single-machine scheduling problem as one that task shortening-rates of processing-times are independent and identically distributed random variables.



Section 3 gives, for a class of deterministic single-machine, the relationship between V-shaped optimal sequence and the monotony about of the penalty function. Several elementary properties of the problem are established in section 4. In section 5, some properties of the problem have been utilized to develop an efficient dynamic programming algorithm for finding exact solution. We conclude this work with remarks and further research.

## II. PROBLEM FORMULATION

In this paper, a single machine is used to process a set of given independent tasks at a time. It is assumed that all tasks are simultaneously available at time 0 and can be processed without interruption on the machine. The optimal common due-date  $\delta^*$  is a decision variable and to be determined. Each task ideally should be completed exactly at this due-date. Penalties are incurred whenever a task is completed earlier or later than time  $\delta^*$ .

In addition, we focus on a task system consisting of  $n$  independent tasks, where the task  $i$  is characterized by a normal processing-time  $a_i$  and a shortening-rate  $b_i$ . Here  $a_i$  is deterministic, nonnegative real and  $b_i$ ,  $i = 1, 2, \dots, n$ , are independent and identically distributed (i.i.d) random variables (r.v) following an arbitrary probability density  $dH(\cdot)$  with a finite mean  $\mu$ . The actual processing-time of the task  $i$  depends on its starting time  $S_i \geq 0$  and is given by:

$$p_i(S_i) = a_i - b_i S_i.$$

We assume that the shortening-rates satisfy the following condition:

$$0 < b_i < \frac{a_{min}}{(\sum_{i=1}^n a_i - a_{min})},$$

where  $a_{min} = \min_{1 \leq i \leq n} a_i$ .

The above condition ensures that all task shortening-rates are distributed on a same interval, the decrease of each task processing-time is less than one unit for every unit delay in its starting time and all task processing-times are positive in a feasible schedule. (see [2] for detailed explanations).

For a given schedule  $\sigma$ , the expected earliness/tardiness, compression and due-date penalties objective function can be written as:

$$f(\sigma, \delta) = \mathbb{E} \sum_{i=1}^n (\alpha E_{\sigma(i)} + \beta T_{\sigma(i)} + \theta z_{\sigma(i)} + \gamma \delta),$$

in which  $\mathbb{E}$  represents the expectation,  $E_{\sigma(i)}$  and  $T_{\sigma(i)}$  represent the earliness and tardiness of the task  $\sigma(i)$  respectively with  $E_{\sigma(i)} = \max\{0, \delta - C_{\sigma(i)}\}$ ,  $T_{\sigma(i)} = \max\{0, C_{\sigma(i)} - \delta\}$ ,  $C_{\sigma(i)}$  is the completion time of the task  $\sigma(i)$ ,  $z_{\sigma(i)} = b_{\sigma(i)} S_{\sigma(i)}$  is the processing-time compression of the task  $\sigma(i)$  while  $\alpha, \beta, \theta, \gamma > 0$  are

respectively the unit earliness, tardiness, compression and due-date penalty.

The per unit costs involved in this paper (earliness, tardiness, compression and common due-date) are all linear. The linear cost functions present a case that is more tractable than that occurring with nonlinear costs. The insight gained from the linear model may be useful when approaching the nonlinear model. Also it is likely that in practice the estimation of costs may introduce more inaccuracies than those occurring with the assumption of linear costs. All of these costs can be regarded as opportunity costs.

Using the three field notation scheme introduced in [9], the scheduling problem considered can be expressed as:

$$1 \left| p_i(S_i) = a_i - b_i S_i, b_i(r.v), 0 < b_i < \frac{a_{min}}{(\sum_{i=1}^n a_i - a_{min})}, \right. \\ \left. \delta_i = \delta \right| \mathbb{E} \sum_{i=1}^n (\alpha E_i + \alpha T_i + \theta z_i + \gamma \delta)$$

For a fixed sequence  $\sigma$ , let  $D_{\sigma(i)}$  denote the deterministic completion time of task  $\sigma(i)$ , which is sequence dependent.

Then:

$$D_{\sigma(i)} = \sum_{j=1}^i a_{\sigma(j)}.$$

Consequently, we have:

$$\begin{aligned} C_{\sigma(i)} &= \sum_{j=1}^i p_{\sigma(j)}(S_{\sigma(j)}) \\ &= \sum_{j=1}^i [a_{\sigma(j)} - z_{\sigma(j)}] \\ &= \sum_{j=1}^i a_{\sigma(j)} - \sum_{j=1}^i z_{\sigma(j)} \\ &= D_{\sigma(i)} - Z_{\sigma(i)}, \end{aligned}$$

where:

$$\begin{aligned} z_{\sigma(j)} &= b_{\sigma(j)} S_{\sigma(j)} \\ &= b_{\sigma(j)} C_{\sigma(j-1)} \end{aligned}$$

$$= b_{\sigma(j)} \sum_{l=1}^{j-1} a_{\sigma(l)} \prod_{k=l+1}^{j-1} (1 - b_{\sigma(k)}),$$

and

$$\begin{aligned} Z_{\sigma(i)} &= \sum_{j=1}^i z_{\sigma(j)} \\ &= \sum_{j=1}^i b_{\sigma(j)} \sum_{l=1}^{j-1} a_{\sigma(l)} \prod_{k=l+1}^{j-1} (1 - b_{\sigma(k)}). \end{aligned}$$

which is the total compression of the first  $i$  tasks in the sequence  $\sigma$ .

In what follows,  $\bar{\mu}^z$  and  $\bar{\mu}^z$  denote respectively the means of  $z_{\sigma(i)}$  and  $Z_{\sigma(i)}$ ,  $\bar{z}$  and  $\bar{z}$  denote respectively the upper bounds of  $z_{\sigma(i)}$  and  $Z_{\sigma(i)}$ .

Clearly we can see that:

$$\begin{aligned}\mathbb{E}[z_{\sigma(j)}] &= \mu \sum_{l=1}^{j-1} a_{\sigma(l)} (1-\mu)^{j-l-1}, \\ \mathbb{E}[Z_{\sigma(i)}] &= \sum_{j=1}^i \mu \sum_{l=1}^{j-1} a_{\sigma(l)} (1-\mu)^{j-l-1}, \\ 0 &\leq z_{\sigma(i)} \leq \\ \frac{a_{\min}}{(\sum_{l=1}^n a_{\sigma(l)} - a_{\min})} \sum_{l=1}^{i-1} a_{\sigma(l)} \left(1 - \frac{a_{\min}}{(\sum_{l=1}^n a_{\sigma(l)} - a_{\min})}\right)^{i-l-1} &= \bar{z}\end{aligned}$$

and

$$0 \leq Z_{\sigma(i)} \leq \frac{a_{\min}}{(\sum_{l=1}^n a_{\sigma(l)} - a_{\min})} \sum_{j=1}^i \sum_{l=1}^{j-1} a_{\sigma(l)} \left(1 - \frac{a_{\min}}{(\sum_{l=1}^n a_{\sigma(l)} - a_{\min})}\right)^{j-l-1} = \bar{z}.$$

Let  $\overline{H_{\sigma(i)}^z}(x)$ ,  $0 \leq x \leq \bar{z}$ , and  $\overline{H_{\sigma(i)}^z}(x)$ ,  $0 \leq x \leq \bar{z}$ , denote respectively the distribution functions of  $z_{\sigma(i)}$  and  $Z_{\sigma(i)}$ . Noted that  $\overline{H_{\sigma(i)}^z}(x)$  (respectively  $\overline{H_{\sigma(i)}^z}(x)$ ) dependent upon the scheduling strategy adopted and  $\overline{H_{\sigma(i)}^z}(0) = 0$  (respectively  $\overline{H_{\sigma(i)}^z}(0) = 0$ ).

Following the same arguments of proof as in [8, Section 4.1], the objective function can be written as:

$$f(\sigma, \delta) = \sum_{i=1}^n J(D_{\sigma(i)}) + \vartheta_{\sigma(i)},$$

where  $J(t)$  is expressed as follows:

$$J(t) = \begin{cases} \alpha(\delta - t + \bar{\mu}^z) & \text{for } 0 \leq t \leq \delta \\ \alpha(\delta - t + \bar{\mu}^z) + \\ (\alpha + \beta) \int_0^{t-\delta} \overline{H_{\sigma(i)}^z}(x) dx & \text{for } \delta < t < \delta + \bar{z} \\ \beta(t - \bar{\mu}^z - \delta) & \text{for } t \geq \delta + \bar{z} \end{cases}$$

and

$$\vartheta_{\sigma(i)} = \theta \bar{\mu}^z + \gamma \delta.$$

### III. DEFINITIONS AND BASIC RESULT

Consider the deterministic single machine scheduling problem with the objective function:  $f(\sigma, \delta) = \sum_{i=1}^n J(D_{\sigma(i)}) + \vartheta_{\sigma(i)}$ , where  $D_{\sigma(i)}$  is the completion time of the  $i$ -th task in the sequence  $\sigma$ ,  $J(t)$  is a continuous function and  $\vartheta_{\sigma(i)}$  is a constant which does not depend on  $D_{\sigma(i)}$ . Then  $J(D_{\sigma(i)}) + \vartheta_{\sigma(i)}$  is the contribution of the  $i$ -th task in the sequence  $\sigma$ . When  $J(t)$  is monotone about some  $T$ , a typical result is the V-shaped property of the optimal schedule. Before we can state this

interesting property, we recall the definitions of monotony about and V-shaped first.

**Definition 1:** A function  $J(t)$  is said to be monotone about  $T$  if it is nonincreasing for  $t \leq T$  and nondecreasing for  $t > T$ .

**Definition 2:** A sequence is said to be V-shaped with respect to processing-times if in the sequence the tasks before and after the task with the smallest processing-time are, respectively, in the non-increasing and non-decreasing orders of processing-times. Thus a V-shaped sequence never contains three consecutive tasks with the largest one located in the middle.

The optimal schedule is V-shaped property implies that once the membership in the two sets is known, the sequence of the tasks within each set can be determined immediately. That is, a search for an optimum must consider only the  $2^n$  ways of forming sets, instead of all  $n$  sequences. Since, V-shaped properties have been utilized to develop efficient algorithms for finding solutions.

**Theorem 1:** If  $J(t)$  is a monotone function about  $T$ , then there exists an V-shaped sequence with respect to the processing-time values that minimizes the cost function  $f(\sigma, \delta)$ .

**Proof:** The proof can be found in [8, Section 3] ■

### IV. ANALYTICAL PROPERTIES

The problem described in section 2 exhibits several interesting properties. These properties are given in this section. They turn out to be useful in the next part of this paper; they allow us to develop an efficient dynamic programming algorithm.

As mentioned in the abstract of this paper, this work extends the researches tackled in [8] and [6] where the same problem is studied except that the deterministic shortening-rate in [6] is here random variable, the random processing-time compression in [8] is now a combination of stochastic with dynamic, while the objective function is extended to cover the objective functions in [8] and [6] by adding together the compression and due-date penalties. Thus, the analysis here (stated properties and their proofs) is somewhat similar (i.e., with some extent) to those in [6] and [8].

In the last paragraph of section 2, we have shown that the objective function  $f(\sigma, \delta)$  can be expressed as:

$$f(\sigma, \delta) = \sum_{i=1}^n J(D_{\sigma(i)}) + \vartheta_{\sigma(i)}.$$

In order to exploit the theorem 1, we can state the following property:

**Property 1:**  $J(t)$  is a monotone function about some  $T$ .

The next property holds immediately from the theorem 1 and the property 1:

**Property 2:** There exists a V-shaped sequence with respect to basic processing-times  $a_i$  that minimizes the objective function  $f(\sigma, \delta)$ .



In the objective function, when  $\gamma < \beta$ , an optimal solution exists in which the minimized due-date  $\delta^*$  coincides with the completion time of a task in position  $K$ . More precisely, we have:

**Property 3:** For any specified sequence  $\sigma$ , there exists an optimal due-date  $\delta^*$  equal to  $C_{\sigma(K)}$  such that  $K = \left\lceil \frac{n\beta - n\gamma}{\alpha + \beta} \right\rceil$  and exactly  $K$  tasks are non-tardy.

When the optimal due-date  $\delta^*$  is considered, the property below gives an equivalent form of the objective function; it provides the positional weights and assigns tasks to the weights in the appropriate way:

**Property 4:** Given the optimal due-date  $\delta^*$  and a task sequence  $\sigma$ , we have:

$\mathbb{E} \sum_{i=1}^n (\alpha E_{\sigma(i)} + \alpha T_{\sigma(i)} + \theta z_{\sigma(i)} + \gamma \delta^*) = \sum_{i=1}^n W_i a_{\sigma(i)}$ , where the coefficients  $W_i$ ,  $1 \leq i \leq n$ , called weights, are as follows:

$$W_i = w_i + w_{i+1}\mu + w_{i+2}\mu(1-\mu) + \dots + w_n\mu(1-\mu)^{n-i-1} \text{ with } w_i = \begin{cases} \alpha(i-1) + n\gamma; & 1 \leq i \leq K \\ \beta(n+1-i); & K < i \leq n \end{cases}$$

and  $w_k = \begin{cases} \theta - \alpha(k-1) - n\gamma; & i+1 \leq k \leq K \\ \theta - \beta(n+1-k); & K < k \leq n \end{cases}$

Using the general argument for minimizing the sum of products, the following property outcomes:

**Property 5.** The sum  $\sum_{i=1}^n W_i a_{\sigma(i)}$  is minimized if the sequence  $(W_1, W_2, \dots, W_n)$  is ordered non-decreasingly and the sequence  $(a_{\sigma(1)}, a_{\sigma(2)}, \dots, a_{\sigma(n)})$  is ordered non-increasingly or vice versa.

Recall first that  $T$  is the time when  $J(t)$  is monotone about. The following property, gives the domain of  $T$ . Specifically:

**Property 6:** Let  $K$  be the integer defined as in property 3, and  $i_0 = \max \left\{ \left\lfloor n - \frac{1}{\mu} - \frac{\theta}{\beta} \right\rfloor; K+1 \right\}$ , then in any optimal schedule  $\sigma$  there hold inequalities  $a_{\sigma(K+1)} \geq \dots \geq a_{\sigma(i_0)}$  and inequalities  $a_{\sigma(i_0)} \leq \dots \leq a_{\sigma(n)}$ . Namely,  $[C_{\sigma(i_0)}, C_{\sigma(K+1)}]$  is the interval when  $T$  belongs.

## V. DETERMINATION OF THE OPTIMAL V-SHAPED SCHEDULE

In this subsection, we will give a dynamic programming algorithm with  $O(n \log n)$  worst-case complexity for solving the problem, based on V-shaped property of the optimal sequence (Property 2), the expression of the objective function  $f(\sigma, \delta)$  writing in terms of property 5, and property 6 which deal with the manner of minimizing a sum of products. Properties 3 and 6 also serve in the construction of the algorithm.

Let  $\sigma^{SPT} = (\sigma^{SPT}(1), \sigma^{SPT}(2), \dots, \sigma^{SPT}(n))$  be the SPT sequence (shortest normal processing-time first). Our goal is to find the optimal due-date  $\delta^*$  and the optimal sequence  $\sigma^*$ , such that:

$$f(\sigma^*, \delta^*) = \min_{\sigma \in \Sigma} \sum_{i=1}^n W_i a_{\sigma(i)},$$

where:

$$\delta^* = C_{\sigma^*(K)} \quad (\text{see, Property 3}),$$

$$W_i = w_i + w_{i+1}\mu + w_{i+2}\mu(1-\mu) + \dots + w_n\mu(1-\mu)^{n-i-1} \quad (\text{see, Property 5}),$$

and  $\Sigma$  is the set of all feasible (V-shaped) permutations of  $\{1, 2, \dots, n\}$ .

From the V-shaped property of the optimal sequence (Property 2) and according to property 6, task  $\sigma^{SPT}(n)$  must be scheduled in the optimal sequence  $\sigma^*$  the first if  $W_1 < W_n$ , or the last otherwise. Generally, let  $\sigma_{k,i}^*$  be the optimal sequence for the tasks of the set  $N_{k,i} = \{\sigma^{SPT}(k), \sigma^{SPT}(k+1), \dots, \sigma^{SPT}(i)\}$ .  $\sigma_{k,i}^*$  is a subsequence of the optimal sequence  $\sigma^*$ . Following  $\sigma_{k,i}^*$ , task  $\sigma^{SPT}(i)$  must be the first if  $W_k < W_i$ , or the last otherwise.

The dynamic programming algorithm presented, iteratively extends a sequence of tasks, appending a new task to it. Each sequence corresponds to a state. A state is a tuple  $(\mathcal{J}, \mathcal{F})$ , where  $\mathcal{J}$  is the subset of tasks with their positions in the optimal sequence  $\sigma^*$  are already determined and  $\mathcal{F}$  is the contribution of the tasks in  $\mathcal{J}$  to the objective function.

If  $N_{k,i}$  is the set of tasks with their positions in the optimal sequence  $\sigma^*$  are unknown,  $(\mathcal{J}, \mathcal{F})$  can be extended by appending  $\sigma^{SPT}(i)$ . When a task  $\sigma^{SPT}(i)$  is appended to extend a state  $(\mathcal{J}, \mathcal{F})$ , a new state  $(\mathcal{J}', \mathcal{F}')$  is generated, where:

$$\mathcal{J}' = \mathcal{J} \cup \{\sigma^{SPT}(i)\}$$

and

$$\mathcal{F}' = \mathcal{F} + \min\{W_k, W_i\} \times a_{\sigma^{SPT}(i)}.$$

By the principle of optimality of dynamic programming, this process leads to an optimal V-shaped sequence.

Based on the above results, we propose for the problem the following dynamic programming algorithm.

### Algorithm.

**Input:** Sequence  $(a_1, a_2, \dots, a_n)$ , numbers  $\mu, \alpha, \beta, \gamma, \theta$ .  
**Output:** an optimal due-date  $\delta^*$ , an optimal schedule  $\sigma^*$ .

- **Step 1:** Arrange tasks in the non-decreasing order of  $a_i$ -values and call the sequence obtained  $\sigma^{SPT} = (\sigma^{SPT}(1), \sigma^{SPT}(2), \dots, \sigma^{SPT}(n))$ ;
- **Step 2:** (Calculate  $K$  and  $i_0$ )  
 $K \leftarrow \max \left\{ 0; \left\lceil \frac{n\beta - n\gamma}{\alpha + \beta} \right\rceil \right\}$ ;

$$i_0 \leftarrow \max \left\{ \left\lfloor n - \frac{1}{\mu} - \frac{\theta}{\beta} \right\rfloor ; K + 1 \right\};$$

► **Step 3:** (Compute  $W_i$ )  
**for**  $i \leftarrow 1$  **to**  $K$  **do**  
 $w_i \leftarrow \theta - \alpha(i - 1) - n\gamma$ ;  
**done**;  
**for**  $i \leftarrow K + 1$  **to**  $n$  **do**  
 $w_i \leftarrow \theta - \beta(n + 1 - i)$ ;  
**done**;  
**for**  $i \leftarrow 1$  **to**  $n$  **do**  
 $W_i \leftarrow w_i + w_{i+1}\mu + w_{i+2}\mu^2(1 - \mu) + \dots + w_n\mu^{n-i}$ ;  
**done**;  
 ► **Step 4:**  
**if**  $i_0 = 1$  **then**  
**begin**  
 $\sigma^* \leftarrow \sigma^{SPT}$ ;  
 $\delta^* \leftarrow C_{\sigma^{SPT}(K)}$ ;  
**goto Step 5**;  
**end**;  
**else** (Construct the sequence  $\sigma^*$  by a backward tracking procedure)  
 4.1.  $\mathcal{T}_{n+1}^1 \leftarrow \emptyset$ ;  
 $\mathcal{T}_{n+1}^2 \leftarrow \emptyset$ ;  
 $|\mathcal{T}_{n+1}^1| \leftarrow 0$ ;  
 4.2. **for**  $i = n$  **down to** **do**  
 $k \leftarrow |\mathcal{T}_{i+1}^1| + 1$ ;  
**if**  $\min\{W_k, W_i\} = W_k$  **then**  
**begin**  
 $\mathcal{T}_i^1 \leftarrow (\mathcal{T}_{i+1}^1, \sigma^{SPT}(i))$ ;  
 $\mathcal{T}_i^2 \leftarrow \mathcal{T}_{i+1}^2$ ;  
 $|\mathcal{T}_i^1| \leftarrow |\mathcal{T}_{i+1}^1| + 1$ ;  
**end**;  
**else** (i.e.,  $\min\{W_k, W_i\} = W_i$ )  
**begin**  
 $\mathcal{T}_i^1 \leftarrow \mathcal{T}_{i+1}^1$ ;  
 $\mathcal{T}_i^2 \leftarrow (\sigma^{SPT}(i), \mathcal{T}_{i+1}^2)$ ;  
 $|\mathcal{T}_i^1| \leftarrow |\mathcal{T}_{i+1}^1|$ ;  
**end**;  
**done**;  
 4.3.  $\sigma^* \leftarrow (\mathcal{T}_2^1, \sigma^{SPT}(1), \mathcal{T}_2^2)$ ;  
 $\delta^* \leftarrow C_{\sigma^*(K)}$ ;  
**Step 5:**  
 Return  $\sigma^*$  and  $\delta^*$ ;

**Theorem 2 :** The sequence  $\sigma^*$  generated by the algorithm is an optimal V-shape sequence for the problem. The complexity of the algorithm is  $O(n \log n)$ .

**Proof:** The Algorithm presented is a dynamic programming algorithm. From the principle of optimality of dynamic programming, the algorithm generates the best V-shaped sequence. Step 1 of the algorithm needs  $O(n \log n)$  times for arranging the tasks. The backward tracking procedure of Step 4 needs  $O(n)$  time. We complete the proof of the theorem. ■

In the following we demonstrate the optimal (V-shaped) sequence on an 8-task problem.

**Example:**

Consider an eight-task problem with the following normal processing-times:  $a_1 = 10, a_2 = 5, a_3 = 14, a_4 = 11, a_5 = 19, a_6 = 7, a_7 = 21$  and  $a_8 = 13$ . Assume that the shortening-rates are random variables, independent and identically distributed in the interval  $\left[0; \frac{a_{\min}}{\sum_{j=1}^8 a_j - a_{\min}}\right] = \left[0; \frac{1}{19}\right]$  with the same mean  $\mu = 0.026$ . The unit penalty factors are:

$$\alpha = 2, \beta = 5, \gamma = 3, \theta = 6.$$

On the basics of the algorithm, the process can be given as follows:

$$\bullet \sigma^{SPT} = (\sigma^{SPT}(1), \sigma^{SPT}(2), \dots, \sigma^{SPT}(8)) = (2, 6, 1, 4, 8, 3, 5, 7).$$

$$(a_{\sigma^{SPT}(1)}, a_{\sigma^{SPT}(2)}, \dots, a_{\sigma^{SPT}(8)}) = (5, 7, 10, 11, 13, 14, 19, 21).$$

$$\bullet K = \max\left\{0; \left\lfloor \frac{n\beta - n\gamma}{\alpha + \beta} \right\rfloor\right\} = \max\{0; 3\} = 3,$$

$$i_0 = \max\left\{\left\lfloor n - \frac{1}{\mu} - \frac{\theta}{\beta} \right\rfloor; K + 1\right\} = \max\{-35; 4\} = 4.$$

So, for any optimal schedule  $\sigma^*$  we have:  $a_{\sigma^*(1)} \geq a_{\sigma^*(2)} \geq a_{\sigma^*(3)}$  and  $a_{\sigma^*(4)} \leq \dots \leq a_{\sigma^*(8)}$ .

$$\bullet W = (W_1, W_2, \dots, W_8) = (21.8383231814, 24.3145001862, 26.8567763719, 24.333445967, 19.689369576, 14.921324, 10.026, 5.0).$$

$$\bullet \mathcal{T}_9^1 = \emptyset, \mathcal{T}_9^2 = \emptyset, |\mathcal{T}_9^1| = 0.$$

(i) for  $i = 8$  we have  $k = |\mathcal{T}_9^1| + 1 = 1$  and  $\min\{W_1, W_8\} = W_8 = 5$  then  $\mathcal{T}_8^1 = \mathcal{T}_9^1, \mathcal{T}_8^2 = (\sigma^{SPT}(8), \mathcal{T}_9^2) = (\sigma^{SPT}(8)) = (7)$  and  $|\mathcal{T}_8^1| = 0$ .

(ii)  $i = 7$ :  $k = 1, \min\{W_1, W_7\} = W_7 = 10.026$   
 $\mathcal{T}_7^1 = \mathcal{T}_8^1, \mathcal{T}_7^2 = (\sigma^{SPT}(7), \mathcal{T}_8^2) = (\sigma^{SPT}(7), \sigma^{SPT}(8)) = (5, 7), |\mathcal{T}_7^1| = 0$ .

(iii)  $i = 6$ :  $k = 1, \min\{W_1, W_6\} = W_6 = 14.921324$   
 $\mathcal{T}_6^1 = \mathcal{T}_7^1,$

$$\mathcal{T}_6^2 = (\sigma^{SPT}(6), \mathcal{T}_7^2) = (\sigma^{SPT}(6), \sigma^{SPT}(7), \sigma^{SPT}(8)) = (3, 5, 7), |\mathcal{T}_6^1| = 0.$$

(iv)  $i = 5$ :  $k = 1, \min\{W_1, W_5\} = W_5 = 19.689369576$

$$\mathcal{T}_5^1 = \mathcal{T}_6^1, \mathcal{T}_5^2 = (\sigma^{SPT}(5), \mathcal{T}_6^2) = (\sigma^{SPT}(5), \sigma^{SPT}(6), \sigma^{SPT}(7), \sigma^{SPT}(8)) = (8, 3, 5, 7), |\mathcal{T}_5^1| = 0.$$

(v)  $i = 4$ :  $k = 1, \min\{W_1, W_4\} = W_1 = 21.8383231814$

$$\mathcal{J}_4^1 = (\mathcal{J}_5^1, \sigma^{SPT}(4)) = (\sigma^{SPT}(4)) = (4), \quad \mathcal{J}_4^2 = \mathcal{J}_5^2, \\ |\mathcal{J}_4^1| = 1.$$

$$(vi) \quad i = 3: \quad k = 2, \quad \min\{W_2, W_4\} = W_2 = 24.3145001862$$

$$\mathcal{J}_3^1 = (\mathcal{J}_4^1, \sigma^{SPT}(3)) = (\sigma^{SPT}(4), \sigma^{SPT}(3)) = (4, 1),$$

$$\mathcal{J}_3^2 = \mathcal{J}_4^2, |\mathcal{J}_3^1| = 2.$$

$$(vii) \quad i = 2: \quad k = 3, \quad \min\{W_3, W_4\} = W_4 = 24.333445967$$

$$\mathcal{J}_2^1 = \mathcal{J}_3^1,$$

$$\mathcal{J}_2^2 = (\sigma^{SPT}(2), \mathcal{J}_3^2) = (\sigma^{SPT}(2), \sigma^{SPT}(5), \sigma^{SPT}(6), \sigma^{SPT}(7), \sigma^{SPT}(8)) =$$

$$(6, 8, 3, 5, 7), |\mathcal{J}_2^1| = 2.$$

$$\text{So the optimal sequence is } \sigma^* = (\mathcal{J}_2^1, \sigma^{SPT}(1), \mathcal{J}_2^2) = (\sigma^{SPT}(4), \sigma^{SPT}(3),$$

$$\sigma^{SPT}(1), \sigma^{SPT}(2), \sigma^{SPT}(5), \sigma^{SPT}(6), \sigma^{SPT}(7), \sigma^{SPT}(8)) = (4, 1, 2, 6, 8, 3, 5, 7) \text{ and the corresponding normal-processing-times sequence is}$$

$$a_{\sigma^*} = (a_{\sigma^*(1)}, a_{\sigma^*(2)}, \dots, a_{\sigma^*(8)}) =$$

$$(11, 10, 5, 7, 13, 14, 19, 21).$$

Thus, the optimal value is then

$$f(\sigma^*, \delta^*) = \sum_{i=1}^8 W_i a_{\sigma^*(i)} = 1548.33890097.$$

and the optimal due-date is

$$\delta^* = C_{\sigma^*(3)} = \sum_{i=1}^3 W_i a_{\sigma^*(i)} = 617.650438717.$$

## VI. CONCLUSION

In this study, the problem of “scheduling a set of independent tasks on a single-machine with random shortening-rates and a common due-date witch is decision variable” is considered. The objective is to find jointly the optimal common due-date, the optimal task sequence and the optimal processing-time compressions so as to minimize the total penalty based on the expected task earliness/tardiness, compression and common due-date.

Several properties of the problem are established and used to develop an efficient dynamic programming algorithm with  $O(n \log n)$  time complexity.

The more interesting further worth researching would be to try to understand the randomness impact of task shortening-rate by analyzing, in a unified fashion, the problem considered in this paper with different objectives of (JIT) production. This made a contribution to further our understanding of their relationship. Finally, a combination of random normal processing-times and random shortening-rates can be a challenge for a new line of research.

## REFERENCES

- [1] T-C.E. Cheng, , Q. Ding, and B.M.T. Lin, “A concise survey of scheduling with time-dependent processing-times,” *European Journal of Operational Research*, vol. 152, no. 1, pp. 1–13, 2004.
- [2] K.I.-J. Ho, J.Y.-T. Leung, and W.-D. Wei, “Complexity of scheduling tasks with time-dependent execution times,” *Information Processing Letters*, vol. 48, pp. 315-320, 1993.
- [3] T-C.E. Cheng, and Q. Ding, “The time dependent makespan problem is strongly NP-complete,” *Computers and Operations Research*, vol. 26, no. 8, pp. 749–754, 1999.
- [4] C-T. Ng, T-C.E. Cheng, A. Bachman, and A. Janiak, “Three scheduling problems with deteriorating tasks to minimize the total completion time,” *Information Processing Letters*, vol. 81, no. 6, pp. 327–333, 2002.
- [5] J. Wang, “Modern Scheduling Problems with Controllable Processing-time,” [D]. *Dalian: Dalian Polytechnic University, (in Chinese)*, 2005.
- [6] T-C.E. Cheng, L-Y. Kang, and C-T. Ng, “Single machine due-date scheduling of tasks with decreasing start-time dependent processing-times,” *International Transactions in Operational Research*, vol. 12 no. 4, pp. 355–366, 2005.
- [7] YU. Ying, SUN. Shi-jie, HE. Long-min, “A dynamic programming algorithm for scheduling problems on earliness award and tardiness penalty with time-dependent processing-time,” *J Shanghai Univ (Engl Ed)*, vol 12, no. 3, pp. 204–209, 2008.
- [8] X.D. Qi, G.Yin, and J.R. Birge, “Single machine scheduling with randomly compressible processing-times,” *Stochastic analysis and applications*, vol 20, no. 3, pp. 591–613, 2002.
- [9] R.L. Graham, E.L. Lawler, J.K. Lenstra, and AHG. Rinnoog Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of Discrete Mathematics*, vol 5, pp. 287-326, 1979.

## SCHEDULING DETERIORATING JOBS: TEN OPEN PROBLEMS

Ali DERBALA

LAMDA-RO laboratory, University Saad Dahlab  
BP 270, Route de Soumaa,  
Blida, Algeria.  
aliderbala@univ-blida.dz

Abdesselem KALI.

University of Khemis Miliana,  
Khemis Miliana, Algeria  
aeskali@gmail.com

### *Abstract:*

We discuss what we consider to be open problems in the area of deteriorating jobs scheduling. We introduce this kind of jobs that can be found in steel production, glass production, fire fighting and maintenance, etc., where any delay in processing a job may increase its completion time. We remind some preliminary concepts for approximation algorithms. We state some open problems and make a contribution towards solving one of them. We are able to identify an equivalent time-dependent scheduling problem and we showed that an expression of a dynamic priority in scheduling is interpreted as a net-reduction in the processing of a job.

*Key words: deteriorating jobs, worst case ratio, approximation algorithms, open problems.*

### I. Introduction

The processing times of jobs may change in the real world. We call jobs those whose processing times increase over time or are an increasing function of their starting time *deteriorating jobs*. Problems of scheduling jobs with start time dependent processing times (deteriorating jobs) attract the attention of many researchers, recently. The scientists introduce new models to describe different aspects of this phenomenon and solve different problems for various criteria. However, in spite of many efforts, there are still some unsolved problems. Examples can be found in steel production, fire fighting and maintenance, etc., where any delay in processing a task may increase its completion time. This model is used in many industrial applications such as:

- In scheduling steel rolling mills, if the temperature of an ingot, while waiting in a buffer between the furnace and the rolling machine, drops below a critical temperature, the ingot needs to be reheated to bring it up to the temperature required for rolling. The reheating time of the ingot depends on its waiting time in the buffer and is thus schedule-dependent.
- The planning of machine maintenance or service.
- The inventory model for deteriorating items with instantaneous supply,
- A perishable product, linear or non-linear increasing demand and shortage. Items deteriorate while in storage.

- Decay in radioactive elements, spoilage in food grain storage, and pilferage from on-hand inventory.

Browne and Yechiali [01] have introduced stochastic deteriorating jobs and have assumed that the actual processing time is  $a_j + b_j \cdot t$  ( $b_j > 0$ ), where  $a_j$  is the basic processing time,  $b_j$  the deteriorating rate, and  $t$  the starting time, of job  $j$ . These authors were inquiring about identifying other equivalent time-dependent scheduling problems. Preliminary concepts in approximation theory are given in section 2. Our motivation for studying this kind of jobs is to bring from the literature ten open problems and to answer a question raised in [01], respectively in sections 3 and 4. We will show that an expression of a dynamic priority is interpreted as a net-reduction in the processing of a job. The reader is referred to the concise survey of scheduling with time-dependent processing times of Cheng et al. [02].

### II. Preliminary concepts for approximation algorithms

A standard way of dealing with NP-hard problems is not to search for an optimal solution, but to go for *near-optimal* solutions [03]. An algorithm that returns near-solutions is called an *approximation algorithm*; if it does this in polynomial time, and then is called a *polynomial time approximation algorithm*.

An approximation algorithm that always returns a near-optimal solution with cost at most a factor  $\rho$  above the optimal cost (where  $\rho$  is called the *worst-case performance guarantee*).

A family of  $(1 + \epsilon)$ -approximation algorithms over all  $\epsilon > 0$  with polynomial running times is called a *polynomial time approximation scheme* or PTAS, for short.

If the time complexity of a PTAS is also polynomial bounded in  $1/\epsilon$ , then it is called a *fully polynomial time approximation scheme* or FPTAS.

For every hard problem, we would like to know whether it possesses a polynomial time approximation algorithm with constant worst-case performance guarantee, or a PTAS, or even an FPTAS. When it is the case, we are concerned by the design and analysis of such polynomial time approximation algorithms and schemes.



### III. The Ten Open problems

Open problems are ordered according respectively to deterministic polynomial time approximation algorithms and stochastic optimal policies. The first seven open problems deal with batch scheduling problem with job deteriorating, availability constraint, jobs having individual critical dates and simple precedence constraints or relations. The last three stochastic scheduling deteriorating tasks problems deal with scheduling problems with exponential functions, identical normal processing times and the problem to determine policies to minimize expected makespan, the sum of weighted completion times.

Barketau et al. [04] studied a batch scheduling problem with job deterioration. In this problem jobs are processed in batches preceded by a setup time and the processing time of a job depends on the time elapsed since the start of the batch to which the job belongs. If this waiting time for job "i" is less than a given threshold value  $D$ , then the job requires a basic processing time otherwise, it requires an extended processing time  $a_i + b_i$ . The objective is to minimize the makespan. They showed that the special case of this problem where  $b_i = b$  is NP-hard in the strong sense, and it is non-approximable with any constant performance guarantee  $\Delta < 3/2$  in polynomial time, unless  $P = NP$ .

For the case with equal basic processing times,  $a_i = a$ , and the case with  $F$  distinct basic processing times, we presented  $O(n \log n)$  and  $O(n^{3F-1} \log n / F^F)$  time solution algorithms, respectively. Let  $m^*$  denote the number of batches in an optimal schedule. They further presented an approximation  $O(n^2 \log n)$  time algorithm with a performance guarantee  $\Delta \leq 1 + \left\lceil \frac{m^*}{2} \right\rceil / m^* \leq 3/2$  for the general problem.

Open problem 1.

*Develop approximation algorithms with a good asymptotic behavior for the general problem  $1/S, p_i = a_i + u_i (s_i - s_b - D)b_i / C_{max}$  and its NP-hard special cases. A modification of the problem with a decreasing step deterioration function instead of an increasing one is worth studying, too.*

Barketau et al.[05] studied a problem with elements of batch scheduling and simple deterioration. Specifically, jobs are processed in batches preceded by a setup time and the processing time of a job depends on the time elapsed since the start of the batch that the job belongs to. A non-negative numbers  $b_i$  is associated with job  $i$ . The processing time of the  $i$ -th job is equal to  $b_i \left( s_i - (s_i^b + S_0) \right)$ , where  $s_i^b$ ,  $s_i$  and  $S_0$  are the starting time of the  $b$ -th batch to which the  $i$ -th job belongs, and the starting time of this job and batch preparation time, respectively. We have shown that the problem is NP-hard. For the special case of the problem with  $b_i = b$  we presented an  $O(N)$  time algorithm solving the problem optimally. For the general case we presented an  $O(N^2)$  time approximation algorithm with a performance guarantee 2.

Open problem 2.

*Develop an approximation algorithms with a good asymptotic behavior for the general problem  $1/S,$*

$$p_i = b_i \left( s_i - (s_i^b + S_0) \right) / C_{max}$$

Ji et al. [06] considered the problem of scheduling deteriorating jobs on a single machine with an availability constraint. They studied the non-resumable case with the objective of minimizing the makespan and total completion time. They showed that both problems are NP-hard in the ordinary sense. For the makespan problem, they presented an optimal approximation algorithm for the on-line case, and a fully polynomial time approximation scheme for the off-line case. For the total completion time problem, they provided a heuristic and evaluated its effectiveness by computational experiments.

Open problem 3.

*The computational results show that the heuristics is efficient in obtaining near-optimal solutions. It will be interesting to find out if an approximation algorithm with a constant worst-case ratio exists for the total completion time problem.*

Extending these problems to parallel machines or flowshops is also an interesting issue. In addition, it is worth studying the problem with the objective of minimizing other scheduling performance criteria.

Leung et al. [07] have shown the strong NP-hardness of the

$$1 | S, p_i = a_i + \text{stp}(s_i - s_i^i - D) b_i | \sum_{i=1}^n C_i$$

problem They gave an  $O(n^2)$  time optimal algorithm for the special case where  $a_i = a$  for all  $1 \leq i \leq n$ , as well as an  $O(n^2)$  time approximation algorithm for the general case. They show that the approximation algorithm has a performance guarantee of 2 when  $b_i \leq D$  for all  $1 \leq i \leq n$ .

Open problem 4.

*What is the performance guarantee of the approximation algorithm for the general case? Is there any better approximation algorithm? Is there any optimal algorithm for the sum of weighted completion times objective when  $a_i = a$  for all  $1 \leq i \leq n$ ?*

For a linear deterioration with a single common critical date, Kubiak and van de Velde [08] have established the computational complexity of the makespan minimization problem. In particular, they have shown that this problem is NP-hard, and developed pseudopolynomial dynamic programming algorithms for it. They have also developed a heuristic and a branch-and-bound algorithm that performs well in view of its simplicity. Their results indicate that the problem with jobs having individual critical dates could be NP-hard in the strong sense.

Open problem 5.

*Develop a fully polynomial approximation scheme for the problem with a single common critical date, if such a scheme exists.*

It is shown, by Gawiejnowicz et al. [09], in this project that if the precedence relations are in the form of chains, a tree (an out-tree or in-tree), a forest or series-parallel, the problem can be solved in polynomial time. It is

shown that in the general case, when the precedence relations are given by an arbitrary directed a-cyclic graph, the problem is NP-hard in the ordinary sense. In the latter case, they shown to be solvable in quadratic time if the processing times are proportional.

Open problem 6.

a. Find good polynomial-time approximation algorithm for the general case, which is NP-hard.

b. Find other forms of deterioration for which simple precedence relations, such as chains or tree, still preserve the property of being polynomial solvable.

Lee and Wu [10] consider the problem of scheduling deteriorating jobs on multiple machines when each machine may have a downtime. The objective is to minimize the makespan. Both the resumable and the non-resumable cases are considered. This approach can be viewed as an extension of the problem proposed by Hsieh and Bricker [11] and Mosheiov [12]. The computational experiments show that the proposed heuristics actually promote the performance of the LDR heuristic algorithm. In addition, the proposed lower bounds actually improves the lower bound by Hsieh and Bricker [11] and Mosheiov [12] when the ratio of the number of jobs to the number of machines is large, and the performance improves as the ratio of the number of jobs to the number of machines increases. However, the proposed lower bound is exactly the same as the one by Hsieh and Bricker [11] and Mosheiov [12] in other cases.

Open problem 7. Future research on developing a tighter lower bound when the ratio of the number of jobs to the number of machines is small might be an interesting issue.

Janiak and Kovalyov [13] established computational complexity of three scheduling problems with exponentially dependent job processing times. They considered minimization of the following cost functions: the makespan, the maximum lateness and the total weighted completion time. In the NP-hardness proofs, they used reductions from problems PARTITION and 3-PARTITION. For solving each of the considered scheduling problems, they constructed and experimentally compared three heuristic algorithms.

It follows from the results obtained in our experiments that the well known scheduling heuristics are efficient to solve the problems with exponentially start time dependent job processing times.

Open problem 8.

Further research can be undertaken to apply proposed heuristics, some other heuristics, local search techniques and enumerative methods for solving real-life scheduling problems with exponential functions of job processing times.

The problems of scheduling a set of tasks with deadlines and increasing rates of processing times on a single machine to minimize the makespan, flow time and maximum lateness have many industrial applications. These problems have remained untackled in the literature. Cheng and Ding [14] showed that the makespan problem  $\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|C_{max}$  is solvable in  $O(n^5)$  time by a dynamic programming algorithm. They also show that the corresponding flow time problem

$\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|\sum C_i$  is equivalent to the makespan problem. As an application of this dynamic programming algorithm, we further show that the problem  $\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|L_{max}$  is solved in  $O(n^6 \log n)$  time. On the other hand, they showed that the general makespan problem  $\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|C_{max}$  is strongly NP-complete by a reduction from 3-Partition. So, the corresponding flow time problem  $\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|\sum C_i$ , and maximum lateness problem, denoted as  $\mathbb{1}|p_i = \alpha_i + w_i s_i, d_i|L_{max}$  are both strongly NP-complete. Thus, they have resolved the computational complexities of six open problems for this class of scheduling model.

Open problem 9.

However, the complexity of the corresponding problems with identical normal processing times is yet to be studied.

$N$  jobs are to be processed sequentially on a single machine. While waiting for processing, jobs deteriorate, causing the random processing requirement of each job to grow at a job-specific rate. Under such conditions, the actual processing times of the jobs are no longer exchangeable random variables and the expected makespan is no longer invariant under any scheduling strategy that disallows idleness. In this paper, Browne and Yechiali [01] analyze the effects of different deterioration schemes and derive optimal scheduling policies that minimize the expected makespan, and, for some models, policies that minimize the variance of the makespan. They also allow for random setup and detaching times. Applications to optimal inventory issuing policies are discussed and extensions are considered.

Open problems 10.

They have established policies to minimize expected makespans for some cases of deteriorating jobs. It would be of great interest to determine policies to minimize (as is usual) the sum of weighted completion times... and is conjectured to be NP-hard, a proof of which still awaits.

#### IV. Main result: make a contribution towards answering a question raised by Brown and Yechiali [01].

These authors [01] were inquiring whether any deterioration or time-dependent functions exist.  $N$  jobs are to be processed sequentially on a single machine. They are supposed to be within the workshop at the time zero. Is associated with each job «  $i$  » an instantaneous priority  $p_i(t)$ , where  $m_i$  and  $M_i$  are real-values,  $t_i^p(t)$  and  $t_i^w(t)$  are the time spent processing job  $i$  and the time spent waiting for processing during  $[0,t]$  respectively. At each time  $t$ , we carry out the job with

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

for  $t=1, 2, \dots$  and  $p_i(0) \in [m_i, M_i]$  the highest priority. This means that the job with the highest priority is



selected among the waiting ones. In a case of tie, Round Robin's rule is employed to break it. Scheduling jobs according to formula (1) is called MBPS, the mean bounded priority scheduling. Mean bounded priority schedulers such as those designed by Haro and Proust [15] have been shown to operate in a fair fashion. At each time  $t$ ,  $t_i^p(t) + t_i^w(t) = t$ .

The formula (1) becomes, for all  $t$ ,  $t = 1, 2, \dots$

$$p_i(t) = M_i + (m_i - M_i) \frac{t_i^p(t)}{t} = M_i - \Delta_i \frac{t_i^p(t)}{t} \quad (2)$$

$$\text{Hence } A_i(t) = t \frac{p_i(t)}{m_i} = (1 + \alpha_i) t_i^p(t) - \alpha_i \cdot t = t_i^p(t) - \alpha_i (t - t_i^p(t)) \quad (3)$$

Where  $\alpha_i = -\frac{M_i}{m_i}$  is a positive real value

(4). We suppose that  $P_i$  is the processing requirement of job  $i$  at  $t = 0$ .

The constant  $\alpha_i = -\frac{M_i}{m_i}$  is the deterioration rate of job  $i$ .

If job  $i$  is not yet completed, its processing residual will grow at rate  $\alpha_i$ .

$\alpha_i (t - t_i^p(t))$  is the increase in  $i$ 's processing residual as a result of the processing given to other jobs during  $[0, t]$ .

$A_i(t) = t \frac{p_i(t)}{m_i}$  can be interpreted as a «net reduction»

in job  $i$  residual processing as a consequence of all processing during  $[0, t]$ , where  $p_i(t)$  is its instantaneous priority. Assume that time evolves in discrete epoch of  $N$ , the set of integer-values. The time evolves in discrete steps of size one. MBPS belongs to the class of preemptive linear deterioration models in which the processing requirement of a job grows linearly as it awaits for processing. It can be considered as a model which belongs to the class of Markov decision processes where:  $U$  is  $\{c, w\}$ , the set of decisions. The jobs are individuals. Job completion will not necessarily occur at integer time points. The deterioration phenomenon means that an increase in processing requirements experienced by jobs will not necessarily be an integer. If  $t$  is a decision time, the next decision for a non completed job chosen for processing at  $t$  will be  $t + 1$  or the time of job  $j$  completion, whichever occurs sooner.

We define the state of job  $i$  at decision time  $t$  by:  $X_i(t) = \begin{cases} C_i \text{ (completion time)} & \text{if } i \text{ has been completed by } t \\ \{A_i(t), B_i(t)\} & \text{otherwise} \end{cases}$

Where  $B_i(t) = \max_{s \in D_i} A_i(t)$ .

It is equals to  $\max_{s \in D_i} t \frac{p_i(t)}{m_i}$  and  $D_i = \{0, 1, \dots, t\}$ .

The state of the process at  $t$  is  $X(t) = \{X_1(t), \dots, X_N(t)\}$ .

The preemptive policy MBPS will be used for choosing a non completed job for processing at each decision epoch.

The criterion assumes a discount factor  $\alpha$ ,  $\alpha \in [0, 1[$  and attempts to minimize the expected total discounted cost.

For any trajectory  $\{x_t, t \in N\}$  and for any policy  $\pi$ , we

define  $V_\pi(i)$  by  $E_\pi \left[ \sum_{t=0}^{\infty} \alpha^t c(x_t, a_t) / x_0 = i \right]$ ,  $i \geq 0$ , the

conditional expected total discounted cost incurred when the policy  $\pi$  is employed and the initial state is  $i$ . We assume that  $c(x_t, a_t)$  are bounded and  $\alpha$  is less than one.

### Conclusion

This work answers the question raised by Browne and Yechiali [1]. An equivalent time-dependent scheduling problem is stated. An expression of a dynamic priority is interpreted as a net-reduction in the processing of a job. Further investigation is needed in order to compare the performance of different assignment policies.

### References

- [01] S. Browne and U. Yechiali (1990). Scheduling deteriorating jobs on a single processor. *Operation Research* 38, 495-4498.
- [02] T.C.E Cheng, Q.Ding and B.M.T.Lin. A concise survey of scheduling with time dependent processing times. *EJOR* 152 (2004), pp.1-13.
- [03] G.Ausiello, P.Crescenzi, G.Gambosi, V.Kann, A.Marchetti-Spaccamela and M.Protasi. Complexity and Approximation. Combinatorial Optimization Problems and Their Approximability Properties. Springer-Verlag Berlin Heidelberg 1999.
- [04] M.S. Barketau · T.C.E. Cheng · C.T. Ng · Vladimir Kotov · Mikhail Y. Kovalyov. Batch scheduling of step deteriorating jobs, *J.Sched* (2008) 11: 17–28.
- [05] M.S. Barketau · T.C.E. Cheng · C.T. Ng · Vladimir Kotov · Mikhail Y. Kovalyov. Batch Scheduling of Deteriorating Products, *Decision Making in Manufacturing and Services* Vol. 1 • 2007 • No. 1–2 • pp. 25–34.
- [06] Min Ji, Young He and T.C.E. Cheng. Scheduling linear deteriorating jobs with an availability constraint on a single machine, *Theoretical Computer Science* 362(2006) 115 – 126.
- [07] Joseph Y.-T. Leung et al., Minimizing sum of completion times for batch scheduling of jobs with deteriorating processing times, *European Journal of Operational Research* 187 (2008) 1090–1099.
- [08] W. Kubiak and S. Van de Velde, Scheduling Deteriorating Jobs To Minimize Makespan, John Wiley & Sons, Inc. *Naval Research Logistics* 45: 511–523, 1998.
- [09] S. Gawiejnowicz, A. Kononov and T-C. Lai, Scheduling deteriorating jobs subject to precedence constraints. Report 112/2001, Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Poland, June 2001.
- [10] Wen-Chiung Lee and Chin-Chia Wu, Multi-machine scheduling with deteriorating jobs and scheduled maintenance, *Applied Mathematical Modeling* 13 (2007) 114–125.
- [11] Hsieh, Y. C., & Bricker, D. L. (1997). Scheduling linearly deteriorating jobs on multiple machines. *Computers and Industrial Engineering*, 32, 727–734.
- [12] G. Mosheiov. A Note on Scheduling Deteriorating Jobs. *Mathematical and Computer Modelling* 41 (2005) 883-886.

- [13] Adam Janiak and Mikhail Y. Kovalyov, Job Sequencing with Exponential Functions of Processing Times, *Informatica*, 2006, Vol. 17, No. 1, 13–24.
- [14] T.C.E. Cheng and Q. Ding, Single machine scheduling with deadlines and increasing rates of processing times, *Acta Informatica* 36, 673–692 (2000).
- [15] Haro,C. and C. Proust, “ Un ordonnancement équitable par priorités bornées,” Colloque Méthodes et outils d’aide à la décision, *MOAD’92* Béjaia, Algérie, 15-17dec, 46-49 ( 1992).



## ANNEXE C

## CALCUL DE LA SUPERFICIE BRULÉE PAR UN INCENDIE

Nous supposons que les terrains forestiers sont plats, ont des angles d'élévation uniformes, les conditions météorologiques et le combustible sont homogènes. Dans cet environnement, le front d'un incendie à des instants donnés prend des formes généralement approximées par des ellipses de différents rapports longueur/largeur.

Pour calculer  $E(t)$ , la superficie brûlée par un incendie, nous rapportons le plan à un repère orthonormé d'axes  $x_1$  et  $x_2$ , où  $x_1$  représente l'axe focal (ou grand axe) de l'ellipse. Sa direction étant dans celle du vent effectif. L'origine du repère est le point d'allumage de l'incendie.

A un instant donné  $t$ , calculé depuis l'instant d'allumage de l'incendie, le front de l'incendie est représenté par une courbe à deux dimensions:

$$\vec{x}(\omega, t) = (x_1(\omega, t), x_2(\omega, t))$$

Où  $\omega$ ,  $0 \leq \omega \leq 2\pi$ , est l'angle entre l'axe  $x_1$  et le vecteur normal  $\vec{n}(\omega) = (\cos \omega, \sin \omega)$ .  $x_1(\omega, t)$  et  $x_2(\omega, t)$  sont des fonctions définissant la forme du front de l'incendie et son tau de propagation  $ROS$ .

Pour la propagation elliptique d'un incendie, les fonctions les plus utilisées sont de la forme

$$\vec{x}(\omega, t) = (x_1(\omega, t), x_2(\omega, t)) = (M(t) \cos \omega, m(t) \sin \omega)$$

où  $M(t)$  et  $m(t)$  sont des fonctions du temps  $t$ .

$E(t)$  peut être calculée comme suit:

$$\begin{aligned}
E(t) &= \frac{1}{2} \int_{\partial E} \left[ -x_2(\omega, t) \frac{\partial x_1(\omega, t)}{\partial \omega} + x_1(\omega, t) \frac{\partial x_2(\omega, t)}{\partial \omega} \right] \\
&= \frac{1}{2} \int_0^{2\pi} (-m(t) \sin \omega, M(t) \cos \omega) (-M(t) \sin \omega, m(t) \cos \omega) d\omega \quad (1) \\
&= \frac{1}{2} \int_0^{2\pi} M(t) m(t) (\sin^2 \omega + \cos^2 \omega) d\omega \\
&= \pi M(t) m(t)
\end{aligned}$$

Où  $\partial E$  est le front de l'incendie

Pour  $\omega = 0$ , la distance parcourue par l'incendie à l'instant  $t$  est donnée dans [55] par

$$M(t) = \frac{A}{B+1} t^{B+1} + c(t)t \quad (2)$$

Dans [56], l'estimation utilisée pour le rapport  $\frac{M_i(t)}{m_i(t)}$ , pratiquement constant dans

le temps, est la suivante

$$\frac{M(t)}{m(t)} = 0.936 e^{0.07127U} + 0.461 e^{-0.043U} = \psi \quad (3)$$

où  $U$  est la vitesse du vent en  $km \times h^{-1}$ .

De (1), (2) et (3), il découle que

$$E(t) = \frac{\pi}{\psi} \left( \frac{A}{B+1} \times t^{B+1} + c(t) \times t \right)^2$$

## ANNEXE D

### CODE SOURCE SCILAB POUR CALCULER LES I.A.D

#### 1. Les données du programme

**NS** : Nombre d'états d'un incendie.

**NI** : Nombre d'instances.

**NF** : Nombre d'incendies dans une instance.

**SF** : Ensemble des incendies.

**A** et **B** : Paramètres estimés en fonction des divers facteurs environnementaux dans chaque zone (la vitesse du vent, le type et la charge du combustible, ...) et / ou du taux de propagation d'équilibre de l'incendie.

**U** : Vitesse du vent dans une zone

**c** : Paramètre non négatif, défini comme la différence entre le taux de propagation de l'incendie basé sur le système FBP et celui basé sur BEHAVE.

**Gamma** : Paramètre dépend de **U**

**alpha** : Taux d'actualisation

**V** : Valeur attribuée à une zone de forêt

**ro** : Taux d'application du processeur

**Q** : Capacité de transport du processeur

**thau** : Temps d'interruption du processeur, au cours de ces opérations pour confiner un incendie, pour aller au point d'eau le plus proche pour recharger.

**T(0,f)** : Temps nécessaire pour le processeur de se déplacer de la base où il est stationné à l'endroit du premier incendie choisi pour être confiné.

**T(f1,f2)** : Temps nécessaire pour que le processeur puisse se rendre au point d'eau le plus proche pour recharger après l'achèvement de ses opérations pour l'incendie **f1** et passer au front de l'incendie **f2**

**r** : date de disponibilité d'un incendie

**cdfnormal**: Fonction de distribution cumulative normale

**f** : Incendie

**C** : Date de fin d'exécution d'un incendie

**s** : Date de début d'exécution d'un incendie

**p** : Durée opératoire d'un incendie

**DAI**: Indice d'Allocation Dynamique d'un incendie

**TM** : Matrice stochastique de transition entre les états d'un incendie

**step** : Etape

**d**: Date d'échéance d'un incendie

**D** : Ecart-type de la loi normale

**Pr** : Probabilité de transition

**LSF** : Dernier incendie choisi pour être confiné

**CTLSF** : Date de fin d'exécution de l'incendie **LSF**

**SCHEDULE** : Séquence d'ordonnancement des incendies

**Tr\_Mat** : Matrice, contient toutes les matrices de probabilité de transition de **SF**

**pos** : Position dans un ordonnancement

## 2. Fichier principal du programme : DAI.sce

```
exec('C:\Users\pc\Desktop\...\Data.sci')
```

```
exec('C:\Users\pc\Desktop\...\Schedule.sci')
```

```
exec('C:\Users\pc\Desktop\...\processing_time.sci')
```

```
exec('C:\Users\pc\Desktop\...\starting_time.sci')
```

```
exec('C:\Users\pc\Desktop\...\completion_time.sci')
```

```
exec('C:\Users\pc\Desktop\...\cdfnormal.sci')
```

```
exec('C:\Users\pc\Desktop\...\transition_matrix.sci')
```

```
exec('C:\Users\pc\Desktop\...\DAI_fire.sci')
```

```
funcprot(0); stacksize('max'); format(10); timer();
```

```
NS=100; NI=1000; time=[]; LS=[]; CTLSF2=[];
```

```
for d=1:50
```

```
    for i=1:NI
```

```
        [r,T,T0,thau,Q,ro,V,alpha,Gamma,c,U,B,A,SF,NF]=Data(i);
```

```
        CTLSF=0; Tr_Mat=TM1(SF); [Sched,CTLSF,LSF]=schedule(NF,SF,Tr_Mat);
```

```
        CTLSF2(i)=CTLSF; LS(i)=length(Sched); time(i)=timer();
```

```
    end
```

```
average_running_time(d)=sum(time)/NI;
```

```
average_Lenght_Schedule(d)=sum(LS)/NI;
```

```
average_completion_time(d)=sum(CTLSF2)/NI;
```

```
end
```

```
    average_running_time, average_Lenght_Schedule, average_completion_time
```

### 3. Fonction pour générer les données du problème : Data.sci

```
function [r, T, T0, thau, Q, ro, V, alpha, Gamma, c, U, B, A, SF, NF]=Data(i);
NF=grand(1,1,"uin",100,100); SF=[1:NF]; A=grand(1,NF,"unf",0,14);
B=grand(1,NF,"unf",1,2); U=grand(1,1,"uin",0,8); c=grand(1,NF,"unf",0,11);
Gamma=0.936*exp(0.07127*U)+0.461*exp(-0.043*U);
alpha=grand(1,1,"unf",0.000001,0.0001); V=grand(1,NF,"unf",100000,200000);
ro=6480000; Q=30000; thau=0.2; T0=grand(1,NF,"unf",0,1);
for j=1:1:(NF-1)
    for i=j+1:1:NF
        T(j,i)=grand(1,1,"unf",0,1); T(i,j)=T(j,i);
    end
end
r=grand(1,NF,"unf",0,1);
endfunction
```

### 4. Fonction pour ordonnancer les incendies: Schedule.sci

```
function [Sched, CTLSF, LSF]=schedule(NF, SF, Tr_Mat)
for pos=1:NF
t=0; indices=[];
    for f=SF
        TM=Tr_Mat(:,1+t:NS+1+t); indices=[indices,DAI(TM)]; t=NS+1+t;
    end
max_DAI=max(indices); i=find(indices==max_DAI); SF=SF(i); CTLSF=C(SF(i));
    if CTLSF>d then break
    else Sched(pos)=SF(i); SF(i)=[]; Tr_Mat(:,(i-1)*(NS+1)+1:i*(NS+1))=[];
    end;
end
endfunction
```

### 5. Fonction pour calculer la durée opératoire d'un incendie : processing\_time.sci

```
function y=p(f)
y=((1.2*%pi)/(Gamma*ro))*(((A(f)*(s(f)^(B(f)+1)))/(B(f)+1))+c(f)*s(f)^2)*sqrt(A(f)*B(
f)*(s(f)^(B(f)-1))+c(f))+int(((1.2*%pi)/(Gamma*Q))*(((A(f)*(s(f)^(B(f)+1)))/(B(f)+1))+
c(f)*s(f)^2)*sqrt(A(f)*B(f)*(s(f)^(B(f)-1))+c(f))*thau
endfunction
```

6. Fonction pour calculer Date de début d'exécution d'un incendie : starting\_time.sci

```
function y=s(f)
    if CTLSF==0 then    y=T0(f)+r(f)
    else    y=CTLSF+T(LSF,f)+r(f)
    end
endfunction
```

7. Fonction pour calculer la date de fin d'exécution d'un incendie: completion\_time.sci

```
function y=C(f)
y=s(f)+p(f)
endfunction
```

8. Fonction pour calculer la fonction de distribution cumulative normale: cdfnormal.sci

(cumulative distribution function of normal distribution , Written by

[Philippe.CASTAGLIOLA@univ-nantes.fr](mailto:Philippe.CASTAGLIOLA@univ-nantes.fr) Université de Nantes & IRCCyN UMR CNRS 6597)

```
function Y=cdfnormal(X, mu, sigma)
[argout,argin]=argn()
    if (argin<1)|(argin>3)
        error("incorrect number of arguments")
    end
    if ~exists("mu","local")
        mu=0
    end
    if ~exists("sigma","local")
        sigma=1
    end
    if sigma<=0
        error("argument "sigma" must be > 0")
    end
    Y=cdfnor("PQ",X,mu*ones(X),sigma*ones(X))
Endfunction
```

9. Matrice des probabilités de transition d'un incendie : transition\_matrix.sci

```

function y=TM(f)
step=linspace(s(f),C(f),NS); step(1)=[]; n=grand(1,1,"uin",1,4); D=(d-r(f))/n; k=1;
for t=step
    if t<=d then
        Pr(k,k+1)=cdfnormal(t,d,D)-cdfnormal(s(f),d,D), PP(k)=1-Pr(k,k+1);
    else
        PP(k)=cdfnormal(t,d,D)-cdfnormal(s(f),d,D); Pr(k,k+1)=1-PP(k);
    end
    k=k+1
end
Pr=[Pr,PP]; Pr=[Pr;zeros(Pr(1,:))]; Dim=size(Pr); Pr(Dim(1),Dim(2))=1; y=Pr;
endfunction
function Tr_Mat=TM1(SF)
Tr_Mat=[];
for f=SF
    Tr_Mat=[Tr_Mat,TM(f)];
end
endfunction

```

10. Fonction pour calculer l'indice d'allocation dynamique d'un incendie: DAI\_fire.sci

```

function y=DAI(TM)
P1=[]; Product=[];
for l=1:NS-1
    P1=[P1,TM(1,l+1)];
    Product=[Product,(alpha^l)*prod(P1)*(1+(alpha/(1-alpha))*TM(l+1,NS+1))]
end
H=1+((alpha*TM(1,NS+1))/(1-alpha))+sum(Product)
if CTLSF==0 then
    y=V(f)/(1-(alpha^(T0(f)+r(f)))*(1-(1-alpha))*H)
else
    y=V(f)/(1-(alpha^(T(LSF,f)+r(f)))*(1-(1-alpha))*H)
end
endfunction

```

## RÉFÉRENCES

1. Kunnathur, A.S., et Gupta, S.K., "Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem", *European Journal of Operational Research*, V. 47, n° 1, (1990), 56 - 64.
2. Sundararaghavan, P.S., et Kunnathur, A.S., "Single machine scheduling with start time dependent processing times: some solvable cases", *European Journal of Operational Research*, V. 78, n° 3, (1994), 394 - 403.
3. Mosheiov, G., "Scheduling jobs under simple linear deterioration", *Computers and Operations Research*", V. 21, n° 6, (1994), 653 - 659.
4. Rachaniotis, N.P., et Pappis, C.P., "Scheduling fire fighting tasks using the concept of deteriorating jobs", *Canadian Journal of Forest Research*, V. 36, n° 3, (2006), 652-658.
5. Pappis, C.P., et Rachaniotis, N.P., "Scheduling a single fire fighting resource with deteriorating fire suppression times and set-up times", *Operational Research International Journal*, V. 10, (2010), 27 - 42.
6. Pappis, C.P., et Rachaniotis, N.P., "Scheduling in a multi-processor environment with deteriorating job processing times and decreasing values : the case of forest fires", *Journal of Heuristics*, V. 16, n° 4, (2010), 617 - 632.
7. Kali, A, "Stochastic scheduling of single forest firefighting processor", *Canadian Journal of Forest Research*, V. 46, n° 3, (2016), 370 - 375
8. Gittins, J., Glazebrook, K., et Weber, R., "Multi-armed Bandit Allocation Indices", John Wiley & Sons, (2011), 293 p.
9. Gittins, J.C., et Jones, D.M., "A dynamic index for the sequential Design of experiments", *Colloquia Mathematica Janes Bolai*, European meeting of statisticians, Budapest, Hungary, (1972), 241- 266.
10. Sonin, I., "The optimal stopping of Markov chain and recursive solution of Poisson and Bellman equations", In *Stochastic Calculus to Mathematical Finance*, (2006), 609 - 621.
11. Pinedo, M., "Scheduling: Theory, algorithms and systems", 5th edition, Springer, (2016).
12. Gupta, S.K., Kunnathur, A.S., et Dandapani, K., "Optimal repayment policies for multiple loans", *Omega*, V. 15, n° 4, (1987), 323 - 330.



13. Gupta, J.N.D., et Gupta, S.K., "Single facility scheduling with nonlinear processing times", *Computers and Industrial Engineering*, V. 14, (1988), 387 - 393.
14. Mosheiov, G., " $\Lambda$ -shaped policies to schedule deteriorating jobs", *Journal of the Operational Research Society*", V. 47, n° 9, (1996), 1184 - 1191.
15. Flapper, S.D.P., Fransoo, J.C., Broekmeulen, R.A.C.M., et Inderfurth, K., "Planning and control of rework in the process industries : a review", *Production Planning and Control*, V. 13, (2002), 26 - 34.
16. Shafransky, Y.M., "On optimal sequencing for deterministic systems with a tree-like partial order", (in Russian), *Vestsi Akad Navuk BSSR, Ser Fiz-Mat Navuk*, (1978), 2:120.
17. Melnikov, O.I., et Shafransky, Y.M., "Parametric problem of scheduling theory", *Kibernetika*, (1979), (in Russian), English translation: *Cybernetics*, (1980), V. 15, 352 - 357.
18. Alidaee, B., et Womer, N.K., "Scheduling with time dependent processing times : review and extensions", *Journal of the Operational Research Society*, V. 50, n° 6, (1999), 711 - 720.
19. Cheng, T-C.E., Ding, Q., et Lin, B.M.T., "A concise survey of scheduling with time-dependent processing times", *European Journal of Operational Research*, V. 152, n° 1, (2004), 1 - 13.
20. Gawiejnowicz, S., "Time-dependent scheduling", Berlin, Springer, (2008), 379 p.
21. Agnetis, A., Billaut, J-C., Gawiejnowicz, S., Pacciarelli, D., et Soukhal, A., "Multiagent Scheduling : Models and Algorithms", Springer, Berlin-Heidelberg, (2014), 281 p.
22. Yin, Y, Cheng, T. C. E., et Wu, C-C, "Editorial: Scheduling with Time-Dependent Processing Times", *Mathematical Problems in Engineering*, (2014).
23. Yin, Y, Cheng, T. C. E., et Wu, C-C, "Editorial: Scheduling with Time-Dependent Processing Times", *Mathematical Problems in Engineering*, (2015).
24. Strusevich, V.A., et Rustogi, K., "Scheduling with Times-Changing Effects and Rate-Modifying Activities". Springer, Berlin-Heidelberg, (2017), 460 p.
25. Tanaev, V.S., Gordon, V.S., et Shafransky, Y.M., "Scheduling Theory. Single-Stage Systems" Dordrecht: Kluwer, (1994), 800 p.
26. Mosheiov, G., "V-shaped policies for scheduling deteriorating jobs", *Operations Research*, V. 39, n° 6, (1991), 979 - 991.

27. Kononov, A, "A single machine scheduling problems with processing times proportional to an arbitrary function", *Discrete Analysis and Operations Research*, V. 5, n° 3, (1998), 17 - 37.
28. Cheng, T-C.E., et Ding, Q., "Single machine scheduling with deadlines and increasing rates of processing times", *Acta Informatica*, V. 36, (2000), n° 9–10, 673 - 692.
29. Cheng, T-C.E., et Ding, Q., "Scheduling start time dependent tasks with deadlines and identical initial processing times on a single machine", *Computers and Operations Research*, V. 30, n° 1, (2003), 51 - 62.
30. Kuo, W-H., et Yang, D-L., "Single-machine scheduling problems with start time dependent processing time", *Computers and Mathematics with Applications*, V. 53, n° 11, (2007), 1658-1664.
31. Alidaee, B., "A heuristic solution procedure to minimize makespan on a single machine with non-linear cost functions", *Journal of the Operational Research Society*, V. 41, n° 11, (1990), 1065–1068.
32. Browne, S., et Yechiali, U., "Scheduling deteriorating jobs on a single processor", *Operations Research*, V. 38, n° 3, (1990), 495 - 498.
33. Janiak, A, et Kovalyov, M.Y., "Job sequencing with exponential functions of processing times", *Informatica*, V. 17, n° 1, (2006), 13 - 14.
34. Mosheiov, G., "Scheduling jobs with step-deterioration; Minimizing makespan on a single- and multi-machine", *Computers and Industrial Engineering*, V. 28, n° 4, (1995), 869 - 879.
35. Kubiak, W., et van de Velde, S.L., "Scheduling deteriorating jobs to minimize makespan", *Naval Research Logistics*, V. 45, n° 5, (1998), 511 - 523.
36. Cai, J-Y., Cai, P., et Zhu, Y., "On a scheduling problem of time deteriorating jobs", *Journal of Complexity*, V. 14, n° 2, (1998), 190 - 209.
37. Voutsinas, T.G., et Pappis, C.P., "Scheduling jobs with values exponentially deteriorating over time", *International Journal of Production Economics*, V. 79, n° 3, (2002), 163 - 169.
38. Rachaniotis, N.P., "Advances on job scheduling with time dependent parameters: theory and industrial applications", In *Electronic Proceedings of the 9th ELA doctorate workshop*, Monchy St. Eloi, France, (2004), 120 - 131.
39. Sullivan, A.L., "Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi- physical models", *International Journal of Wildland Fire*, V. 18, (2009), 349- 368.

40. Sullivan, A.L., "Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models", *International Journal of Wildland Fire*, V. 18, (2009), 369 - 386.
41. Sullivan, A.L., "Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models", *International Journal of Wildland Fire*, V. 18, 387 - 403.
42. Hirsch, K.G., "Canadian Forest Fire Behavior Prediction System, Interactive Training and Reference", University of British Columbia Press, (1998).
43. Burgan, R.E., Rothermel, R.C., "BEHAVE: Fire Behavior Prediction and Fuel Modeling System—FUEL subsystem", USDA For. Serv. Gen. Tech. Rep. INT-167, (1984).
44. Finney, M.A., "FARSITE: Fire Area Simulator-model development and evaluation", Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, (2004), 47 p
45. Linn, R.R., "A transport model for prediction of wildfire behaviour", PhD Thesis LA-13334-T, Los Alamos National Laboratory. Reissue of PhD Thesis accepted by Department of Mechanical Engineering, New Mexico State University, (1997).
46. Mell, W., Jenkins, M.A., Gould J., et Cheney, P., "A physics-based approach to modelling grassland fires", *International Journal of Wildland Fire*, V. 16, n° 1, (2007), 1 - 22.
47. Martell, D.L., "A review of operational research studies in forest fire management", *Canadian Journal of Forest Research*, V. 12, (1982), 119-140.
48. Martell, D.L., Gunn, E.A., et Weintraub, A., "Forest management challenges for operational researchers", *European Journal of Operational Research*, V. 104, (1998), 1-17.
49. Minas, J.P., Hearne, J.W., et Handmer, J.W., "A review of operations research (OR) methods applicable to wildfire management. *International Journal of Wildland Fire*, V. 21, n° 3, (2012), 189 - 196.
50. Martell, D.L., "A review of recent forest and wildland fire management decision support systems research" *Curr. Forest. Rep.*, V. 1, n° 2, (2015), 128 - 137.
51. Vorisis, D., "Definition of the needs of suppress groups as concern of the prediction of the behaviour of the wild land fires". *International Symposium, Forest Fires: Needs and Innovations, EC, DG XII, Athens, Greece.* (1999), 18 - 19.

52. Kalabokidis, K.D., Gatzogiannis, S. et Galatsidas, S., "Introducing wildfire into forest management planning: towards a conceptual approach", *Forest Ecology and Management*, V. 158, (2002), 41 - 50.
53. Graham, R.I., Lawler, E.L., Lenstra, J.K., et Rinnooy Kan, A.H.G, "Optimization and approximation in deterministic sequencing and scheduling: A survey", *Annals of Discrete Mathematics*, V. 5, (1979), 287-326.
54. McArthur, A.G., "The effect of time on fire behaviour and fire suppression problems", *S.A. Emergency Fire Serv., Keswick, South Aust. E.F.S., Manual*, 3-6, (1968), 10 - 13.
55. McAlpine, R.S. et Wakimoto, R.H., "The acceleration of fire from point source to equilibrium spread", *Forest Science*, V 37, n° 5, (1991), 1314 - 1317.
56. McAlpine, R.S., et Xanthopoulos, G., "Predicted vs. observed fire spread rates in ponderosa pine fuel beds: a test of American and Canadian systems", In *Proceedings of the 10th Conference on Fire and Forest Meteorology*, Ottawa, Ontario, 17–21 April 1989. Forestry Canada, Ottawa, Ontario, (1998), 287 - 294.
57. Rachaniotis, Nikos P., "Contribution à la théorie d'ordonnancement et à ses applications en production et environnement", thèse doctorat, Université du Pirée, Athènes, Grèce, 2007. (en grec).
58. Katsanos, A, "Suppression of forest fires by air", *Ministry of Agriculture, Department of Forest Research, Athens, Greece*. V. 40, (1978), 13 - 73.
59. McNaughton, R., "Scheduling with deadlines and loss functions", *Management Science*, V. 6, (1959) 1–12.
60. Brunelle, E, "Analyse et calibration d'un modèle multi-échelle pour la simulation de feux de forêt", Thèse Doctorat en Mathématiques appliquées, Département de mathématiques et de statistique, Faculté des arts et des sciences, Université de Montréal, Canada, (2007), 140 p.
61. Whittle, P., "Multi-armed bandits and the Gittins index", *J. Roy. statist Soc. Ser. B* 42, 2, (1980), 143- 149.
62. Glazebrook, K.D., Hodge, D.J., Kirkbride, C. et Mintyal, R. J., "Stochastic scheduling: A short history of index policies and new approaches to index generation for dynamic resource allocation", *Journal of Scheduling*, V. 17, n° 5, (2014), 407 - 425.
63. Chakravorty, J, et Mahajan, A., "Multi-Armed Bandits, Gittins Index, and Its Calculation", *Methods and Applications of Statistics in Clinical Trials: Planning, Analysis, and Inferential Methods*, John Wiley & Sons, (2014). 416 - 435.
64. Katehakis, M.N., and Robbins, H.E. 1995. Sequential choice from several populations. *In Proc. Natl. Acad. Sci.* **92**: 8584–8585. doi:10.1073/pnas.92.19.8584.

65. Burnetas, A.N., and Katehakis, M.N. 1993. On sequencing two types of tasks on a single processor under incomplete information. *Prob. Eng. Inf. Sci.* **7**(1): 85–119. doi:10.1017/S0269964800002801.
66. Burnetas, A.N., and Katehakis, M.N. 1997. Optimal adaptive policies for Markov decision processes. *Math. Oper. Res.* **22**(1): 222–255. doi:10.1287/moor.22.1.222.
67. Blake, IF. 1979. *An introduction to applied probability*. John Wiley and Sons, New York.
68. Katehakis, M.N., and Veinott, A.F., Jr. 1987. The multi-armed bandit problem: decomposition and computation. *Math. Oper. Res.* **12**(2): 262–268. doi:10.1287/moor.12.2.262.