الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية العلوم
Faculté des Sciences

قسم الرياضيات
Département de mathématiques

Mémoire de Projet de Fin d'Études

Pour l'obtention du diplôme de Master en Mathématiques

option

modélisation stochastique et statistique

Thème

Gestion optimale d'une centrale électrique & Problème de maintenance Par Les processus de décision markovien

Présenté Par : **ZEKRI YASSINE & LEKHAL SAMI**

Devant le jury composé de :

Mr Chellali Mustapha	**PROF**	**U.S.D. Blida**	**Président**
Mme Z.Dahmane	**MAA**	**U.S.D. Blida**	**Promotrice**
Mr Frihi Rédhouane	**MAA**	**U.S.D. Blida**	**Examinateur**

Année Universitaire 2016-2017

REMERCIEMENTS

Nous remercions Dieu de nous avoir donné la santé et le courage
Pour inspirer la connaissance et le savoir.

Nos plus vifs remerciements vont à notre promotrice ***Mme***
Z. DAHMANE *qui a bien voulu diriger ce travail.*

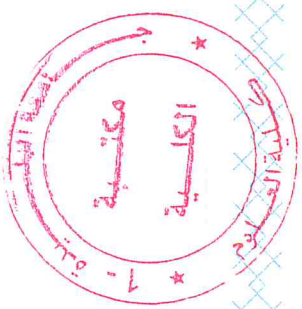
Une pensée pleine de reconnaissance à ***Mme N.OUKID*** *et à tous les enseignants du Département de mathématique pour leurs efforts et leurs collaborations lors de notre cursus à l'université*

Des remerciements vont également à ***ISSAM*** *et à celles et ceux qui nous ont apporté, de prés ou de loin, orientation, soutien et aide dans la réalisation et la concrétisation de ce travail.*

Nous remercions les membres du jury qui ont bien voulu juger ce travail,
Nous les assurons de notre profonde gratitude.

Merci beaucoup

LEKHAL.S & ZEKRI.Y



Dédicaces

Je dédie ce modeste travail à :

Mes chères parents, pour leurs soutient moral et financier et d'être la lumière de ma vie, qui m'ont toujours encouragé pour terminer mes études dans de bonnes conditions, en leur espérant une longue vie et que ***Dieu*** *les garde.*

A ma grand-mère.

A mes frères.

A mes sœurs.

A tous les membres de ma grande famille « ***ZEKRI*** *».*

A mes chers amis et collègues.

A mon promotrice ***Mme Z- DAHMANE****, et mon binôme* ***SAMI***

A toute la promotion ***2016-2017*** *mathématique sans exception.*

****ZEKRI YASSINE ****

Dédicaces

Je dédie ce modeste travail à :

Mes chères parents, pour leurs soutient moral et financier et d'être la lumière de ma vie, qui m'ont toujours encouragé pour terminer mes études dans de bonnes conditions, en leur espérant une longue vie et que ***Dieu*** *les garde.*

A mes frères.

A tous les membres de ma grande famille.

A mes chers amis et collègues.

A ma promotrice ***Mme Dahmane****, et mon binôme* ***Zekri Yacine***

A toute la promotion ***2016-2017*** *processus stochastique et statistique sans exception.*

****Lekhal Sami****

Résumé :

Ce mémoire s'inscrit sous le thème des processus stochastique, plus précisément des processus de décision markovien (markovien decision processes,MDP) issu de la théorie de la décision et de la théorie des probabilités .

Le modèle MDP peut être vu comme une chaine de Markov à la quelle on ajoute une composante décisionnelle.

Le but n'est pas d'optimiser une décision isolée, mais de déterminer la suite d'actions (politique) qui minimise une certaine fonction de coût. L'incertain est représenté sous forme de probabilités de transition supposées connues.

Parmi les méthodes utilisées, pour la recherche de politique optimale, nous allons découvrir deux algorithmes, « policy-iteration algorithm» et «value-iteration algorithm ».Nous allons également vérifier l'avantage et l'inconvénient de chacun deux à travers deux exemples : le problème de maintenance et le problème de la gestion d'une centrale électrique.

Abstract:

This memory thesis is based on the theme of stochastic processes, more precisely Markov decision processes derived from the theory of decision and the theory of probabilities.

The model MDP can be seen as a Markov chain to which a decisional component is added.

The objective is not to optimize an isolated decision, but to determine the sequence of actions (policy) that minimizes a certain cost function. The uncertainty is represented as supposedly known transition probabilities.

Among the methods used, for the optimal policy search, we will discover two algorithms, « policy-iteration algorithm» and «value-iteration algorithm ». We will also check the advantage and the disadvantage of each two through two examples: the problem of maintenance and the problem of managing a power plant.

ملخص

تستند هذه الأطروحة على موضوع العمليات العشوائية، وبصورة أدق عمليات نمذجة القرار لماركوف . مستمدة من نظرية القرار ونظرية الاحتمالات.

هذا النموذج يمكن أن يرى على أساس سلسلة ماركوف التي تتم إضافة عنصر القرار.

الهدف ليس تحسين القرار ، بل تحديد تسلسل الإجراءات (السياسة) الذي يقلل من عملية التكلفة . ويمثل على أنه احتمالات انتقال مفترضة معروفة.

من بين الطرق المستعملة لإيجاد السياسة المثلى تعرفنا على طريقتين " خوارزمية تكرار السياسة" و " خوارزمية تكرار القيمة " و تطرقنا أيضا لميزة وعيب الاثنين من خلال مثالين مشكلة الصيانة و مشكلة إدارة محطة توليد الكهرباء .

Table des matiéres

Principaux notions et symboles

I : l'ensemble des états

a : l'ensemble des actions

$A(i)$: l'ensemble des actions possibles en l'état « i »

$c_i(a)$ = fonction de coût

R = politique stationnaire

$p_{ij}(R_i)$ = probabilités de transition lorsque la politique R_i est utilisée

π_j = la distrubution (probabilité) stationnaire

$V_n(i, R)$ =l'esperance du cout total sur les n premiers instants de decision si i est l'état initial et R la politique utiliseé

$g_i(R)$ = la fonction coût moyen

$\Delta(i, a, R)$ = la différence de coûts attendus total sur une période infiniment longue

$T_i(R)$ = l'espérance du temps jusqu'au premier retour a l'état r en utilisant la politique R à l'état initial i

$T_r(R)$ = la longueur attendue d'un cycle

$K_i(R)$ =l'espérance du cout encours jusqu'au premier retour a l'état r en utilisant la politique R à l'état initial i

$w_i(R)$ =la fonction de valeur relative particulière

$v_i - v_j$ =la différence entre les coûts attendus totaux

$\overline{R}$ =la nouvelle politique de comparaison

g^* =coût moyenne minimale à long terme par unité de temps.

t : le temps

Liste des illustrations, graphiques et tableaux

Introduction

Dans l'analyse de nombreux systèmes opérationnels, les concepts d'état d'un système et de transition d'état revêtent une importance fondamentale. Pour les systèmes dynamiques qui évoluent selon une distribution de probabilité donnée, le modèle de Markov est souvent approprié, cependant dans de nombreuses situations en environnement incertain, les transitions du système d'un état à un autre peuvent être contrôlées moyennant une séquence d'actions, le modèle correspondant est appelé *processus séquentiel de décision markovien* qui est à la base *de la programmation dynamique stochastique [TJIMS][1].*

Le mot *programmation* ne doit être compris dans le sens qu'on lui donne en informatique, il signifie précisément *résolution de problèmes.* Le problème est en général mis sous forme de "programme mathématique" où à chaque décision possible a été associée une variable. On cherche une solution c'est à dire un ensemble de décisions qui est optimal vis-à-vis d'un critère appelé fonction coût (ou fonction économique).

Le mot *dynamique* signifie que le temps intervient d'une façon cruciale dans la résolution:dans de nombreuses applications il s'agit essentiellement d'aider a prendre des décisions échelonnées dans le temps, nous verrons qu'on outre le fondement même de la méthode est une optimisation récursive "période après période".

Le processus de décision markovien (markovien décision processus MDP) permet la modélisation d'un large éventail de problèmes d'optimisations. Il trouve des applications en Gestion de Stock, Maintenance, Allocation de Ressources, Ordonnancement, Télécommunication et dans certains problèmes d'analyse de systèmes informatique

Dans la Section 1.2, nous présentons les éléments de base du modèle de décision Markovien à temps discret. Apres avoir donnée une définition de " politique stationnaire" à la Section 1.3,la procédure d'amélioration de la politique est discuté à la Section 1.4, cette procédure est la clé de divers algorithmes de recherche de politique

de cout moyen optimale. Les valeurs dites relatives d'une politique donnée jouent un rôle important dans la procédure d'amélioration. Les valeurs relatives et leur interprétation fait l'objet de la Section 1.5, Dans la Section 1.6 nous présentons l'algorithme d'itération de politique qui génère une séquence de politiques améliorées.La Section 1.7 traite de la méthode alternative d'itération de la valeur qui évite la résolution lourde des systèmes d'équations linéaires, mais implique seuls les calculs récursifs.

Le chapitre 2 contient l'essentiel de notre contribution. Nous présentons d'abord le problème de maintenance dans la Section 2.1 avec le modèle et la solution obtenue par les deux méthodes. Le problème d'électricité fera l'objet de la Section 2.2, ce dernier est traité par une seule méthode " policy –iteration algorithm"

CHAPITRE I

PROCESSUS DE DÉCISION MARKOVIEN (CAS DISCRET)

1.1 Définitions

Un processus de décision markovien (MDP) est un processus de contrôle stochastique discret défini par :

un ensemble d'états I, qui peut être fini, dénombrable ou continu. Cet ensemble définit l'environnement tel que perçu par l'agent .

un ensemble d'actions A, qui peut être fini, dénombrable ou continu et dans lequel l'agent choisit les interactions qu'il effectue avec l'environnement .

une fonction de transition $P_{ij}(a)$, cette fonction représente la probabilité de se retrouver dans l'état j en effectuant l'action a, sachant que l'on était à l'instant d'avant dans l'état i.

une fonction de coût $C_i(a)$ (fonction de gain) : elle définit la récompense (positive ou négative) reçue par l'agent. Cette fonction permet de déterminer le(les) but(s) à atteindre et les éventuelles zones dangereuses de l'environnement.

À chaque étape, le processus est dans un certain état i, et l'agent choisit une action a . La probabilité que le processus arrive à l'état j est déterminé par l'action choisie. Plus précisément, elle est décrite par la fonction de transition d'états $P_{ij}(a)$.

Quand le processus passe de l'état i à l'état j avec l'action a, l'agent gagne une récompense $C_i(a)$.

1.2 Le modèle

Considérons un système qui peut être modélisé comme un processus stochastique discret avec la propriété markovienne (i.e., chaîne de Markov discrète *annexe 1*). En tout moment, le système se retrouve dans un des $(M+1)$ états possibles : $\{0, \ldots, M\}$.

À chaque fois que nous observons le système (processus), il faut prendre une décision,dans un ensemble de décisions disponibles $\{1, \ldots, K\}$

Nous avons considéré un système dynamique qui évolue au fil du temps selon une loi de probabililé l'hypothèse markovienne. Cette hypothèse indique que le prochain état à visiter ne dépend que du présent état du système.

Dans ce chapitre, nous abordons un système dynamique *(annexe 2)* évolutif dans le temps où la loi de probabililé du mouvement peut être contrôlée en prenant des décisions. En outre, les coûts sont engagés (ou les gains sont gagnés) en conséquence des décisions qui sont réalisés lorsque le système évolue avec le temps.

Nous présentons maintenant le modèle de décision de Markov. Considérons un système dynamique qui est observé à des instants $t = 0, 1,$. . . . a chaque instant d'observation, le système est classé dans l'un des nombre possible d'états et par la suite une décision doit être fait. L'ensemble des états possibles est indiqué par I Pour chaque état $i \in I$, un ensemble $A(i)$ de décisions ou d'actions est donné. Les espaces d'état I et les actions $A(i)$ sont supposés être finis.Les conséquences économiques des décisions prises à les temps de révision (périodes de décision) se reflètent dans les coûts. cette dynamique contrôlée système s'appelle un modèle de Markov à temps discret.

Si, au moment de la décision, l'action a est choisie dans l'état i, alors indépendamment de passer du système, ce qui suit se produit :

(a) un coût immédiat $c_i(a)$est engagé,

(b) À l'étape suivante de la décision, le système sera dans l'état j avec probabilité

$p_{ij}(a)$ avec $\sum p_{ij}(a) = 1 \quad i\ ,\ j \in I.$

Notons que la seule étape coûte $c_i(a)$ et les probabilités de transition en une étape $p_{ij}(a)$ sont supposées être homogènes.

$c_i(a)$ représente souvent le coût prévu engagé jusqu'au prochain instant de décision lorsque l'action a est choisie dans l'état i. En outre, il convient de souligner que le choix de l'espace d'états et les actions dépend souvent de la structure des coûts du problème spécifique considéré. De nombreux problèmes pratiques de contrôle peuvent être modélisés par les processus de décision markovien par un choix approprié de l'espace d'états et les actions (voir exemple section 2.1 : problème de maintenance) .

1.3 Politiques stationnaires

Politique (policy)

Une politique est une stratégie (choix d'une action ou décision) pour chaque état, c'est un ensemble de règles **"if state then action".**

Politiques stationnaires

L'ensemble des actions choisies par une certaine politique à l'état i est noté par $A(i)$. Ainsi, l'action choisie par une politique peut, par exemple, dépendre de l'histoire du processus jusqu' à cet instant.

Elle peut être aléatoire dans le sens où elle choisit une action « a» avec une certaine probabilité P_a, a une action de $A(i)$. Une sous-classe importante parmi toutes les classes des politiques est celle des politiques stationnaires.

Une politique R est dite stationnaire si elle n'est pas aléatoire et l'action choisie à l'instant t dépend seulement de l'état du processus en cet instant.

Une politique stationnaire est une fonction de l'espace d'état dans l'espace d'action.

Il découle facilement que si une politique stationnaire R est employée, alors la suite des états $\{X_t, t = 0, 1, 2, ...\}$ forme une chaîne de Markov avec des probabilités de transition $P_{ij} = P_{ij}[R(i)]$ et ainsi le processus est appelé processus Markovien de

décision.

Politique optimale

une politique optimale est celle qui maximise (ou minimise) les récompenses.

Notre but est de déterminer une politiques optimale d'exécution d'un processus pour un critère ou une fonction objective donnée.

1.4 L'idée de la politique amélioreé (the policy-improvement idea)

Dans cette section, nous établirons un résultat clé qui sous-entend les différents algorithmes pour le calcul d'une politique de coût moyen optimale, avant de faire cela, nous discutons le coût moyen à long terme par unité de temps pour une politique stationnaire.

Le coût moyen pour une politique stationnaire donnée

Fixer une politique stationnaire R. Selon la politique R chaque fois que l'action $a = R_i$ est prise à l'état i. Le processus $\{X_n\}$ décrivant l'état du système aux époques de décision est une chaîne de Markov avec des probabilités de transition $p_{ij}(R_i)$ i , j $\in I$.lorsque la politique R est utilisée. On définit les probabilités de transition $p_{ij}^{(n)}$ sur n instant de décision par :

$p_{ij}^{(n)}(R) = P\{X_n = j | X_0 = i\}, i$, $j \in I$. et $n = 1, 2, , ,$

Notez que $p_{ij}^{(1)}(R) = p_{ij}(R_i)$ par les équations

$$p_{ij}^{(n)}(R) = \sum_{k \in I} p_{ik}^{(n-1)}(R) p_{kj}(R_k) \qquad \text{(1-4-1)}$$

$p_{ij}^{(0)}(R) = 1$ pour $j = i$ et $p_{ij}^{(0)}(R)$=0 pour $j \neq i$

On définit également les fonctions cout $V_n(i, R)$ par :

$V_n(i, R)$=l'esperance du cout total sur les n premiers instants de decision si i est l'état initial et R la politique utiliseé .

Évidemment, nous avons

$V_n(i,R) = \sum_{t=0}^{n-1} \sum_{j\in I} p_{ij}^{(t)}(R) c_j(R_j)$ (1-4-2)

Ensuite, nous définissons la fonction coût moyen $g_i(R)$ par:

$g_i(R)$ =représente l'esperance du cout moyen par unité de temps à long- terme

$g_i(R) = \lim_{n\to\infty} \frac{1}{n} V_n(i,R)$ $i \in I$ (1-4-3)

Cette limite existe et représente l'esperance du cout moyen par unité de temps lorsque le système est contrôlé par la politique R à l'état initiale i.

Le coût moyen à long terme par unité de temps $= g_i(R)$ (1-4-4)

Propriété de la chaine unique (unichain assumption)

La distrubution (probabilité) stationnaire π_j existe sous l'hypothèse de la chaine de markov irreductible (hypothèse de la chaine unique" unichain assumption",voir propriété 3.5 Bruno bayna [2])

$\pi_j(R) = \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} p_{ij}^{(n)}(R)$ pour tout $j \in I$ (1-4-5)

Autrement dit, le regime stationnaire existe uniquement si l'espace des états forme une chaine unique (l'espace des états ne peut pas etre réparti en deux ensembles disjoints fermés).

Les $\pi_j(R)$ sont la solution unique au système des équations d'équilibre

$\pi_j(R) = \sum_{i\in I} p_{ij}(R_i)\pi_i(R),\ j \in I$ (1-4-6)

avec $\sum_{j\in I} \pi_j(R) = 1$ à partir de (1-4-2) (1-4-3) (1-4-5)

$g_i(R) = g(R)$ pour tout $i \in I$ avec

$g(R) = \sum_{j\in I} c_j(R_j)\pi_j(R)$ (1-4-7)

L'idée de la poltique amélioreé

Une politique stationnaire R^* est considérée comme optimale si

$g_i(R^*) \leq g_i(R)$

Pour chaque politique stationnaire (est la règle qui prescrit une action unique chaque fois que le système se trouve à l'état i). On admet sans preuve qu'une politique stationnaire optimale existe toujours. De plus la politique R^* n'est pas seulement optimale parmi la classe de politiques stationnaire, mais elle est également optimale

parmi la classe de toutes les politiques envisageables.

Dans la plupart des applications, il n'est pas possible de trouver une politique optimale en calculant le cout moyen de chaque politique stationnaire. Par exemple, si le nombre d'états est N et qu'il y a deux actions dans chaque état, alors le nombre de politiques stationnaires possibles est de 2^N et ce nombre augmente rapidement au-delà de toute limite pratique. Cependant, plusieurs algorithmes peuvent être donnés qui conduisent de manière efficace à une politique optimale, "policy iteration "et "value iteration " sont les algorithmes les plus utilisés.

Fixons une politique stationnaire R. On suppose que la chaîne de Markov $\{X_n\}$associé à la politique R est unique, le coût moyen $g_i(R) = g(R)$ indépendamment de l'état initial $i \in I$. Le point de départ est la relation évidente $\lim_{n\to\infty} V_n(i,R)/n = g(R)$ pour tout i, où $V_n(i,R)$ l'esperance du cout total sur les n premiers instants de decision si i est l'état initial et R la politique utiliseé .

$V_n(i,R) \approx ng(R) + v_i(R)$ Pour n grand (1-4-8)

Noter que $v_i(R) - v_j(R) \approx V_n(i,R) - V_n(j,R)$ pour n grand ,$v_i(R) - v_j(R)$ mesure la différence dans les coûts totaux attendus en commençant par l'état i plutôt que l'état j, étant donné que la politique R est suivie. Cela explique le nom des valeurs relatives pour le $v_i(R)$. Nous avons l'équation de récurrence

$V_n(i,R) = c_i(R_i) + \sum_{j\in I} p_{ij}(R_i)V_{n-1}(j,R)$,$n \geq 1$ et $i \in I$

Avec $V_0(i,R) = 0$. Cette équation suit par conditionnement sur l'état suivant qui se produit lorsque l'action $a = R_i$ est faite à l'état i sur n instant des décisions. En remplaçant l'expansion asymptotique (1-4-8) dans l'équation de récurrence, on trouve, après avoir annulé les termes communs:

$g(R) + v_i(R) \approx c_i(R_i) + \sum_{j\in I} p_{ij}(R_i)v_j(R)$ $i \in I$ (1-4-9)

L'idée derrière la procédure d'amélioration de la politique R donnée est de considérer la différence de coûts suivante:

$\Delta(i,a,R)$ =la différence de coûts attendus total sur une période infiniment longue.

Cette différence est égale à zéro lorsque l'action $a = R_i$ est choisie. Cette différence est donnée par :

$\Delta(i,a,R)=\lim_{n\to\infty}\left[c_i(a)+\sum_{j\in I}p_{ij}(a)V_{n-1}(j,R)-\left\{c_i(R_i)+\sum_{j=1}p_{ij}(R_i)V_{n-1}(j,R)\right\}\right]$.

En remplaçant (1-4-8) par l'expression entre parenthèses, on constate que pour n grand cette expression est approximativement égale à-

$c_i(a)+\sum_{j\in I}p_{ij}(a)v_j(R)-(n-1)g(R)-\left\{c_i(R_i)+\sum_{j=1}p_{ij}(R_i)v_j(R)-(n-1)g(R)\right\}$.

Cela donne :

$\Delta(i,a,R)\approx c_i(a)+\sum_{j=1}p_{ij}(R_i)v_j(R)-c_i(R_i)-\sum_{j=1}p_{ij}(R_i)v_j(R)$

Ainsi, en utilisant (1-4-9),

$\Delta(i,a,R)\approx c_i(a)+\sum_{j=1}p_{ij}(R_i)v_j(R)-g(R)-v_i(R)$

Cette relation et la définition de $\Delta(i,a,R)$ nous suggèrent de rechercher une action à l'état i afin que la quantité

$$c_i(a)-g(R)+\sum_{j\in I}p_{ij}(a)v_j(R) \qquad (1\text{-}4\text{-}10)$$

Soit t aussi petite que possible. La quantité dans (1-4-10) s'appelle la quantité de la politique améliorée, la discussion heuristique ci-dessus suggère un Théorème principal qui sera la base pour les algorithmes qui seront discutés plus tard. Une preuve directe de ce théorème peut être donné sans utiliser l'une des hypothèses heuristiques mentionnées ci-dessus.

Théorème 1-4-1 (théorème d'amélioration)[1] :

Soient les nombres g et v_i $i \in I$. Supposons que la politique stationnaire $\overline{R}$ ait la propriété

$$c_i(\overline{R})-g+\sum_{j\in I}p_{ij}(\overline{R})v_j \le v_i \text{ pour } i\in I \qquad (1\text{-}4\text{-}11)$$

Alors,le coût moyen à long terme de la politique $\overline{R}$ satisfait

$$g_i(\overline{R}) \le g \quad ,i\in I \qquad (1\text{-}4\text{-}12)$$

Où le signe d'inégalité strict occupe (1-4-12) pour $i = r$ lorsque l'état r est récurrent en vertu de la politique et le signe d'inégalité stricte contient (1-4-11) pour $i = r$. Le résultat est également vrai lorsque l'inégalité signe en (1-4-11) et (1-4-12) est

inversée.

1.5 *La fonction de la valeur relative(The relative value function)*

Dans la section 1-4, nous avons introduit de manière heuristique les valeurs relatives d'une politique stationnaire R donnée. Dans cette section, nous traitons la fonction de la valeur relative. Cela sera fait pour le cas d'une chaîne unique R. Soit un état récurrent de la politique R. In vue de l'hypothèse de la chaîne unique, la chaîne de Markov $\{X_n\}$ associée à la politique R visiteront l'état r après des transitions finement nombreuses, quel que soit l'état initial. Ainsi, on peut définir, pour chaque état $i \in I$

$T_i(R)$ = l'espérance du temps jusqu'au premier retour a l'état r en utilisant la politique R à l'état initial i

$T_r(R)$ = la longueur attendue d'un cycle,pour chaque $i \in I$.

$K_i(R)$ =l'espérance du cout encours jusqu'au premier retour a l'état r en utilisant la politique R à l'état initial i.

Nous utilisons la convention que $K_i(R)$ inclut le coût engagé lors du démarrage de l'état mais exclut le coût engagé lors du retour à l'état r, par la théorie du renewal reward processus, le coût moyen par unité de temps est égal aux coûts prévus encourus en un cycle divisé sur la longueur attendue d'un cycle et donc

$g(R) = \frac{K_r(R)}{T_r(R)}$

Ensuite, nous définissons la fonction de valeur relative particulière

$$w_i(R) = K_i(R) - g(R)T_i(R), \qquad i \in I \qquad \text{(1-5-1)}$$

Note, en conséquence de (1-4-1), la normalisation

$w_r(R) = 0.$

Conformément au résultat heuristique (1-4-9), le théorème suivant montre que le coût moyen $g = g(R)$ et les valeurs relatives $v_i = w_i(R)$, $i \in I$ vérifient le système

d'équations linéaires.

Théorème 1-5-1[1]:

Soit R une politique stationnaire donnée telle que la chaine de markov associé $\{X_n\}$ ne peut pas être repartie en deux ensembles disjoints fermes.Alors :

(a)- Le coût moyen $g(R)$ et les valeurs relatives $w_i(R)$, $i \in I$, verifient les conditions suivantes du système d'équations linéaires

$v_i = c_i(R_i) - g + \sum_{j \in I} p_{ij}(R_i)v_j, \qquad i \in I \quad (1\text{-}5\text{-}2)$

(b)- Laissez les nombres g et v_i, $i \in I$, soit une solution à (1-5-2). Alors $g = g(R)$

Et, pour un certain constant c, $v_i = w_i(R) + c$, $i \in I$.

(c)- Soit un état arbitrairement choisi.Ensuite,les équations linéaires (1-5-2) ensemble avec l'équation de normalisation $v_s = 0$ ont une solution unique.

Interprétation des valeurs relatives:

Les équations (1-5-2) sont appelées les équations de détermination de la valeur. La fonction de valeur relative v_i, $i \in I$ est unique jusqu'à une constante additive,l'unique solution (1-5-1) peut être interprétée comme le coût total attendu jusqu'à ce que retourner à l'état r lorsque la politique R est utilisée et les coûts en une étape sont donnés par $c'_i(a) = c_i(a) - g$ avec $g = g(R)$ si la chaîne Markov $\{X_n\}$ associée à la politique R est apériodique, deux autres interprétations peuvent être données à la fonction de valeur relative. la première interprétation est que, pour tous les deux états i, $i \in I$

$v_i - v_j$ =La différence entre les coûts attendus totaux sur une base infinie longue période en commençant par l'état i plutôt que dans état j lorsque vous utilisez la politique R.

En d'autres termes, $v_i - v_j$ est le montant maximal qu'une personne rationnelle est disposée à payer pour démarrer le système dans l'état j plutôt que dans l'état i lorsque le système est contrôlé par la règle R,Cette interprétation est une conséquence

simple de (1-5-3). En utilisant l'hypothèse que la chaîne de Markov $\{X_n\}$ est apériodique,nous avons cela $\lim_{m\to\infty} p_{ij}^{(m)}(R)$ existe ,en outre, cette limite est indépendante de l'état initial i, puisque R est chaine unique. ainsi, par (1-5-3),

$$v_i = \lim_{m\to\infty} \{V_m(i,R) - mg\} + \sum_{j\in I} \pi_j(R) v_j. \qquad (1\text{-}5\text{-}5)$$

Cela implique que $v_i - v_j = \lim_{m\to\infty} \{V_m(i,R) - V_m(j,R)\}$,une interprétation spéciale s'applique à la fonction de valeur relative v_j $i \in I$ avec la propriété $\sum_{j\in I} \pi_j(R) v_j = 0$,comme la fonction de valeur relative est unique jusqu'à une constante additive, il existe une fonction de valeur relative unique avec cette propriété. Indiquez cette fonction de valeur relative par h_i, $i \in I$. Ensuite, par (1-5-5),

$$h_i = \lim_{m\to\infty} \{V_m(i,R) - mg\}. \qquad (1\text{-}5\text{-}6)$$

Le biais h_i peut également être interprété comme la différence entre les coûts prévus totaux entre le système dont l'état initial est i et le système dont l'état initial est distribué selon la répartition à l'équilibre $\{\pi_j(R), j \in I\}$ lorsque les deux systèmes sont contrôlé par la politique R, ce dernier système s'appelle le système stationnaire. Ce système a la propriété qu'à toute décision l'époque de l'état est distribuée comme $\{\pi_j(R)\}$ ainsi, pour le système stationnaire, le coût prévu encouru à n'importe quel l'époque de décision est égale $\sum_{j\in I} c_j(R_j)\pi_j(R)$ étant le coût moyen $g = g(R)$de la politique R. par conséquent, dans le système stationnaire, les coûts totaux attendus sur le premier m époques de décision sont égales à mg. Cela donne l'interprétation ci-dessus du biais h_i puissance.

1.6 Algorithme d'itération par politiques (Policy-iteration algorithm)

Les valeurs relatives associées à une politique donnée R fournissent un outil pour la construction d'une nouvelle politique $\overline{R}$ dont le coût moyen n'est pas supérieur à celui de la politique actuelle R. Afin d'améliorer une politique R donnée dont le coût moyen $g(R)$ et les valeurs relatives $v_i(R)$, $i \in I$ ont été calculés, nous appliquons le

Théorème (1-4-1) avec $g = g(R)$ et $v_i = v_i(R)$, $i \in I$. En construisant une nouvelle politique R telle que, pour chaque état $i \in I$,

$$c_i(\overline{R_i}) - g(R) + \sum_{j \in I} p_{ij}(\overline{R_i}) v_j \leq v_i \qquad (1\text{-}6\text{-}1)$$

On obtient une règle améliorée R selon $g(\overline{R}) \leq g(R)$. En construisant une politique améliorée R il est important de réaliser que pour chaque état i séparément une action $\overline{R_i}$ satisfaisant (1-6-1) peut être déterminé, nous indiquons que cette flexibilité de la procédure d'amélioration des politiques peut être exploitée des applications spécifiques pour générer une séquence de politiques améliorées dans une sous-classe de politiques ayant une structure simple. Une façon particulière de trouver pour l'état $i \in I$ l'action $\overline{R_i}$ satisfaisant (1-6-1) est de minimiser :

$$c_i(a) - g(R) + \sum_{j \in I} p_{ij}(a) v_j(R) \qquad (1\text{-}6\text{-}2)$$

Par rapport $a \in A(i)$. Notant que l'expression dans (1-6-1) est égale à $v_i(R)$ pour $a = R_i$ alors (1-6-1) est satisfaite pour l'action R_i qui minimise (1-6-2) par rapport $a \in A(i)$, nous sommes maintenant en mesure de formuler les éléments suivants de l'algorithme.

Policy-iteration algorithm[1]

Étape 0 (initialisation), choisir une politique stationnaire R

Étape 1 (étape de détermination de la valeur), pour la règle actuelle R, calculer l'unique solution $\{g(R), v_i(R)\}$ au système d'équations linéaires suivantes :

$v_i = c_i(R_i) - g(R) + \sum_{j \in I} p_{ij}(R_i) v_j, \quad i \in I$

$v_s = 0,$

Où s est un statut arbitrairement choisi.

Étape 2 (étape d'amélioration de la politique),pour chaque état $i \in I$, déterminer une action a_i donnant le minimum en $\min_{a \in A(i)} \left\{ c_i(a) - g(R) + \sum_{j \in I} p_{ij}(a) v_j(R) \right\}$.

La nouvelle politique stationnaire R est obtenue en choisissant $R_i = a_i$ pour tout $i \in I$ avec laquelle R_i est choisi égal à l'ancienne action R_i lorsque cette action minimise la quantité d'amélioration de la politique.

Étape 3 (test de convergence). Si la nouvelle politique $\overline{R} = R$, l'algorithme est arrêté avec la politique R. Sinon passer à l'étape 1 avec R remplacé par $\overline{R}$.

L'algorithme d'itération de politique converge après un nombre fini d'itérations à une politique optimale de coût moyen.

Remarque 1-6-1 :

l'algorithme d'itération par politiques est également connu par "l'algorithme des améliorations successives de Howard[3] ".

L'équation moyenne d'optimalité des coûts

Puisque l'algorithme d'itération de politique converge après plusieurs itérations,il existe des nombres g^* et $v_i^* i \in I$ tel que:

$$v_i^* = \min_{a \in A(i)} \left\{ c_i(a) - g^* + \sum_{j \in I} p_{ij}(a) v_j^* \right\}. \quad i \in I \qquad (1\text{–}6\text{-}3)$$

Cette équation fonctionnelle s'appelle l'équation moyenne d'optimisation des coûts. En utilisant le Théorème (1-4-1), nous pouvons vérifier directement que toute politique stationnaire R^* pour laquelle l'action R_i^* minimise le côté droit de (1-6-3) pour tout $i \in I$, est un coût moyen optimal. Pour voir cela, notons que

$$v_i^* = c_i(R_i^*) - g^* + \sum_{j \in I} p_{ij}(R_i^*) v_j^*, \quad i \in I \qquad (1\text{–}6\text{-}4)$$

et

$$v_i^* \leq c_i(a) - g^* + \sum_{j \in I} p_{ij}(a) v_j^*, \quad a \in A(i) \text{ et } i \in I \qquad (1\text{–}6\text{-}5)$$

L'égalité (1–6-4) et le Théorème (1-4-1) impliquent que $g(R^*) = g^*$, R une politique stationnaire,prendre $a = R_i$ dans (1–6-5) pour tout $i \in I$ et appliquer le Théorème 1-4-1, On trouve $g(\overline{R}) \geq g^*$. En d'autres termes, $g(R^*) \leq g(\overline{R})$ pour toute politique stationnaire R. Cela montre non seulement que la politique R^* est un coût moyen optimal mais montre également que la constant g^* dans (1–6-3) est déterminé de manière unique en tant que coût moyen minimum par unité de temps, il est indiqué sans preuve que la fonction v_i^*, $i \in I$, pour (1–6-3) est uniquement déterminé jusqu'à une constante additive.

Ensuite, l'algorithme d'itération de politique est appliqué pour calculer un coût

moyen optimal pour le problème de maintenance (voir section 2.1)

1.7 Algorithme d'itération par valeurs (value-iteration algorithm)

La notion de l'algorithme d'itérations par valeurs et de la formulation de la programmation linéaire dans les deux cas, il faut qu'à chaque itération d'un système d'équations linéaires de la même taille que l'espace d'état est résolu. En général, ce seront des calculs fastidieux pour un grand espace d'état et rend ces algorithmes de calculs à grande échelle d'intérêt pour les problèmes de décision markovienns. Dans cette section, nous allons voir un autre algorithme qui évite de résoudre les systèmes d'équations linéaires mais utilise plutôt l'approche de la solution récursive à partir de la programmation dynamique. Cette méthode est l'algorithme d'itération par valeur qui calcule récursivement une séquence de fonctions d'approximation de la valeur moyenne minimale coût par unité de temps. La valeur des fonctions limites inférieure et supérieure sur le coût moyen minimal et sous une certaine condition périodicité ces limites convergent pour le coût moyen minimal. Le nombre d'itérations est dépendant du problème et généralement l'augmentation du nombre de membres du problème à l'examen. Un autre avantage important d'itération de la valeur, c'est qu'il est habituellement facile d'écrire un code pour des applications spécifiques. En exploitant la structure de l'application particulière de la mémoire de l'ordinateur évite généralement l'un des problèmes qui peuvent être rencontrés lors de l'utilisation de la notion d'itération. L'itération de la valeur n'est pas seulement une méthode puissante pour les chaînes de Markov, mais c'est aussi un outil pour calculer des limites sur les mesures du rendement à une seule chaîne de Markov. Dans cette section l'algorithme d'itération par valeur sera analysé sous l'hypothèse de la chaîne unique selon cette hypothèse le coût moyen par unité de temps est indépendant de l'état initial.

g^* =la moyenne minimale des coûts à long terme par unité de temps.

La valeur d'itération calcule l'algorithme-récursivement pour $n = 1, 2, ...$ la fonction de valeur $v_n(i)$ à partir de

$$v_n(i) = \min_{a \in A(i)} \left\{ c_i(a) + \sum_{j \in I} p_{ij}(a) v_{n-1}(j) \right\}. \quad i \in I \tag{1-7-1}$$

En commençant par une fonction choisie arbitrairement $v_0(i)$, $i \in I$ la quantité $v_n(i)$ peut être interprétée comme le minimum des coûts totaux prévus avec n périodes de gauche à l'horizon de temps lorsque l'état actuel est i et un coût de la borne $v_0(j)$ est engagé lorsque le système se retrouve au niveau de l'état j

Intuitivement, on pourrait s'attendre à ce que la différence en une étape $v_n(i) - v_{n-1}(i)$ soit très proche de la moyenne minimal coût par unité de temps et que la politique de l'arrêt dont l'action de minimiser le côté droit de (1-7-1) Pour tout i soit très proche en coût par rapport au coût moyen minimal. Cependant, ces questions semblent être plutôt subtil pour le critère de coût moyen en raison de l'effet de périodicité possible dans le processus de décision. Avant d'expliquer cela plus en détail, nous nous intéressons à un opérateur qui est induite par l'équation de récurrence (1-7-1). L'opérateur T ajoute à chaque fonction $v = (v_i, i \in I)$ une fonction Tv dont la composante i IIème $(Tv)_i$ est définie par:

$$(Tv)_i = \min_{a \in A(i)} \left\{ c_i(a) + \sum_{j \in I} p_{ij}(a) v_j \right\}, i \in I \tag{1-7-2}$$

Notez que $(Tv)_i = v_n(i)$ si $v_i = v_{n-1}(i)$, $i \in I$.

Le théorème suivant joue un rôle clé dans l'algorithme d'itération par valeurs

Théorème(1-7-1)[1]:

Supposons que l'hypothèse de la chaine unique est satisfaite. Soit $v = (v_i)$. Définir la politique stationnaire $R(v)$ ainsi qu'une politique qui s'ajoute à chaque état $i \in I$ une action $a = R_i(v)$ qui minimise le côté droit de (1-7-2). Puis

$$\min_{i \in I} \{(Tv)_i - v_i\} \leq g* \leq g_s(R(v)) \leq \max_{i \in I} \{(Tv)_i - v_i\} \tag{1-7-3}$$

Pour tout $s \in I$, où g^* est la moyenne à long terme minimal coût par unité de temps et de $g_s(R(v))$ désigne la moyenne à long terme coût par unité de temps sous politiques $R(v)$ lorsque l'état initial est s.

preuve:

Pour prouver la première inégalité, supposons une politique quelconque R. Par la définition de $(Tv)_i$, nous avons pour tout état $i \in I$ que

$(Tv)_i \leq c_i(a) + \sum_{j\in I} p_{ij}(a)v_j \qquad a \in A(i)$ (1-7-4)

le signe de l'égalité est pour $a = R_i(v)$, le choix d'un $a = R_i$ en (1-7-4) donne

$(Tv)_i \leq c_i(R_i) + \sum_{i\in I} p_{ij}(R_i)v_j \quad i \in I$ (1-7-5)

Définissions de la limite inférieure

$m = \min_{i\in I}\{(Tv)_i - v_i\}$

Puisque $m \leq (Tv)_i - v_i$ pour tout i, il découle de (1-7-5) que $m + v_i \leq c_i(R_i) + \sum_{j\in I} p_{ij}(a)v_j$ pour tout $i \in I$, et donc $c_i(R_i) - m + \sum_{j\in I} p_{ij}(R_i)v_j \geq v_i, \quad i \in I$

Une application du Théorème (1-4-1) que donne maintenant

$g_i(R) \geq m, \quad i \in I$

Cette inégalité est vraie pour chaque politique R et $g* = \min_R g_i(R) \geq m$ ce qui prouve la première inégalité dans (1-7-3). La preuve de la dernière inégalité dans (1-7-3) est très similaire. Par la définition de la politique $R(v)$,

$(Tv)_i = c_i(R_i(v)) + \sum_{j\in I} p_{ij}(R_i)v_j, \quad i \in I$ (1-7-6)

Déterminons la limite supérieure

$M = \max_{i\in I}\{(Tv)_i - v_i\}$

Puisque $M \geq (Tv)_i - v_i$ pour tout $i \in I$, on obtient de (1-7-6) que

$c_i(R_i(v)) - M + \sum_{j\in I} p_{ij}(R_i(v))v_j \leq v_i \qquad i \in I$

Nous avons maintenant formulé l'algorithme d'itération de la valeur. Dans la formulation, il n'existe pas de restriction à supposer que

$c_i(a) > 0$ pour tous $i \in I$ et $a \in A(i)$.

Sinon, ajouter une constante positive pour chaque $c_i(a)$. Cela affecte le coût moyen de chaque politique par la même constante.

Algorithme d'itération de la valeur[1] :

Étape 0 (initialisation). choisir $V_0(i)$, $i \in I$ avec $0 \leq V_0(i) \leq \min_a c_i(a)$ soit

$n := 1$.

Étape 1 (étape de la valeur-itération). Pour chaque état $i \in I$, calculer

$V_n(i) = \min_{a \in A(i)} \left\{ c_i(a) + \sum_{j \in I} p_{ij}(a) V_{n-1}(j) \right\}$

Soit $R(n)$ une politique stationnaire telle que l'action $a = R_i(n)$ minimise la côté droit de l'équation pour $V_n(i)$ pour chaque état i.

Étape 2 (limites sur les coûts minimaux). Calcule les limites

$m_n = \min_{i \in I} \{V_n(i) - V_{n-1}(i)\}, \qquad M_n = \max_{i \in I} \{V_n(i) - V_{n-1}(i)\},$

Étape 3 (test d'arrêt). si

$0 \leq M_n - m_n \leq \epsilon m_n$

Avec $\epsilon > 0$ un numéro de précision spécifié (par exemple $\epsilon = 10^{-3}$), ou s'arrête avec la politique $R(n)$.

Étape 4 (suite). $N := n + 1$ et répéter l'étape 1.

Par le Théorème (1-6-1), nous avons

$$0 \leq \frac{g_i(R(n)) - g*}{g*} \leq \frac{M_n - m_n}{m_n} \leq \epsilon, \quad i \in I \tag{1-6-7}$$

Lorsque l'algorithme est arrêté après n itérations avec la politique $R(n)$. En d'autres mots, le coût moyen de la politique $R(n)$ ne peut pas dépasser plus de $100\epsilon\%$ de le coût moyen théoriquement minimal lorsque les limites m_n et M_n satisfont $0 \leq M_n - m_n \leq \epsilon m_n$ dans les applications pratiques, on est généralement satisfait d'une politique dont le coût moyen est suffisamment proche du coût moyen théoriquement minimal.

Remarque :

L'algorithme d'itérations par valeurs est également connu par « algorithme de la valeur itérée »

Convergence des limites

La question restante est de savoir si les limites inférieures et supérieures m_n et M_n convergent à la même limite afin que l'algorithme soit arrêté après plusieurs itérations [4]. La réponse est oui, seulement si une certaine condition d'apériodicité

est satisfaite. En général m_n et M_n n'ont pas besoin d'avoir la même limite, comme il peut être vu par l'exemple suivant.

Considérons le problème trivial de décision markovien avec deux états 1 et 2 et un seul action a_0 dans chaque état.Les coûts et les probabilités de transition sont donnés par $c_1(a_0) = 1, c_2(a_0) = 0, p_{12}(a_0) = p_{21}(a_0) = 1$ et $p_{11}(a_0) = p_{22}(a_0) = 0$ ensuite, le système se déplace entre les états 1 et 2. Il est facile a vérifié que $V_{2k}(1) = V_{2k}(2) = k$ et $V_{2k-1}(2) = k - 1$ pour tout $k \geq 1$ Par conséquent $m_n = 0$ et $M_n = 1$. Pour tout n, impliquant que les séquences $\{M_n\}$ et $\{m_n\}$Ont des limites différentes. La raison du comportement oscillant de $V_n(i) - V_{n-1}(i)$ est la périodicité de la chaîne de Markov décrivant l'état de la chaîne de Markov système. Le prochain Théorème donne des conditions suffisantes pour la convergence des algorithme d'itérations de valeurs.

Théorème (1-7-2)

Supposons que l'hypothèse de la chaîne unique se maintienne et que pour chaque coût moyen optimal, la chaîne Markov associée $\{X_n\}$ est apériodique. Ensuite, il existe des constantes finies $\alpha > 0$ et $0 < \beta < 1$ telles que $|M_n - m_n| \leq \alpha\beta^n, \quad n \geq 1$

En particulier,$\lim_{n\to\infty} M_n = \lim_{n\to\infty} m_n = g^*$.

Nous prouvons le résultat intéressant que les séquences $\{m_n\}$ et $\{M_n\}$ sont toujours monotone indépendamment de la structure en chaîne des chaînes de Markov.

Théorème (1-7-3)

Dans l'algorithme d'itération par valeurs, le bas et le haut les limites satisfont

$m_{k+1} \geq m_k$ et $M_{k+1} \leq M_k$ pour tout $k \geq 1$

Preuve

Selon la définition de la politique $R(n)$,

$V_n(i) = c_i(R_i(n)) + \sum_{j\in I} p_{ij}(R_i(n))V_{n-1}(j) \quad i \in I$ (1-7-8)

De la même manière que (1-7-5) a été obtenue, nous trouvons pour toute politique R que $c_i(R_i) + \sum_{j\in I} p_{ij}(R_i)V_{n-1}(j) \geq V_n(i) \quad i \in I$ (1–7-9)

Prendre $n = k$ dans (1-6-8) et prendre $n = k + 1$ et $R = R(k)$ dans (1-7-9) donne

$V_{k+1}(i) - V_k(i) \leq \sum_{j \in I} p_{ij}(R_i(k)) \{V_k(j) - V_{k-1}(j)\}, \quad i \in I$ (1-7-10)

De même, en prenant $n = k+1$ dans (1-7-8) et en prenant $n = k$ et $R = R(k+1)$ dans (1-7-9), nous trouvons

$V_{k+1}(i) - V_k(i) \geq \sum_{j \in I} p_{ij}(R_i(k+1)) \{V_k(j) - V_{k-1}(j)\}, \quad i \in I$ (1-7-11)

Puisque $V_k(j) - V_{k-1}(j) \leq M_k$ pour tout $j \in I$ et $\sum_{j \in I} p_{ij}(R_i(k)) = 1$ il suit à partir de (1-7-10) que $V_{k+1}(i) - V_k(i) \leq M_k$ pour tout $i \in I$ cela donne $M_{k+1} \leq M_k$ de même, nous obtenons de (1-7-11) que $m_{k+1} \geq m_k$.

CHAPITRE II

APPLICATIONS ET PROGRAMMATION

2.1 Le problème de la maintenance

Au début de chaque journée, une pièce d'équipement est inspectée pour révéler son véritable état de fonctionnement.

L'équipement sera trouvé dans l'une des conditions de travail $i = 1, ..., N$, où la condition de travail i est meilleur que l'état de fonctionnement $i + 1$.

L'équipement se détériore dans le temps, si l'actuel état de fonctionnement est i et aucune réparation n'est fait, alors au début de la prochaine journée, le matériel sera dans la condition de travail j (l'état j) avec probabilité q_{ij}.

Il est supposé que $q_{ij} = 0$ pour $j < i$ et $\sum_{j \geq i} q_{ij} = 1$.

La condition de travail $i = N$ représente un dysfonctionnement qui nécessite une réparation forcée qui prend deux jours, pour les états intermédiaires i avec $1 < i < N$ il y a un choix entre réparation préventive de l'équipement ou bien laisser l'appareil(l'équipement) fonctionner pendant une journée.

Une réparation préventive ne prend qu'un seul jour et un système réparé se retrouve à la condition de travail $i = 1$. Le coût d'une réparation en cas de panne est C_f et le coût d'une réparation préventive dans la condition de travail i est C_{pi}.

Nous voulons déterminer une règle de maintenance qui minimise les coûts de réparation moyen à long terme par jour.

Modélisation:

Ce problème peut être mis dans le cadre d'un processus de markov à temps discret.

Puisque une réparation forcée prend deux jours et l'état du système doit être défini

au début de chaque jour, donc nous avons besoin d'un état auxiliaire de la situation dans laquelle une réparation est déjà en cours pour une journée.

Ainsi, l'ensemble des états possibles du système est choisi comme $I = \{1, 2, ..., N, N+1\}$, l'état i avec $1 \leq i \leq N$ correspond à la situation dans laquelle l'inspection révèle une condition de travail i, alors que l'état $N + 1$ correspond à la situation où une réparation est déjà en cours pour une journée.

Définir les actions :

$$a = \begin{cases} 0 & \text{Si aucune réparation n'est effectuée} \\ 1 & \text{Si une réparation préventive est effectuée} \\ 2 & \text{Si une réparation forcée est effectuée} \end{cases}$$

L'ensemble des actions possibles à l'état « i» est choisi comme

$A(1) = \{0\}$,

$A(i) = \{0, 1\}$ pour $1 < i < N$,

$A(N) = A(N + 1) = \{2\}$.

Les probabilités de transition d'une étape $p_{ij}(a)$ sont données par :

$p_{ij}(0) = q_{ij}$ pour $1 \leq i < N$,

$p_{i1}(1) = 1$ pour $1 < i < N$,

$p_{N,N+1}(2) = p_{N+1,1}(2) = 1$ et l'autre $p_{ij}(a) = 0$.

Les coûts d'une étape sont donnés par :

$c_i(0) = 0, \quad c_i(1) = c_{pi}, \quad c_N(2) = c_f \quad$ et $\quad c_{N+1}(2) = 0;$

Exemple : Il est supposé que le nombre de conditions de travail possibles est égal à $N = 5$. Les coûts de réparation sont donnés par

$C_f = 10$, $C_{p2} = 7$, $C_{p3} = 7$ et $C_{p4} = 5$.

Les probabilités de détérioration sont données dans le tableau suivant

i/j	1	2	3	4	5
1	0.90	0.10	0	0	0
2	0	0.80	0.10	0.05	0.05
3	0	0	0.70	0.10	0.20
4	0	0	0	0.50	0.50

Tableau 2.1.1 les probabilits de detrioration

Résolution:

2.1.1 Algorithme d'itération par politiques (policy-iteration)

L'algorithme d'itération par politique est initialisée avec la politique $R^{(1)} = (0, 0, 0, 0, 2, 2)$, qui prescrit la réparation uniquement dans les états 5 et 6.

Dans les calculs ci-dessous, la politique d'amélioration de la quantité est abrégé en : $T_i(a, R) = C_i(a) - g(R) + \sum_{j \in I} p_{ij}(a) v_j(R)$

Quand la politique actuelle est R remarque toujours que $T_i(a, R) = v_i(R)$ pour $a = R_i$.

Itération 1

Étape 1 (détermination de la valeur): $v_i(R) = C_i(a) - g(R) + \sum_{j \in I} p_{ij}(a) v_j(R)$

Le coût moyen et la valeur relative de la politique $R^{(1)} = (0, 0, 0, 0, 2, 2)$ sont calculées en résolvant les équations linéaires

$v_1 = 0 - g + 0.9v_1 + 0.1v_2$

$v_2 = 0 - g + 0.8v_2 + 0.1v_3 + 0.05v_4 + 0.05v_5$

$v_3 = 0 - g + 0.7v_3 + 0.1v_4 + 0.2v_5$

$v_4 = 0 - 9 + 0.5v_4 + 0.5v_5$

$v_5 = 10 - g + v_6$

$v_6 = 0 - g + v_1$

$v_6 = 0$

Où $s = 6$ est choisie pour l'équation de normalisation $= 0$.

. La solution de ces équations linéaires est donné par :

$g(R^{(1)}) = 0.5128, v_1(R^{(1)}) = 0.5128, v_2(R^{(1)}) = 5.6410, v_3(R^{(1)}) = 7.4359,$

$v_4(R^{(1)}) = 8.4615, v_5(R^{(1)}) = 9.4872, v_6(R^{(1)}) = 0.$

Étape 2 (amélioration des politiques):

Le test quantité $T_i(a, R^{(1)})$ a les valeurs:

$T_2(0, R^{(1)}) = v_2(R^{(1)}) = 5.6410$

$T_2(1, R^{(1)}) = C_{p2} - g(R^{(1)}) + 0.8v_2(R^{(1)}) + 0.1v_3(R^{(1)}) + 0.05v_4(R^{(1)}) + 0.05v_5(R^{(1)}) =$ 7.0000

$T_3(0, R^{(1)}) = v_3(R^{(1)}) = 7.4359$

$T_3(1, R^{(1)}) = C_{p3} - g(R^{(1)}) + 0.7v_3(R^{(1)}) + 0.1v_4(R^{(1)}) + 02v_5(R^{(1)}) = 7.0000$

$T_4(0, R^{(1)}) = v_4(R^{(1)}) = 8.4615$

$T_4(1, R^{(1)}) = C_{p4} - g(R^{(1)}) + 0.5v_4(R^{(1)}) + 0.5v_5(R^{(1)}) = 5.0000.$

Cela produit la nouvelle politique $R^{(2)} = (0, 0, 1, 1, 2, 2)$ de choisir pour chaque état i l'une qui minimise les $T_i(a, R^{(1)})$

Étape 3 (convergence): La nouvelle politique $R^{(2)}$ est différente de la précédente politique $R^{(1)}$ et donc une autre itération est effectuée.

Itération 2

Étape 1 ((détermination de la valeur): le coût moyen et la valeur relative de la politique

$R^{(2)} = (0, 0, 1, 1, 2, 2)$ sont calculées en résolvant les équations linéaires

$v_1 = 0 - g + 0.9v_1 + 0.1v_2$

$v_2 = 0 - g + 0.8v_2 + 0.1v_3 + 0.05v_4 + 0.05v_5$

$v_3 = 7 - g + v_1$

$v_4 = 5 - g + v_1$

$v_5 = 10 - g + v_6$

$v_6 = 0 - g + v_1$

$v_6 = 0.$

La solution de ces équations linéaires est donné par :

$g(R^{(2)}) = 0.4462, v_1(R^{(2)}) = 0.4462, v_2(R^{(2)}) = 4.9077, v_3(R^{(2)}) = 7.000, v_4(R^{(2)}) = 5.0000, v_5(R^{(2)}) = 9.5538, v_6(R^{(2)}) = 0.$

Étape 2 (amélioration des politiques). Le test quantité $T_i(a, R^{(2)})$ a les valeurs :

$T_2(0, R^{(2)}) = 4.9077, T_2(1, R^{(2)}) = 7.0000, T_3(0, R^{(2)}) = 6.8646, T_3(1, R^{(2)}) = 7.0000, T_4(0, R^{(2)}) = 6.8307, T_4(1, R^{(2)}) = 5.0000.$

Cela produit la nouvelle politique $R^{(3)} = (0, 0, 0, 1, 2, 2)$.

Étape 3 (convergence): La nouvelle politique $R^{(3)}$est différente de la précédente politique $R^{(2)}$ et donc une autre itération est effectuée.

Iteration 3

Étape 1 ((détermination de la valeur): le coût moyen et la valeur relative de la politique

$R^{(3)} = (0, 0, 0, 1, 2, 2)$ sont calculées en résolvant les équations linéaires

$v_1 = 0 - g + 0.9v_1 + 0.1v_2$

$v_2 = 0 - g + 0.8v_2 + 0.1v_3 + 0.05v_4 + 0.05v_5$

$v_3 = 0 - g + 0.7v_3 + 0.1v_4 + 0.2v_5$

$v_4 = 5 - g + v_1$

$v_5 = 10 - g + v_6$

$v_6 = 0 - g + v_1$

$v_6 = 0.$

La solution de ces équations linéaires est donnée par :

$g(R^{(3)}) = 0.4338, v_1(R^{(3)}) = 0.4338, v_2(R^{(3)}) = 4.7717, v_3(R^{(3)}) = 6.5982, v_4(R^{(3)}) = 5.0000, v_5(R^{(3)}) = 9.5662, v_6(R^{(3)}) = 0.$

Étape 2 (amélioration des politiques): Le test quantité $T_i(a, R^{(3)})$ à les valeurs :

$T_2(0, R^{(3)}) = 4.7717, T_2(1, R^{(3)}) = 7,\ T_3(0, R^{(3)}) = 6.5987,$

$T_3(1, R^{(3)}) = 7.0000,\ T_4(0, R^{(3)}) = 6.8493,\ T_5(1, R^{(3)}) = 5.0000.$

Cela produit la nouvelle politique $R^{(4)} = (0, 0, 0, 1, 2, 2)$.

Étape 3 (convergence):

La nouvelle politique $R^{(4)}$ est identique à la précédente politique $R^{(3)}$ et donc c'est la politique optimale.

Le coût moyen optimal est $0,4338$ par jour.

2.1.2 Algorithme d'itération par valeurs (value-iteration)

Pour le problème de maintenance l'équation de récurrence (1.6.1) devient:

$v_n(1) = 0 + \sum_{j=1}^{N} q_{1j} v_{n-1}(j),$

$v_n(i) = \min\left\{0 + \sum_{j=i}^{N} q_{ij} v_{n-1}(j), C_{pi} + v_{n-1}(1)\right\},\ 1 < i < N,$

$v_n(N) = C_f + v_{n-1}(N+1),$

$v_n(N+1) = 0 + v_{n-1}(1).$

Nous avons appliqué l'algorithme d'itération par valeurs pour les données numériques du tableau 2.1.1(pour $N = 5$),avec l'exactitude nombre $\varepsilon = 10^{-3}$.

Étape 0:

En prenant $V_0(i) = 0$ pour $i = 1, 2, ...6$

$n = 1$

Étape 1: calculons $V_1(i)$ pour $i = 1, 2, ...6$

$V_1(1) = 0 + \sum_{j=1}^{5} q_{1j} v_0(j) = 0,$

$\mathrm{v}_1(2) = \min\left\{0 + \sum_{j=2}^{5} q_{2j} v_0(j), C_{p2} + v_0(1)\right\} = \min\{0, 7\} = 0,$

$\mathrm{v}_1(3) = \min\left\{0 + \sum_{j=3}^{5} q_{3j} v_0(j), C_{p3} + v_0(1)\right\} = \min\{0, 7\} = 0,$

$\mathrm{v}_1(4) = \min\left\{0 + \sum_{j=4}^{5} q_{4j} v_0(j), C_{p4} + v_0(1)\right\} = \min\{0, 5\} = 0,$

$\mathrm{v}_1(5) = C_f + v_0(6) = 10,$

$\mathrm{v}_1(6) = 0 + v_0(1) = 0.$

Étape 2:

$m_n = \min\{V_1(i) - V_0(i)\} = 0,$

$M_n = \max\{V_1(i) - V_0(i)\} = 10.$

Étape 3:

$M_n - m_n = 10 \notin [0, \varepsilon m_n]$, passer a l'étape suivant

Étape 4:

$n = 2$

Étape 1: calculons $V_2(i)$ pour $i = 1, 2, ...6$

$V_2(1) = 0 + \sum_{j=1}^{5} q_{1j} v_0(j) = 0,$

$v_2(2) = \min\left\{0 + \sum_{j=2}^{5} q_{2j} v_0(j), C_{p2} + v_0(1)\right\} = \min\{0, 7\} = 0,$

$v_2(3) = \min\left\{0 + \sum_{j=3}^{5} q_{3j} v_0(j), C_{p3} + v_0(1)\right\} = \min\{0, 7\} = 0,$

$v_2(4) = \min\left\{0 + \sum_{j=4}^{5} q_{4j} v_0(j), C_{p4} + v_0(1)\right\} = \min\{0, 5\} = 0,$

$v_2(5) = C_f + v_0(6) = 10,$

$v_2(6) = 0 + v_0(1) = 0.$

Étape 2:

$m_n = \min\{V_2(i) - V_1(i)\} = 0,$

$M_n = \max\{V_2(i) - V_1(i)\} = 10.$

Étape 3:

$M_n - m_n = 10 \notin [0, \varepsilon m_n]$, passer a l'étape suivant

Étape 4:

$n = 3...$

Conclusion:

L'algorithme d'itération par valeurs est arrêté après $n = 28$ itérations avec la politique stationnaire $R^{(n)} = (0, 0, 0, 1, 2, 2)$ qui prescrive une réparation préventive uniquement dans l'état 4, avec les limites inférieure et supérieure $m_n = 0,4336$ et $M_n = 0,4340$ respectivement. Le coût moyen d'une politique $R^{(n)}$ est estimé par $1/2(m_n + M_n) = 0,4338$ et ce coût ne peut pas s'écarter du coût moyen minimal théorique de plus de $0,1\%$. En fait $R^{(n)}$ est optimale comme nous le savons à partir des résultats obtenus par le 1^{er} algorithme (itération par politique).

Voici l'affichage de la politique optimale et le coût moyen optimal avec l'algorithme d'itération par valeurs pour les données de l'exemple 2.1.1

```
Etap : 27 ********************************************************************************

V27=(9.030575,13.369263,15.193833,13.596650,18.162642,8.596650)
V28=(9.464443,13.802759,15.627876,14.030575,18.596649,9.030575)
 la valeur max=0.434043
 la valeur min=0.433496
R(27)=(000122)

Etap : 28 ********************************************************************************

V28=(9.464443,13.802759,15.627876,14.030575,18.596649,9.030575)
V29=(9.898274,14.236357,16.061901,14.464443,19.030575,9.464443)
 la valeur max=0.434025
 la valeur min=0.433598
R(28)=(000122)
la politique optimal estR(28)=(0 0 0 1 2 2)
**********le cout mouyen aptimal egal: 0.433811

Process returned 10 (0xA)   execution time : 70.887 s
Press any key to continue.
```

Figure 2.1.1 : L'affichage de la politique optimale et le cout moyen optimal

Pour se faire une idée de la force avec le nombre requis d'itérations dépend de ε, nous avons appliqué une valeur standard-itération pour $\varepsilon = 10^{-2}$ et $\varepsilon = 10^{-4}$. Pour ces choix de la précision nombre ε, standard value itération requis 21 et 35 itérations respectivement (voir les figures 2.1.2 et 2.1.3).

```
R(20)=(000122)

Etap : 21 ********************************************************************************

V21=(6.425008,10.772409,12.594608,10.989975,15.554631,5.989974)
V22=(6.859748,11.204618,13.026150,11.425008,15.989975,6.425008)
 la valeur max=0.435344
 la valeur min=0.431541
R(21)=(000122)
la politique optimal estR(21)=(0 0 0 1 2 2)
**********le cout mouyen aptimal egal: 0.433443

Process returned 10 (0xA)   execution time : 65.349 s
Press any key to continue.
```

Figure 2.1.2 : L'affichage de la politique optimale et le cout moyen optimal avec $\varepsilon = 10^{-2}$

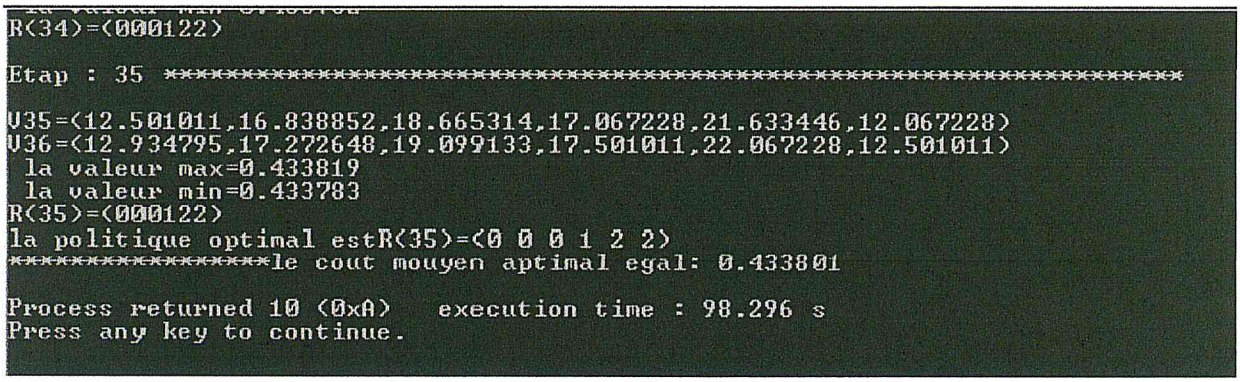


Figure 2.1.3 : L'affichage de la politique optimale et le cout moyen optimal avec $\varepsilon = 10^{-4}$

2.2 *Le problème de gestion d'une centrale électrique*

Considérons une centrale électrique a deux générateurs $j = 1, 2$ pour produire de l'électricité. La quantité d'électricité nécessaire pendant la journée est variable, les 24 heures d'une journée sont subdivisées en six périodes consécutives de 4 heures chacune. La quantité d'électricité requise au cours de la période k est d_k kWh pour $k = 1, ..., 6$. D'autre part, le générateur j a une capacité de générer de C_j kWh par période pour $j = 1, 2$. Un excès d'électricité produit pendant une période ne peut pas être utilisé pour la prochaine période, au début de chaque période k il faut déterminer les générateurs à utiliser pour cette période (un seul générateur, lequel ? ou les deux générateurs ?).

Les coûts imposés sont les suivants:

- Un coût d'exploitation r_j du générateur j pour chaque période d'utilisation.

- Un coût d'installation S_j est engendré à chaque fois que le générateur j est activé (mis en marche) après avoir été inactif pendant un certain temps.

Nous souhaitons déterminer une règle de contrôle qui minimise le coût moyen à

long terme par jour.

Modélisation

Ce problème peut être mis dans le cadre d'un modèle de décision markovien à temps discret.

Dans notre cas, l'état du système peut être décrit par le couple (k, y) telle que:

la première variable k indique la période en cours $(k = 1, 2, ...6)$ et la seconde variable y indique les générateurs opérationnels $(y \in \{1, 2, 3\})$

$y = 1$ veut dire que le premier générateur est opérationnel, $y = 2$ veut dire que le deuxième générateur est opérationnel et $y = 3$ veut dire que les deux générateurs sont opérationnels.

Donc l'espace des états S s'écrit:

$$S = \{(k, y)/k = 1, ..., 6 \text{ et } y = 1, 2, 3\}.$$

Définir les actions:

$$\text{a}=\begin{cases} 1 & \text{si seulement le premiere générateur est utilisé} \\ 2 & \text{si seulement le deuxième générateur est utilisé} \\ 3 & \text{si les deux générateurs sont utilisé} \end{cases}$$

L'ensemble des actions possibles à l'état (k, y) choisi comme

$A\{(k, y)\} = \{y\}$ si $c_y \geq d_k$ pour $y = 1, 2, 3$ et $k = 1, 2, ...6$ (selon la demande de chaque période)

Exemple 2.2.1 [1]

Les demandes d_k pour chaque periode, les capacités c_y pour chaque générateur ainsi que les couts S, r sont données par :

$d_1 = 20,\ d_2 = 40\ ,\ d_3 = 60,\ d_4 = 90,\ d_5 = 70,\ d_6 = 30,$

$c_1 = 40,\ c_2 = 60,\ \ r_1 = 1000\ , r_2 = 1100,\ S_1 = 500,\ S_2 = 300.$

Résolution

L'ensemble des actions possibles est :

$A\{(1, y)\} = \{1, 2, 3\}, A\{(2, y)\} = \{1, 2, 3\}, A\{(3, y)\} = \{, 2, 3\}, A\{(4, y)\} = \{3\},$

$A\{(5,y)\} = \{3\}, A\{(6,y)\} = \{1,2,3\}$.

Les coûts $C_i(j)$ si le générateur i est installé et on veut activé le générateur j est:

$C_1(1) = 1000$, $C_1(2) = 1400$, $C_1(3) = 2400$, $C_2(1) = 1500$, $C_2(2) = 1100$,

$C_2(3) = 2600$, $C_3(1) = 1000$, $C_3(2) = 1100$, $C_3(3) = 2100$.

2.2.1 Algorithme d'itération par politiques (policy-iteration)

L'algorithme est initialisé avec la politique $R^{(1)} = \{(1,1),(2,1),(3,2),(4,3),(5,3),(6,1)\}$.

la politique d'amélioration de la quantité est abrégée en

$T_i(a,R) = C_i(a) - g(R) + \sum_{j\in I} p_{ij}(a)v_j(R)$

Quand la politique actuelle est R. remarque que toujours $T_i(a,R) = v_i(R)$ pour $a = R_i$.

Itération 1

Étape 1 (détermination de la valeur): $v_i(R) = C_i(a) - g(R) + \sum_{j\in I} p_{ij}(a)v_j(R)$

Le coût moyen et la valeur relative de la politique $R^{(1)} = \{(1,1),(2,1),(3,2),(4,3),(5,3),(6,1)\}$ sont calculées en résolvant les équations linéaires $v_i(a,R) = C_i(a) - g(R) + \sum_{j\in I} p_{ij}(a)v_j(R)$

$v_{(1,1)} = 1000 - g + v_{(2,1)}$

$v_{(1,2)} = 1500 - g + v_{(2,1)}$

$v_{(1,3)} = 1000 - g + v_{(2,1)}$

$v_{(2,1)} = 1000 - g + v_{(3,1)}$

$v_{(2,2)} = 1500 - g + v_{(3,1)}$

$v_{(2,3)} = 1000 - g + v_{(3,1)}$

$v_{(3,1)} = 1400 - g + v_{(4,2)}$

$v_{(3,2)} = 1100 - g + v_{(4,2)}$

$v_{(3,3)} = 1100 - g + v_{(4,2)}$

$v_{(4,1)} = 2400 - g + v_{(5,3)}$

$v_{(4,2)} = 2600 - g + v_{(5,3)}$

$v_{(4,3)} = 2100 - g + v_{(5,3)}$

$v_{(5,1)} = 2400 - g + v_{(6,3)}$

$v_{(5,2)} = 2600 - g + v_{(6,3)}$

$v_{(5,3)} = 2100 - g + v_{(6,3)}$

$v_{(6,1)} = 1000 - g + v_{(1,1)}$

$v_{(6,2)} = 1500 - g + v_{(1,1)}$

$v_{(6,3)} = 1000 - g + v_{(1,1)}$

$v_{(1,1)} = 0$ est choisie pour l'équation de normalisation $= 0$.

La solution de ces équations linéaires est donné par :

$g(R^{(1)}) = 1516.67, v_{(1,1)} = 0, v_{(1,2)} = 500,\ v_{(1,3)} = 0, v_{(2,1)} = 516.67,$

$v_{(2,2)} = 1016.67, v_{(2,3)} = 516.67, v_{(3,1)} = 1033.33, v_{(3,2)} = 733.33, v_{(3,3)} = 733.33,$

$v_{(4,1)} = 950, v_{(4,2)} = 1150, v_{(4,3)} = 650, v_{(5,1)} = 366.67, v_{(5,2)} = 566.67,$

$v_{(5,3)} = 66.67, v_{(6,1)} = -516.67, v_{(6,2)} = -16.67, v_{(6,3)} = -516.67.$

Étape 2 (amélioration des politiques)

pour la 1^{er}*periode*

$T_{(1,1)} = \min\{v_{(1;1)}, s_2 + r_2 - g + v_{(2;2)}, s_2 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{0, 900, 1400\} = 0$

$T_{(1,2)} = \min\{v_{(1;2)}, r_2 - g + v_{(2;2)}, s_1 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{500, 600, 1600\} = 500$

$T_{(1,3)} = \min\{v_{(1;3)}, r_2 - g + v_{(2;2)}, r_1 + r_2 - g + v_{(2;3)}\} = \min\{0, 600, 1100\} = 0$

donc le minimun de la 1^{er}periode est 0 (1^{er}générateur)

pour la 2^{eme}periode

$T_{(2,1)} = \min\{v_{(2;1)}, s_2 + r_2 - g + v_{(3;2)}, s_2 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{516.67, 616.67, 1616.67\} = 516.67$

$T_{(2,2)} = \min\{v_{(2;2)}, r_2 - g + v_{(3;2)}, s_1 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{1016.67, 316.67, 1816.67\} = 316.67$

$T_{(2,3)} = \min\{v_{(2;3)}, r_2 - g + v_{(3;2)}, r_1 + r_2 - g + v_{(3;3)}\} = \min\{516.67, 316.67, 1316.67\} = 316.67$

donc le minimun de la 2^{eme}periode est 316.67 (2^{eme}générateur)

pour la 3^{eme}periode on ne peut pas utilise le 1^{er}générateur donc

$T_{(3,1)} = \min\{v_{(3;1)}, s_2 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{1033.33, 1533.33\} = 1033.33$

$T_{(3,2)} = \min\{v_{(3;2)}, s_1 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{733.33, 1733.33\} = 733.33$

$T_{(3,3)} = \min\{v_{(3;3)}, r_1 + r_2 - g + v_{(4;3)}\} = \min\{733.33, 1233.33\} = 733.33$

donc le minimun de la 3^{eme}periode est 733.33 (2^{eme}générateur).

pour la 4^{eme}et la 5^{eme}periode on utilise les deux générateurs forcement.

pour la 6^{eme}periode

$T_{(6,1)} = \min\{v_{(6;1)}, s_2+r_2-g+v_{(1;2)}, s_2+r_1+r_2-g+v_{(1;3)}\} = \min\{-516,67, 383.33, 883.33\} = -516,67$

$T_{(6,2)} = \min\{v_{(6;2)}, r_2-g+v_{(1;2)}, s_1+r_1+r_2-g+v_{(1;3)}\} = \min\{-16.67, 83.33, 1083.33\} = -16.67$

$T_{(6,3)} = \min\{v_{(6;3)}, r_2-g+v_{(1;2)}, r_1+r_2-g+v_{(1;3)}\} = \min\{-516.67, 83.33, 583.33\} = -516.67$

donc le minimun de la 6^{eme}periode est -516.67 (1^{er}générateur)

Étape 3 (convergence): La nouvelle politique $R^{(2)}$ ={(1, 1), (2, 2), (3, 2), (4, 3), (5, 3), (6, 1)}est différente de la précédente politique $R^{(1)}$ et donc une autre itération est effectuée.

Itération 2

Étape 1 (détermination de la valeur):

Le coût moyen et la valeur relative de la politique $R^{(2)} = \{(1, 1), (2, 2), (3, 2), (4, 3), (5, 3), (6, 1)\}$ sont calculées en résolvant les équations linéaires:

$v_{(1,1)} = 1000 - g + v_{(2,1)}$

$v_{(1,2)} = 1500 - g + v_{(2,1)}$

$v_{(1,3)} = 1000 - g + v_{(2,1)}$

$v_{(2,1)} = 1400 - g + v_{(3,2)}$

$v_{(2,2)} = 1100 - g + v_{(3,2)}$

$v_{(2,3)} = 1100 - g + v_{(3,2)}$

$v_{(3,1)} = 1400 - g + v_{(4,2)}$

$v_{(3,2)} = 1100 - g + v_{(4,2)}$

$v_{(3,3)} = 1100 - g + v_{(4,2)}$

$v_{(4,1)} = 2400 - g + v_{(5,3)}$

$v_{(4,2)} = 2600 - g + v_{(5,3)}$

$v_{(4,3)} = 2100 - g + v_{(5,3)}$

$v_{(5,1)} = 2400 - g + v_{(6,3)}$

$v_{(5,2)} = 2600 - g + v_{(6,3)}$

$v_{(5,3)} = 2100 - g + v_{(6,3)}$

$v_{(6,1)} = 1000 - g + v_{(1,1)}$

$v_{(6,2)} = 1500 - g + v_{(1,1)}$

$v_{(6,3)} = 1000 - g + v_{(1,1)}$

$v_{(1,1)} = 0$ est choisie pour l'équation de normalisation$= 0$.

La solution de ces équations linéaires est donné par :

$g(R^{(2)}) = 1533.33, v_{(1,1)} = 0, v_{(1,2)} = 500, v_{(1,3)} = 0, v_{(2,1)} = 533.33,$

$v_{(2,2)} = 233.33, v_{(2,3)} = 233.33, v_{(3,1)} = 966.66, v_{(3,2)} = 666.66$, $v_{(3,3)} = 666.66$,

$v_{(4,1)} = 900, v_{(4,2)} = 1100, v_{(4,3)} = 600, v_{(5,1)} = 333.33$, $v_{(5,2)} = 533.33,$

$v_{(5,3)} = 33.33, v_{(6,1)} = -533.33$, $v_{(6,2)} = -33.33, v_{(6,3)} = -533, 33.$

Étape 2 (amélioration des politiques):

pour la $1^{er} periode$

$T_{(1,1)} = \min\{v_{(1;1)}, s_{2+}r_2 - g + v_{(2;2)}, s_2 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{0, 100, 1100\} = 0$

$T_{(1,2)} = \min\{v_{(1;2)}, r_2 - g + v_{(2;2)}, s_1 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{500, -200, 1300\} = -200$

$T_{(1,3)} = \min\{v_{(1;3)}, r_2 - g + v_{(2;2)}, r_1 + r_2 - g + v_{(2;3)}\} = \min\{0, -200, 800\} = -200$

donc le minimun de la 1^{er}periode est -200 (2^{eme}générateur)

pour la 2^{eme}periode

$T_{(2,1)} = \min\{r_1 - g + v_{(3;1)}, v_{(2;1)}, s_2 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{433.33, 533.33, 1533.33\} = 433.33$

$T_{(2,2)} = \min\{r_1 + s_1 - g + v_{(3;1)}, v_{(2;2)}, s_1 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{933.33, 233.33, 1733.33\} =$

233.33

$T_{(2,3)} = \min\{r_1 - g + v_{(3;1)}, v_{(2;3)}, r_1 + r_2 - g + v_{(3;3)}\} = \min\{433.33, 233.33, 1233.33\} =$ 233.33

donc le minimun de la 2^{eme}periode est 316.67 (2^{eme}générateur)

pour la 3^{eme}periode on ne peut pas utilise le 1^{er}générateur donc:

$T_{(3,1)} = \min\{v_{(3;1)}, s_2 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{966.66, 1466.66\} = 966.66$

$T_{(3,2)} = \min\{v_{(3;2)}, s_1 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{666.66, 1666.66\} = 666.66$

$T_{(3,3)} = \min\{v_{(3;3)}, r_1 + r_2 - g + v_{(4;3)}\} = \min\{666.66, 1166.66\} = 666.66$

donc le minimun de la 3^{eme}periode est 733.33 (2^{eme}générateur)

pour la 4^{eme}et la 5^{eme}periode on utilise les deux générateurs forcement

pour la 6^{eme}periode

$T_{(6,1)} = \min\{v_{(6;1)}, s_{2+}r_2 - g + v_{(1;2)}, s_2 + r_1 + r_2 - g + v_{(1;3)}\} = \min\{-533.33, 366.66, 866.66\} =$ -533.33

$T_{(6,2)} = \min\{v_{(6;2)}, r_2 - g + v_{(1;2)}, s_1 + r_1 + r_2 - g + v_{(1;3)}\} = \min\{-33.33, 66.66, 1066.66\} =$ -33.33

$T_{(6,3)} = \min\{v_{(6;3)}, r_2 - g + v_{(1;2)}, r_1 + r_2 - g + v_{(1;3)}\} = \min\{-533.33, 66.66, 566.66\} =$ -533.33

donc le minimun de la 6^{eme}periode est -533.33 (1^{er}générateur)

Étape 3 (convergence): La nouvelle politique $R^{(3)} = \{(1,2),(2,2),(3,2),(4,3),(5,3),(6,1)\}$est différente de la précédente politique $R^{(2)}$ et donc une autre itération est effectuée.

Itération 3

Étape 1 (détermination de la valeur):

Le coût moyen et la valeur relative de la politique $R^{(3)} = \{(1,2),(2,2),(3,2),(4,3),(5,3),(6,1)\}$ sont calculées en résolvant les équations linéaires:

$v_{(1,1)} = 1000 - g + v_{(2,2)}$

$v_{(1,2)} = 1100 - g + v_{(2,2)}$

$v_{(1,3)} = 1100 - g + v_{(2,2)}$

$v_{(2,1)} = 1400 - g + v_{(3,2)}$

$v_{(2,2)} = 1100 - g + v_{(3,2)}$

$v_{(2,3)} = 1100 - g + v_{(3,2)}$

$v_{(3,1)} = 1400 - g + v_{(4,2)}$

$v_{(3,2)} = 1100 - g + v_{(4,2)}$

$v_{(3,3)} = 1100 - g + v_{(4,2)}$

$v_{(4,1)} = 2400 - g + v_{(5,3)}$

$v_{(4,2)} = 2600 - g + v_{(5,3)}$

$v_{(4,3)} = 2100 - g + v_{(5,3)}$

$v_{(5,1)} = 2400 - g + v_{(6,3)}$

$v_{(5,2)} = 2600 - g + v_{(6,3)}$

$v_{(5,3)} = 2100 - g + v_{(6,3)}$

$v_{(6,1)} = 1000 - g + v_{(1,1)}$

$v_{(6,2)} = 1500 - g + v_{(1,1)}$

$v_{(6,3)} = 1000 - g + v_{(1,1)}$

$v_{(1,1)} = 0$ est choisie pour l'équation de normalisation$= 0$.

La solution de ces équations linéaires est donné par :

$g(R^{(3)}) = 1550, v_{(1,1)} = 0, v_{(1,2)} = -300, v_{(1,3)} = -300, v_{(2,1)} = 450,\ v_{(2,2)} = 150,$

$v_{(2,3)} = 150, v_{(3,1)} = 900, v_{(3,2)} = 600\ , v_{(3,3)} = 600, v_{(4,1)} = 850, v_{(4,2)} = 1050,$

$v_{(4,3)} = 550, v_{(5,1)} = 300, v_{(5,2)} = 500, v_{(5,3)} = 0, v_{(6,1)} = -550, v_{(6,2)} = -50, v_{(6,3)} = -550.$

Étape 2 (amélioration des politiques):

pour la $1^{er} periode$

$T_{(1,1)} = \min\{r_1 - g + v_{(2;1)}, v_{(1;1)}, s_2 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{-100, 0, 1000\} = -100$

$T_{(1,2)} = \min\{r_1 + s_1 - g + v_{(2;1)}, v_{(1;2)}, s_1 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{400, -300, 1200\} = -300$

$T_{(1,3)} = \min\{r_1 - g + v_{(2;1)}, v_{(1;3)}, r_1 + r_2 - g + v_{(2;3)}\} = \min\{-100, -300, 700\} = -300$

donc le minimun de la 1^{er}periode est -300 (2^{eme}générateur)

pour la 2^{eme}periode

$T_{(2,1)} = \min\{r_1 - g + v_{(3;1)}, v_{(2;1)}, s_2 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{350, 450, 1450\} = 350$

$T_{(2,2)} = \min\{r_1 + s_1 - g + v_{(3;1)}, v_{(2;2)}, s_1 + r_1 + r_2 - g + v_{(3;3)}\} = \min\{850, 150, 1650\} = 150$

$T_{(2,3)} = \min\{r_1 - g + v_{(3;1)}, v_{(2;3)}, r_1 + r_2 - g + v_{(3;3)}\} = \min\{350, 150, 1650\} = 150$

donc le minimun de la 2^{eme}periode est 150 (2^{eme}générateur)

pour la 3^{eme}periode on ne peut pas utilise le 1^{er}générateur donc:

$T_{(3,1)} = \min\{v_{(3;1)}, s_2 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{900, 1400\} = 900$

$T_{(3,2)} = \min\{v_{(3;2)}, s_1 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{600, 1600\} = 600$

$T_{(3,3)} = \min\{v_{(3;3)}, r_1 + r_2 - g + v_{(4;3)}\} = \min\{600, 1100\} = 600$

donc le minimun de la 3^{eme}periode est 600 (2^{eme}générateur)

pour la 4^{eme}et la 5^{eme}periode on utilise les deux générateurs forcement

pour la 6^{eme}periode

$T_{(6,1)} = \min\{v_{(6;1)}, s_2 + r_2 - g + v_{(1;2)}, s_2 + r_1 + r_2 - g + v_{(1;3)}\} = \min\{-550, -450, 550\} = -550$

$T_{(6,2)} = \min\{v_{(6;2)}, r_2 - g + v_{(1;2)}, s_1 + r_1 + r_2 - g + v_{(1;3)}\} = \min\{-50, -750, 750\} = -50$

$T_{(6,3)} = \min\{v_{(6;3)}, r_2 - g + v_{(1;2)}, r_1 + r_2 - g + v_{(1;3)}\} = \min\{-550, -750, 250\} = -750$

donc le minimun de la 6^{eme}periode est -533.33 (2^{eme}générateur)

Étape 3 (convergence): La nouvelle politique $R^{(4)} = \{(1,2), (2,2), (3,2), (4,3), (5,3), (6,2)\}$ est différente de la précédente politique $R^{(3)}$ et donc une autre itération est effectuée.

Itération 4

Étape 1 (détermination de la valeur):

Le coût moyen et la valeur relative de la politique $R^{(4)} = \{(1,2), (2,2), (3,2), (4,3), (5,3), (6,2)\}$ sont calculées en résolvant les équations linéaires:

$v_{(1,1)} = 1000 - g + v_{(2,2)}$

$v_{(1,2)} = 1100 - g + v_{(2,2)}$

$v_{(1,3)} = 1100 - g + v_{(2,2)}$

$v_{(2,1)} = 1400 - g + v_{(3,2)}$

$v_{(2,2)} = 1100 - g + v_{(3,2)}$

$v_{(2,3)} = 1100 - g + v_{(3,2)}$

$v_{(3,1)} = 1400 - g + v_{(4,2)}$

$v_{(3,2)} = 1100 - g + v_{(4,2)}$

$v_{(3,3)} = 1100 - g + v_{(4,2)}$

$v_{(4,1)} = 2400 - g + v_{(5,3)}$

$v_{(4,2)} = 2600 - g + v_{(5,3)}$

$v_{(4,3)} = 2100 - g + v_{(5,3)}$

$v_{(5,1)} = 2400 - g + v_{(6,3)}$

$v_{(5,2)} = 2600 - g + v_{(6,3)}$

$v_{(5,3)} = 2100 - g + v_{(6,3)}$

$v_{(6,1)} = 1400 - g + v_{(1,2)}$

$v_{(6,2)} = 1100 - g + v_{(1,2)}$

$v_{(6,3)} = 1100 - g + v_{(1,2)}$

$v_{(1,1)} = 0$ est choisie pour l'équation de normalisation$= 0$.

La solution de ces équations linéaires est donné par :

$g(R^{(4)}) = 1516.66, v_{(1,1)} = 0, v_{(1,2)} = -300, v_{(1,3)} = -300, v_{(2,1)} = 416.66,$

$v_{(2,2)} = 116.66, v_{(2,3)} = 116.66, v_{(3,1)} = 833.33, v_{(3,2)} = 533.33$, $v_{(3,3)} = 533.33$,

$v_{(4,1)} = 750, v_{(4,2)} = 950, v_{(4,3)} = 450, v_{(5,1)} = 166.66$, $v_{(5,2)} = 366.66$,

$v_{(5,3)} = -133.33$, $v_{(6,1)} = -416.66, v_{(6,2)} = -716.66, v_{(6,3)} = -716.66$.

Étape 2 (amélioration des politiques):

pour la 1^{er}*periode*

$T_{(1,1)} = \min\{v_{(1;1)} - g + v_{(2;1)}, v_{(1;1)}, s_2 + r_1 + r_2 - g + v_{(2;3)}\} = \min\{-100, 0, 1000\} =$

-100

$T_{(1,2)} = \min\{r_1+s_1-g+v_{(2;1)}, v_{(1;2)}, s_1+r_1+r_2-g+v_{(2;3)}\} = \min\{400, -300, 1200\} =$
-300

$T_{(1,3)} = \min\{v_{(1;1)} - g + v_{(2;1)}, v_{(1;3)}, r_1 + r_2 - g + v_{(2;3)}\} = \min\{-100, -300, 700\} =$
-300

pour la 2^{eme}periode

$T_{(2,1)} = \min\{r_1-g+v_{(3;1)}, v_{(2;1)}, s_2+r_1+r_2-g+v_{(3;3)}\} = \min\{316.66, 416.66, 1416.66\} =$
316.66

$T_{(2,2)} = \min\{r_1+s_1-g+v_{(3;1)}, v_{(2;2)}, s_1+r_1+r_2-g+v_{(3;3)}\} = \min\{816.66, 116.66, 1616.66\} =$
116.66

$T_{(2,3)} = \min\{r_1-g+v_{(3;1)}, v_{(2;3)}, r_1+r_2-g+v_{(3;3)}\} = \min\{316.66, 116.66, 1116.66\} =$
116.66

pour la 3^{eme}periode on ne peut pas utilise le 1^{er}générateur donc:

$T_{(3,1)} = \min\{v_{(3;1)}, s_2 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{833.33, 1333.33\} = 833.33$

$T_{(3,2)} = \min\{v_{(3;2)}, s_1 + r_1 + r_2 - g + v_{(4;3)}\} = \min\{533.33, 1533.33\} = 533.33$

$T_{(3,3)} = \min\{v_{(3;3)}, r_1 + r_2 - g + v_{(4;3)}\} = \min\{533.33, 1033.33\} = 533.33$

pour la 4^{eme}et la 5^{eme}periode on utilise les deux générateurs forcement

pour la 6^{eme}periode

$T_{(6,1)} = \min\{r_1-g+v_{(1;1)}, v_{(6;1)}, s_2+r_1+r_2-g+v_{(1;3)}\} = \min\{-516.66, -416.66, 583.33\} =$
-516.66

$T_{(6,2)} = \min\{r_1+s_1-g+v_{(1;1)}, v_{(6;2)}, s_1+r_1+r_2-g+v_{(1;3)}\} = \min\{-16.66, -716.66, 783.33\} =$
-716.66

$T_{(6,3)} = \min\{r_1-g+v_{(1;1)}, v_{(6;3)}, r_1+r_2-g+v_{(1;3)}\} = \min\{-516.66, -716.66, 283.33\} =$
-716.66

Étape 3 (convergence): La nouvelle politique $R^{(5)} = \{(1,2),(2,2),(3,2),(4,3),(5,3),(6,2)\}$ est identique à la précédente politique $R^{(4)}$ et donc c'est la politique optimale .

Conclusion :

L'algorithme d'itération par politique converge après 4 itérations avec la politique stationnaire $R^{(4)} = \{(1,2),(2,2),(3,2),(4,3),(5,3),(6,2)\}$ qui impose l'utilisation du générateur 2 dans les périodes 1, 2, 3 , 6 et l'ensemble des deux générateurs dans les périodes 4 et 5, avec un coût moyen minimal $g^* = g(R^{(4)}) = 1516.66$ par jour.

Voici l'affichage de la politique optimale et le coût moyen optimale avec l'algorithme d'itération par politiques pour les données de l'exemple 2.2.1

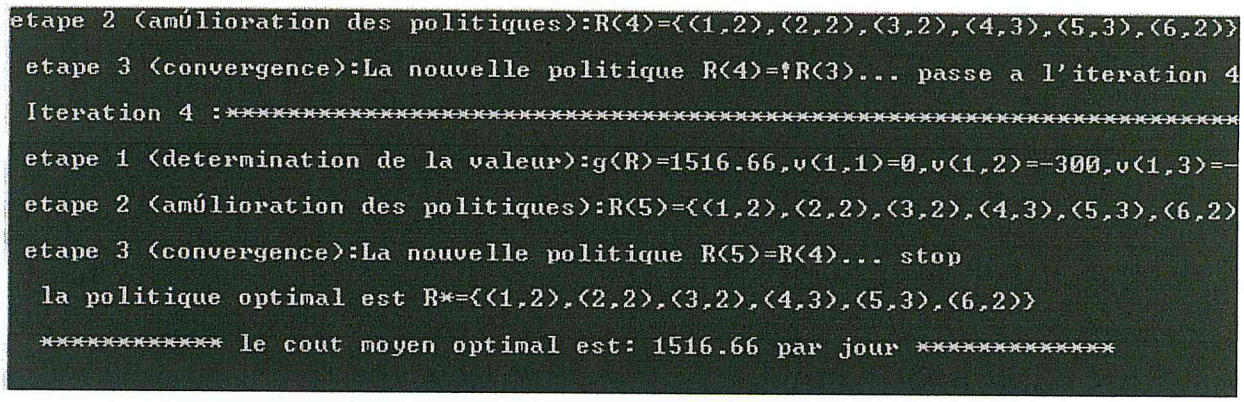


Figure 2.2.1 : L'affichage de la politique optimale et le cout moyen optimal

Conclusion

Le modèle de décision markovienne semble être la meilleure façon de résoudre un large éventail de problème d'optimisation. Parmi les méthodes utilisées pour la recherche de politique optimale, deux algorithmes : "Policy-itération algorithm " et "value-itération algorithm" nous intéressant particulièrement, le premier est robuste, converge après un nombre fini d'itérations. On peut presque dire que le coût moyen généré par cet algorithme converge vers l'optimum au moins avec une vitesse exponentiel, l'inconvénient est qu'il nécessite à chaque étape la résolution d'un système linéaire de même dimension que l'espace des états, "value-itération algorithm" n'est pas aussi robuste que le premier ,ne converge pas aussi rapidement ,mais il est plus simple à appliquer. Il permet également de contourner la difficulté citée précédemment.

La découverte de ces deux algorithmes (technique) s'est réalisé à travers deux applications: le problème de maintenance et le problème d'électricité. Le premier problème est entièrement résolu par TIJMS [1], l'auteur a proposé un modèle de "processus de décision markovien", sur la base de ce dernier les deux algorithmes ont été adaptés. La lecture de ce problème de maintenance, nous a permis de comprendre le mécanisme de chaque méthode (algorithme) à travers les deux programmes que nous avons réalisés et enfin de constater l'avantage et l'inconvénient de chacune d'elle.

Notre second objectif était de résoudre le problème d'électricité. Ce dernier est totalement différent du premier. Introduire les " processus de décision markovienne " nécessite entre autre définition de l'état du système.

Dans notre cas, l'état du système est défini par le couple de variables (k, y), la 1^{ere} variable indique la période et la seconde indique lequel des générateurs est opérationnel, ceci à considérablement compliqué la définition des probabilités de transition

d'un état à un autre. D'autres problèmes se sont posés lors de l'adaptation de " Policy-itération algorithm " à notre cas, plus précisément l'étape d'amélioration de la politique.

Heureusement pour nous, l'objectif de recherche de politique optimale de gestion de la centrale électrique a été atteint en un nombre fini d'itérations.

Enfin, nous pensons par ce travail contribuer à la compréhension des processus de décision ainsi qu'a montrer leur originalité pour aborder les problèmes de contrôle.

Réferences

[1] Henk C. Tijms, A First Course in Stochastic Models, John Wiley & Sons, (2003)

[2] Bruno baynat, Théorie des files d'attente:des chaînes de Markov aux réseaux à forme produit, hermes Science Publications, (1970)

[3] Dahmane zineb, Méthodes Séquentielles et Processus de Décision semi-markovien en Reconnaissance de Formes, USDB, (1999)

[4] Lemdani Rachid, Résolution des problemes d'ordonnancement stochastique par les processus bandits, USDB, (2008)

Les annexes

Annexe.1 Processus de Markov à temps discret

Processus stochastique

Un processus stochastique est une famille de variables aléatoires X(t) où t est un paramètre réel prenant ses valeurs dans ensemble T. En général, T est dénombrable et représente le temps, le processus est alors dit discret.

Chaîne de Markov

Si une suite stochastique vérifie la propriété de Markov .

Propriété de Markov

un processus stochastique vérifie la propriété de Markov si et seulement si la distribution conditionnelle de probabilité des états futurs, étant donné les états passés et l'état présent, ne dépend en fait que de l'état présent et non pas des états passés (absence de « mémoire »).

$P(x_{n+1} = j | x_0 = i_0, x_1 = i_1, ...x_{n-1} = i_{n-1}, x_n = i) = P(x_{n+1} = j | x_n = i)$

Chaîne de Markov stationnaire

une chaîne de Markov est dite stationnaire si la probabilité de transition entre les états est indépendante du temps, plus formellement si pour tout t et k :

$P(x_t = i | x_{t-1} = j) = P(x_{t+k} = i | x_{t+k-1} = j)$.

Processus de Markov à temps discret

En mathématiques, une chaîne de Markov est un processus de Markov à temps discret, ou à temps discret et à espace d'états discret. Un processus de Markov est un processus stochastique possédant la propriété de Markov : l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et n'est pas dépendante des états antérieurs (le système n'a pas de « mémoire »). Les processus de Markov portent le nom de leur inventeur, Andreï Markov.

Un processus de Markov à temps discret est une séquence X_0, X_1, X_2, X_3.....de

variables aléatoires à valeurs dans l'espace des états, qu'on notera E dans la suite. La valeur $X_n$est l'état du processus à l'instant n Les applications où l'espace d'états E est fini ou dénombrable sont innombrables : on parle alors de chaîne de Markov ou de chaînes de Markov à espace d'états discret. Les propriétés essentielles des processus de Markov généraux, par exemple les propriétés de récurrence et d'ergodicité, s'énoncent ou se démontrent plus simplement dans le cas des chaînes de Markov à espace d'états discret. Cet article concerne précisément les chaînes de Markov à espace d'états discret.

Récurrence et transience d'une chaîne de Markov

Un état i d'une chaine de markov $x = (x_n)_{n\geq 0}$ est dit récurrent si une trajectoire « typique » de la chaîne de Markov passe par i une infinité de fois, sinon l'état i est dit transient. Ces propriétés de transience ou de récurrence sont souvent partagées par tous les états de la chaîne X par exemple quand la chaîne X est irréductible : en ce cas c'est la chaîne de Markov qui est dite transitoire ou récurrente.

Une chaîne de Markov est irréductible si et seulement si son graphe est fortement connexe, i.e. si pour tout couple $i \neq j$ de sommets du graphe il existe un chemin de i à j et un chemin de j à i.

unichain

Un unichain est une chaîne de Markov à l'état fini qui contient un seul récurrent classe plus, peut-être, certains états transitoires. En unichain ergodique est un unichain pour lequel le la classe récurrente est ergodique, un unichain, comme nous le verrons, est la généralisation naturelle d'une chaîne récurrente pour permettre certains comportements transitoires initiaux sans perturber le comportement asymptotique à long terme de la chaîne récurrente .

la théorie du renouvellement

Un processus de renouvellement a pour fonction de dénombrer les occurrences d'un phénomène donné, lorsque les délais entre deux occurrences consécutives sont

des variables aléatoires indépendantes et identiquement distribuées. Il peut s'agir de compter le nombre de pannes d'un matériel électronique en théorie de la fiabilité (le matériel est alors renouvelé après chaque panne, d'où la dénomination), de dénombrer les arrivées de clients dans une file d'attente, de recenser les occurrences d'un sinistre pour une compagnie d'assurances...

Annexe.2 Programmation dynamique

la programmation dynamique est une méthode algorithmique pour résoudre des problèmes d'optimisation. Le concept a été introduit au début des années 1950 par Richard Bellman. À l'époque, le terme « programmation » signifie planification et ordonnancement. La programmation dynamique consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires.

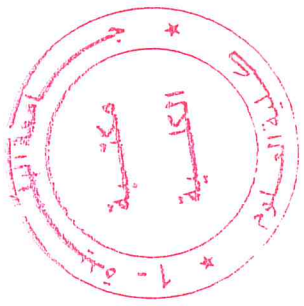
La programmation dynamique s'appuie sur un principe simple, appelé le principe d'optimalité de Bellman: toute solution optimale s'appuie elle-même sur des sous-problèmes résolus localement de façon optimale. Concrètement, cela signifie que l'on peut déduire une ou la solution optimale d'un problème en combinant des solutions optimales d'une série de sous-problèmes. Les solutions des problèmes sont calculées de manière ascendante, c'est-à-dire qu'on débute par les solutions des sous-problèmes les plus petits pour ensuite déduire progressivement les solutions de l'ensemble.

Annexe.3 Programmation

Vous trouvez ci-dessous le programme élaboré en utilisant le langage C pour Le problème de maintenance

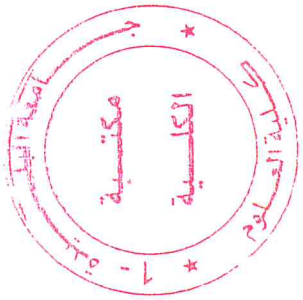
ALGORITHME D'ITERATION DE LA POLITIQUE

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
```

```
int i ,N ,j,n=1;
float X,cou;
float e=0.001,min=0,max=0;
float som=0 ,som1=0;
float test=0;
printf ("donnez le nombre de conditions de travail N: ") ;
scanf ("%d",&N);
float C[N];
int R[N];
float V0[N],V1[N];
float Mat[N-1][N];
//la politique intiale R(1)
for (i=0;i<N-1; i++)
{
R[i]=0;
}
R[N-1]=2;
R[N]=2;
printf("la Politique intiale \n");
printf("R(%d)=(",n);
for(i=0;i<=N;i++)
{
printf("%d",R[i]);
}
printf(")");
printf("\n");
//les Couts de maintence
```

```
for (i=1 ; i<N; i++)
{
if(i==N-1)
{
printf("donnez Cf%d: ",i+1);
scanf ("%f",&C[i]);
}
else
{
printf("donnez Cp%d: ",i+1);
scanf ("%f",&C[i]) ;
}
}
C[0]=0;
C[N]=0;
for(i=0;i<=N;i++)
{
printf("%f ",C[i]);
}
printf("\n");
// les probabilité de transition
printf("\n donnez le tableau de transition \n");
for(i=0;i<N-1;i++)
{
for(j=0;j<N;j++)
{
printf ("donnez l'etat %d%d : ",i+1,j+1) ;
```



```
scanf ("%f", &Mat[i][j]) ;
}
}
for(i=0;i<N-1;i++)
{
for(j=0;j<N;j++)
{
printf ("%f ",Mat[i][j]) ;
}
printf("\n");
}
//les etaps V
printf("\n");
for (i=0 ; i<=N; i++)
{
V0[i]=0;
}
for(i=0;i<=N;i++)
{
printf("%f ",V0[i]);
}
//remplir tableu V1
while(max-min>min*e||test==0)
//
{
printf("\n iteration : %d **********************************************************
for(i=0;i<N;i++)
```

```
{
if(i==0)
{
for(j=i;j<N;j++)
{
som1=som1+(Mat[i][j]*V0[j]);
}
V1[i]=som1;
}
if(i==N-1)
{
V1[N-1]=C[4]+V0[N];
V1[N]=V0[0];
}
if(i>0&&i<N-1)
{
for(j=i;j<N;j++)
{
som=som+(Mat[i][j]*V0[j]);
}
X=C[i]+V0[0];
if(X>=som)
V1[i]=som;
else{V1[i]=X; R[i]=1;}
}
som1=0;
som=0;
```

```
}
test=1;
printf("\n");
printf("V%d=(",n);
for(i=0;i<=N;i++){
if(i<N)
printf("%f,",V0[i]);
else printf("%f",V0[i]);
}
printf(")");
printf("\n");
printf("V%d=(",n+1);
for(i=0;i<=N;i++){
if(i<N)
printf("%f,",V1[i]);
else printf("%f",V1[i]);
}
printf(")");
min=V1[0]-V0[0];
max=V1[0]-V0[0];
for(i=1;i<=N;i++)
{
if(min>=V1[i]-V0[i])
min=V1[i]-V0[i];
if(max<=V1[i]-V0[i])
max=V1[i]-V0[i];
}
```

```
printf("\n la valeur max=%f",max);
printf("\n la valeur min=%f",min);
n++;
if(test==1)
{
for(i=0;i<=N;i++)
{
V0[i]=V1[i];
V1[i]=0;
}
}
printf("\n");
printf("R(%d)=(",n-1);
for(i=0;i<=N;i++)
{
printf("%d",R[i]);
}
printf(")");
printf("\n");
}
printf("la politique optimal est ");
printf("R(%d)",n-1);
printf("\n");
cou=(max+min)/2;
printf("*****************");
printf("le cout mouyen aptimal egal: %f",cou);
printf("\n"); }
```