

**RAIRO Operations Research**

RAIRO Oper. Res. **38** (2004) 215–225

DOI: 10.1051/ro:2004024

**UTILISATION DE LA PROGRAMMATION DYNAMIQUE  
DANS LA MODÉLISATION DE LA PÊCHERIE  
DE LA SARDINE AU MAROC**

MOHAMMED ABBAD<sup>1</sup>, GHALI M. ABDALLAOUI<sup>2</sup> ET  
GHIZLANE BENOMAR<sup>2</sup>

Communiqué par Franco Giannessi

**Abstract.** In this paper we use the dynamic programming approach to model the Moroccan sardine fishery system. We show how the elements of the dynamic programming method such as steps, states and actions are used. The proposed model determines the harvest quantity of sardine in each fishing season in order to maximize the total yield over some definite periods. Some numerical tests of the model, in the deterministic case, are presented and their results show that the approach used is promising. Besides determining the optimal harvest policy, our model can address many important issues such as the impacts of introducing new fishery technologies and finding the best dates and periods of the biological break. This can be done by parametric analysis of the appropriate data of the model. Finally, some data are revealed to be crucial to the model validity and require special attention

---

Received Juillet 12, 2001.

<sup>1</sup> Université Mohammed V, Faculté des Sciences, Département de Mathématiques et Informatique BP 1014 Rabat, Maroc ; e-mail : [abbad@fsr.ac.ma](mailto:abbad@fsr.ac.ma)

<sup>2</sup> Institut National de Statistique et d'Économie Appliquée, Rabat, Maroc ; e-mail : [abdallaoui@insea.ac.ma](mailto:abdallaoui@insea.ac.ma)

© EDP Sciences 2004

**Résumé.** Dans cet article on modélise le système de la pêche de la sardine au Maroc par la méthode de la programmation dynamique. On montre comment les éléments de la programmation dynamique tels que les étapes, états et actions sont utilisés. Le modèle proposé calcule la quantité de sardine à pêcher durant chaque saison de pêche dans le but de maximiser la récolte totale sur un certain nombre de périodes. Des tests, dans le cas déterministe, sont présentés et leurs résultats montrent que l'approche proposée est prometteuse. Notre modèle peut répondre à des problèmes intéressants tels que l'impact d'introduction de nouvelles technologies de pêche, et la détermination des meilleures périodes et dates du repos biologique. Ceci peut être fait par une analyse paramétrique des données appropriées du modèle. Enfin, on mentionne des données sensibles à la validité du modèle et qui nécessitent un traitement spécial.

**Mots Clés.** Pêcherie, sardine, modélisation, programmation dynamique.

## 1. INTRODUCTION

La pêche représente un secteur de grande importance pour l'économie marocaine. Notre patrimoine halieutique est riche en ressources naturelles qui sont de plus en plus exploitées. Parmi les principales raisons de la surexploitation que connaissent les pêcheries dans les mers marocaines on peut citer :

- l'absence de plans d'aménagement et de gestion des pêcheries pour réglementer cette dernière ;
- l'évolution des techniques de pêche qui permettent d'augmenter l'effort de pêche ;
- la forte pression de la part des flottes étrangères utilisant des bateaux ainsi que des techniques de pêche très performantes, et qui en plus échappent en général au contrôle de débarquement.

D'où l'intérêt de créer des outils de planification permettant d'évaluer les conséquences des actions et mesures relatives au secteur des pêches maritimes.

Un de ces outils, est la Programmation Dynamique [2]. La méthode de la programmation dynamique consiste à décomposer le système à étudier en périodes consécutives (par exemple saisons de pêche). Au début de chaque période, et selon l'état du système (la biomasse), on décide de l'action à entreprendre (la quantité de biomasse à pêcher). Le choix d'une action doit prendre en compte deux facteurs importants : le gain immédiat qui en résultera (valeur de la quantité pêchée), et le nouvel état du système auquel il aboutirait au début de la période suivante. (Ce nouvel état dépend de la quantité pêchée, de l'environnement, de l'aspect des proies et prédateurs, et des stades biologiques.) Le processus continuera de la même manière jusqu'à un horizon infini ou fini choisi par l'utilisateur.

Le problème est de trouver la meilleure suite d'actions qui optimise une certaine fonction objectif. Dans la littérature, il existe de bons algorithmes pour calculer une telle suite optimale d'actions [4].

Ces techniques ont été utilisées dans plusieurs pays tels que le Canada, les États-Unis, la Norvège, etc. (voir [5, 6]), et ont donné des résultats satisfaisants. Dans le cadre d'un projet de fin d'étude à l'INSEA (Institut National de Statistique et d'Économie Appliquée) et en collaboration avec l'INRH (Institut National de Recherche Halieutique) [1], nous avons utilisé la programmation dynamique dans le cas de la pêche de la sardine. Les résultats obtenus semblent prometteurs.

## 2. PROGRAMMATION DYNAMIQUE

On peut utiliser la programmation dynamique [2] dans le cas où un problème d'optimisation se décompose en  $T$  étapes telles qu'au début de chaque étape  $i = 1, 2, \dots, T$ , un preneur de décisions observe l'état du processus  $s_i \in [a_i, b_i]$  et choisit une décision (ou action)  $u_i \in U_i$ . Ensuite deux choses vont se produire : un gain immédiat  $r_i(s_i, u_i)$  en résultera, et le processus passera à un nouvel état  $s'_{i+1}$  avec une loi de probabilité  $p_i(\cdot | s_i, u_i)$ . Ainsi, le processus se trouvera dans l'état  $s'_{i+1}$  au début de l'étape  $i + 1$ , et le preneur de décision choisit une nouvelle action, et ainsi de suite le processus continuera de cette manière jusqu'à la dernière étape  $T$ .

Le problème consiste à déterminer une suite de décisions qui donne le maximum des gains immédiats accumulés plus une valeur de l'état final. La méthode de calcul d'une telle suite de décision optimale se fait par la résolution de l'équation d'optimalité :

$$V_i(s_i) = \max_{u_i \in U_i} \left\{ r_i(s_i, u_i) + \beta \int_{a_{i+1}}^{b_{i+1}} V_{i+1}(s_{i+1}) p_i(ds_{i+1} | s_i, u_i) \right\}, \quad (1)$$

où  $V_{T+1}(s_{T+1})$  sont des valeurs choisies pour les états finaux possibles, et  $\beta \in [0, 1)$  étant le facteur d'actualisation.

La résolution de l'équation d'optimalité se fait d'une manière régressive : à chaque étape  $i = T, T - 1, \dots, 1$  et pour chaque état  $s_i$  on calcule une décision optimale  $u_i^*(s_i)$  et la valeur optimale  $V_i(s_i)$ . La valeur optimale du problème original sera alors  $V_1(s_1)$  si l'état initial du processus est  $s_1$ .

## 3. MODÉLISATION DE LA PÊCHERIE DE LA SARDINE

D'après la section précédente, pour formuler un problème par la programmation dynamique il faut alors définir les éléments suivants : étapes, décisions, gains immédiats, valeurs des états finaux possibles et les lois de probabilités des états postérieurs.

Dans cette section, nous allons définir chaque élément dans le but de formuler la pêche de la sardine par la programmation dynamique.

Le choix de la sardine a été motivé par son importance dans l'économie nationale (75 % du total des captures côtières) et de la disponibilité des données.

Dans cette modélisation, une étape  $i$  représente le début de l'année  $i$ .

Pour définir les états, on a considéré la structure de la répartition de la sardine par âge :

Age en années	1	2	3	4	5	6	7
Pourcentage	53	37	6	1	1	1	1

La durée de vie de la sardine est de 7 ans. L'âge de maturité sexuelle de la sardine est de 2 ans. D'après le tableau ci-dessus, on constate que la population de sardine âgée d'une année (juvéniles) représente plus que la moitié de la biomasse totale. La population de sardine âgée de 2 ans est de 37 %, tandis que celle âgée de plus de 2 ans ne représente que 10 %. Cette observation sur la structure de la répartition de la sardine par âge nous a conduit à définir un état  $s_i$  par un couple  $(x_i, y_i)$  où  $x_i$  est la biomasse des juvéniles (âge 1) et  $y_i$  est la biomasse des adultes (âge 2 et plus).

Une décision  $u_i$  au début de l'année  $i$  représente le pourcentage de l'utilisation de la capacité de pêche disponible. Dans ce cas  $U_i = [0, 1]$ .

Le gain immédiat est alors formulé par :

$$r_i((x_i, y_i), u_i) = \lambda_i h(y_i, u_i), \quad (2)$$

où  $\lambda_i$  est le prix du poisson au début de l'année  $i$  et la fonction  $h(y_i, u_i)$  représente la quantité pêchée. Comme dans plusieurs références (par exemple [5]), nous retenons la fonction de Cobb-Douglas (on pourrait utiliser une fonction plus appropriée si elle était disponible) :

$$h(y_i, u_i) = qy_i u_i, \quad (3)$$

où  $q$  est le taux de capture. D'après les données disponibles, nous avons estimé  $q$  à 12,68 %.

Quant à l'état postérieur  $(x_{i+1}, y_{i+1})$ , la biomasse des juvéniles  $x_{i+1}$  au début de l'année  $i + 1$  va être une fonction de la biomasse des adultes  $y_i$  de l'année  $i$ . Une étude élaborée par le comité des pêches pour l'atlantique Centre-Est [3] a permis de constater que la fonction de recrutement qui calcule la biomasse des juvéniles en fonction des adultes est donnée par la formule de Chapman :

$$x_{i+1} =: R(y_i) = Ay_i^{\frac{3}{2}} - By_i^2, \quad (4)$$

où  $A = 3,753 \times 10^{-3}$  et  $B = 0,00194 \times 10^{-3}$ .

Puisque la fonction de recrutement dépend en général de plusieurs facteurs aléatoires tels que l'environnement, la température, et d'autres facteurs biologiques, on a ajouté une perturbation de loi normale :

$$x_{i+1} =: R(y_i) = Ay_i^{\frac{3}{2}} - By_i^2 + G(y_i), \quad (5)$$

où  $G(y_i)$  est une variable aléatoire qui suit une loi normale  $N(0, \sigma^2(y_i))$  de moyenne 0 et de variance  $\sigma^2(y_i)$ .

La biomasse des adultes  $y_{i+1}$  au début de l'année  $i + 1$  se calcule d'une manière naturelle par :

$$y_{i+1} = x_i + y_i - M_1(x_i, u_i) - M_2(y_i, u_i) - h(y_i, u_i), \tag{6}$$

où  $M_1(x_i, u_i)$  ( $M_2(x_i, u_i)$ ) est la biomasse des juvéniles (adultes) qui meurent de cause naturelle ou de pêche.

Notons que  $x_i - M_1(x_i, u_i)$  est la biomasse des survivants des juvéniles. Dans plusieurs travaux sur les pêcheries, on trouve la formule :

$$x_i - M_1(x_i, u_i) = x_i \exp^{-N_1 - P_1(u_i)} \tag{7}$$

où  $N_1$  est la mortalité naturelle et  $P_1(u_i)$  est la mortalité par pêche.

De même pour les adultes, on a :

$$y_i - M_2(y_i, u_i) = y_i \exp^{-N_2 - P_2(u_i)} \tag{8}$$

où  $N_2$  est la mortalité naturelle et  $P_2(u_i)$  est la mortalité par pêche pour les adultes.

Après discussion avec des chercheurs dans le domaine de la pêche de la sardine, on a utilisé  $N_1 = N_2 = 0,6$  et on a choisi  $P_1(u_i) = F_1 u_i$  et  $P_2(u_i) = F_2 u_i$  avec  $F_1 = F_2 = 0,4$ .

#### 4. APPROCHE DE CALCUL

D'après la section précédente, l'équation d'optimalité à résoudre est :

$$V_i(x_i, y_i) = \max_{u_i \in [0,1]} \left\{ \lambda_i h(y_i, u_i) + \beta \int V_{i+1}(x_{i+1}, y_{i+1}) p_i(d(x_{i+1}, y_{i+1}) | (x_i, y_i), u_i) \right\} \tag{9}$$

où  $\{V_{T+1}(x_{T+1}, y_{T+1})\}$  sont des valeurs données, et

$$x_{i+1} = Ay_i^{\frac{3}{2}} - By_i^2 + G(y_i) := R(y_i) \tag{10}$$

$$y_{i+1} = x_i \exp^{-N_1 - P_1(u_i)} + y_i \exp^{-N_2 - P_2(u_i)} - h(y_i, u_i) := B(x_i, y_i, u_i). \tag{11}$$

Dans ce qui suit, on présente la procédure de résolution de l'équation d'optimalité ci-dessus. On commence par  $i = T$ , on fixe  $(x_T, y_T)$ , puis pour chaque  $u_T$ , on calcule la fonction de répartition de  $(x_{T+1}, y_{T+1})$  à l'aide de (5). Puisque les valeurs  $V_{T+1}(x_{T+1}, y_{T+1})$  sont connues, on calcule à l'aide de (9), une décision optimale  $u_T^*(x_T, y_T)$  et la valeur optimale  $V_T(x_T, y_T)$ . Ensuite, on passe à  $i = T - 1$ .

Puisque les valeurs  $\{V_T(x_T, y_T)\}$  sont déjà calculées dans l'étape  $i = T$ , on répète une procédure similaire à celle présentée dans l'étape précédente pour calculer  $u_{T-1}^*(x_{T-1}, y_{T-1})$  et  $V_{T-1}(x_{T-1}, y_{T-1})$ . Ainsi de suite, on poursuit les calculs jusqu'à  $i = 1$ .

Puisque les états  $(x_i, y_i)$  sont continus, la résolution de l'équation d'optimalité présente des difficultés. Un problème de recherche intéressant est d'étudier l'existence et l'unicité de l'équation d'optimalité et d'en déterminer une solution analytique. Néanmoins, on peut résoudre l'équation d'optimalité en passant à la discrétisation des états  $(x_i, y_i)$ . Ainsi, on peut fournir une solution approchée du problème original.

Sur la base de la biomasse actuelle, on a utilisé les états discrets  $(j, j')$  définis par :

- $j, j' = 1$  correspond à une biomasse en tonnes dans  $[0, 500\,000[$ ;
- $j, j' = 2$  correspond à une biomasse en tonnes dans  $[500\,000, 1\,000\,000[$ ;
- $j, j' = 3$  correspond à une biomasse en tonnes dans  $[1\,000\,000, 1\,500\,000[$ ;
- $j, j' = 4$  correspond à une biomasse en tonnes dans  $[1\,500\,000, 2\,000\,000[$ ;
- $j, j' = 5$  correspond à une biomasse en tonnes dans  $[2\,000\,000, 2\,500\,000[$ ;
- $j, j' = 6$  correspond à une biomasse en tonnes dans  $[2\,500\,000, 3\,000\,000[$ ;
- $j, j' = 7$  correspond à une biomasse en tonnes dans  $[3\,000\,000, 3\,500\,000[$ .

Par exemple l'état  $(4, 3)$  signifie que la biomasse des juvéniles est dans l'intervalle  $[1\,500\,000, 2\,000\,000[$  et la biomasse des adultes est dans l'intervalle  $[1\,000\,000, 1\,500\,000[$ .

Pour chaque étape  $i$ , si l'état est  $(j, j')$  alors la valeur numérique de  $x_i$  ( $y_i$ ) qui sera utilisée dans les calculs est le milieu de l'intervalle correspondant à  $j$  ( $j'$ ). Par exemple pour une étape donnée  $i$ , si l'état est  $(4, 3)$  alors  $x_i = 1\,750\,000$  et  $y_i = 1\,250\,000$ .

En ce qui concerne les actions, on a utilisé la discrétisation suivante :  $u_i \in \bar{U} := \{0, 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1\}$ .

Pour tout  $j$ , on note par  $m_j$  le milieu de l'intervalle  $I_j$  correspondant à  $j$ ; et pour toute biomasse  $x \in [0, 3\,500\,000[$  on note par  $k(x) \in 1, \dots, 7$  le nombre correspondant à l'intervalle  $I_{k(x)}$  contenant  $x$ .

Dans le cas de la discrétisation ci-dessus, l'équation (9) devient :

$$V_i(j, j') = \max_{u_i \in \bar{U}} \left\{ \lambda_i h(m_{j'}, u_i) + \beta \sum_{l=1}^7 \sum_{l'=1}^7 p_i((l, l') | (j, j'), u_i) V_{i+1}(l, l') \right\} \quad (12)$$

où  $\{V_{T+1}(j, j')\}$  sont des valeurs données, et  $p_i((l, l')|(j, j'), u_i) = \text{Probabilité}\{R(m_j) \in I_l, B(m_j, m_{j'}, u_i) \in I_{l'}\}$ .

Dans le cas déterministe (c'est-à-dire  $G = 0$ ), l'équation (12) devient :

$$V_i(j, j') = \max_{u_i \in \bar{U}} \{\lambda_i h(m_{j'}, u_i) + \beta V_{i+1}(k(R(m_j)), k(B(m_j, m_{j'}, u_i)))\} \quad (13)$$

où  $\{V_{T+1}(j, j')\}$  sont des valeurs données.

Dans le but de tester notre modèle, on a considéré le cas discret déterministe. Dans ce cas, on a commencé par étudier deux scénarios. Le premier scénario consiste à ne rien pêcher qui se traduit par  $u_i = 0$  pour tout  $i$ . Tandis que le deuxième scénario consiste à utiliser toute la capacité de pêche, ce qui se traduit par  $u_i = 1$  pour tout  $i$ .

Avec un stock initial de 2 000 000 tonnes pour les juvéniles et 2 000 000 pour les adultes, on a constaté que pour le premier scénario la biomasse totale se stabilise à 5 000 000 tonnes environ. Alors que le deuxième scénario donne un épuisement de stock après une dizaine d'années.

Bien que les résultats dépendent des données utilisées, nous croyons que la tendance observée de la stabilisation et de l'épuisement du stock pour les deux scénarios relativement aux données que nous avons considérées reste valable avec des données plus fiables.

Il est clair que les deux scénarios ne sont pas rentables. Le premier donne un stock important qui n'est pas exploité et le deuxième conduit vers un épuisement.

Ainsi, on est amené à déterminer une suite de décisions  $u_1^*, u_2^*, \dots, u_T^*$  qui permet une gestion rationnelle de la pêcherie de la sardine. Ceci se fait à l'aide de l'équation de la programmation dynamique.

En annexe, on fournit le programme écrit en langage C qui permet la résolution de l'équation de la programmation dynamique (13). Avec les données dont nous disposons, le programme a donné les résultats ci-dessous dans le cas où on souhaite que la biomasse à la fin de l'horizon considéré soit proche de la biomasse actuelle.

Horizon  $T = 10$  années :

Étape	État	Action
1	(4,4)	40 %
2	(6,5)	100 %
3	(5,3)	40 %
4	(6,4)	90 %
5	(5,3)	40 %
6	(6,4)	90 %
7	(5,3)	40 %
8	(6,4)	90 %
9	(5,3)	100 %
10	(6,3)	100%
11	(5,3)	-

Horizon  $T = 15$  années :

Étape	État	Action
1	(4,4)	40 %
2	(6,5)	100 %
3	(5,3)	40 %
4	(6,4)	90 %
5	(5,3)	40 %
6	(6,4)	90 %
7	(5,3)	40 %
8	(6,4)	90 %
9	(5,3)	40 %
10	(6,4)	90 %
11	(5,3)	40 %
12	(6,4)	90 %
13	(5,3)	100 %
14	(6,3)	70 %
15	(5,3)	100 %
16	(6,3)	-

Dans les tableaux ci-dessus une solution optimale consiste alors à utiliser 40 % de la capacité de pêche sachant que l'état initial est (4,4), l'état suivant sera alors (6,5) et dans ce cas il faut utiliser toute la capacité de pêche, l'état suivant sera alors (5,3) et l'action à utiliser est 0,4, et ainsi de suite.

## 5. CONCLUSION ET PERSPECTIVES

Dans cet article, on a montré que la programmation dynamique peut être utilisée pour aider à la décision dans la gestion des pêcheries. Nous avons pris comme exemple le cas de la sardine mais le modèle pourra être appliqué à d'autres espèces une fois que les données sont disponibles.

Avec une paramétrisation des données appropriées, le modèle que nous avons développé peut :

- 1) évaluer l'impact de l'introduction de nouvelles technologies de pêche ;
- 2) déterminer les dates d'introduction de nouvelles capacités de pêche et leurs valeurs ;
- 3) déterminer les périodes et évaluer les conséquences des repos biologiques.

Il est important d'insister sur le fait que les résultats de tout modèle dépendent de la qualité des données et paramètres utilisés. Pour que notre modèle donne de meilleurs résultats et par conséquent pour fournir des recommandations au niveau de la gestion des pêcheries, les données suivantes nécessitent des études de



fiabilité particulières : fonction de recrutement, biomasse initiale, taux de captures, mortalité naturelle et par pêche, prix ou valeur du poisson à étudier.

### ANNEXE : PROGRAMME EN LANGAGE C

```

#include<stdio.h> #include<conio.h> #include<math.h>
#include<iostream.h>
int T=10; double A=3.753*pow10(-3); double B=0.00194*pow10(-3);
double n1=0.6,n2=0.6,f1=0.4,f2=0.4,q=0.1268;
double G[8][8],v[11][8][8],a[11][8][8];

int max_tab(double tab[],int n,double *max) { int j=0; (*max)=tab[0];
for(int i=1;i<n;i++)
if(tab[i]>(*max)) {*max=tab[i];j=i;}
return j; }

void saisie_matG() {int i,j;
for(i=0;i<=7;i++)
for(j=0;j<=7;j++)
G[i][j]=0;
G[4][4]=pow10(20);}

void init_matv() {
for(int i=0;i<=T+1;i++)
for(int j=0;j<8;j++)
for(int k=0;k<8;k++)
v[i][j][k]=0;
for(j=0;j<8;j++)
for(k=0;k<8;k++)
v[T+1][j][k]=G[j][k]; }

double
X[8]=0,250000,750000,1250000,1750000,2250000,2750000,3250000;

int valeur_etat(double x) {int etat;
if(x>=0 && x<=500000) etat=1;
else if(x>500000 && x<=1000000) etat=2;
else if(x>1000000 && x<=1500000) etat=3;
else if(x>1500000 && x<=2000000) etat=4;
else if(x>2000000 && x<=2500000) etat=5;
else if(x>2500000 && x<=3000000) etat=6;
else if(x>3000000) etat=7;
return etat; }

```

```

    int p1(int y) {double r; double y1=X[y];
r=(A*pow(y1,1.5))-(B*y1*y1); return valeur_etat(r); }

    int p2(int x,int y,double k) {int i;
double x1=X[x],y1=X[y];
double s=(x1*exp(-n1-(f1*k)))+(y1*exp(-n2-(f2*k)))-q*y1*k;
i=valeur_etat(s);
return i; }

    double K[]={0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1}; void
calcul()
{double t[11];
int k,j;
double max;
for(int i=T;i>=1;i-)
for(int x=1;x<=7;x++)
for(int y=1;y<=7;y++)
{ k=0;j=0;
while(k<=10)
{ int x1=p1(y);
int y1=p2(x,y,K[k]);
t[j]=(q*X[y]*K[k])+v[i+1][x1][y1];j++;
k++; }
int c=max_tab(t,j,&max);
v[i][x][y]=max; a[i][x][y]=K[c]; } }

    void affichage (double a[][8][8])
{for(int i=1;i<=T;i++)
{printf(" \ n ");
printf(" L'ETAPE %i : \ n ",i);
for(int j=1;j<=7;j++)
{for(int k=1;k<=7;k++)
printf(" \ t%.1f ",a[i][j][k]);
printf(" \ n "); }
getch(); }

    void main() {
textbackground(1);
clrscr();

    int i,j,s,r;
double x,y,A[4][11];
for(i=0;i<4;i++)
for(j=0;j<T+1;j++)
A[i][j]=0;

```

```

saisie_matG();
init_matv();
calcul();
affichage(a);
printf(" donnez le couple (x,y) ");
cin>>x;cin>>y;

    r=valeur_etat(x);
s=valeur_etat(y);
for(i=1;i<T+1;i++)
{ A[1][i]=a[i][r][s];
A[0][i]=i;
r=p1(r);
s=p2(r,s,A[1][i]);
A[2][i]=r;
A[3][i]=s; }

    printf(" \ n le chemin optimal pour le couple (%.0f,%.0f) ",x,y);
printf(" \ n \ n "); for(i=1;i<T+1;i++) printf(" %.1f ",A[1][i] );
printf(" (4,4)→ "); for(i=1;i<T+1;i++)
printf(" (%.0f,%.0f)→ ",A[2][i],A[3][i]);
getch(); }

```

*Remerciements.* Nous remercions tous les chercheurs du département d'évaluation des ressources de l'Institut National de Recherche Halieutique (INRH), et plus particulièrement Monsieur A. Idelhaj pour leurs conseils et leurs discussions enrichissantes.

## RÉFÉRENCES

- [1] G. Benomar, *Utilisation de la programmation dynamique dans la modélisation d'une pêcherie : Cas de la sardine au Maroc*. Projet de fin d'études dirigé par M. Abbad et G.M. Abdallaoui, INSEA (1999).
- [2] D. Bertsekas, S.E. Shreve, *Stochastic Optimal Control : The Discrete Time Case*. Academic Press, New York (1978).
- [3] Comité des pêches pour l'atlantique Centre-Est, Rapport des groupes de travail *ad hoc* sur la sardine et sur les chinchards et les aquereaux dans la région nord du COPACE. FAO (1990).
- [4] L.C.M. Kallenberg, *Linear Programming and Finite Markovian Control Problems*. Mathematical Centre Tracts, 148, Amsterdam (1983).
- [5] C.J. Walters, Some Dynamic Programming Applications in Fisheries Management. *Dyn. Program. Appl.* (1979) 233–246.
- [6] D.J. White, Real Applications of Markov Decision Processes. *Interfaces* **15** (1985) 73–83.