

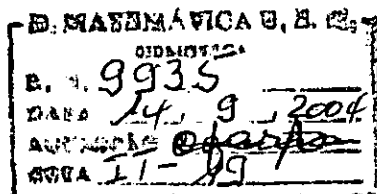
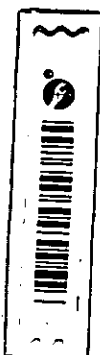
UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

MODELO PARA GESTÃO DE RECURSOS RENOVÁVEIS

Supervisores: **dr. Mário F. Getimane**
 Engº José Grachane

Autor: **Ilda Mónica Come**



IT-19

DEDICATÓRIA

Este trabalho é dedicado aos meus pais, que sempre me incentivaram a estudar.

Ilda Mónica Come

AGRADECIMENTOS

Estes agradecimentos vão para aquelas pessoas que me apoiaram, quando precisei e que grandemente contribuíram para a realização deste trabalho.

Ao **dr. Mário Getimane** e ao **Eng^o José Grachane**, meus supervisores, quero agradecer-vos muito pela dedicação e disposição na supervisão desta tese. A vós o meu muito obrigado.

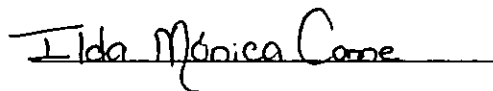
A minha gratidão é estendida ainda ao **dr. Bonifácio José**, aos meus colegas **dr^a Brígida da Cruz**, **dr. Lúcio Artiel** e **dr. Isaiás Nhavane**.

Quero por último agradecer aos funcionários da biblioteca do DMI que directa ou indirectamente contribuíram para a realização deste trabalho. Muito obrigado!

Ilda Mónica Come

DECLARAÇÃO DE HONRA

Declaro por minha honra que este trabalho é resultado das minha própria investigação, e que o mesmo foi realizado para ser submetido como trabalho de **Licenciatura em Informática** na Universidade Eduardo Mondlane.



Ilda Mónica Come

Maputo, Junho de 1999

RESUMO

Existem diversos estudos de aplicação da Programação dinâmica (PD) na gestão de recursos renováveis, onde o objectivo principal é a geração de políticas óptimas de controlo de modo a produzir maiores benefícios ou reduzir custos.

O presente trabalho, trata de um modelo para gestão de recursos renováveis, com um nível populacional crítico (s_{\min}) abaixo do qual nenhuma exploração do recurso deve ser feita, de modo a evitar a sua extinção.

Pretende-se com este trabalho, através da aplicação da técnica de programação dinâmica, descrever a estrutura de solução de um programa dinâmico.

Para a realização do trabalho, considerou-se um modelo de controlo com perturbações aleatórias independentes e identicamente distribuídas, obedecendo a distribuição uniforme, e o espaço de estados e acções conjuntos em IR.

A primeira parte do trabalho ocupa-se da introdução dos conceitos básicos de PD, bem como de alguns resultados matemáticos e de PD sobre as propriedades estruturais dos modelos de PD, onde se enfatiza a monotonia das funções de valor $V_n(s)$ e a continuidade dos maximizadores $f_n(s)$.

Como parte final do trabalho, simula-se o estado final do recurso e verifica-se que a distribuição probabilística do estado final depende do nível crítico s_{\min} .

Do estudo feito, conclui-se que, aplicando o teorema básico de controlo, as funções de valor, ou seja o ganho obtido ao fim de N períodos, são crescentes o que significa que com o nível populacional inicialmente fixo, o ganho cresce com o número de períodos de decisão.

INDICE

1. Introdução	1
1.1 Objectivos	3
2. Material e Métodos	4
3. Resultados	5
3.1 Modelos de PD	5
3.1.1 Definição Básica de um MEDPD	9
3.1.2 Formulação do problema de Optimização de um MEDPD	10
3.1.2.1 Formulação do Problema de optimização de um MEDPD em termos de acções.	10
3.1.2.2 Formulação do Problema de optimização de um MEDPD em termos de políticas	12
3.1.3 Análise das duas formulações	15
3.1.4 Teorema Básico para a maximização dos Ganhos para um MEDPD	16
3.1.5 Procedimento de Solução de Problemas de PD	18
3.1.5.1 O Procedimento <i>Backward</i>	19
3.2 Modelos de PD com Distúrbâncias Independentes e Identicamente Distribuídas	20
3.2.1 Definição Básica	20
3.2.2 Teorema Básico para um MC com Distúrbâncias i.i.d. e espaço de distúrbância finito.	22

3.3 Estudo das Propriedades estruturais de um MEDPD	22
3.3.1 Funções côncavas e convexas	23
3.3.2 Funções de Valor.....	24
3.3.3 Maximizadores.....	26
3.3.3.1 Existência de políticas ótimas e continuidade de funções de valor e dos maximizadores	28
3.4 Modelo para Gestão de Recursos Renováveis com estado crítico (smin)	32
3.4.1 Formulação do Modelo	33
3.4.2 Estudo das Propriedades estruturais do Modelo	39
3.4.3 Estudo da Distribuição do estado final do sistema.....	44
4. Discussão.....	48
5.1 Conclusões	50
5.2 Recomendações.....	50
6.1 Referências Bibliográficas	52
7.1 Anexo A	54
7.2 Anexo B	78
7.3 Anexo C	81

CAPÍTULO I

1. Introdução

A Programação Dinâmica (PD) é um procedimento matemático desenhado essencialmente para aumentar a eficiência dos cálculos em problemas de programação matemática, decompondo-os em subproblemas facilmente computáveis.

Em PD, os cálculos são realizados em etapas, dividindo o problema global em subproblemas. Cada subproblema é tratado separadamente com o objectivo de reduzir o volume de cálculos. Já que os subproblemas são interdependentes, deve-se procurar um procedimento para ligar os cálculos de modo a garantir que a solução para uma etapa, em combinação com as soluções das outras etapas, dê uma solução óptima do problema global.

Especificamente a PD resolve o problema por estágios, sendo cada estágio caracterizado por uma única variável a otimizar. Os cálculos nos diferentes estágios são ligados por cálculo recursivo.

O nome de **Programação Dinâmica**, talvez tenha sido sugerido pela sua utilização em problemas que envolvem tomadas de decisão ao longo do tempo. Além disso, o procedimento é utilizado também em situações não dependentes do tempo.

A principal teoria de base na PD é o princípio da optimização, que na sua essência nos diz como é que um problema devidamente decomposto vai ser resolvido em estágios usando o cálculo recursivo.

Nos problemas de PD, o modelo a ser usado depende essencialmente da formulação desses mesmos problemas (Fiadeiro, 1992).

Ideia básica

A ideia básica de PD é a resolução de problemas de optimização através de uma progressão. Por exemplo, tem-se um problema P , que à primeira vista é difícil resolvê-lo directamente. Procura-se uma maneira de dividir o problema numa série de subproblemas manejáveis que conduzem à solução do problema P . Existem $n+1$ subproblemas na série designados por $P_0, P_1, P_2, \dots, P_n$. O primeiro subproblema (P_0) deve ser o subproblema que se sabe como resolvê-lo. O segundo subproblema (P_1) deve ser o problema que se pode resolver sob condição de ter a solução de P_0 , igualmente, P_2 deve ser o problema que se pode resolver conhecendo as soluções de P_0 e P_1 , etc. Ao se resolver o subproblema final (P_n na série), estar-se-à a solucionar o problema P , isto é, uma vez que se tem a sequência de subproblemas, irão se resolver começando com P_0 , até se atingir o subproblema final P_n e usa-se a solução de P_n combinada com as soluções de P_0, P_1, \dots, P_{n-1} para determinar a solução do problema P . Então a ideia básica de PD resume-se no seguinte:

Para resolver um determinado problema P , resolve-se uma sequência de subproblemas P_0, P_1, \dots, P_n , onde:

1. Inicialmente: sabe-se como resolver P_0 .
2. Progressivamente: sabe-se como resolver P_n dada a solução de P_0, P_1, \dots, P_{n-1} para $n > 0$.

3. Finalmente: dada a solução de P_n , pode-se determinar directamente a solução do problema P.

A representação do problema é feita de forma abstracta ou simbólica, conhecida por modelo matemático. Tendo um modelo matemático apropriado, deve-se escolher uma técnica de optimização para resolver o problema. A habilidade de encontrar um modelo apropriado, é adquirida somente por experiência. Alguns autores, usam o termo “Arte de Programação Dinâmica”, isto é, a arte de encontrar um modelo apropriado.

1.1 Objectivos

1.1.1 Objectivos Gerais

O trabalho tem como objectivo principal descrever a aplicabilidade da técnica de programação dinâmica na gestão de recursos renováveis, focando o interesse na exploração óptima de populações biológicas, por exemplo recursos pesqueiros.

1.1.2 Objectivos Específicos

- Descrever a estrutura de solução de um programa dinâmico;
- Fazer o estudo das propriedades estruturais dos modelos de programação dinâmica, concretamente o estudo das funções de valor ($V_n(s)$) e dos maximizadores ($f_n(s)$).
- Avaliar o nível de recurso atingido no fim do planeamento através do estudo da distribuição do estado final do sistema.

CAPÍTULO II

2. Material e Métodos

A metodologia deste trabalho é baseada nos princípios de Programação Dinâmica, onde se tem como objecto de estudo uma população biológica (por exemplo peixes). De referir que, não se fez um estudo prático do modelo, mas apenas tomou-se em conta aos modelos existentes relacionados com este tipo de investigação.

O trabalho comportou duas fases de estudo, onde na primeira faz-se uma descrição dos conceitos ligados a Programação Dinâmica e a segunda fase trata do modelo de gestão de recursos renováveis com um nível crítico s_{\min} onde o espaço de acções e dos estados são conjuntos em IR.

Para a efectivação do trabalho recorreu-se à bibliografia disponível na biblioteca do Departamento de Matemática e Informática, aos trabalhos de licenciatura dos estudantes Brígida da Cruz e Isaiás Nhavane, às aulas de esclarecimento de dúvidas, referências bibliográficas fornecidas pelos supervisores da tese, ao uso do computador e da linguagem de programação TURBO PASCAL versão 6.0 para a demonstração computacional dos resultados. As referências bibliográficas são ilustradas no final do trabalho.

CAPÍTULO III

3. Resultados

3.1 Modelos de PD

Em PD existem dois modelos básicos a saber:

Modelo Determinístico

Neste tipo de modelos, especificando os valores das variáveis de decisão, pode-se prever o resultado do modelo. Não há variáveis incontroláveis ou aleatórias.

Modelo Estocástico ou Probabilístico

Estes modelos contêm variáveis aleatórias que não podem ser controladas e que os respectivos valores são dados pelas distribuições de probabilidades. O resultado da decisão tomada não é previsível devido a factores aleatórios que podem “deturpá-lo”.

Em cada um destes modelos, pode-se considerar o caso estacionário e o não estacionário. Diz-se que um modelo é estacionário quando os componentes básicos do modelo não dependem do tempo. Caso contrário, o modelo diz-se não estacionário.

À seguir apresenta-se uma descrição dos componentes básicos de um Modelo de Programação Dinâmica e alguns resultados teóricos, seguindo para tal, Hinderer, (1993).

Definição 3.1

Sejam B e C conjuntos não vazios e $M \subset B \times C$. Diz-se que $M(b)$ é secção de M ou simplesmente b -secção de M se $M(b) \subset M$ onde $b \in B$ ou seja

$M(b) = \{c : (b,c) \in M\}$. Graficamente temos:

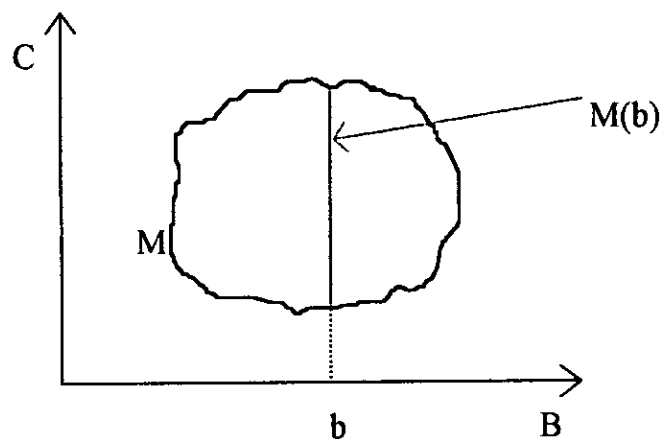


Figura 1 – b -secção de M

Componentes de um Modelo Estacionário Determinístico de Programação Dinâmica (MEDPD)

Tem-se, inicialmente, um número N de períodos chamado *horizonte de planeamento*. O sistema inicia em algum estado s_0 chamado *estado inicial*

pertencente a algum conjunto S , chamado *espaço de estados*. O sistema move-se através de S sob influência de acções (decisões) tomadas em cada período $n = 0, 1, \dots, N-1$ de um conjunto A chamado *espaço de acções*. Em cada período, uma decisão é tomada. Assume-se que o estado seguinte do sistema é conhecido. Em geral, se o sistema no período n , estiver no estado s , nem todas as acções do espaço de acções A podem ser tomadas, mas somente aquelas que pertencem a um certo subconjunto $D(s)$. Portanto, chama-se $D(s)$ ao *conjunto de acções admissíveis do estado s* , e $D := \{(s, a) \in S \times A : a \in D(s)\}$ de *conjunto de restrições*.

A transição de um estado para outro é descrita por uma função $T : D \rightarrow S$ chamada *função de transição* que tem o seguinte significado:

Se no período n o sistema estiver no estado s_n , e se a acção $a_n \in D(s_n)$ for tomada, então o sistema move-se para um novo estado $s_{n+1} := T(s_n, a_n)$.

Ao longo do trabalho irá se fazer referência aos conceitos de estágio e período.

Para o seu melhor entendimento, vejamos o seguinte exemplo:

Supõe-se que se tem um horizonte $N = 3$ (ver figura 2).

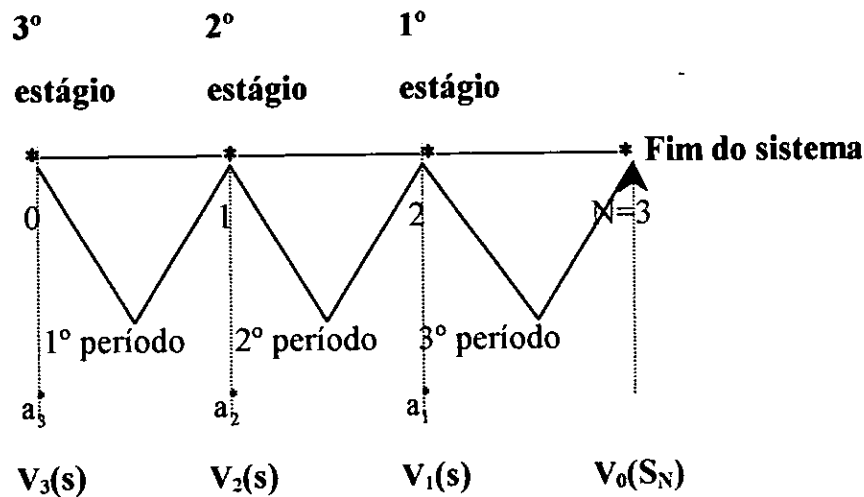


Figura 2 – Noção de estágio e período

Os pontos 0,1,2 e 3 representam os períodos e é sobre estes pontos onde as decisões devem ser tomadas. Uma particularidade é que a numeração dos períodos é feita para frente (da esquerda para a direita). O intervalo entre os períodos $n-1$ e n , designa-se n -ésimo período. Na resolução do problema, parte-se do último período até ao período inicial (da direita para a esquerda). A esta numeração do último até ao primeiro, designa-se estágio. O conceito de estágio está ligado com as acções óptimas (a_3^* , a_2^* , a_1^*), tomadas em cada período. Portanto, a_2^* é a decisão tomada no estágio 2, isto é, decisão tomada quando falta um período para o processo terminar. Neste exemplo, o estágio 1 corresponde ao período 3, o estágio 2 ao período 2 e o estágio 3 ao período 1. No último período, não se toma nenhuma decisão, pois, está-se no fim do sistema.

Se no estágio n estamos no estado s_n e tomámos a acção a_n , teremos o *ganho de um estágio*, $r(s_n, a_n) \in \mathbb{R}$. Se o sistema termina no período N no estado S_N , então um *ganho final* $V_0(S_N)$ é obtido.

Normalmente, o ganho obtido em diferentes períodos terá diferentes valores devido ao juro e/ou desvalorização. Este facto é levado em consideração pelo chamado factor de desconto $\beta \in \mathbb{R}^+$; isto significa que o ganho $r(s_n, a_n)$ obtido no período n e o ganho terminal $V_0(S_N)$ entram no balanço relativo ao período inicial como $\beta^n r(s_n, a_n)$ e $\beta^N V_0(S_N)$ respectivamente.

Resumindo, chega-se à seguinte definição:

3.1.1 Definição Básica de um MEDPD

Um MEDPD é uma sequência $(S, A, D, T, r, V_0, \beta)$ onde:

- a) S é um conjunto não vazio chamado *espaço de estados*;
- b) A é um conjunto não vazio chamado *espaço de acções*;
- c) D é um subconjunto de $S \times A$, tal que todas as secções $D(s)$ de D são não vazias. D é chamado *conjunto restrição* e $D(s)$ *conjunto de acções admissíveis* para o estado s ;

- d) T é uma função de D em S , chamada *função de transição*;
- e) r é uma função finita em D , chamada *função do ganho num estágio*;
- f) V_0 é uma função finita em S , chamada *função do ganho terminal*; e
- g) β é um número real positivo chamado *factor de desconto*.

3.1.2 Formulação do problema de Optimização de um MEDPD

Apresenta-se a seguir duas formulações do problema de optimização determinadas por um MEDPD, uma em termos de acções e outra em termos de políticas.

3.1.2.1 Formulação do Problema de optimização de um MEDPD em termos de acções

Dado um estado inicial s_0 arbitrário, diz-se que a sequência de acções

$(a_0, a_1, \dots, a_{n-1}) \in A^N$ é *admissível* em s_0 se:

$$a_0 \in D(s_0),$$

$$a_1 \in D(s_1), \text{ onde } s_1 := T(s_0, a_0),$$

$$a_2 \in D(s_2), \text{ onde } s_2 := T(s_1, a_1),$$

.....

$$a_{N-1} \in D(s_{N-1}), \text{ onde } s_{N-1} := T(s_{N-2}, a_{N-2});$$

Sendo $S_N := T(s_{N-1}, a_{N-1})$, a sequência (s_1, s_2, \dots, s_N) é chamada *sequência de estados gerados por s_0 e sequência de acções admissíveis* $y := (a_n)_0^{N-1}$. s_n é uma função de s_0 e de $y : s_n = s_{ny}(s_0)$. O conjunto de sequências de acções admissíveis de s_0 será denotado por $A^N(s_0)$.

Para cada estado inicial $s_0 \in S$ e cada sequência de acções $y = (a_n)_0^{N-1} \in A^N(s_0)$, o *ganho total de N estágios* é definido por,

$$V_{ny}(s_0) := \sum_{n=0}^{N-1} \beta^n r(s_n, a_n) + \beta^N V_0(S_N) \in \mathbb{R}, \quad (1)$$

onde $(s_n)_1^N = (s_{ny}(s_0))_1^N$ é a sequência de estados gerada por s_0 e y .

Portanto, a formulação do problema de optimização em termos de acções para um **MEDPD** é a seguinte:

I. Calcular $\text{Sup} \{ V_{Ny}(s_0) : y \in A^N(s_0) \}$

II. Encontrar se possível o ponto de máximo da função $y \rightarrow V_{Ny}(s_0)$ onde

$$y \in A^N(s_0).$$

Observação:

O número $V_N(s_0)$, que pode ser igual a $+\infty$, é o ganho máximo de N estágios. A função $V_N : S \rightarrow \mathbb{R}$ é chamada *função de valor de N estágios de um MEDPD* e joga um papel muito importante em toda a teoria.

3.1.2.2 Formulação do Problema de optimização de um MEDPD em termos de políticas

A formulação do problema de optimização em termos de acções, está directamente ligada a um estado inicial. Isto significa que, sempre que tivermos um novo estado inicial, temos que refazer os cálculos (obter nova sequência de acções óptimas) ligadas ao novo estado inicial. Em termos de políticas, temos uma situação diferente, pois existe a chamada regra de decisão que para cada estado dá a solução.

A formulação do problema de optimização em termos de políticas é indispensável quando se trata de programas dinâmicos estocásticos. Em vez de se decidir que sequência de acções tomar (em relação ao estado inicial), determina-se que acção $\phi_n(s)$ tomar estando no estado s , no período n . Portanto, não é indispensável conhecer o estado inicial. Este raciocínio conduz à seguinte definição:

Definição 3.2

1. Uma função $\phi : S \rightarrow A$ tal que $\phi_n(s) \in D(s)$ para todo $s \in S$ é chamada *regra de decisão (r.d.)*;

2. Uma sequência $\pi = (\phi_0, \phi_1, \dots, \phi_{N-1})$ de N regras de decisão é chamada *política de N estágios*;
3. F é o conjunto de todas as regras de decisão; e
4. F^N é o conjunto de todas as políticas de N estágios e não depende do estado inicial.

A condição $\phi_n(s) \in D(s)$ para r.d.'s, garante que ao se usar uma política de N estágios arbitrária, selecciona-se em cada período e em cada estado somente acções admissíveis. Se se inicia em s_0 e usa-se uma política $\pi = (\phi)_0^{N-1} \in F^N$, então o sistema move-se sucessivamente através da sequência de estados

$$\begin{aligned}
 s_1 &:= T(s_0, \phi_0(s_0)), \\
 s_2 &:= T(s_1, \phi_1(s_1)), \\
 &\dots\dots\dots \\
 s_N &:= T(s_{N-1}, \phi_{N-1}(s_{N-1})),
 \end{aligned} \tag{2}$$

s_n , onde $1 \leq n \leq N$ depende de s_0 e π , ou seja $s_n = s_n \pi(s_0)$. Na forma mais geral tem-se

$$s_{n+1} := T(s_n, \phi_n(s_n)), \quad 0 \leq n \leq N-1 \tag{3}$$

Designa-se $(s_n)_1^N$ de *sequência de estados gerada por s_0 e política π* e $(\phi_n(s_n))_0^{N-1}$ de *sequência de acções geradas por s_0 e π* . s_n é também a sequência de estados gerada por s_0 e pela sequência de acções $(\phi_n(s_n))_0^{N-1}$.

O *ganho total de N estágios sob a política π e estado inicial s_0* , é definido por:

$$V_{N\pi}(s_0) := \sum_{n=0}^{N-1} \beta^n r(s_n, \phi_n(s_n)) + \beta^N V_0(S_N). \quad (4)$$

Portanto, a formulação do problema de otimização em termos de políticas de um *MEDPD* é a seguinte:

- I. Calcular $\text{Sup} \{V_{N\pi}(s_0) : \pi \in F^N\}$
- II. Encontrar se possível uma política s_0 -optimal, isto é, o ponto máximo da função $\pi \rightarrow V_{N\pi}(s_0)$.
- III. Encontrar se possível uma política óptima, isto é, uma política que é s_0 -optimal para cada estado inicial s_0 .

As duas formulações são equivalentes através da seguinte proposição:

Proposição 3.1

Seja s_0 um estado inicial arbitrário. É válido o seguinte:

- I. $\text{Sup} \{V_{N\gamma}(s_0) : \gamma \in A^N(s_0)\} = \text{Sup} \{V_{N\pi}(s_0) : \pi \in F^N\}$

II. Se π^* é uma política de N estágios e se y^* é a sequência de acções geradas por s_0 e π^* , então π^* é s_0 -optimal, se e só se, y^* é s_0 -optimal.

3.1.3 Análise das duas Formulações

I. Resultados teóricos são muitas vezes mais fáceis de formular em termos de políticas do que de sequências de acções. Em problemas práticos (aplicações), a formulação em termos de acções é mais importante.

II. Maximizar sobre um conjunto de sequências de acções, à primeira vista parece mais fácil do que maximizar sobre um conjunto de políticas, pois na primeira temos sequências finitas de pontos enquanto que na última temos sequências finitas de funções do espaço de estados dentro do espaço de acções. Esta impressão é reforçada pelo facto de que em geral o conjunto F^N , de políticas de N estágios tem maior cardinalidade do que os conjuntos $A^N(s_0)$. Como consequência, é frequente uma abundância de políticas s_0 -optimais em casos onde existe relativamente poucas sequências de acções s_0 -optimais. Além disso a propriedade das políticas serem compostas de maximizadores, resulta num conceito extremamente útil.

III. Em Programas Dinâmicos Estocásticos, o estado e a acção momentânea não determinam unicamente o futuro. Neste caso, a formulação em termos de políticas é aplicável.

3.1.4 Teorema Básico para a maximização dos Ganhos para um MEDPD

I. É válida a *iteração de valor*:

As *funções de valor* V_n podem ser calculadas recursivamente do seguinte modo:

$$\begin{aligned} V_n(s) &= \text{Sup}_{a \in D(s)} \{r(s,a) + \beta V_{n-1}(T(s,a))\} \quad s \in S, n \in N. & (9) \\ &= \text{Sup}_{f \in F} \{r_f(s) + \beta V_{n-1}(T_f(s))\} \quad s \in S, n \in N. \end{aligned}$$

II. É válido o seguinte *critério de optimalidade*:

Se $f_n(s)$ é um ponto máximo da função

$$a \rightarrow w_n(s,a) := r(s,a) + \beta V_{n-1}(T(s,a)) \quad 0 \leq n \leq N-1 \text{ e } s \in S \quad (10)$$

então a política de N estágios $\pi^* = (f_n)_N^1$ é ótima. Para um $s_0^* = s_0$ se $(s_n^*)_1^N$ é a sequência de estados gerada por s_0 e π^* , então a sequência de acções,

$$a_n^* := f_{N-n}(s_n^*), \quad 0 \leq n \leq N-1 \text{ é ótima em relação à } s_0.$$

Observações:

- I. A *Iteração de valor* é válida se existe uma política ótima (para cada horizonte) ou não.

- II. f_n em II, denota a r.d. que é usada n estágios antes do processo terminar, ao passo que ϕ_n irá sempre denotar a r.d. que é usada n períodos depois do processo começar. Portanto, as duas políticas de N estágios $(f_n)_N^I$ e $(\phi_n)_0^{N-1}$ coincidem se, e só se, $f_n = \phi_{N-n}$ para $1 \leq n \leq N$.

Definição 3.3

- I. O conjunto $D_n^*(s)$ constituído de pontos máximos da função

$$a \rightarrow w_n(s,a) \tag{11}$$

é chamado *conjunto de acções óptimas no estado s e estágio n* .

- II. A regra de decisão f é chamada *maximizador no estágio n* se $f(s) \in D_n^*(s)$ para todo $s \in S$.

Nota que, por iteração de valor, f é um maximizador no estágio n , se e só se,

$$w_n(s, f(s)) = V_n(s), \quad s \in S$$

Com a definição anterior, o critério de optimalidade pode ser enunciado da seguinte maneira:

Se para $0 \leq n \leq N-1$, f_n é um maximizador no estágio n , então $(f_n)_N^1$ é ótima.

É possível aplicar estes resultados para o problema de minimização, bastando para tal efectuar as correspondentes modificações.

3.1.5 Procedimento de Solução de Problemas de PD

Na resolução de problemas de optimização, a solução inicia com o problema de um estágio e vão-se adicionando sequencialmente uma série de problemas de um estágio até se otimizar o problema global. O procedimento de solução é baseado nos procedimentos *backward* e *forward*. No primeiro procedimento, o cálculo das funções de valor $V_n(s)$ e dos maximizadores $f_n(s)$, é feita a partir do último período (primeiro estágio) até ao primeiro período (último estágio), obtendo-se o óptimo em cada estágio. No segundo procedimento, obtém-se a sequência de acções óptimas, recursivamente, com base nos maximizadores do processo anterior.

Explica-se pormenorizadamente o procedimento *backward*, pois este será usado para a solução do modelo em estudo.

3.1.5.1 O Procedimento *Backward*

O primeiro passo neste procedimento consiste em resolver um problema de um estágio a partir do último período (isto é, estágio $n = 1$) do problema original, calculando para cada $s \in S$ o conjunto $D_1^*(s)$ de acções óptimas e o ganho total máximo $V_1(s)$. No segundo passo, os dois últimos períodos são tratados como um problema de um estágio, onde a função de ganho terminal V_0 é substituída por V_1 . Assim obtemos para cada $s \in S$ o conjunto $D_2^*(s)$ de acções óptimas e o ganho máximo de dois estágios $V_2(s)$, etc. Aqui vê-se claramente como é que o problema global de N estágios é dividido em N problemas paramétricos interligados de um estágio e todos contendo a mesma estrutura. Depois de N passos, o Procedimento *Backward* pára e a função de valor V_N é obtida. Além disso, a r.d. f_n é obviamente um maximizador no estágio n se, e só se, $f_n(s) \in D_n^*(s)$ para todo s . Portanto, se todos os conjuntos $D_n^*(s)$, $1 \leq n \leq N$, $s \in S$, são não vazios, então obtém-se uma política de N estágios óptima consistindo de maximizadores f_n seleccionando $f_n(s)$ arbitrariamente de $D_n^*(s)$. Desta maneira, conhecendo todos os conjuntos $D_n^*(s)$ produz-se todas as políticas óptimas de N estágios consistindo de maximizadores.

De referir que, o procedimento *Forward* é usado para o cálculo de todas as sequências de acções óptimas depois de completado o procedimeto *backward*.

3.2 Modelos de PD com Distúrbâncias Independentes e Identicamente Distribuídas

O caso mais simples do modelo estocástico de N estágios é aquele em que a transição de s_n para s_{n+1} é especificada por uma função de transição T que é “disturbada” por uma variável aleatória X_{n+1} que toma valores num conjunto finito M e onde as variáveis X_1, X_2, \dots, X_N são independentes e identicamente distribuídas (i.i.d.). Assume-se que os X_i 's são definidos no espaço probabilístico (Ω, P) .

A lei de transição do modelo estocástico é uma função T de $D \times M$ em S com o seguinte significado: Se no período n está-se no estado s_n e toma-se a acção a_n e se a distúrbância X_{n+1} assume o valor x_{n+1} então o sistema move-se para um novo estado aleatório,

$$X_{n+1} := T(s_n, a_n, x_{n+1}), \quad 0 \leq n \leq N-1 \quad (12)$$

3.2.1 Definição Básica

Um *Modelo de Controlo* (MC) com distúrbâncias i.i.d. e espaço de distúrbâncias finito é uma sequência $(S, A, D, M, q, T, r, V_0, \beta)$ do seguinte tipo:

- I. S, A, D, T, r, V_0 e β têm o mesmo significado e interpretação como no MEDPD;

II. M é um conjunto finito de *valores de perturbações*;

III. q é uma função *probabilística de perturbações i.i.d*; e

IV. T é uma função de $D \times M$ em S , chamada *função de transição estocástica*.

O *ganho obtido no estágio* N pelo processo de decisão $(\zeta_{n\pi})_0^N$ quando este inicia em s , é uma variável aleatória real em (Ω, P) da seguinte forma:

$$R_{N\pi}(s, Y) := \sum_{n=0}^{N-1} \beta^n r(\zeta_{n\pi}, \varphi_n(\zeta_{n\pi})) + \beta^N V_0(\zeta_{N\pi}),$$

(13)

onde $Y := (X_i)_i^N$ ($\zeta_{n\pi}$ é uma função de s e Y)

O *ganho esperado no estágio* N é definido pelo número real,

$$V_{N\pi}(s) := ER_{N\pi}(s, Y) = \sum_{y \in M^N} R_{N\pi}(s, y) \cdot P(Y=y). \quad (14)$$

A *função de valor* $V_N : S \rightarrow \mathbb{R}$, no estágio N é definida por:

$$s \rightarrow V_N(s) := \text{Sup}\{V_{N\pi}(s) : \pi \in F^N\}, N \in \mathbb{N}. \quad (15)$$

A noção de maximizador é similar à correspondente noção para MEDPD.

3.2.2 Teorema Básico para um MC com Distúrbâncias i.i.d. e espaço de distúrbância finito.

É válido o seguinte:

I. *Iteração de valor (IV)*

$$V_n(s) = \sup_{a \in D(s)} \{r(s,a) + \beta \cdot EV_{n-1}(T(s,a,X))\} \quad n \in \mathbb{N}, s \in S$$

II. *Crítério de Optimalidade (CO)*

Se f_n é um maximizador no estágio n , para $0 \leq n \leq N-1$, então, $(f_n)_N^1$ é óptimo para um Programa Dinâmico de N estágios DP_N .

3.3 Estudo das Propriedades estruturais de um MEDPD

No estudo de modelos de programação dinâmica, muitas vezes não é fácil obter-se uma solução analítica. Somente um número reduzido destes modelos possui uma solução analítica. Portanto, recorre-se ao estudo das propriedades estruturais de modo a obter-se resultados qualitativos e em particular resultados que podem ajudar na simplificação dos cálculos.

Nesta sessão, apresentam-se lemas e teoremas matemáticos e de programação dinâmica (sem demonstração) que servirão de suporte no estudo de propriedades

estruturais do modelo a estudar. As demonstrações dos teoremas, dos lemas, etc, poderão ser encontradas no livro de K. Hinderer (1993), bem como em outros livros referenciados na bibliografia.

3.3.1 Funções côncavas e convexas (Roberts & Varberg, 1973)

Definição 3.4

Chama-se convexa à função $f: I \rightarrow \mathbb{R}$ se :

$$f[\lambda.x + (1-\lambda).y] \leq \lambda.f(x) + (1-\lambda).f(y), \text{ onde } x, y \in I \text{ e } \lambda \in (0,1)$$

Definição 3.5

Uma função $f: I \rightarrow \mathbb{R}$, chama-se côncava se:

$$f[\lambda.x + (1-\lambda).y] \geq \lambda.f(x) + (1-\lambda).f(y), \text{ onde } x, y \in I \text{ e } \lambda \in (0,1)$$

Definição 3.6

Diz-se que a função $-f(x)$ é convexa se, e só se, a função $f(x)$ for côncava.

Lema 3.1

- i) A composição de duas funções crescentes, é uma função crescente
- ii) A soma de duas funções crescentes quando definidas, é uma função crescente
- iii) Se v é uma função crescente e $c \in \mathbb{R}^+$ então, $c.v$ é crescente.
- iv) O produto de duas funções não negativas, é crescente
- v) A função positiva v é crescente se, e só se, $1/v$ for crescente

Teorema 3.1

Seja $f : I \rightarrow \mathbb{R}$ e $g : I \rightarrow \mathbb{R}$ funções convexas, então $f+g$ e $f.g$ são convexas.

Teorema 3.2

Seja $f : I \rightarrow \mathbb{R}$ e $g : I \rightarrow \mathbb{R}$ funções não negativas, decrescentes [crescentes] e convexas, então $h(x) = f(x).g(x)$ é uma função não negativa decrescente [crescente] e convexa.

Teorema 3.3

Se a função $f : [a; b] \rightarrow \mathbb{R}$ for convexa, então ela atinge o seu máximo no ponto a ou no ponto b .

Teorema 3.4

Se a função $f : [a; b] \rightarrow \mathbb{R}$ for côncava, então ela atinge o seu mínimo no ponto a ou no ponto b .

3.3.2 Funções de Valor

Teorema 3.5

Se $V_0 \leq V_1$ [$V_0 \geq V_1$], então $n \rightarrow V_n(s)$ é crescente [decrescente] para todo $s \in S$.

Teorema 3.6 (Monotonia de funções de valor)

Supõe-se que S é estruturado. Então, $s \rightarrow V_n(s)$ é crescente para todo $n \in \mathbb{N}$ se:

- i. $D(\cdot)$ é crescente.
- ii. $s \rightarrow T(s, a, x)$ é crescente em D_a ; $D_a := \{s \in S : (s, a) \in D\}$ para todo a e todo $x \in M$.
- iii. $s \rightarrow r(s, a)$ é crescente e
- iv. V_0 é crescente

Teorema 3.7 (Concavidade de funções de valor)

Supõe-se que:

- i. D é convexo.
- ii. $D(s)$ é limitado em todo s .
- iii. T é crescente.
- iv. V_0 e $r(\cdot, a)$ são côncavas

então, V_n e w_n são finitas e côncavas para todo $n \in \mathbb{N}$

Lema 3.2 (Supremo de uma família de funções crescentes)

Seja S um conjunto estruturado e w uma função em D . Se $s \rightarrow D(s)$ é crescente e se $s \rightarrow w(s, a)$ é crescente em todo a , então, $s \rightarrow w^*(s) := \text{Sup} \{w(s, a) : a \in D(s)\}$ é crescente em s .

Definição 3.7

Diz-se que a função w tem diferenças crescentes se, e só se, $s \rightarrow w(s, a') - w(s, a)$ é crescente em $s \in D : (s, a) \in D, (s, a') \in D$, para todo $a \leq a'$.

Teorema 3.8 (Concavidade de funções de valor crescentes)

Assume-se que:

- i. D é convexa.
- ii. $D(s)$ é limitada em todo s .
- iii. $D(\cdot)$ é crescente.
- iv. $T(\cdot, a)$, $r(\cdot, a)$ e V_0 são crescentes em todo a .
- v. T , r e V_0 são côncavas.

Então V_n e W_n são côncavas (e V_n é crescente) em todo $n \in \mathbb{N}$

Teorema 3.9 (Convexidade de funções de valor)

Assume-se que:

- i. S é convexa
- ii. A é um conjunto arbitrário e $D(s) = A$ para todo s .
- iii. $r(\cdot, a)$, $a \in A$ e V_0 são convexas e [crescentes]
- iv. V_n é finita para todo n . Então, V_n é convexa [e crescente] para todo $n \in \mathbb{N}$.

3.3.3 Maximizadores

Considere-se um problema de optimização P . Dada uma função w em D , pretende-se maximizar para cada parâmetro s em S uma função $a \rightarrow w(s, a)$ em $D(s)$. Assume-se que para cada $s \in S$ a função $w(s, \cdot)$ tem o menor ponto máximo $g(s)$ e o maior ponto máximo $h(s)$. Isto conduz-nos a seguinte definição:

Definição 3.8

Uma regra de decisão para D é uma função f de S em A tal que $f(s) \in D(s)$ para todo s . Um maximizador de w é uma regra de decisão tal que $f(s)$ é o ponto máximo de $a \rightarrow w(s,a)$ para todo s . Assim g e h são os menores e maior maximizadores de w , respectivamente.

Definição 3.9

Diz-se que a correspondência $B(\cdot)$ de S em A é monotamente completa se $s \leq s'$, $a \in B(s)$, $a' \in B(s')$ e $a \leq a'$ implica $a' \in B(s)$ e $a \in B(s')$.

Lema 3.3

Seja $B(\cdot)$ uma correspondência monotamente completa de S em A . Se $B(s)$ tem um menor [maior] elemento $f(s)$, $s \in S$, então f é crescente.

Definição 3.10

Uma função finita w em D tem diferenças crescentes se $s \rightarrow w(s,a) - w(s,a')$ é crescente em $D_a \cap D_{a'}$, onde $a < a'$.

Observação: A função finita w em D tem diferenças crescentes se, e só se, $w(s,a) - w(s,a') \leq w(s',a) - w(s',a')$ onde $s \leq s'$, $a \leq a'$.

Teorema 3.10

Assume-se que $S \subset \mathbb{R}$, $A \subset \mathbb{R}$, D tem a forma de intervalo com d_1 e d_2 crescentes e que $w : D \rightarrow \mathbb{R}$ tem diferenças crescentes. Se $w(s, \cdot)$ tem o menor [maior] ponto máximo $g(s)$ [$h(s)$], então g é crescente [h é crescente].

Teorema 3.11

Sejam S e A conjuntos estruturados e supõe-se que:

- i. $D(\cdot)$ é monotamente completa.
- ii. w tem diferenças crescentes

Então, cada menor e cada maior maximizador de w é crescente.

Lema 3.4

Assume-se que $S \subset \mathbb{R}^n$, $A = \mathbb{R}$ e $D(s) := [0, d(s)]$ para uma função contínua d de S em \mathbb{R} . Se $w : D \rightarrow \mathbb{R}$ é contínua e limitada, então $s \rightarrow w^*(s) := \max_{0 \leq a \leq d(s)} w(s,a)$ é contínua.

Definição 3.11

Denota-se por $ILIP(K)$ o conjunto de todas as funções crescentes f em S que são contínuas com a constante LIPSCHITZ K .

3.3.3.1 Existência de políticas ótimas e continuidade de funções de valor e dos maximizadores

A existência de políticas ótimas é óbvia se o conjunto $D(s)$ de acções admissíveis é finito em todo s .

Assume-se que os espaços de acções e de estados são métricos e (S, d_1) e (A, d_2) e D são dotados de um máximo $D((s,a),(s',d')) := \max(d_1(s,s'), d_2(a,a'))$.

A existência de um maximizador no estágio n , é garantida se $D(s)$ é compacto e se se puder mostrar que $LV_{n-1}(s, \cdot)$ é contínua, em $s \in S$. Para isso, basta usar o teorema sobre as funções contínuas (Hinderer, 1993).

Recorde-se que uma função v num espaço métrico M é chamada semi-contínua superior se $\limsup v(x_n) \leq v(x)$, $n \rightarrow \infty$ para cada sequência (x_n) em M que converge para x , e é chamada semi-contínua inferior, se $-v$ é semi-contínua superior. Em particular, v é contínua se, e só se, é semi-contínua superior e semi-contínua inferior.

Definição 3.12

Seja $D(\cdot)$ uma correspondência de S em A .

- i. $D(\cdot)$ é chamada quasi-contínua se tiver as seguintes propriedades:

Se $s_n \rightarrow s$ e $a_n \in D(s_n)$ para todo $n \in \mathbb{N}$, então $(a_n)_n^\infty$ tem uma sequência de valores em $D(s)$.

- ii. $D(\cdot)$ é chamada semi-contínua inferior se tiver as seguintes propriedades:

Se $s_n \rightarrow s$, então cada ponto em $D(s)$ é um grupo de valores de uma sequência de pontos $a_n \in D(s_n)$ para todo $n \in \mathbb{N}$.

- iii. $D(\cdot)$ é chamada contínua se for quasi-contínua e semi-contínua inferior.

Lema 3.5

- i. Assume-se que A é um intervalo em \mathbb{R} e que:

$D(s) = [c(s), d(s)]$, $s \in S$, para funções c e d de S em A , $d \geq c$.

Se c é semi-contínua inferior, e d é semi-contínua superior, então $D(\cdot)$ é quasi-contínua; se c é semi-contínua superior e d é semi-contínua inferior, então

- $D(\cdot)$ é semi-contínua inferior; Portanto se c e d são contínuas então $D(\cdot)$ é contínua.
- ii. Se A é compacto, então $D(\cdot)$ é quasi-contínua se, e só se, D é fechado.
 - iii. Se $D(s) = A$ para todo s , então $D(\cdot)$ é semi-contínua inferior; se D é compacto, então $D(\cdot)$ é contínua.
 - iv. Seja g uma função de S em A com $s \rightarrow G(s) := \{g(s)\}$. Então g é contínua se $G(\cdot)$ é quasi-contínua se, e só se, $G(\cdot)$ é semi-contínua inferior se, e só se, $G(\cdot)$ é contínua.
 - v. Se $D_1(\cdot)$ é fechado, se $D(\cdot)$ é quasi-contínua e se $D_1 \subset D$, então $D_1(\cdot)$ é quasi-contínua.

Lema 3.6

- i. Se $D(\cdot)$ é quasi-contínua e se w é semi-contínua superior, então w^* é semi-contínua superior. Se w é finita, então w^* é finita.
- ii. Se $D(\cdot)$ e w são semi-contínuas inferiores, então w^* é semi-contínua inferior.
- iii. Se $D(\cdot)$ e w são contínuas, então w^* é contínua.

Lema 3.7

- i. Se $D(\cdot)$ é quasi-contínua e se w é semi-contínua superior, então w tem um maximizador. Se $A \subset \mathbb{R}$, então w tem um menor e um maior maximizador.
- ii. Seja $D(\cdot)$ e w contínuas. Se w tem um único maximizador f , então f é contínua. Se $A \subset \mathbb{R}$, então o menor [maior] maximizador de w é semi-contínua inferior [semi-contínua superior].

Teorema 3.12

Assume-se que:

- i. $D(\cdot)$ é quasi-contínua [contínua]
- ii. T é contínua
- iii. r e V_0 são semi-contínuas superiores [contínuas]
- iv. Então, V_n e LV_{n-1} são finitas e semi-contínuas superiores [contínuas] em todo n , e existe um maximizador em cada estágio. Se $A \subset \mathbb{R}$, então existe um maior e um menor maximizador em cada estágio.

Definição 3.13

Os operadores $L, U_f, f \in F$, e U transformam as funções v em S em funções Lv em D , $U_f v$ em S e Uv em S como se segue:

$$I. (Lv)(s,a) := r(s,a) + \beta v(T(s,a)),$$

$$II. (U_f v)(s) := Lv(s, f(s)) = r_f(s) + \beta v(T(s,a)),$$

$$III. (Uv)(s) := \sup Lv(s,a) = \sup U_f v(s), \text{ onde } a \in D(s) \text{ e } f \in F.$$

Corolário 3.1

Assume-se que $D(\cdot)$, T , r e V_0 são contínuas, então existe um maximizador f_n em cada estágio. Logo,

- i. Se f_n é único, então f_n é contínua.

- ii. Se $A \subset \mathbb{R}$, então o menor [maior] maximizador no estágio n é semi-contínua inferior [superior].

3.4 Modelo para Gestão de Recursos Renováveis com estado crítico

(s_{\min})

Os problemas de gestão de recursos renováveis, surgem quando temos que gerir um determinado recurso (ex: uma população biológica) de modo a obter um ganho máximo num certo horizonte N . Para este modelo, aplicar-se-à o método de optimização sequencial de programação dinâmica, onde se tem como objecto de estudo uma população constituída de peixes. Propõe-se um modelo de programação dinâmica com perturbações aleatórias, uma vez que a população é sujeita à factores ambientais que ocorrem ao acaso, que podem influenciar no crescimento e/ou extermínio da mesma. Deste modo, introduz-se neste modelo um estado crítico s_{\min} que designa o nível de segurança do recurso, abaixo do qual nenhuma exploração do recurso deve ser feita.

O modelo de gestão de recursos renováveis com um estado crítico s_0 , onde se considera o tempo e os conjuntos de estados e das acções discretos, foi estudado no trabalho de licenciatura da estudante Brígida da Cruz em 1977, (Cruz, 1997).

No presente trabalho, far-se-à o estudo do mesmo modelo, considerando o tempo discreto e os conjuntos de estados e de acções intervalos em \mathbb{R} .

3.4.1 Formulação do Modelo

1. s é o nível da população e representa o estado do sistema. s_n representa o nível da população num certo período n . Portanto, S representa o conjunto de todos os níveis populacionais e denota-se por $S \in [0; S_{\max}]$, $S_{\max} \in \mathbb{R}$, onde S_{\max} designa o nível populacional máximo.
2. A acção a_n representa o nível populacional que resta depois da exploração do recurso. $A = S = [0; S_{\max}]$, $S_{\max} \in \mathbb{R}$ é o conjunto de todos os níveis atingidos depois da exploração e $(s - a)$ é a exploração do recurso.
3. $D(s)$ é o conjunto de todas as acções admissíveis quando o nível da população for s , isto é, $D(s) = [0; s]$. Mais precisamente,

$$D(s) = \begin{cases} \{s_{\min}, \dots, s\}, & \text{se } s_{\min} \leq s \leq s_{\max} \\ s, & \text{se } 0 \leq s \leq s_{\min} \end{cases}$$

onde s_{\min} designa o nível crítico (segurança do recurso) e s_{\max} o nível populacional máximo.

4. O conjunto restrição $D = \{(s,a) \in [0; S_{\max}]^2 : 0 \leq a \leq s\}$.

5. O crescimento populacional está sujeito a factores ambientais que ocorrem ao acaso. Portanto, X_n representa a perturbação aleatória que influencia o crescimento da população em cada período e toma valores finitos num conjunto M . Supõe-se que os x_i 's são estocasticamente independentes e identicamente distribuídos.
6. A transição do estado s_n para o estado s_{n+1} é representada por uma função de transição T de $D \times M$ em S com o seguinte significado: Se o nível da população num certo período n , for s_n tendo em conta que a acção a_n foi tomada e ocorre uma perturbação aleatória x_{n+1} , o sistema move-se para um novo estado aleatório,

$$s_{n+1} := T(s_n, a_n, x_{n+1}).$$

No modelo, assume-se que $T(s, a, x)$ não depende do estado s , isto é, quando se faz a gestão do recurso no período $n=2$, não importa saber o que aconteceu no período anterior. Portanto, se ζ_n é um estado aleatório no período n e a_n o nível da população depois da exploração do recurso, então, o próximo estado será $\zeta_{n+1} := T(a_n, x_{n+1})$. Este modelo chama-se invariante, pois a função de transição $T(s, a, x)$ não depende de s .

7. A função $r : D \rightarrow \mathbb{R}$, chamada função do ganho tem a seguinte interpretação: Se o nível da população no período n for s_n e se uma acção a_n for tomada, então obtém-se um ganho $r_n(s_n, a_n)$.

8. V_0 representa a função do ganho terminal e $V_0(S_N)$ representa o balanço final da população antes de uma nova gestão.
9. O número $\beta \in (0,1)$ representa o factor de desconto.

Duma maneira geral, ao se fazer a gestão ao fim de N períodos, é natural que se faça uma avaliação dos ganhos ou perdas obtidos. Na gestão de recursos renováveis, como o objectivo é maximizar o nível do recurso no fim do planeamento, interessa avaliar em quanto o recurso aumentou ao fim de N períodos de tempo. A avaliação do nível do recurso no fim do planeamento, será feita com base no estudo das funções de valor $V_n(s)$ e dos maximizadores $f_n(s)$. Portanto, aplicando o teorema básico para um MC com perturbações aleatórias i.i.d., tem-se:

- a) Iteração de valor

$$V_n(s) = \max \{ r(s,a) + \beta E[V_{n-1}(T(a,X))] \} \text{ com } a \in [0;s]$$

- b) Critério de Optimalidade

Se $f_n(s)$ é um maximizador no estágio n , $1 \leq n \leq IN$, então, a política é ótima para o modelo de PD_N .

Para a simulação do modelo, supõe-se que:

- a) $r(s,a) = \sqrt{s-a}$;
- b) $\beta = 0,98$;
- c) A variável aleatória X contínua, obedece a distribuição uniforme tomando valores no intervalo $[0; 2]$;
- d) A função de transição $T(a,X) = ax$;
- e) O nível de segurança $s_{\min} = 2$;
- f) O nível populacional máximo é $S_{\max} = 10$;
- g) O conjunto de todos os níveis populacionais é $S \in [0; 10]$; e
- h) $V_0 = 0$.

Sabe-se da estatística que a densidade de uma variável aleatória distribuída uniformemente no intervalo $[a; b]$, é dada por:

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0 & , \text{ caso contrário} \end{cases},$$

e a esperança matemática de uma variável aleatória contínua é dada por:

$$E[x] = \int_{-\infty}^{+\infty} xf(x)dx$$

Portanto, para um certo nível populacional, a função de valor será dada por:

$$V_n(s) = \text{Sup}_{a \in [0, s]} \{ r(s, a) + \beta E[V_{n-1}(T(a, X))] \}$$
$$= \text{Sup}_{a \in [0, s]} \left\{ r(s, a) + \beta \int_a^b [V_{n-1}(T(a, X))] \cdot \frac{1}{b-a} dx \right\}$$

Substituindo pelos correspondentes valores,

$$f(x) = \begin{cases} \frac{1}{2}, & 0 \leq x \leq 2 \\ 0, & \text{caso contrário} \end{cases}$$

tem-se

$$V_n(s) = \text{Sup}_{a \in D(s)} \left\{ \sqrt{s-a} + \beta \int_0^2 [V_{n-1}(ax_i)] \cdot \frac{1}{2} dx_i \right\}$$

Os x_i 's são gerados aleatoriamente, obedecendo aos critérios de geração de variáveis aleatórias contínuas, usando a distribuição uniforme, (Law & Kelton, 1991, P. 485).

Da posse do programa computacional (ANEXO A) desenvolvido, obtiveram-se as seguintes tabelas de função de valor ($V_n(s)$) e dos maximizadores ($f_n(s)$):

$V_n(s)$

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	1.41	1.73	2.00	2.24	2.45	2.65	2.83
0.00	0.00	0.69	1.69	2.11	2.43	2.71	2.98	3.22	3.44	3.65
0.00	0.34	1.03	2.03	2.45	2.76	3.06	3.33	3.58	3.81	4.03
0.00	0.51	1.20	2.20	2.61	2.93	3.23	3.50	3.75	3.99	4.21
0.00	0.59	1.28	2.28	2.69	3.01	3.32	3.58	3.84	4.07	4.30
0.00	0.63	1.32	2.32	2.73	3.05	3.36	3.62	3.88	4.12	4.34
0.00	0.65	1.34	2.34	2.75	3.07	3.38	3.64	3.90	4.14	4.36
0.00	0.66	1.35	2.35	2.76	3.08	3.39	3.65	3.91	4.15	4.37
0.00	0.66	1.35	2.35	2.77	3.09	3.39	3.66	3.92	4.15	4.38
0.00	0.66	1.36	2.36	2.77	3.09	3.39	3.66	3.92	4.16	4.38

Tabela 1: Valores das funções de valor $V_n(s)$ com $\beta = 0,98$

$f_n(s)$

0	1	2	2	2	2	2	2	2	2	2
0	1	2	2	2	2	3	3	3	4	4
0	1	2	2	2	3	3	4	4	4	4
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5
0	1	2	2	2	3	3	4	4	4	5

Tabela 2: Valores dos maximizadores $f_n(s)$.

3.4.2 Estudo das Propriedades estruturais do Modelo

Nesta sessão aplicam-se, ao modelo em estudo, resultados matemáticos e de PD apresentados na secção 3.3.

1. Segundo o **Teorema 3.5**, a função $n \rightarrow V_n(s)$ é crescente para todo $s \in S$. No modelo, isto significa que para um nível populacional inicial fixo s , o ganho máximo V_n cresce com o número de períodos de decisão.

Demonstração:

a) Supõe-se que $V_0 \equiv 0$ e $r \geq 0$, então,

$$V_1(s) = \max_{a \in D(s)} \{r(s,a) + \beta E[V_0(T(a,X))]\} = \max_{a \in D(s)} r(s,a) \geq 0 = V_0(s)$$

b) Supõe-se ainda que $V_n(s) \geq V_{n-1}(s)$ para todo $s \in S$ e $n \in \mathbb{N}$.

logo,

$$\begin{aligned} V_{n+1}(s) &= \max_{a \in D(s)} \{r(s,a) + \beta E[V_n(T(a,X))]\} \geq \max_{a \in D(s)} \{r(s,a) + \beta E[V_{n-1}(T(a,X))]\} \\ &= V_n(s). \end{aligned} \quad \square$$

Este resultado é ilustrado também graficamente (gráfico 1, ANEXO B)

2. Do **Teorema 3.6**, tem-se que $s \rightarrow V_n(s)$ é crescente para todo $n \in \mathbb{N}$. Isto significa que, o ganho máximo num certo horizonte N cresce em relação ao nível populacional, obedecendo as seguintes condições:

- a) $r(s,a)$ e V_0 crescentes
- b) $T(a, X)$ crescente em D_a

Demonstração:

a) Pela condição do teorema, V_0 é crescente.

Supondo que $V_n(s') \geq V_n(s)$ para $s' \geq s$

Deve-se mostrar que $V_{n+1}(s') \geq V_{n+1}(s)$ para $s' > s$.

$$V_{n+1}(s') = \max_{a \in D(s')} \{r(s',a) + \beta E[V_n(T(s',X))]\}$$

$$V_{n+1}(s) = \max_{a \in D(s)} \{r(s,a) + \beta E[V_n(T(a,X))]\}$$

Verificam-se os seguintes casos:

- $0 \leq s < s' \leq s_{\min}$

neste caso:

a) $D(s) = \{s\}$ e $D(s') = \{s'\}$

b) $r(s,a) = r(s',a) = 0$.

c) $T(s',X) \geq T(s,X)$, pois, pela condição do teorema $T(.,X)$ é crescente, logo

$$V_n(T(s',X)) \geq V_n(T(s,X)) .$$

De a), b) e c) tem-se

$$V_{n+1}(s') = \beta E[V_n(T(s',X))] \geq \beta E[V_n(T(s,X))] = V_{n+1}(s)$$

- $s_{\min} \leq s < s' \leq s_{\max}$

Neste caso $r(s,a) \leq r(s',a)$ e $D(s) \subseteq D(s')$.

Já que o modelo em estudo é invariante, segue que:

$$V_{n+1}(s') \geq V_{n+1}(s)$$

- $0 \leq s \leq s_{\min}$ e $s_{\min} \leq s' \leq s_{\max}$

Do 1º e 2º caso, tem-se

$$V_{n+1}(s) \leq V_{n+1}(s_{\min}) \leq V_{n+1}(s').$$

□

Este resultado é ilustrado também graficamente (Gráfico 2, ANEXO B)

3. Do Teorema 3.9 tem-se que $V_n(s)$ é finita e convexa para todo $n \in \mathbb{IN}$.

Demonstração:

Usando a indução, vamos provar que V_n é finita e convexa em $n \in \mathbb{IN}$.

Supõe-se que para $n=0$, V_n é finita e convexa

Supõe-se ainda que V_{n-1} definida em $[0,s]$ é convexa pelo Teorema 3.1.

Através do **Teorema 3.3**, $a \rightarrow w_n(s,a)$ assume em $[0,s]$ o máximo em $a=0$ ou $a=s$.
Daí, $V_n = \sup_{a \in D(s)} w_n(s,a) = \max(w_n(s,0), w_n(s,s)) < \infty$. As funções $s \rightarrow w_n(s,0)$ e $s \rightarrow w_n(s,s)$ são convexas, já que r e V_{n-1} são convexas. Logo $s \rightarrow f_n(s)$ é um maximizador. \square

4. Assume-se que r e V_0 são côncavas. Então,

i) $s \rightarrow V_n(s)$ e $a \rightarrow w_n(s,a)$, $s \in S$, são côncavas.

ii) Se existe um menor maximizador f_n no estágio n , então $s \rightarrow f_n(s)$ é crescente.

Observação:

Se r e V_0 são contínuas em $s=0$ e $s=S_{\max}$, então o menor maximizador $f_n(s)$ no estágio n , pode ser encontrado maximizando as funções $a \rightarrow w_n(s,a)$ em $[0, s]$,

$S \in [0, S_{\max}]$.

Demonstração

i) Vamos demonstrar que $V_n(s)$ e $a \rightarrow w_n(s,a)$, são côncavas. Para tal, vamos supõe-se que $V_{n-1}(s)$ é côncava. Daí, $(s,a) \rightarrow V_{n-1}(s,a)$ é côncava. Como r é côncava, e

$(s,a) \rightarrow w_n(s,a)$, no conjunto D , pelo **Teorema 3.1**. Logo $a \rightarrow w_n(s,a)$ é côncava.

$s \rightarrow \text{Sup } w_n(s,a)$, é finita e côncava pelo **Teorema 3.8**, como $D(s) = [0, s]$ é limitada. Pelo **Teorema 3.1**, segue que V_n é côncava. \square

ii) Para um n fixo tem-se $V_n(s) = \text{Sup}_{a \in [0,s]} \{r(s,a) + \beta E[V_{n-1}(T(a,X))]\}$

$$:= \text{Sup}_{a \in [0,s]} \{w_n(s,a)\}.$$

Usando o **Teorema 3.10**, temos que provar que,

$$w(s,a) - w(s,a') \leq w(s',a) - w(s',a') \quad (a)$$

Substituindo pelos correspondentes valores, tem-se

$$w_n(s,a) = r(s,a) + \beta E[V_{n-1}(T(a,X))] \quad (1)$$

$$w_n(s,a') = r(s,a') + \beta E[V_{n-1}(T(a',X))] \quad (2)$$

$$w_n(s',a) = r(s',a) + \beta E[V_{n-1}(T(a,X))] \quad (3)$$

$$w_n(s',a') = r(s',a') + \beta E[V_{n-1}(T(a',X))] \quad (4)$$

substituindo em (a) por 1, 2, 3 e 4 tem-se

$$r(s,a) + \beta E[V_{n-1}(T(a,X))] - r(s,a') - \beta E[V_{n-1}(T(a',X))] \leq r(s',a) + \beta E[V_{n-1}(T(a,X))] - r(s',a') - \beta E[V_{n-1}(T(a',X))]$$

$$r(s,a) - r(s,a') \leq r(s',a) - r(s',a') \quad (b)$$

Como $s < s'$ e $a < a'$ e supondo que

$s' = s + 1$ e $a' = a + 1$, substituindo em b tem-se

$$2r(s,a) \leq r(s+1,a) + r(s,a+1)$$

pela concavidade de r , a desigualdade anterior é válida, o que significa que w_n tem diferenças crescentes e logo a função $s \rightarrow f_n(s)$ é crescente. \square

5. Se r e V_0 são contínuas, então $s \rightarrow V_n(s)$ é contínua.

Demonstração

Vamos supor que $s \rightarrow V_k(s)$ é contínua, então

$(s,a) \rightarrow w_n(s,a) := V_k(s) = \text{Sup}_{a \in [0,s]} \{r(s,a) + \beta E[V_{k-1}(T(a,X))]\}$ é contínua.

Usando o **lema 3.4**, com $n=2$, $d(s) := s$, vê-se que $\text{Sup}_{a \in D(s)} \{w_n(s,a)\}$ é contínua e portanto, $V_{k+1}(s) = \text{Sup}_{a \in [0,s]} \{r(s,a) + \beta E[V_k(T(a,X))]\}$, mostra que $s \rightarrow V_{k+1}(s)$ é contínua. \square

3.4.3 Estudo da Distribuição do estado final do sistema

No modelo em estudo, supõe-se a existência de factores ambientais aleatórios (distúrbâncias) que influenciam o crescimento da população, daí que não se pode prever qual será o nível populacional no fim do planeamento. É de interesse,

portanto, a informação do tipo qual a probabilidade do nível populacional cair num certo intervalo escolhido; e como é que essa distribuição varia em relação ao nível de segurança s_{\min} e ao nível populacional inicial s_0 .

Para este estudo, define-se uma função de transição,

$$s_{n+1} := T(s_n, x_{n+1}) = a_n \times x_{n+1} \quad (*)$$

onde a_n designa o nível populacional que resta depois da exploração do recurso e x_{n+1} é o valor da perturbância. Assume-se que a variável aleatória X obedece a distribuição uniforme contínua no intervalo $[0; 2]$. Se a variável aleatória X tomar valores no intervalo $[0; 1)$, significa que houve uma diminuição do tamanho (nível populacional). Se a variável aleatória X tomar o valor 1, significa que o nível populacional manteve e se a variável aleatória X tomar valores no intervalo $(1; 2]$, significa que o nível populacional aumentou.

Para simular os possíveis estados finais fixa-se um estado inicial s_0 . O s_N ao fim de N períodos é uma variável aleatória cujos valores serão calculados segundo a equação seguinte:

$$S_N := T(a_{n-1}, x_n) = a_{n-1} \times x_n \quad (**)$$

Fazendo uma realização da variável X e a partir do estado inicial s_0 fixo, na matriz dos maximizadores extrai-se o vector das acções óptimas e usando a equação (**), simula-se o primeiro estado possível final. Os x_i 's ($i=1, 2, \dots, n$), valores da variável

aleatória X são gerados aleatoriamente, usando a função RANDOM. Mas como o interesse é obter uma série de possíveis valores do estado final, repete-se a experiência um número r de provas independentes.

A partir da série de valores possíveis da variável aleatória em estudo, pode-se determinar a ocorrência de cada um dos valores (frequência absoluta).

Designa-se por nr_i a frequência absoluta dos valores da variável aleatória s_N . A razão entre a frequência absoluta e o número de valores do estado final simulados dá a frequência relativa e esta, aproxima-se à probabilidade de ao fim de N períodos o nível populacional ser igual ao valor cuja frequência absoluta é nr_i , matematicamente tem-se,

$$\text{freqrel} = nr_i / r \quad (***)$$

onde freqrel é a frequência relativa

Através da execução do programa (ANEXO A) pode-se obter os seguintes resultados:

Considerando que o estado inicial $s_0 = 3$ e gerando uma sequência de variáveis aleatórias, tem-se:

- Possíveis estados finais:

2, 0, 1, 2, 1, 1, 1, 1, 2, 1

A frequência absoluta, Gmurman (1983), de cada valor será:

$$\text{FREQ}[0] = 1$$

$$\text{FREQ}[1]= 6$$

$$\text{FREQ}[2]= 3$$

Com o auxílio da distribuição de frequências, é possível conhecer a probabilidade da cada valor (usando a fórmula (***)).

$$\text{frel}[0]= 0.10$$

$$\text{frel}[1]= 0.60$$

$$\text{frel}[2]= 0.30$$

O gráfico 3 (ANEXO B), mostra a distribuição de frequências relativas dos possíveis estados finais.

CAPÍTULO IV

4. Discussão

Da posse do programa computacional desenvolvido, e à luz do que seria na prática, analisa-se o modelo para gestão de recursos renováveis com um nível crítico s_{\min} .

1. Num problema prático, os ganhos crescem quanto mais forem os períodos de decisão. No modelo, as funções de valor $V_n(s)$ crescem quanto maior for o horizonte de planeamento. Isto é suportado pelos Teoremas 3.5 e 3.6. Portanto, este resultado vai de acordo com a intuição humana.
2. O facto dos maximizadores serem crescentes em relação a s , é muito importante, pois há uma tendência de conservação do recurso para que não se extinga.
3. O crescimento populacional é influenciado por factores ambientais (disturbâncias) que ocorrem ao acaso e que influenciam no seu crescimento. Nesta circunstância, já que esses factores são descritos através de uma distribuição probabilística (uniforme contínua), deve-se observar os critérios de geração de variáveis aleatórias contínuas. Interessa aqui referenciar que os critérios de geração das variáveis aleatórias, que neste caso representam o factor perturbador do crescimento da população, influenciam de facto no resultado do modelo em estudo, concretamente nos ganhos obtidos ($V_n(s)$). Em particular, a escolha da semente para a geração de números aleatórios deve ser cuidada.

4. Na simulação do estado final do sistema, ao fim de N períodos de decisão, verifica-se uma relação entre o nível crítico e a distribuição probabilística, o que leva a concluir que os valores gerados aleatoriamente dependem do estado inicial s_0 fixado. Em termos práticos, isto significa que quanto maior for o nível de segurança de recurso, menor será a probabilidade da sua extinção.

CAPÍTULO V

5. Conclusões e recomendações

5.1 Conclusões

Dentro das limitações da pesquisa e com base nas referências bibliográficas, a metodologia de maximização do ganho tem como principais conclusões:

- Aplicando o teorema básico para um MC com perturbações aleatórias i.i.d. (Iteração de valor), verifica-se que as funções de valor $V_n(s)$ crescem com o número de períodos de decisão;
- Os maximizadores são crescentes em n ;
- Na simulação do estado final do sistema a distribuição probabilística da Variável Aleatória depende do estado inicial.

Portanto, os resultados deste estudo, correspondem com os resultados obtidos no trabalho feito em 1997 nesta área (Cruz, 1997).

5.2 Recomendações

Para o desenvolvimento de futuros trabalhos relacionados com o modelo para gestão de recursos renováveis com estado crítico s_{min} recomenda-se:

- A investigação de um MC com perturbações aleatórias i.i.d., obedecendo a outras distribuições;
- O estudo deste modelo considerando o caso não estacionário;

- Que seja feito um estudo comparativo deste modelo com os demais que não tenham um nível de recurso crítico.

CAPÍTULO VI

6.1 Referências Bibliográficas

1. Bellman R. (1957). Dynamic Programming, 340 pp. London.
2. Bellman R. & Dreyfus E. (1962). Applied Dynamic Programming,. New Jersey.
3. Bertsekas D.P. (1987). Dynamic Programming, "Deterministic and Stochastic Models, New Jersey.
4. Gmurman, V.E., (1983). Teoria de Probabilidades e Estatística Matemática, Moscou.
5. Nemhauser G.L. (1966). Dynamic Programming, U.S.A.
6. Cruz, B. (1997). Gestão de Recursos Renováveis. Tese de Licenciatura. Maputo, UEM.
7. Nhavane, I., (1996). A sequential Allocation Model with random opportunities. Licenciatura Thesis. Maputo, UEM.
8. Clarke, G.M. & Cooke, D. (1992). A Basic Course in Statistics, 3rd Edition, Great Britain.
9. Crawshaw, J. & Chambers, J. (1990). A Concise Course in A-Level Statistics, 2nd Edition, Great Britain.
10. Gottfried, B.S. (1988). Programação em PASCAL. Portugal, Editora McGraw-Hill.
11. O'Keefe, R. & Davies, R. (1989). Simulation Modelling with PASCAL, Prentice Hall International (UK).

12. Law, A.M. & Kelton, W.D. (1991). Simulation Modelling & Analysis, 2nd Edition, McGraw-Hill International Editions.
13. Hinderer, K. (1984). On the structure of solutions of stochastic dynamic programs, in: Proceeding of the Seventh Conference on Probability Theory, Brasov 1984.
14. Hinderer, K. (1991). Increasing Lipschitz continuous maximizers of some Dynamic Programs. Annals of Operational Research Nr. 29, p. 565-585.
15. Hinderer, K. (1993). Lecture Notes on Stochastic Dynamic Programming.
16. Jaquette, David L. (1972). A Discrete Time Population Control Model.
17. Fiadeiro, P. M. (1992). Análise Operacional. Portugal, Serviço de Publicações Escolares.
18. Roberts, A. W. & Varberg, D. E. (1973). Convex Functions. Academic Press.

CAPÍTULO VII

7.1 ANEXO A – Programa Computacional

PROGRAM Gestao_recursos_renovaveis;

USES crt, graph, cp3;

CONST

Smax = 10; {Limite máximo do recurso existente}
n = 10; {Número máximo de iterações}
nr = 10; {Número máximo de iterações na simulação do estado
final do sistema}
smin = 2; {Nível de segurança do recurso}
max_stream_num = 32;

TYPE

Matriz = array[0..n, 0..Smax] of real;
Matriz1 = array[1..n, 0..Smax] of integer;
vector = array[0..n] of real;
vector1 = array[1..nr] of integer;
vector2 = array[0..smax] of real;
Stream_num = 1..max_stream_num;
cardinal = 0..maxint;
X = array[0..n] of real;

VAR

Aopt : Matriz1; {Acções óptimas}
V : Matriz; {Valores de ganho}
beta : real; {Factor de desconto}
N_ : integer; {Horizonte de planeamento}
a : integer; {Nível populacional que resta depois da exploração do recurso}
s : integer; {Nível populacional disponível num certo período}
original_seeds, seeds: array[stream_num] of cardinal;
xi : X; {Valores da disturbância}
aux : real;
index,i : integer;
freqrel : vector; {sequência de frequências relativas}
s0 : integer; {estado inicial}
max,xn : real;
color : integer;
sN : vector1; {sequência de estados finais na Simulação}
freqab : vector2; {sequência de frequências absolutas}
univar : vector; {sequência de valores da realização da variável X}
minimo, maximo : real;
fim : boolean;
y : real;

Procedure DESCRICAO;

begin

clrscr;

gotoxy (10,10);

writeln (' MODELO PARA A GESTAO DE RECURSOS RENOVAVEIS :');

gotoxy(10,11);

writeln ('-----');

writeln;

gotoxy(7,13);

writeln ('O programa calcula a sequência de acções óptimas e a função');

gotoxy(7,14);

writeln ('do ganho, com a existência de um nível de segurança, smin.');

gotoxy(23,24);

writeln('Pressione Enter para continuar ');

readln

end;

(*-----*)

Procedure MENU (var index: integer);

begin

clrscr;

writeln(' MENU ');

writeln;

writeln(' 1. Entrada de dados');

```
writeln;  
writeln(' 2. Tabela de Função de valor Vn(s)');  
writeln;  
writeln(' 3. Tabela de Maximizadores fn(s)');  
writeln;  
writeln(' 4. Saída gráfica de Vn(s)');  
writeln;  
writeln(' 5. Saída grafica de Vn com diferentes valores de s');  
writeln;  
writeln(' 6. Saída grafica de fn(s)');  
writeln;  
writeln(' 7. Simulação do estado final');  
writeln;  
writeln(' 8. Histograma de Frequências');  
writeln;  
writeln(' 9. Saída');  
writeln;  
write ('opção: ');  
readln(index);  
end;
```

(*-----*)


```
Procedure ENTRADA_DADOS (var m:integer; var beta: real);  
begin  
  clrscr;  
  repeat  
    writeln('Introduza o horizonte de planeamento N_ <= 10');  
    readln(N_);  
    writeln('Introduza o factor de desconto 0 < beta <= 1');  
    readln(beta);  
  until(N_ > 0) and (N_ <= n) and (beta > 0) and (beta < 1);  
end;
```

(*-----*)

```
function VALIDAR : boolean;  
begin  
  if(N_ > 0) and (N_ <= n) and (beta > 0) and (beta <= 1) then  
    VALIDAR := true  
  else  
    VALIDAR := false  
end;
```

(*-----*)

{Geração de números aleatórios no intervalo [0.0, 1.0]}

function RND (s: stream_num) : real;

const

mult = 3993;

divid = maxint;

add = 1;

var aux : integer;

begin

aux := seeds[s]*mult+add;

seeds[s] := aux mod divid;

if seeds[s]<0 then seeds[s] := seeds[s]+maxint+1;

RND := seeds[s]/divid;

end;

(*-----*)

{Retorna as sementes para a geração de números aleatórios}

procedure MAKE_STREAMS;

var i : stream_num;

begin

for i := 1 to max_stream_num do

original_seeds[i] := i*1000+7;

seeds := original_seeds;

end;

(*-----*)

{Permite gerar variáveis aleatórias uniformemente distribuídas no intervalo [a,b]}

```
function UNIFORM (a,b: real; s: stream_num) : real;
begin
  uniform := a+(b-a)*rnd(s);
end;
```

(*-----*)

{Ordena uma sequência de valores em ordem crescente}

```
procedure BUBBLE_SORT (var xi: X);
var aux : real;
    i,j : integer;
begin
  for i := 2 to n do
    for j := 1 to i-1 do
      if (xi[j] > xi[i]) then
        begin
          aux := xi[i];
          xi[i] := xi[j];
          xi[j] := aux;
        end;
    end;
end;
```

(*-----*)

{Gera e ordena variáveis aleatórias uniformemente distribuídas no intervalo
[0,0 ; 2,0] usando a 32ª semente}

Procedure GERA_VARIAVEL;

begin

MAKE_STREAMS;

for i := 0 to n do

begin

xi[i] := abs(uniform(0,0 ; 2,0,seeds[32]));

end;

BUBBLE_SORT(xi);

end;

(*-----*)

{ Calculo de $V_n(s) = [r(s,a) + \text{beta} * E[V_{n-1}(T(a,X))]]$ }

function Vns (i,s,a: integer) : real;

var si,j : integer;

Integral : real;

xn,y : real;

begin

GERA_VARIAVEL;

for j := 0 to n do

begin

```
xn := xi[j];
y := a*xn;
si := round(y);
Integral := 1/2*V[i-1,si]; { E[Vn-1(T(a,X))] }
end;
Vns := sqrt(s-a) + beta * Integral;
end;

(*-----*)
{Simulação do estado final do sistema}

procedure SIMULACAO (var sN: vector1; var univar: vector);
var i,j,s : integer;
    xn,si : real;
begin
    RANDOMIZE; {Procedimento embutido no computador, que inicializa um
gerador de números aleatórios, com um valor aleatório}
    for i := 1 to nr do
        begin
            s := s0;
            for j := 1 to n do
                begin
                    xn := (RANDOM(N_))/10; {A função RANDOM retorna um nº aleatório}
                    si := Aopt[n-j+1,s]*xn; {estados aleatórios}
                end;
            end;
        end;
end;
```

```
s := round(si);
if s > smax then s := smax;
  sN[i] := s; {Sequência de estados finais}
  univar[i] := xn;
end
end;

(*-----*)
{Cálculo de frequências relativas e absolutas}
procedure FREQUENCIA (VAR freqab: vector2; var freqrel: vector);
var
  i,j : integer;
begin
  for i := 0 to nr do
    freqab[i] := 0;
    for i := 1 to nr do
      begin
        j := sN[i];
        freqab[j]:=freqab[j]+1;
      end;
      readln;
    writeln('Frequencias absolutas ');
    writeln;
    for i := 0 to n do
      writeln('freqab['i,']' = ',freqab[i]:0);
```

```
readln;
writeln('Frequencias relativas');
writeln;
for j := 0 to nr do
  begin
    freqrel[j] := freqab[j]/nr;
    writeln('frel['j,']' = ',freqrel[j]:1:2);
  end;
writeln;
readln;
end;
```

(*-----*)

```
procedure Ds (var V: matriz); {Iteração de valor}
var i,s,a : integer;
    valor : real;
begin
  for s := 0 to smax do V[0,s] := 0;
  for i := 1 to n do
    begin
      for s := 0 to smax do
        begin
          if s < smin then
```

```
begin
  a := s;
  if s = 0 then V[i,s] := beta * V[i-1,0]
  else V[i,s] := Vns(i,s,a);
  Aopt[i,s] := a
end
else
begin
  aux := 0;
  for a := smin to s do
    begin
      valor := Vns(i,s,a);
      if valor >= aux then
        begin Aopt[i,s] := a;
          aux := valor
        end;
      V[i,s] := aux;
    end;
  end;
end;
end;
end;
end;

(*-----*)
```



```
procedure SAIDA_Vns (V : matriz);
var i,j : integer;
begin
  clrscr;
  gotoxy(15,2);
  write('TABELA DO GANHO ESPERADO MAXIMO Vn(s) ');
  gotoxy(5,4);
  write('s');
  j := 1;
  for i := 0 to smax do
  begin
    j := j+7;
    gotoxy(j,4);
    write(i)
  end;
  gotoxy(4,5);
  write('n');
  for i := 0 to n do
  begin
    gotoxy(4,6+i);
    write(i);
  end;
  gotoxy(6,6);
  for i := 0 to n do
```

```
begin
  gotoxy(6,i+6);
  for j := 0 to smax do
    write(V[i,j]:5:2,' ');
  writeln;
  writeln;
end;
readln;
end;
```

(*-----*)

```
procedure SAIDA_fns (Aopt : matriz1);
var i,j : integer;
begin
  clrscr;
  gotoxy(25,2);
  write('TABELA DOS MAXIMIZADORES fn(s) ');
  gotoxy(4,4);
  write('s');
  gotoxy(6,4);
  for i := 0 to smax do
    write(i,' ');
    gotoxy(3,5);
    write('n');
```

```
for i := 1 to n do
begin
  gotoxy(3,5+i);
  write(i);
end;
for i := 1 to n do
begin
  gotoxy(6,5+i);
  for j := 0 to smax do
    write(Aopt[i,j], ' ');
  writeln;
  writeln;
end;
readln;
end;

(*-----*)
```

```
Procedure IMPRESSAO;
var
  cc: char;
begin
  CpStatlin ('Hardcopy (Y/N) ?');
  cc := upcase(readkey);
  if cc = 'Y' then
```

```
begin
  cpstatlin("");
  cpHardcopy(true,HPLJII,Landscape,"");
end;
end;
```

(*-----*)

```
procedure MAIOR;
  var i,j : integer;

begin
  minimo := V[0,0];
  maximo := V[0,0];
  for i := 0 to n do
    for j := 1 to smax do begin
      if (V[i,j] < minimo) then
        minimo := V[i,j];
      if (V[i,j] > maximo) then
        maximo := V[i,j]
    end;
  end;
```

(*-----*)

```
procedure GRAFICO_Vns;
var code : integer;
begin
  MAIOR;
  cpinit(14);
  Rcmask(1);
  Rcrange(0,0,smax+1,maximo+0.5);
  Rctitle('GRAFICO DA FUNCAO Vn(s) COM DIFERENTES VALORES DE n
');
  RcSetDecPlaces(2);
  Rcxax(5,num,'s');
  Rcyax(5,num,'Vn(s)');
  for a := 0 to smax do
  begin
    cpcolor(a+1,0);
    for s := 0 to n-1 do
    begin
      Rcmask(s,V[a,s],ord('*'));
      Rcdraw(s,V[a,s],s+1,V[a,s+1]);
    end;
    if (a < 10) then
    begin
      code := ord(a+49);
      Rcmask(smax-0.1,V[a,smax]+0.13,code);
    end
  end
```

```
else  
begin  
  code := ord(49);  
  Rcmak(smax-0.25,V[a,smax]+0.03,code);  
  Rcmak(n-0.1,V[a,n]+0.03,a+38);  
end;  
end;  
  IMPRESSAO;  
readln;  
cpexit  
end;
```

(*-----*)

```
procedure GRAFICO_Vsn;  
var  
  i,code : integer;  
begin  
  MAIOR;  
  cpinit(14);  
  Rcmak(1);  
  Rcrange(0,minimo,smax,maximo+0.5);  
  Rctitle('GRAFICO DO GANHO MAXIMO Vn(s) COM DIFERENTES  
VALORES DE s');  
  Rcxax(6,num,'n');
```

```
Rcyax(6,num,'Vn(s)');
for i := 0 to n-1 do
begin
  for s := 0 to n-1 do
  begin
    if (i < 10) then
    begin
      code := ord(s+49);
      cpcolor(s+1,0);
      Rcmak(i,V[i,s],ord('*'));
      Rcdraw(i,V[i,s],i+1,V[i+1,s]);
      Rcmak(n-0.25,V[n,s]+0.05,code);
    end
  else
  begin
    code := ord(50);
    Rcmak(n-0.25,V[n,smax]+0.05,code);
    Rcmak(n-0.1,V[n,smax]+0.3,i+38);
  end;
end;
end;
end;
  IMPRESSAO;
readln;
cpexit
end;
```

(*-----*)

```
procedure GRAFICO_fns;
var cc : char;
begin
  cpinit(14);
  Rcmask(1);
  Rcrange(-1,-1,smax,smax+1);
  Rctitle('GRAFICO DE MAXIMIZADORES fn(s) ');
  Rcxax(6,num,'s');
  Rcyax(7,num,'fn(s)');
  for a := 1 to n do
  begin
    cpcolor(a,0);
    for s := 0 to n-1 do
    begin
      Rcmask(s,Aopt[a,s],ord('*'));
      Rcdraw(s,Aopt[a,s],s+1,Aopt[a,s+1]);
    end;
  end;
  IMPRESSAO;
  readln;
  cpexit
end;
```

(*-----*)


```
Procedure HISTOGRAMA(frel:vector);
var
  xi1,yi1 : longint;
  xi2,yi2 : longint;
  xr,yr   : real;
  i       : integer;
  us      : string[10];
begin
  cpinit(14);
  Rcmask(1);
  Rcrange(0-1, 0, 11,1.1);
  Rctitle('Histograma de Frequências ');
  Cpcolor(red, black);
  Rcframe;
  RcSetDecPlaces (0);
  Rcxax(5,num,'freqab');
  RcSetDecPlaces (2);
  Rcyax(4,num,'freqrel');
  Cpcolor (cyan, black);
  SetFillStyle (1,1);
  for i := 0 to nr do
  begin
    xr := freqrel[i];
    RcScaleR2W (Screen, i-0.15, 0, xi1, yi1);
    RcScaleR2W (Screen, i+0.15, xr, xi2, yi2);
```

```
Bar3D (xi1, yi1, xi2, yi2,(xi2-xi1+1) div 2,topon);  
RcMoveTo (i-0.4, xr+0.06);  
Str (xr:5:1,us);  
OutText (us);  
end;  
    IMPRESSAO;  
readln;  
cpexit  
end;
```

(*-----PROGRAMA PRINCIPAL-----*)

```
Begin  
    fim := false;  
    DESCRICAO;  
    repeat  
        MENU(index);  
        case index of  
            1 : begin  
                ENTRADA_DADOS (N_,beta);  
                While (not VALIDAR) do  
                    begin  
                        Writeln('Dados invalidos');  
                        readln;  
                        ENTRADA_DADOS (N_,beta);
```

```
        end;
        Ds (V);
    end;
2: SAIDA_Vns (V);
3: SAIDA_fns (Aopt);
4: GRAFICO_Vns;
5: GRAFICO_Vsn;
6: GRAFICO_fns;
7: Begin
    writeln ('Introduza o valor de s0:');
    readln(s0);
    SIMULACAO (sN,univar);
    writeln('Vector dos possíveis estados finais');
    for i := 1 to nr do
        write(sN[i], ' ');
        writeln;
        readln;
        writeln('Vector dos Valores da Realizacao da Variavel X');
        for i:= 1 to n do
            write(univar[i]:2:2, ' ');
            writeln;
            readln;
            FREQUENCIA(freqab,freqrel);
        end;
    end;
```

8: HISTOGRAMA(freqrel);

9 : fim := true;

end;

until fim = true;

end.

7.2 ANEXO B - Gráficos

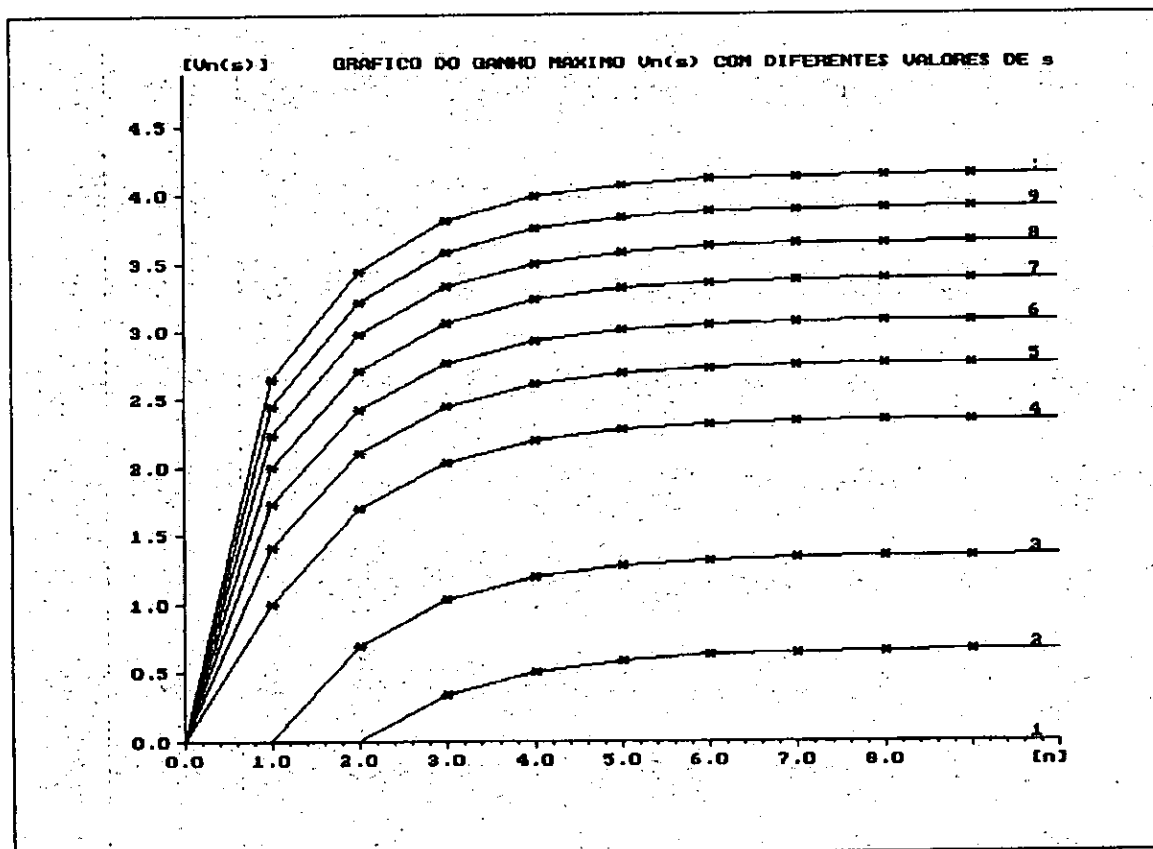


Gráfico 1: Função de $n \rightarrow V_n(s)$, com $\beta = 0.98$

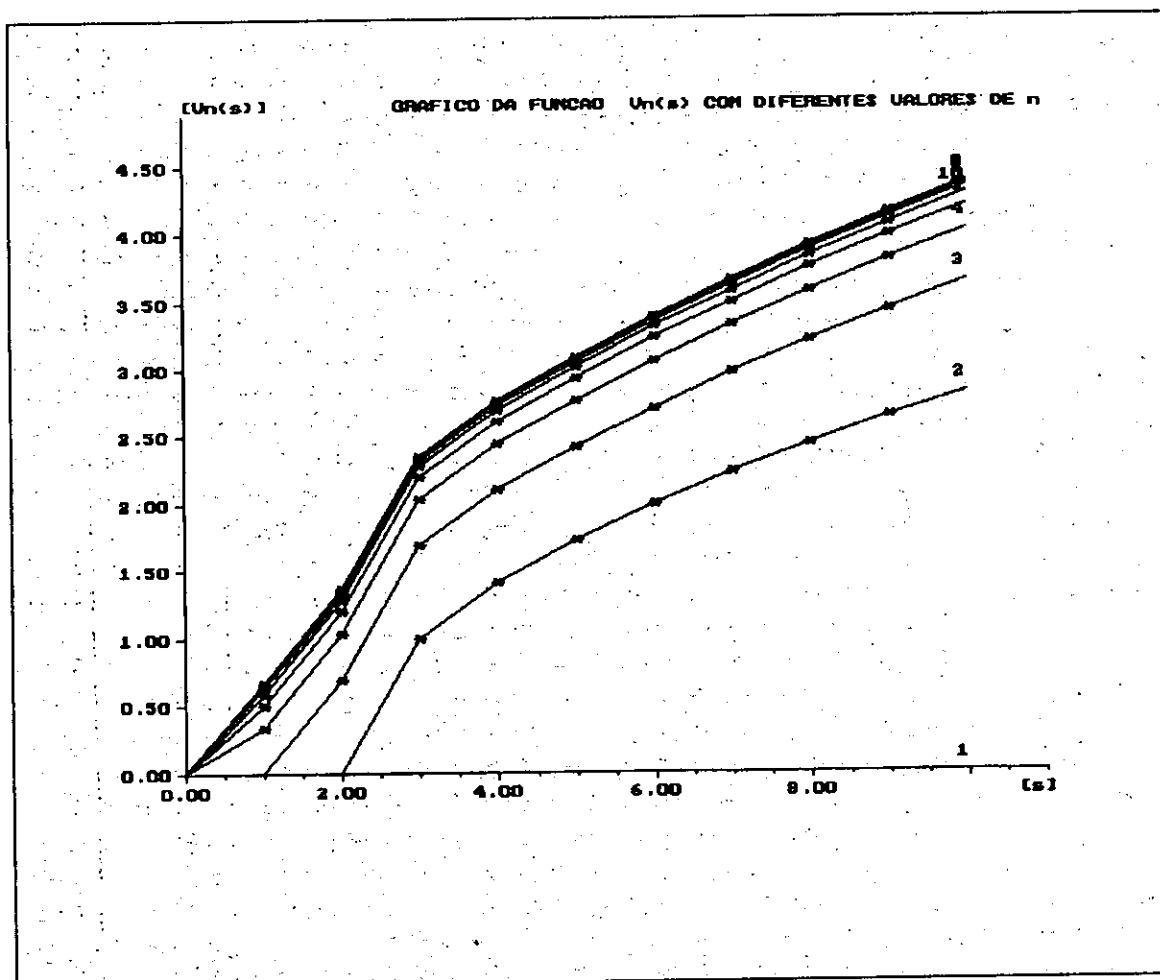


Gráfico 2: função de $s \rightarrow V_n(s)$ com $\beta = 0.98$

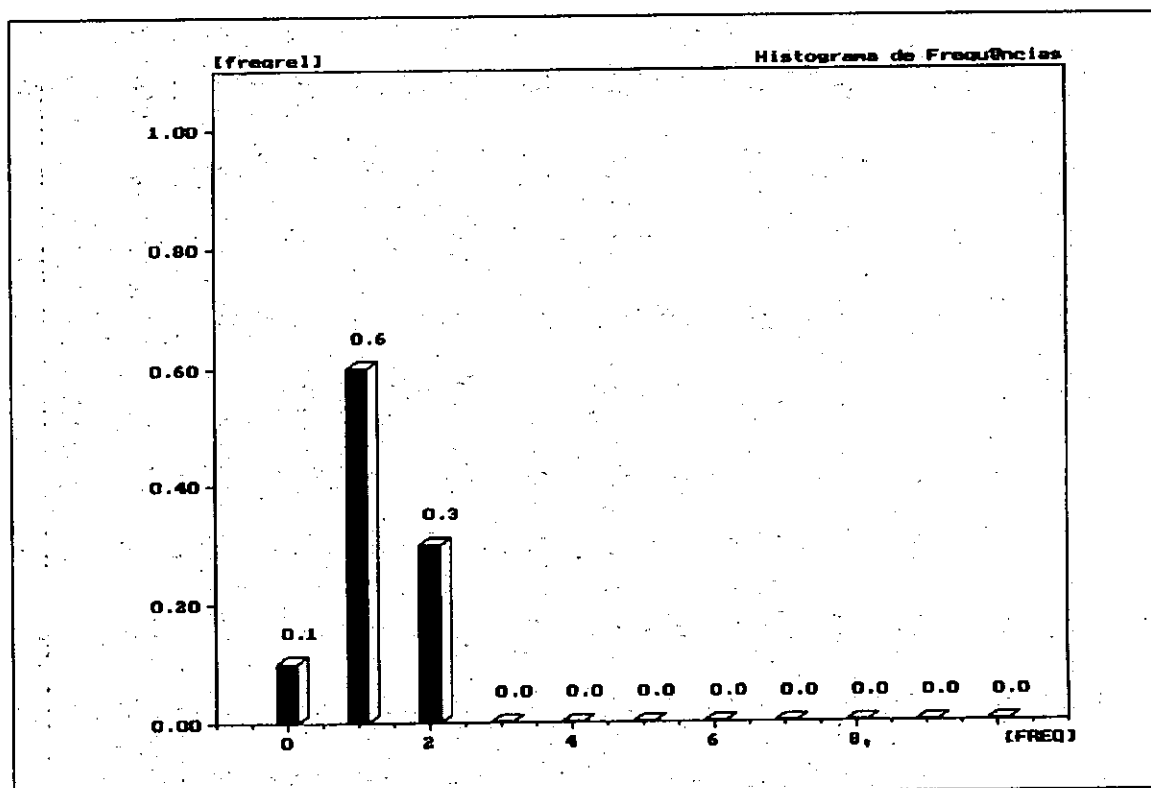


Gráfico 3: Distribuição das frequências dos possíveis estados finais, com $s_0 = 3$

7.3 ANEXO C – Notações usadas e seu significado

- $(a_n)_0^{N-1} = (a_0, a_1, \dots, a_{n-1})$;
- $(f_n)_N^1 = (f_N, f_{N-1}, \dots, f_1)$;
- $\mathbb{R}_+ = [0, +\infty[$;
- $\mathbb{R}^+ =]0, +\infty[$;
- $[0, 0 ; 1, 0) = [0, 0 ; 1, 0[$;
- $(1, 0 ; 2, 0] =]1, 0 ; 2, 0]$;
- $(0, 1) =]0, 1[$;
- $X \sim U(0, 2)$ Variável aleatória X distribuída uniformemente no intervalo $[0, 2]$;
- $E[X]$ Esperança matemática de uma variável aleatória;
- $[0, S_{\max}]^2$ é o produto cartesiano de $S \times A = [0, S_{\max}] \times [0, S_{\max}]$;
- Para funções S em A usa-se a notação $\begin{cases} x \rightarrow f(x) \\ S \rightarrow A \end{cases}$
ou $f(\cdot)$. $f(x)$ denota somente o valor de f em x, e não a função f;
- $D(\cdot) = D(s)$, onde $s \in \mathbb{R}$;
- $T(\cdot, a) = T(s, a)$, onde $s \in \mathbb{R}$;
- $r(\cdot, a) = r(s, a)$, onde $s \in \mathbb{R}$;
- $w(s, \cdot) = w(s, a)$, onde $a \in \mathbb{R}$;