RENATO DE SOUZA GOMES

O ALGORITMO DE ELIMINAÇÃO PARA O PROBLEMA DA PARADA ÓTIMA EM CADEIAS DE MARKOV

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2006

RENATO DE SOUZA GOMES

O ALGORITMO DE ELIMINAÇÃO PARA O PROBLEMA DA PARADA ÓTIMA EM CADEIAS DE MARKOV

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Otimização Combinatória

Orientador:

Prof. Lucas Monteiro Chaves

LAVRAS
MINAS GERAIS – BRASIL
2006

..

Ficha Catalográfica

Gomes, Renato de Souza

O Algoritmo de Eliminação Para o Problema da Parada Ótima em Cadeias de Markov/Renato de Souza Gomes. Lavras - Minas Gerais, 2006. 62p : il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Problema da Parada Ótima. 2. Cadeias de Markov. 3.Problema da Secretária. I. GOMES, R. S. II. Universidade Federal de Lavras. III. Título.

RENATO DE SOUZA GOMES

O ALGORITMO DE ELIMINAÇÃO PARA O PROBLEMA DA PARADA ÓTIMA EM CADEIAS DE MARKOV

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 21 de setembro de 2006

____
Msc. Prof. Deive Ciro de Oliveira

____
Msc. Devanil Jaques de Souza

____
Prof. Dr. Lucas Monteiro Chaves

LAVRAS
MINAS GERAIS – BRASIL
2006

Aos Meus pais Aguinaldo Gomes de Almeida e Rita de Souza Almeida.

A minha irmã Valéria Gomes de Almeida.

A meu irmão André de Souza Gomes.

Dedico.

AGRADECIMENTOS

A Deus pela vida e proteção; aos meus pais Aguinaldo Gomes de Almeida e Rita de Souza Almeida pelo apoio incondicional, pelo amor e dedicação; a meu irmão André de Souza Gomes e minha irmã Valéria Gomes de Almeida pela amizade e companhia.

Ao orientador Lucas Monteiro Chaves**,** pela paciência e competência com que conduziu meus estudos.

Aos familiares e amigos pelo afeto e carinho.

Aos professores que participaram de toda minha trajetória acadêmica.

Resumo

Este trabalho apresenta um estudo didático sobre o problema da parada ótima (OSP) em cadeias de Markov. incluindo o método de redução de estados, a equação de Bellman e o método da iteração na solução do OSP. Mostra também uma solução ótima para o problema da secretária. O ambiente computacional utilizado é o software livre R, em razão de seus recursos estatísticos.

Palavras-chave: Parada Ótima, Cadeias de Markov, Problema da Secretária.

Abstract

This work presents a didactic study on the Optimal Stopping Problem (OSP) in Marcov chains. Including the method of reduction of states, the equation of Bellman and the method of the iteration in the solution of the OSP. It also snows the optimal solution for the problem of the secretary. The computational environment used is the free software R, in reason of its statistical resources.

Key words: Optimal Stoping, Markov Chain, Secretary Problem.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Em vários problemas práticos, como por exemplo, problemas no campo financeiro, é de grande interesse saber qual é o momento em que devemos parar um investimento para receber o maior retorno possível. Esse é um problema matemático conhecido como parada ótima, que nos permite encontrar uma regra de parada que maximiza o lucro. Este trabalho estuda o problema de parada ótima aplicada às cadeias de Markov.

Uma cadeia de Markov é um processo aleatório onde os estados anteriores são irrelevantes para a predição dos estados futuros, dado que o estado atual é conhecido. Uma maneira simples de visualizar um tipo específico de cadeia de Markov é através de um grafo dirigido no qual cada aresta é rotulada com as probabilidades de transição de um estado a outro sendo estes estados representados por nós. Se um sistema está no estado y no tempo n, então a probabilidade de que ele se mova para o estado x no tempo $n + 1$ não depende de n, e somente depende do estado atual y em que o sistema se encontra.

A solução computacional de problemas de parada ótima é em geral de custo elevado e, se o número de estados é grande, pode ser muito cara.

Neste trabalho foi estudado um algoritmo de eliminação, que consiste em eliminar os estados onde certamente não é ótimo parar. Neste caso a complexidade computacional é reduzida obtendo resultados em tempos menores.

O algoritmo foi proposto por Sonin, (1999) e publicado em um extenso artigo com várias aplicações. Uma aplicação ao problema clássico da secretária é desenvolvida.

O trabalho está estruturado da seguinte forma:

- Capítulo 1: Contextualização do problema tratado, motivação pelo qual o trabalho foi realizado e objetivo.

- No Capítulo 2: Definição de processos estocásticos.

- No Capítulo 3: Apresentação das Cadeias de Markov de maneira didática.

- No Capítulo 4: Uma revisão de literatura sobre o problema da parada ótima.

- No Capítulo 5: O problema da secretária e uma solução ótima para o mesmo

- No Capítulo 6: Todos os procedimentos metodológicos do trabalho.

- No Capítulo 7: Apresentação de alguns resultados obtidos com o estudo do problema da parada ótima bem como sua aplicação na solução do problema da secretária.

- No Capítulo 8: Conclusões obtidas nesse trabalho.

- No Capítilo 9: Referências bibliográficas.

- No Capítulo 10: Anexos.

2 PROCESSOS ESTOCÁSTICOS

Os processos estocásticos são famílias arbitrárias de variáveis aleatórias X_t indexadas por t onde $t \in T$ (sendo T um conjunto qualquer, geralmente $\mathbb{N}$+). O índice t pode ser visto como um indexador de tempo ou espaço. Os resultados assumidos por X_t (contra-domíno de X_t) são dentados como conjunto dos estados do processo. Os processos estocásticos são definidos de maneira estritamente formal através do Teorema de kolmogorov(Teorema Fundamental dos Processos Estocásticos) apresentado abaixo (Biswas, 1995).

Definição 2.1 Se a função distribuição de probabilidade conjunta das variáveis aleatórias $Xt_1, Xt_2,..., Xt_n$ é conhecida para todo n enumerável positivo, e para todo conjunto de valores t_1, t_2,... t_n onde t_k qualquer, pertence a um conjunto T, podemos denotar o conjunto destas variáveis, { Xt_i }, por *Processo Estocástico*.

Exemplo 2.1 ***Processo Estocático.*** Um conjunto de variáveis aleatórias independentes da forma X_t, que possuem uma distribuição conjunta normal multivariada, onde t $\in$ $\mathbb{Z}$+.

Existem vários tipos de processos estocásticos. Como exemplos, temos os Processos de Poisson, Gaussianos, de Martingale (Biswas, 1995). Vamos utilizar neste trabalho um processo estocástico particular denominado Cadeia de Markov.

3 CADEIAS DE MARKOV

Em 1907, Markov definiu e investigou propriedades que são hoje conhecidas como processos de Markov. A principal característica dos processos de Markov é que o modo como toda a história passada afeta o futuro está completamente resumido no valor atual do processo.

Muitos dos processos que ocorrem na natureza e na sociedade podem ser estudados (pelo menos em primeira aproximação) como se o fenômeno estudado passasse, a partir de um estado inicial, por uma seqüência de estados, em que a transição de um determinado estado para o seguinte ocorre segundo certa probabilidade. No caso em que esta probabilidade de transição depende apenas do estado em que o fenômeno se encontra e do estado a seguir, o tal processo será chamado *processo de Markov* e uma seqüência de estados seguindo este processo será denominada uma *cadeia de Markov*. (Boldrini et al., 1980).

Uma cadeia de Markov, ou processo de Markov, é um processo no qual a probabilidade de um sistema estar em um determinado estado em um dado período de observação depende apenas do estado no período de observação anterior. Considere um sistema que, em cada instante, está em apenas um entre um número finito de estados. Por exemplo, o tempo em determinada região pode estar chuvoso ou seco; uma pessoa pode ser fumante ou não fumante, podemos freqüentar ou não uma universidade; vivemos em uma área urbana ou rural; pertencemos à classe baixa, média ou alta. Ao longo do tempo o sistema pode mudar de um estado para outro: vamos supor que o estado do sistema é observado em período fixo de tempo (por exemplo, uma vez por dia, em cada hora, e assim por diante). Em muitas aplicações conhecemos o estado atual do sistema e queremos prever o estado no próximo período de observação ou em algum período futuro. Podemos prever, muitas vezes, a probabilidade de o sistema estar em determinado estado em um período futuro de observação a partir de sua história pregressa (Kolman, 1999).

Isto é, (Ferrari, 1987), uma cadeia de Markov é um processo $\{X(n)\}$, sobre um espaço de estados S finito ou enumerável. A variável $X(n)$ indica a posição (aleatória) do processo no instante n. Estes processos têm a seguinte propriedade: a distribuição do salto em tempo $n + 1$ depende só da posição $X(n)$ e não de como se chegou nessa posição nem mesmo de n. Isso pode ser descrito por uma matriz $\mathbf{P}$ com entradas $P_{x,y} := P(X(n+1) = y \mid X(n) = x)$ = probabilidade de saltar a y dado que o processo está em x.

Supondo que um sistema tem n estados possíveis, $a_1, a_2, ..., na_,$ para cada $i = 1, 2, ...,$

n, j = 1, 2,..., n, seja p_{ij} a probabilidade que, se o sistema esteja no estado j em um determinado período de observação então ele estará no estado i no próximo período de observação; p_{ij} é chamado de uma probabilidade de transição. Alem disso p_{ij} se aplica a todos os períodos de tempo, isto é, não muda com o tempo.

Como p_{ij} é uma probabilidade, temos que ter:

$$0 \le p_{ij} \le 1 \ (1 \le i, j \le n).$$

Alem disso, se o sistema está no estado j em um determinado período de observações, então ele tem que estar em um dos n estados possíveis (ele pode permanecer no estado j) no próximo período de observação, Logo.

$$p_{1j} + p_{2j} + \dots + p_{nj} = 1.$$

É conveniente escrever as probabilidades de transição em uma matriz n x n, T = [p_{ij}], chamada matriz de transição. Da cadeia de Markov. Outros nomes para a matriz de transição são matriz de Markov, matriz estocástica e matriz de probabilidade. Os elementos da matriz **T** são não-negativos e a soma de cada uma de suas colunas, pela equação anterior é 1.

A matriz de das probabilidades de transição (matriz estocástica) é dada por:

$$\mathbf{T} = \begin{vmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{n3} \end{vmatrix}$$

(Observe que $p_{ij} \ge 0$, e que a soma de cada coluna deve ser 1.)

Os Exemplos e teoremas a seguir foram retirados do livro do kolman, (1999).

Exemplo 3.1 Suponha que o tempo em determinada cidade é chuvoso ou seco. Como resultado do grande número de registros existentes, determinou-se que a probabilidade de se ter um dia chuvoso logo após um dia seco é de 1/3, e a probabilidade de se ter um dia chuvoso logo após um dia chuvoso é de ½ .

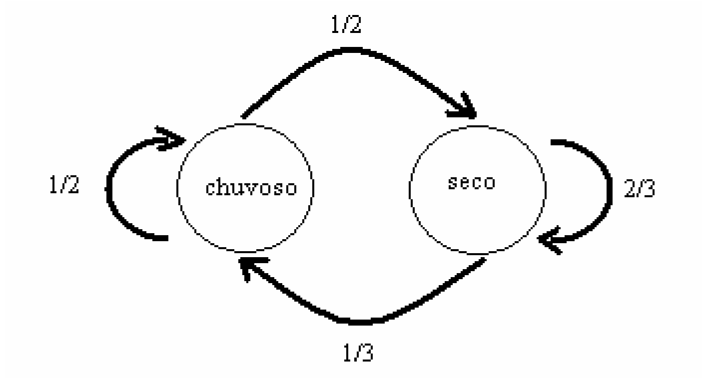


FIGURA 1. Grafo representando a mudança de tempo de uma cidade.

Se representar por D o estado de um dia seco e por R o de um dia chuvoso, então a matriz de transição dessa cadeia de Markov é:

$$T = \begin{array}{c} \begin{array}{cc} \text{D} & \text{R} \end{array} \\ \left| \begin{array}{cc} 2/3 & 1/2 \\ 1/3 & 1/2 \end{array} \right| \end{array} \begin{array}{c} \\ \text{D} \\ \text{R} \end{array}$$

Exemplo 3.2. Uma organização que faz pesquisa de mercado está estudando um grupo grande de compradores de café que compram um pacote por semana. Descobriu-se que 50 por cento dos que utilizam atualmente a marca *A* vão comprar novamente a marca *A* na próxima semana, enquanto que 25 por cento vão mudar para a marca *B* e 25 por cento vão mudar para outra marca. Entre os que usam atualmente a marca *B*, 30 por cento vão comprar novamente a marca *B* na próxima semana, enquanto que 60 por cento vão mudar para a marca *A* e 10 por cento vão mudar para outra marca. Entre os que usam atualmente outra marca, 30 por cento vão continuar com outra marca na próxima semana, 40 por cento vão mudar para *A* e 30 por cento vão mudar para *B*. Os estados A, B e D correspondem às marcas *A*, *B* e outras, respectivamente. A probabilidade de que uma pessoa usando *A* vai mudar para *B* é de 0,25, a probabilidade de que uma pessoa usando *B* vai continuar com *B* é de 0,30 e assim por diante. Então, a matriz de transição dessa cadeia de Markov é:

$$T = \begin{array}{cccc} A & B & D & \\ \left|\begin{matrix} 0.50 & 0.60 & 0.40 \\ 0.25 & 0.30 & 0.30 \\ 0.25 & 0.10 & 0.30 \end{matrix}\right| & & & \begin{matrix} A \\ B \\ D \end{matrix} \end{array}$$

Vamos agora usar a matriz de transição do processo de Markov para determinar a probabilidade de o sistema estar em qualquer dos *n* estados em períodos futuros de tempo.

Vamos representar por

$$\mathbf{x}^{(k)} = \left|\begin{matrix} p_1^{(k)} \\ p_2^{(k)} \\ \vdots \\ p_n^{(k)} \end{matrix}\right| \quad (k \geq 0),$$

o vertor de estado do processo de Markov no período de observação k, onde $p_j^{(k)}$ é a probabilidade de que o sistema esteja no estado *j* no período de observação *k*. o vetor de estado $x^{(0),}$ no período de observação 0, é chamado de vetor de estado inicial.

Teorema 3.1. Se *T* é a matriz de transição de um processo de Markov, então o vetor de estado $x^{(k+1),}$ no (*k*+1)-ésimo período de observação, pode ser determinado a partir do vetor de estados $x^{(k)}$, no *k*-ésimo período de observação, por:

$$\mathbf{x}^{(k+1)} = T\mathrm{x}^{(k)}$$

Da equação anterior temos

$$\mathbf{x}^{(1)} = T\mathbf{x}^{(0)}$$
$$\mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = T(\mathrm{T}\mathbf{x}^{(0)}) = T^2\mathbf{x}^{(0)}$$
$$\mathbf{x}^{(3)} = T\mathbf{x}^{(2)} = T(T^2\mathbf{x}^{(0)}) = T^3\mathbf{x}^{(0)},$$

em geral,

$$\mathbf{x}^{(n)} = T^n\mathbf{x}^{(0)}.$$

portanto, a matriz de transição e o vetor de estado inicial determinam completamente todos os outros vetores de estado.

Exemplo 3.3. Considere o exemplo 1 de novo. Suponha que, ao iniciar nossas observações (dia 0), o dia esta seco, de modo que o vetor de estado inicial é

$$\mathbf{x}^{(0)} = \begin{vmatrix} 1 \\ 0 \end{vmatrix}$$

Então, o vetor de estado no dia 1 (o dia seguinte ao inicio de nossas observações) é

$$\mathbf{x}^{1} = T\mathbf{x}^{(0)} = \begin{vmatrix} 0.67 & 0.5 \\ 0.33 & 0.5 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \begin{vmatrix} 0.67 \\ 0.33 \end{vmatrix}$$

Logo, a probabilidade de não chover no dia 1 é 0,67 e a probabilidade de chover é 0,33. Analogamente,

$$\mathbf{x}^{(2)} = T\mathbf{x}^{(1)} = \begin{vmatrix} 0.67 & 0.5 \\ 0.33 & 0.5 \end{vmatrix} \begin{vmatrix} 0.67 \\ 0.33 \end{vmatrix} = \begin{vmatrix} 0{,}614 \\ 0{,}386 \end{vmatrix}$$

$$\mathbf{x}^{(3)} = T\mathbf{x}^{(2)} = \begin{vmatrix} 0.67 & 0.5 \\ 0.33 & 0.5 \end{vmatrix} \begin{vmatrix} 0{,}614 \\ 0{,}386 \end{vmatrix} = \begin{vmatrix} 0{,}604 \\ 0{,}396 \end{vmatrix}$$

$$\mathbf{x}^{(4)} = T\mathbf{x}^{(3)} = \begin{vmatrix} 0.67 & 0.5 \\ 0.33 & 0.5 \end{vmatrix} \begin{vmatrix} 0{,}604 \\ 0{,}396 \end{vmatrix} = \begin{vmatrix} 0{,}603 \\ 0{,}397 \end{vmatrix}$$

$$\mathbf{x}^{(5)} = T\mathbf{x}^{(4)} = \begin{vmatrix} 0.67 & 0.5 \\ 0.33 & 0.5 \end{vmatrix} \begin{vmatrix} 0{,}603 \\ 0{,}397 \end{vmatrix} = \begin{vmatrix} 0{,}603 \\ 0{,}397 \end{vmatrix}$$

A partir do quarto dia, o vetor de estado do sistema é sempre o mesmo,

$$\begin{vmatrix} 0{,}603 \\ 0{,}397 \end{vmatrix}$$

isso significa que, a partir do quarto dia, fica seco aproximadamente 60 por cento do tempo e chove aproximadamente 40 por cento do tempo.

À medida que o numero de observações aumenta, os vetores de estado tendem a um vetor fixo. Nesse caso dizemos que o processo de Markov atingiu o equilíbrio. O vetor fixo é chamado de vetor de estado estacionário. Os processos de Markov são usados, em geral, para determinar o comportamento de um sistema depois de um longo período de tempo; por exemplo, a parte do mercado que certo produtor pode esperar manter em uma base mais ou menos permanente. Portanto, a questão se um processo de Markov atinge ou não equilíbrio é de importância primordial. O próximo exemplo mostra que nem todo processo de Markov atinge equilíbrio.

No entanto, se exigirmos que a matriz de transição de um processo de Markov satisfaça uma propriedade bastante razoável, obteremos uma classe grande de processos de markov, muitos dos quais aparecem em aplicações praticas, que atingem um estado de equilíbrio.

Uma matriz de transição *T* de um processo de Markov é dita regular se todos os elementos de alguma potência de *T* são números positivos. Um processo de Markov é dito regular se sua matriz de transição é regular.

Teorema 3.2: Se *T* é matriz de transição de um processo de Markov regular, então:

(a) Quando n → ∞, T^n tende à matriz

$$A = \begin{vmatrix} u_1 & u_1 & \cdots & u_1 \\ u_2 & u_2 & \cdots & u_2 \\ \vdots & \vdots & & \vdots \\ u_n & u_n & \cdots & u_n \end{vmatrix}$$

Que tem todas as colunas iguais;

(b) Todas as colunas

$$\mathbf{u} = \begin{vmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{vmatrix}$$

de A são vetores de probabilidade com todos os elementos positivos, isto é $u_i > 0$ ($1 \leq i \leq n$) e

$$u_1 + u_2 + \ldots + u_n = 1.$$

Prova do seguinte resultado:

Teorema 3.4: Se T é uma matriz de transição regular e se A e **u** são como no teorema anterior, então:

(a) Qualquer que seja o vetor de probabilidade **x**, $T^n\mathbf{x} \rightarrow \mathbf{u}$ quando $n \rightarrow \infty$, de modo que é um vetor de estado estacionário;

(b) O vetor de estado estacionário **u** é o único vetor de probabilidade que satisfaz a equação matricial $T\mathbf{u} = \mathbf{u}$.

Demonstração:

(a) Seja

$$\mathbf{x} = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix}$$

um vetor de probabilidade. Como $T^n \rightarrow A$ quando $n \rightarrow \infty$, temos,

$$T^n\mathbf{x} \rightarrow A\mathbf{x}.$$

Por outro lado,

$$\begin{vmatrix} u_1 & u_1 & \dots & u_1 \\ u_2 & u_2 & \dots & u_2 \\ \vdots & \vdots & & \vdots \\ u_n & u_n & \dots & u_n \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix} = \begin{vmatrix} u_1x_1 + u_1x_2 + \dots + u_1x_n \\ u_2x_1 + u_2x_2 + \dots + u_2x_n \\ \vdots \\ u_nx_1 + u_nx_2 + \dots + u_nx_n \end{vmatrix} = \begin{vmatrix} u_1(x_1 + x_2 + \dots + x_n) \\ u_2(x_1 + x_2 + \dots + x_n) \\ \vdots \\ u_n(x_1 + x_2 + \dots + x_n) \end{vmatrix} = \begin{vmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{vmatrix}$$

Já que $x_1 + x_2 + ... + x_n = 1$. Logo $T^n\mathbf{x} \rightarrow \mathbf{u}$.

(b) Como $T^n \rightarrow$ A, também temos $T^{n+1} \rightarrow A$. No entanto,

$$T^{n+1} = TT^n,$$

de modo que $T^{n+1} \rightarrow TA$. Logo, $TA = A$. Igualando as colunas correspondentes dessa equação, temos $T\mathbf{u} = \mathbf{u}$. Para mostrar a unicidade de **u**, suponha que **v** é outro vetor de probabilidade tal que $T\mathbf{v} = \mathbf{v}$. De (a), $T^n\mathbf{v} \rightarrow \mathbf{u}$ e, como $T\mathbf{v} = \mathbf{v}$, temos $T^n\mathbf{v} = \mathbf{v}$ para todo n. portanto, $\mathbf{v} = \mathbf{u}$.

4 O PROBLEMA DA PARADA ÓTIMA (OSP)

O problema da parada ótima (OSP) é um dos mais importantes campos estudados dentro da teoria geral do controle estocástico. Ele foi inicializado dentro do contexto de tempo de parada ótima em estatística Baysiana no trabalho clássico de Wald and Wolfowitz, (1948), citado por (Sonin, 1999). O problema geral de parada ótima de processos randômicos em tempo discreto foi formulado por Snell em 1953 (Sonin, 1999).

$M = (X, P, g)$ denota o problema da parada ótima (para uma cadeia de Markov) com espaço de estados finitos X, matriz de transição $P = \{p(x, y)\}$ e *função retorno g*. A função v satisfaz a equação (ótima) de Bellman para um ponto inicial x,

$$v(x) = \max(g(x), Tv(x)),$$

onde $g(x)$ é o retorno de parada no ponto x e T é o operador médio, dado pela fórmula $Tf(x) = \sum_y p(x,y)f(y)$. A equação de Bellman satisfaz $v(x) \geq g(x)$, $v(x) \geq \boldsymbol{T}v(x)$ para todo x $\in X$. Os estados onde o valor de $v(x)$ é iguail a $g(x)$, são estados de parada, ou seja, o momento que devemos parar de percorrer a cadeia de Markov e receber o valor g(x), (Sonin, 1999).

Outro método para resolver o tempo de parada ótima é o método da iteração (Sonin, 1999).

$$v_{n+1}(x) = \max(g(x), \boldsymbol{T}v_n(x)), v_0(x) = g(x).$$

$v_0(x) \leq v_1(x) \leq \ldots$ e a seqüência $v_n(x)$ converge para $v(x)$. Claro que esse é equivalente para o OSP considerando o intervalo finito $[0,n]$.

4.1 O Algoritmo de Eliminação para o Problema da Parada Ótima em Cadeias de Markov

O algoritmo é baseado na idéia de eliminar os estados onde a parada não é ótima e a definição de uma outra cadeia com as probabilidades alteradas.

Vamos assumir o modelo de Markov M_1, como o par (X_1, P_1), onde X_1 é um espaço de estados finitos e $P_1 = \{p_1(x; y)\}$ é a matriz de transição, é dado (Z_n) como a cadeia de Markov especificada pelo modelo M_1. Temos $D \subset X_1$ e vamos introduzir na seqüência de Markov os tempos $\tau_1, \tau_2, \ldots, \tau_n, \ldots$ os momentos do primeiro, segundo e assim por diante visitas (de Z_n) para $X_1 \backslash D$ (X_1 sem D), isto é:

$$\tau_1 = \min\{k > 0, Z_k \in (X_1 \backslash D)\},$$

$$\tau_{n+1} = \min\{k > \tau_n, Z_k \in (X_1 \backslash D)\}, 1 \leq \tau_1 < \tau_2 < \ldots .$$

Temos $X_2 \equiv (X_1 \setminus D)$ e considere a seqüência randômica $Y_n = Z_{\tau_n}, n = 1, 2, \ldots$. Denotamos por $u_1(z, X_2, \cdot)$ a distribuição da cadeia de Markov (Z_n) para o modelo inicial M_1 no momento τ_1 da primeira visita para X_2(primeira saída de D) começando em z, $z \in D$.

Sonin, (1999), A seqüência randômica (Y_n) é a cadeia de Markov em um modelo $M_2 = (X_2, P_2)$, onde a matriz de transição $P_2 = \{p_2(i,j)\}$ é dada pela fórmula:

$$p_2(x,y) = p_1(x,y) + \sum_{z \in D} p_1(x,z) u_1(z, X_2, y), \quad (x, y \in X_2).$$

Chamamos o modelo M_2, o modelo D-reduzido de M_1.

Um caso importante é quando D consiste de somente um ponto não absorvente z. Neste caso a fórmula (1) toma a forma,

$$p_2(x,\cdot) = p_1(x,\cdot) + p_1(x,z)(1 - p_1(z,z))^{-1} p_1(z,\cdot), \quad (x \in X_2).$$

Embora no OSP, possa ser difícil encontrar os estados onde é ótimo parar, é relativamente fácil encontrar estados onde é ótimo não parar. Por exemplo, é ótimo não parar em todos os estados onde Tg (.) > g (.), isto é, o retorno esperado ao dar mais um passo é maior do que o retorno de parada. Geralmente, é ótimo não parar em estados onde o retorno esperado ao realizar certo número de passos é maior do que o retorno de parada. Agora podemos excluir tais estados e recalcular a matriz de transição utilizando:

$$p_2(x,\cdot) = p_1(x,\cdot) + p_1(x,z)(1 - p_1(z,z))^{-1} p_1(z,\cdot), \quad (x \in X_2).$$

para eliminar um estado, ou:

$$p_2(x,y) = p_1(x,y) + \sum_{z \in D} p_1(x,z) u_1(z, X_2, y), \quad (x, y \in X_2).$$

para eliminar um subconjunto de estados maior que um (Sonin, 1999).

Um exemplo do método de redução do número de estados pode ser mostrado a seguir:

Seja a cadeia de Markov da figura 2,

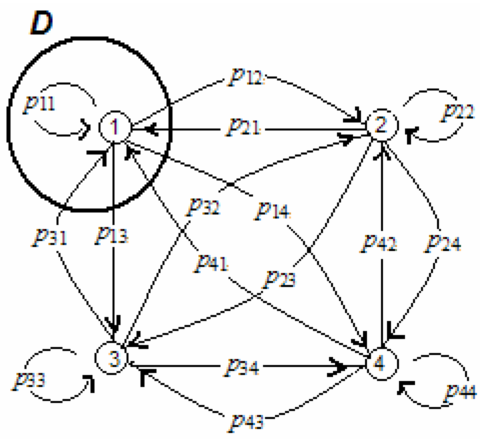


Figura 2. Grafo com estado a ser eliminados.

eliminando o subconjunto *D,* que neste exemplo, possui somente um estado, o novo grafo seria o da figura 3, com as novas probabilidades.

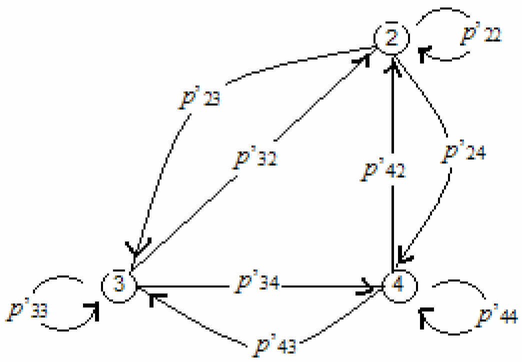


Figura 3. Grafo com estados eliminados.

As novas probabilidades podem ser encontradas da seguinte forma:

$p'_{22} = p_{22} + (p_{21})\ (1\text{-}p_{11})^{-1}(p_{12})$

$p'_{23} = p_{23} + (p_{21})\ (1\text{-}p_{11})^{-1}(p_{13})$

$p'_{24} = p_{24} + (p_{21})\ (1\text{-}p_{11})^{-1}(p_{14})$

$p'_{32} = p_{32} + (p_{31})\ (1\text{-}p_{11})^{-1}(p_{12})$

$p'_{33} = p_{33} + (p_{31})\ (1\text{-}p_{11})^{-1}(p_{13})$

$p'_{34} = p_{34} + (p_{31})\ (1\text{-}p_{11})^{-1}(p_{14})$

$p'_{42} = p_{42} + (p_{41})\ (1\text{-}p_{11})^{-1}(p_{12})$

$p'_{43} = p_{43} + (p_{41})\ (1\text{-}p_{11})^{-1}(p_{14})$

$p'_{44} = p_{44} + (p_{41})\ (1\text{-}p_{11})^{-1}(p_{14})$

5 O PROBLEMA DA SECRETÁRIA

Na coluna de jogos Matemáticos de Martin Gardner, na edição de fevereiro de 1960 do periódico *Scientific American*, publicou-se um problema simples que veio a ser conhecido como o Problema da Secretária. Pode-se afirmar que esse problema, objeto de estudo de probabilistas e estatísticos, fundou um campo dentro da otimização estocástica de enormes proporções (Fergusson, 1989 citado por Brighenti, 2003).

O problema é o seguinte: deseja–se contratar uma nova secretária, e apresentam-se n candidatas interessadas no trabalho. As candidatas são entrevistadas em uma ordem aleatória, mas, imediatamente após cada entrevista, decide-se entre aceitar ou rejeitar a candidata; rejeitada a candidata, não se pode aceitá-la depois, e, uma vez aceita, todas as outras são rejeitadas. Em particular, se rejeitadas as (n-1) primeiras candidatas, aceita-se automaticamente a candidata n (Freeman, 1983). Nesta situação, qual a estratégia a ser adotada para maximizar a probabilidade de se contratar a melhor secretária? Ou, pelo menos, uma suficiente qualificada?

Lindley (1961) foi o primeiro a publicar uma solução para esse problema utilizando programação dinâmica e teoria da decisão. Posteriormente Dynkin (1963), citado por Landin (1983), apresentou uma outra solução utilizando o conceito de tempo ótimo de parada para cadeias de Markov. A estratégia ótima conhecida para o problema corresponde a observar um número predeterminado de objetos sem interromper o processo e pará-lo logo que apareça um objeto melhor que os anteriores. Essa estratégia é baseada em postos relativos.

O problema da Secretária é tratado também como o Problema do Matrimônio ou Problema da Princesa (Landim, 1983), onde uma princesa deseja selecionar entre seus pretendentes o mais qualificado. Neste caso, o conhecimento a priori do número de pretendentes pode não ser verossímil, mas a segunda hipótese, que estabelece a impossibilidade de reconsiderar um objeto anteriormente rejeitado, parece plausível.

Davis, (1973) trata o mesmo problema como Dilema do Policial, no qual cinco pessoas suspeitas de um crime mantêm encontro secreto no porão de um edifício. Do lado de fora, um policial, com ordens de seguir o chefe do bando, espera que eles se dispersem. O policial sabe que o homem em que está interessado é o mais alto do grupo, único meio que dispõe para distingui-lo dos demais. Se os suspeitos deixam o encontro em ordem aleatória, qual a melhor estratégia a ser adotada pelo policia? Se adotar a melhor estratégia, qual a possibilidade de ser efetivamente o chefe a pessoa que ele vier a seguir?

5.1 Estratégia por Postos Relativos

Essa abordagem será apresentada seguindo o exposto em Landim (1983).

Considerando as candidatas como n objetos a serem inspecionados e representando-os como pontos da reta, tem-se que um ponto à direita de outro indica ser aquele melhor do que este. E definindo-os por $a_1 < a_2 < ... < a_n$. Escolhe-se ao acaso um destes pontos, denominado w_1. Assim, w_1 será igual à a_j, para $1 \leq j \leq n$, com probabilidade 1/n. Em seguida, escolhe-se, ao acaso, um entre os (n-1) pontos restantes. Este ponto será o w_2. Desta forma obtem-se uma seqüência (w_1, w_2, .., w_n). Cada seqüência possível é uma permutação dos elementos a_1,..., a_n. Como as seqüências são equiprováveis e existem n! permutações de n objetos distintos, cada seqüência terá probabilidade 1/(n!) de ocorrer.

Após terem sido inspecionados k objetos, a única informação que se possui é a ordem relativa de w_1,... , w_k. Neste momento, deve-se decidir entre selecionar w_k e interromper o processo ou prosseguir e rejeitar definitivamente w_k. Esta estratégia é então baseada em postos relativos, ou seja, a cada objeto inspecionado, é atribuído um posto relativo às inspeções realizadas nos objetos anteriores. Deseja-se maximizar a probabilidade de selecionar a_n.

Uma estratégia consiste em selecionar o primeiro ponto examinado: w_1. Neste caso, ter-se-ia selecionado corretamente o melhor objeto se a_n tiver sido inspecionado primeiro. A probabilidade de escolher a_n na primeira inspeção é igual a 1/n. Quando n é grade essa probabilidade tende a zero. Pode-se formular agora uma pergunta interessante. Existe uma estratégia em que a probabilidade de sucesso não tente a zero quando n tende a infito?

Suponha n par e considere a seguinte estratégia: rejeitar sistematicamente os n/2 primeiros objetos; em seguida, selecionar o primeiro objeto melhor do que todos os anteriores. Com esta estratégia, selecionar-se o melhor objeto, por exemplo, se a_{n-1} estiver entre os n/2 primeiros objetos examinados e a_n estiver na segunda metade a ser examinada, ou seja, fixando a_{n-1} e a_n, têm-se (n-2)! seqüências possíveis. Como a_{n-1} assumir n/2 posições diferentes, assim como a_n, então o total de seqüências que satisfaz as condições é $(n/2)^2$ (n-2)!(Figura 4).

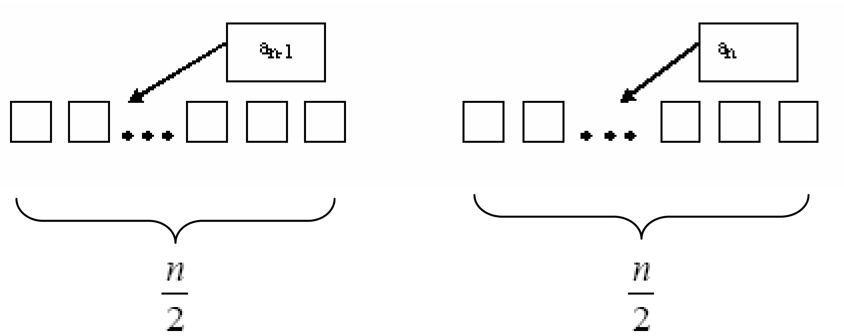


FIGURA 4. Esquema de seqüências com sucesso, em que a_{n-1} aparece na primeira metade as amostra e a_n na segunda.

Portanto a probabilidade mínima da selecionar o melhor objeto com esta estratégia é

$$P_{[sucesso]} \geqslant \left(\frac{n}{2}\right)^2 \frac{(n-2)!}{n!} = \frac{n}{4(n-1)}$$

A probabilidade de sucesso pode ser maior que o valor encontrado, pois não foram levadas em consideração outras situações de sucesso, como por exmplo, a_{n-1} e a_n estarem na segunda metade e a_n aparecer antes de a_{n-1} (Figura 5).

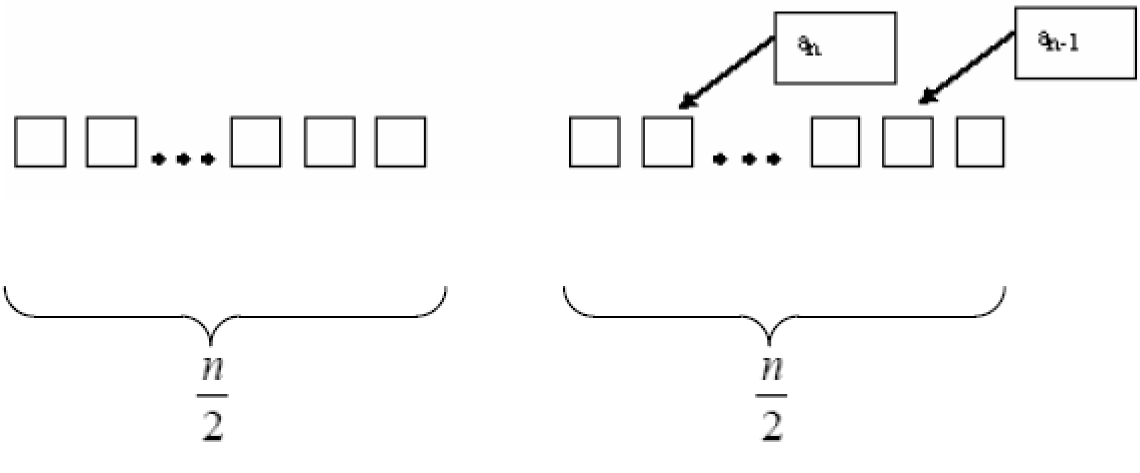


FIGURA 5. Esquema de seqüências com sucesso, em que a_{n-1} e a_n aparecem na segunda metade da amostra.

Exemplificando a Estratégia Exposta:

Suponham-se 10 candidatas apresentadas aleatoriamente para uma entrevista. Admitindo que os pontos sejam atribuídos de 1 a 10, pela estratégia proposta de ir até a metade, as entrevistas seriam realizadas com as 5 primeiras candidatas e a partir dos postos relativos atribuídos a estas, escolher a primeira melhor que as anteriores.

Existiriam 10! seqüências possíveis de entrevista. Destas existem apenas $(10/2)^2$ $(10-2)! = 5^2 8! = 1.008.000$ seqüências em que a candidata de posto 9 é entrevistada entre as cinco primeiras, enquanto a candidata de posto 10 encontra-se nas próximas cinco entrevistas.

E a probabilidade de sucesso seguindo a estratégia seria

$$P_{[sucesso]} \geq \frac{((5)^2(8)!)}{(10!)} = \frac{10}{(4(9))} = 0{,}2778$$

Alguns exemplos de possíveis seqüências de postos relativos:

a) 2; 7; 9;1; 6; 3; 5; 10; 8; 4.

b) 3; 5; 7; 8; 4; 6; 10; 1; 9; 2.

c) 6; 2; 1; 5; 4; 7; 8; 10; 3; 9.

d) 8; 4; 3; 2; 10; 1; 5; 9; 7; 6.

Seguindo a estratégia o caso (a) é de sucesso, sendo o processo interrompido na oitava candidata. O caso (b) é um exemplo de sucesso que não está incluído no cálculo da probabilidade de sucesso. Nos casos (c) e (d) ocorre fracasso, escolhendo no caso (c) a candidata de posto 7 interrompendo o processo escolhendo na sexta entrevista, no caso (d), escolhendo a última.

A estratégia de ir até a metade pressupõe o conhecimento de n, mas não o de a_n. Desta forma, obtêm-se uma estratégia com probabilidade que não tende a 0 quando cresce para $+\infty$, pois:

$$\lim_{(n \to \infty)} P_{[sucesso]} = \lim_{(n \to \infty)} \left(\frac{n}{(4(n-1))}\right) = \frac{1}{4}$$

Esta estratégia ainda não é a ótima, isto é, existe outra estratégia com a probabilidade de sucesso maior. A estratégia ótima é aquela em que a $P_{[sucesso]}$ é máxima.

Para encontrar a estratégia ótima deve-se verificar se o ponto w_k é máximo. Isto ocorre se $w_j \leq w_k$ para $1 \leq j \leq k$. quando w_k não é máximo, w_k é certamente diferente de a_n e o processo deve continuar. A decisão de interromper ou não o processo, quando aparecer um ponto de máximo w_k , só depende do índice do ponto de máximo e não da disposição relativa dos pontos anteriores, $w_1,..., w_{k-1}$. De fato, sendo w_k um ponto máximo, $w_k = a_n$, se $w_k > w_j$ para $k < j \leq n$. Pelo contrário, $w_k \neq a_n$ se existe $k+1 < j \leq n$ $w_j > w_k$. Portanto, a estratégia depende do número de pontos máximo. A estratégia obterá sucesso se parar no ultimo ponto de máximo.

Seja X_j o índice do j-ésimo ponto de máximo. Desta forma, como w_1 é sempre um ponto de máximo, $X_1(w) = 1$ e $X_2(w) = 2$ se $w_2 > w_1$; caso contrário, $X_2(w) > 2$.

Para cada seqüência w o numero de pontos máximo varia. Para contornar esta dificuldade, introduz-se um novo ponto δ. Se a seqüência w possuir $j \leq n$ pontos máximos, então $X_k(w) = \delta$ para todo $k > j$.

Deseja-se estudar a evolução da seqüência $(X_j)_{j \geq 1}$ para encontrar um método que indique quando o processo de inspeção deve ser interrompido. A cada aparecimento de um novo ponto máximo e, portanto, a cada novo valor da seqüência (X_j) deve-se decidir entre o prosseguimento ou não da inspeção (Figura 6).

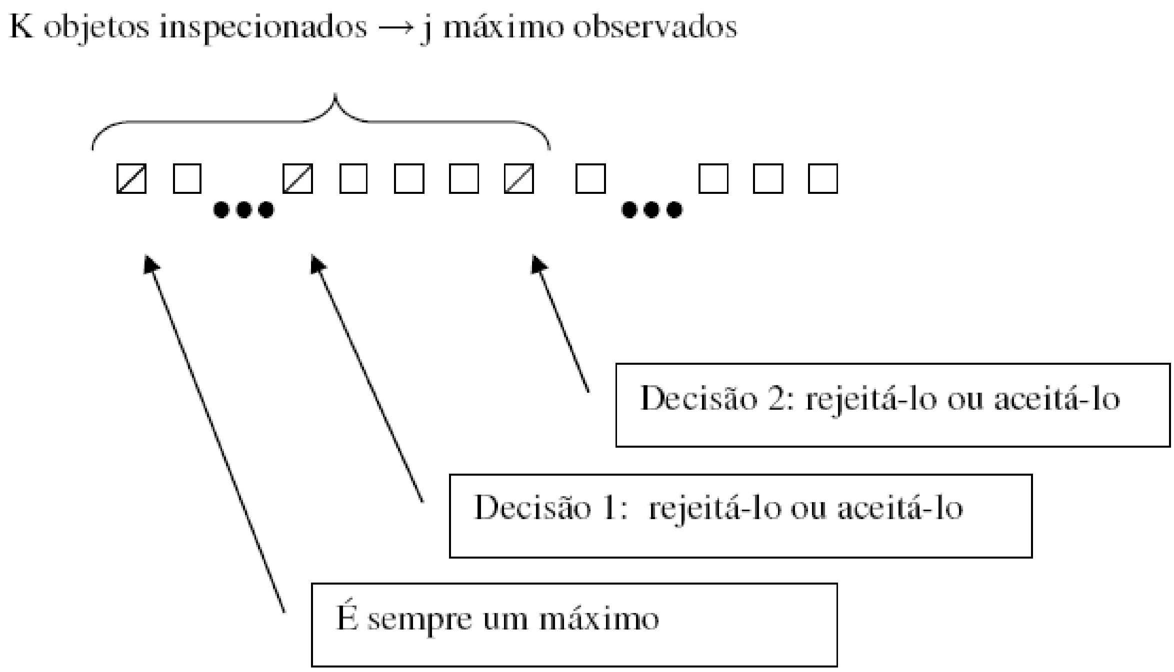


FIGURA 6. Esquema da inspeção de seqüencial por postos relativos.

Considere-se um instante k no qual aparece um novo ponto de máximo w_k. Como se trata de um ponto máximo, existe um inteiro $j \leq k$ com $X_j = k$, onde j indica o número de pontos máximos examinados até a etapa k. $w_k = a_n$ quando e somente quando não aparecer outro ponto máximo; portanto, se e somente se $X_{j+1} = \delta$.

A probabilidade de w_{k+1} ser um ponto máximo, quando w_k é também um ponto máximo, é dado por $P[X_{j+1} = k + 1 \mid X_j = k] = 1/(k+1)$, que representa a probabilidade de w_{k+1} pertencer a qualquer um dos intervalos formados pelos pontos ordenados (Figura 7).

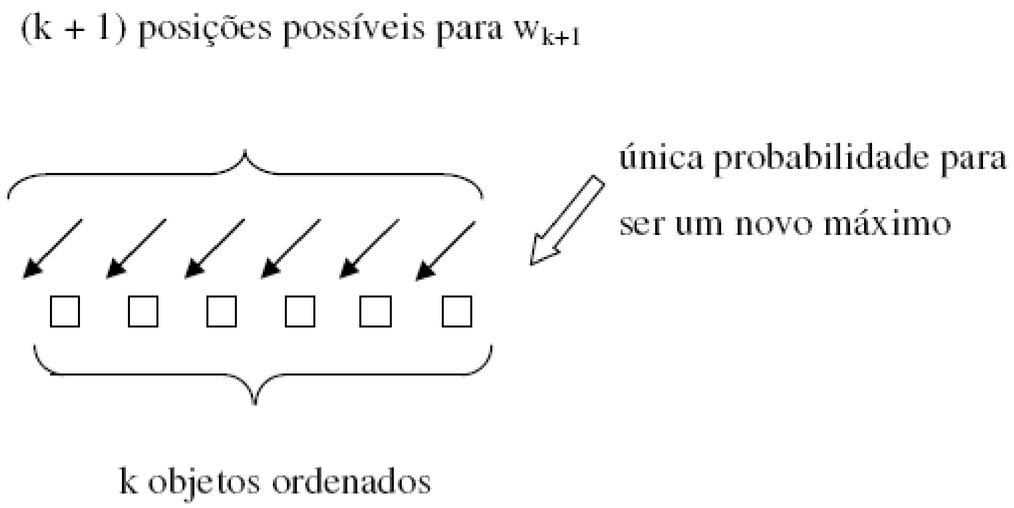


FIGURA 7. Posições possíveis para o objeto w_{k+1}.

A probabilidade de w_{k+1} não ser máximo é igual a $k/(k+1)$. Uma vez que o ponto w_{k+1} foi inspecionado, a probabilidade de w_{k+2} pertencer a qualquer um dos (k+2) intervalos formados é $1/(k+2)$. Assim, a probabilidade de w_{k+2} não ser um ponto, máximo, conhecidos os valores de $w_1,\ldots, w_{k-1}$ é $(k+1)/(k+2)$. Prosseguindo desta forma até a última etapa k + i, onde aparece um ponto máximo, tem-se

$$P[X_{j+1}=k+i|X_j=k]=\left(\frac{k}{k+1}\right)\left(\frac{k+1}{k+2}\right)\cdots\left(\frac{k+i-2}{k+i-1}\right)\left(\frac{1}{k+i}\right)=\left(\frac{k}{(k+i-1)(k+1)}\right).$$

Por outro lado $X_j = k$ e $X_{j+1} = \delta$ quando não aparecer nenhum ponto de máximo após w_k. Logo

$$P[X_{j+1}=\delta|X_j=k]=\left(\frac{k}{k+1}\right)\left(\frac{k+1}{k+2}\right)\cdots\left(\frac{n-2}{n-1}\right)\left(\frac{n-1}{n}\right)=\left(\frac{k}{n}\right).$$

O cálculo das probabilidades não envolve o número de pontos máximos ocorrido até a etapa k nem a distribuição relativa de $w_1,\ldots, w_{k-1}$, e, portanto, dos valores de $X_1, \ldots, X_{j-1}$. Deste modo, para toda seqüência crescente

$$P[X_{j+1} = y \mid X_1 = 1,\ldots, X_{j-1} = k_{j-1}, X_j = k] = P[X_{j+1} = y \mid X_j = k]$$

$$= \begin{cases} k/((y-1)y) & \text{se } 1 \le k \le n,\ k+1 \le y \le n; \\ k/n & \text{se } 1 \le k \le n, y = \delta; \\ 1 & \text{se } k = \delta,\ y = \delta; \\ 0 & \text{caso contrário} \end{cases}$$

Uma seqüência de variáveis aleatórias $(X_j)_{j \ge 1}$ que segue as probabilidades citadas, definidas num espaço de probabilidade, é chamado de cadeia de Markov.

Seja $1 \le k \le n$ e suponha que $X_j = k$. A probabilidade do processo pular para δ é k/n. Portanto, estando X_j em k, a probabilidade de selecionar o melhor objeto interrompendo o processo neste instante é k/n. Por outro lado, pode-se preferir aguardar um novo máximo, esperando que apareça algum. Isto representa aguardar um novo salto do processo (X_j), e então interromper o processo. Nesta alternativa, seleciona-se o melhor objeto se $X_{j+1} \in \{k + 1,\ldots, n\}$ e $X_{j+2} = \delta$, portanto com probabilidade

$$P[X_{j+1} \in \{k+1,\ldots, n\}, X_{j+2} = \delta \mid X_j = k] =$$

$$= \sum_{i=k+1}^{n} P[X_{j+1}=i, X_{j+2}=\delta | X_j=k]$$

$$= \sum_{i=k+1}^{n} P[X_{j+2}=\delta |, X_j=k, X_{j+1}=i] P[X_{j+1}=i | X_j=k]$$

$$= \sum_{i=k+1}^{n} P[X_{j+2}=\delta |, X_{j+1}=i] P[X_{j+1}=i] P[X_j=k]$$

$$\sum_{i=k+1}^{n} \frac{1}{n} \frac{k}{i(i-1)} = \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i} = \left(\sum_{i=k}^{n-1} \frac{1}{i} \right) P[X_{j+1}=\delta | X_j=k].$$

Portanto, se $\sum_{i=k}^{n-1} \frac{1}{i} < 1$ é melhor interromper o processo quando X_j assume o valor k, pois a probabilidade de encontrar um novo máximo tende a diminuir; caso contrário, é preferível aguardar um novo ponto máximo.

A seqüência $(T_k)_{k\geq 1} = \sum_{i=k}^{n-1} \frac{1}{i}$ decresce para valores crescentes de k; logo existe um

inteiro k_n que verifica a relação

$$\frac{1}{k_n+1}+\ldots+\frac{1}{n-1}<1<\frac{1}{k_n}+\ldots+\frac{1}{n-1}$$

A relação corresponde à soma das áreas mostradas na figura 8.

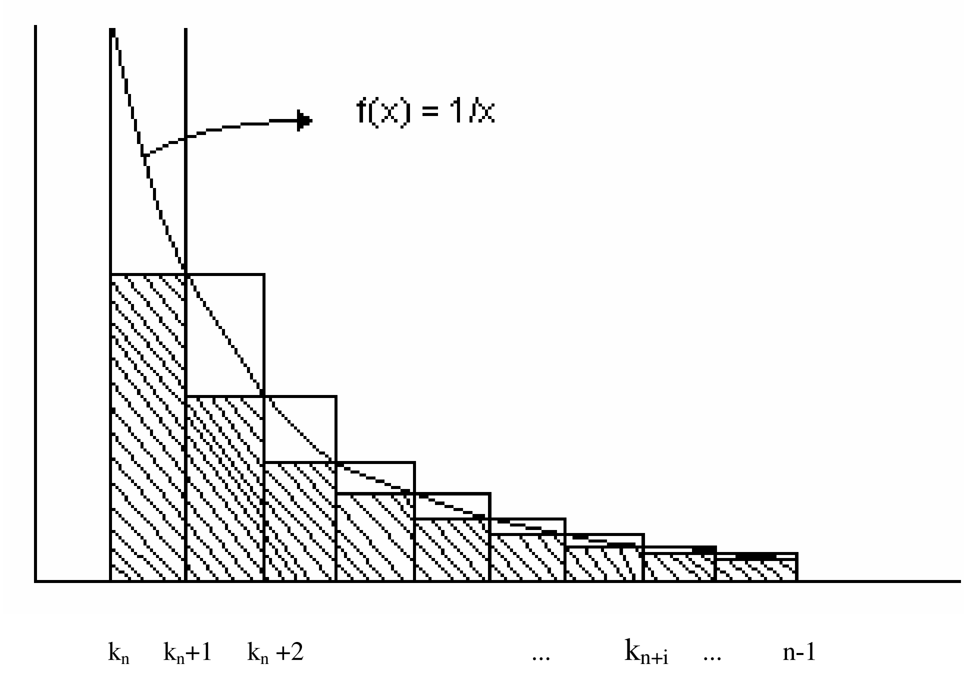


FIGURA 8. Somatório das áreas correspondentes à relação de k_n.

A estratégia ótima corresponde então a observar as k_n primeiros objetos sem interromper o processo e em seguida pará-lo logo que apareça um objeto melhor que todos os anteriores.

O comportamento assintótico de k_n/n é dado por

$$\int_{k_n+1}^{n}\left(\frac{1}{x}\right)dx<1<\int_{k_n-1}^{n-1}\left(\frac{1}{x}\right)dx$$

$$\ln\left(\frac{n}{k_n+1}\right)<1<\ln\left(\frac{n-1}{k_n-1}\right)$$

$$\frac{n}{k_n+1}<\mathrm{e}<\frac{n-1}{k_n-1}$$

$$\frac{1}{\mathrm{e}}-\frac{1}{n}<\frac{k_n}{n}<\frac{1}{\mathrm{e}}+\frac{1}{n}$$

Assintoticamente a estratégia ótima consiste em aguardar aproximadamente n/e objetos e escolher então o primeiro melhor que todos os anteriores.

Este estratégia é a melhor possível. A demonstração completa deste resultado utiliza a teoria dos Martingais e pode ser encontrada em Landim (1983).

Exemplo do cálculo de k_n

Realizando a inspeção de 8 objetos, o valor de k_n que a relação é $k_n = 3$, pois:

$$\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}<1<\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}$$

$$0{,}759 < 1 < 1{,}093$$

ou $$k_n\approx\frac{n}{e}\approx\frac{8}{2{,}7}=2{,}96$$

Ou seja, inspecionam-se os três primeiros objetos, avaliando-se a característica de interesse. Atribui-se um posto relativo a cada um deles e a partir daí escolhe o primeiro objeto com posto relativo maior que os anteriores.

5.2 Probabilidade de sucesso

A probabilidade da estratégia ótima selecionar a_n na inspeção w_k é dada por

$$P_{[sucesso]}=\left(\frac{k_n}{k_n+1}\right)\left(\frac{k_n+1}{k_n+2}\right)\cdots\left(\frac{k-2}{k-1}\right)\frac{1}{k}=\frac{k_n}{(k-1)k}.$$

logo,

$$P_{[sucesso]}=P[X_j\in\{1,\ldots,n\},X_{j+1},=\delta]=$$

$$\sum_{k=k_n+1}^{n}P[X_j=k,X_{j+1}=\delta]=\sum_{k=k_n+1}^{n}P[X_{j+1}=\delta|X_j=k]P[X,=k]=\sum_{k=k_n+1}^{n}\frac{k}{n}\frac{k_n}{(k-1)k}=\frac{k_n}{n}\sum_{k=k_n}^{n-1}\frac{1}{k}$$

como $\sum_{k=k_n}^{n-1}\frac{1}{k}\approx 1$ e $\frac{1}{e}-\frac{1}{n}<\frac{k_n}{n}<\frac{1}{e}\frac{1}{n}$

$$P_{[sucesso]}=\frac{k_n}{n}\sum_{k=k_n}^{n-1}\frac{1}{k}\approx\frac{1}{e}.$$

6 METODOLOGIA

Foi estudado o algoritmo de eliminação de estados (Sonin, 1999) para o problema da parada ótima em cadeias de Markov e sua aplicação na solução do problema da secretária. Assim como a implementação dessa solução.

Para eliminar os estados onde a parada não é ótima em uma cadeia de Markov, foi utilizado o método de redução apresentado por Sonin, (1999).

Utilizou-se a equação de Bellman, para resolver o problema da parada ótima em cadeias de Markov. Em algumas cadeias a solução foi encontrada utilizando o procedimento backward (ordem inversa), em outras a solução foi encontrada utilizando o método da iteração. Implementações e testes foram realizados em várias cadeias, onde a solução encontrada retorna os estados onde é ótimo parar.

Em seguida foi apresentado e implementado uma solução ótima para o problema da secretária, utilizando a eliminação de estados e a solução da equação de Bellman.

Todas as implementações foram feitas utilizando o software livre R no laboratório de Ciência da computação da Universidade Federal de Lavras.

7 RESULTADOS E DISCUSSÃO

7.1 Redução do Número de Estados

O método de eliminação foi aplicado na matriz de Markov abaixo, utilizando o programa cujo código está em anexo (Anexo A).

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.2	0.1	0.3	0.5	0.3
[2,]	0.2	0.2	0.1	0.1	0.2
[3,]	0.1	0.4	0.4	0.1	0.1
[4,]	0.3	0.1	0.1	0.1	0.3
[5,]	0.2	0.2	0.1	0.2	0.1

Aqui temos a nova matriz com os estados 1 e 2 eliminados e as probabilidades redistribuídas.

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0	0	0.0000000	0.0000000	0.0000000
[2,]	0	0	0.0000000	0.0000000	0.0000000
[3,]	0	0	0.5071429	0.2404762	0.2642857
[4,]	0	0	0.2642857	0.3642857	0.4785714
[5,]	0	0	0.2285714	0.3952381	0.2571429

Abaixo temos as representações das matrizes em forma de grafos.

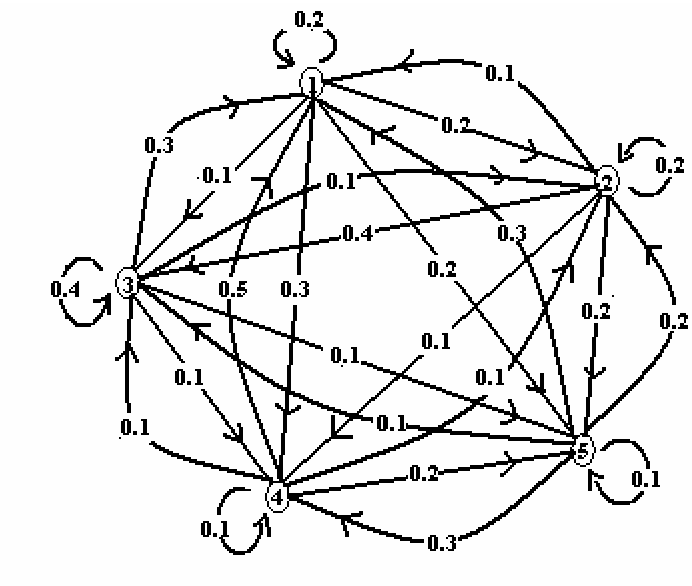


FIGURA 9: Grafo que representa a matriz de entrada antes dos estados serem eliminados.

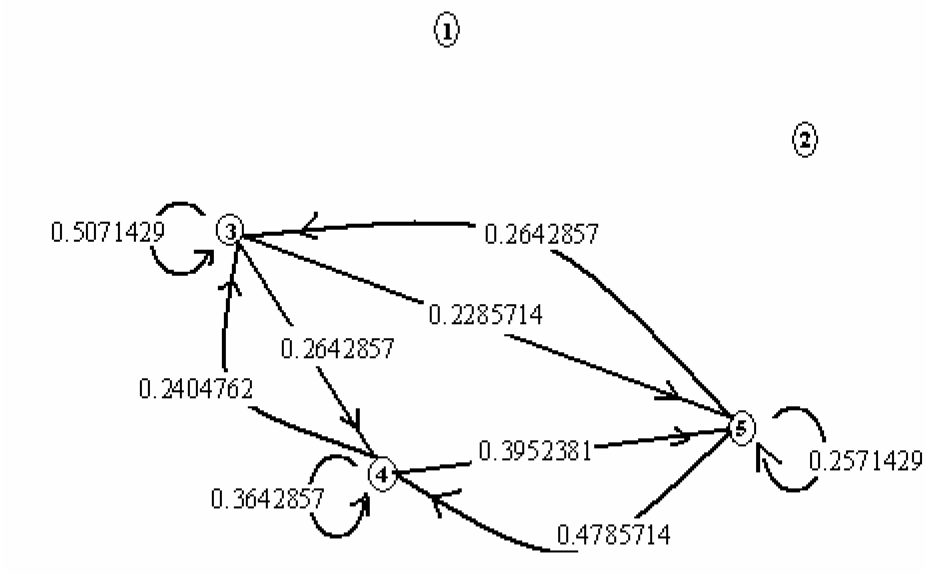


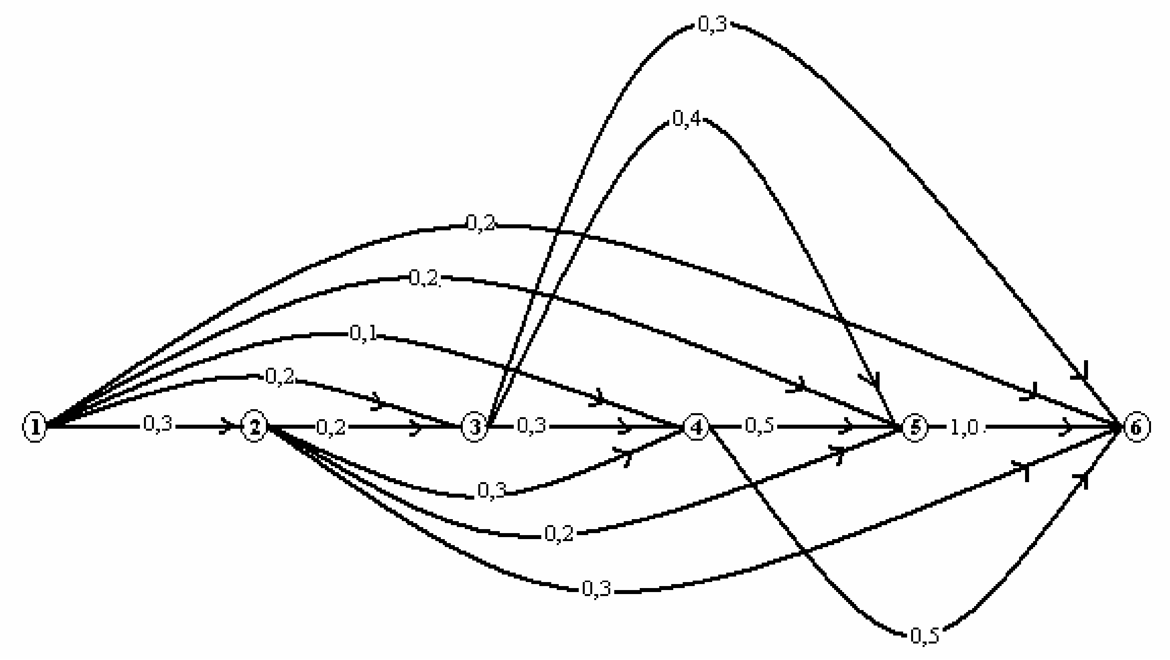
FIGURA 10: Grafo que representa a matriz de entrada depois dos estados serem eliminados

Podemos observar que os estados *1* e *2* da cadeia de Markov foram eliminados e as probabilidades da cadeia foram redistribuídas.

7.2 O Problema da Parada Ótima (OSP)

A Equação de Bellman ($v(x) = \max(g(x), Tv(x))$) nos dá uma solução direta para o problema da parada ótima, porém ela é aplicada em situações específicas onde as probabilidades de transições possuem uma estrutura muito simples.

Um exemplo é o Problema da Secretária que pode ser reduzido para o OSP com $X = \{1,2,...,n\}$ e a Cadeia de Markov que satisfaz a seguinte propriedade $P(x,y) = 0$ se $y \leq x$(ver figura). Neste caso o valor de $v(x)$ pode ser calculado seqüencialmente começando de $v(n)$ na ordem inversa (procedimento conhecido por Backward).



$g(1) = 4$; $g(2) = 6$; $g(3) = 5$; $g(4) = 10$; $g(5) = 5$; $g(6) = 3$

FIGURA 11: Grafo que representa uma variação do problema da secretária.

$$\boldsymbol{v(x) = \max \{ g(x), Tv(x) \}}$$

$v(6) = \max\{ g(6), Tv(6) \} = 3$

$v(5) = \max\{ g(5), Tv(5) \} = \max\{ g(5), 1 * v(6) \} = \max \{ 5, 3 \} = 5$

$v(4) = \max\{ g(4), Tv(4) \} = \max\{ g(4), (0,5 * v(5) + 0,5 * v(6)) \} = \max\{ 10, (2,5 + 1,5) \} = 10$

$v(3) = \max\{ g(3), Tv(3) \} = \max\{ g(3), (0,3 * v(4) + 0,4 * v(5) + 0,3 * v(6)) \} = \max\{ 5,$ (

3 + 2 + 0,9) } = 5,9

$v(2)$ = max{ $g(2)$, $Tv(2)$ } = max{ $g(2)$, (0,2*$v(3)$ +0,3*v(4) + 0,2 * $v(5)$ + 0,3 * $v(6)$) } = max{ 6, (1,18+3+1+0,9) } = 6,08

$v(1)$ = max{ $g(1)$, $Tv(1)$ } = max{ $g(1)$, (0,3*$v(2)$+ 0,2*$v(3)$ +0,1*$v(4)$ + 0,2 * $v(5)$ + 0,2 * $v(6)$) } = max{ 4, (1,824+1,18+ 1+1+0,6) } = 5,604

7.2.1 Resolução pelo método da iteração

$v_{n+1}(x) = \max(g(x), \boldsymbol{T}v_n(x)), v_0(x) = g(x)$

$Tg(x_i) = g(x_i)p(x_i,x_i) + g(x_{i+1})p(x_i,x_{i+1}) + \ldots + g(x_n)p(x_i,x_n)$

$Tg(1) = 5,4$

$Tg(2) = 5,9$

$Tg(3) = 5,9$

$Tg(4) = 4,0$

$Tg(5) = 3,0$

$Tg(6) = 0,0$

$v_1(1)$ = max{ $g(1)$, $Tg(1)$ } = max{4, (5,4)} = 5,4 $\neq v(1)$

$v_1(2)$ = max{ $g(2)$, $Tg(2)$ } = max{6, (5,9)} = 5,9 $\neq v(2)$

$v_1(3)$ = max{ $g(3)$, $Tg(3)$ } = max{5, (5,9)} = 5,9 = $v(3)$

$v_1(4)$ = max{ $g(4)$, $Tg(4)$ } = max{10, (4,0)} = 10,0 = $v(4)$

$v_1(5)$ = max{ $g(5)$, $Tg(5)$ } = max{5, (3,0)} = 5,0 = $v(5)$

$v_1(6)$ = max{ $g(6)$, $Tg(6)$ } = max{3, (0,0)} = 3,0 = $v(6)$

Pode se observar que $v_1(1)$ e $v_2(2)$ são diferentes de $v(1)$ e $v(2)$ (valores obtidos com a equação de Bellman) respectivamente. É preciso, então, que faça, mais iterações até os valores ficarem iguais. A partir desse momento os valores não se alteram aplicando mais iterações.

$v_2(1)$ = max{ $g(1)$, $Tv_1(1)$ } = max{4, (0,3*5,9+0,2*5,9+0,1*10+0,2*5+0,2*3)}= max{4, (5,55)} = 5,55 $\neq v(1)$

$v_2(2)$ = max{ $g(2)$, $Tv_1(2)$ } = max{6, (0,2* 5,9+ 0,3*10 + 0,2*5 + 0,3*3)}= max{6, (6,08)} = 6,08 = $v(2)$

Observe que $v_2(1)$ ainda não é igual a $v(1)$, é preciso fazer mais iterações.

$v_3(1)$ = max{ $g(1)$, $Tv_2(1)$ } = max{4, (0,3*6,08+ 0,2*5,9 +0,1*10 +0,2* 5+0,2*3)}= 5,604 = max{4, (5,604)} = 5,604 = $v(1)$

A partir deste momento o valor de $v_n(x)$ permanece o mesmo se continuar com as iterações.

O valor de $v(x)$ obtido representa o valor do jogo no estado x, ou seja, a esperança de ganho utilizando uma estratégia ótima a partir do estado x.

A estratégia ótima é parar onde $v(x)$ é igual a $g(x)$. Para o exemplo esses estados são 4, 5 e 6.

7.2.2 Valor ótimo utilizando o procedimento Backward

Na figura abaixo temos uma cadeia de Markov representada por uma árvore binária em que suas arestas estão rotuladas pela probabilidade de estar em um estado e ir para o outro. Os valores sublinhados são os valores $g(x)$, os retornos de cada estado.

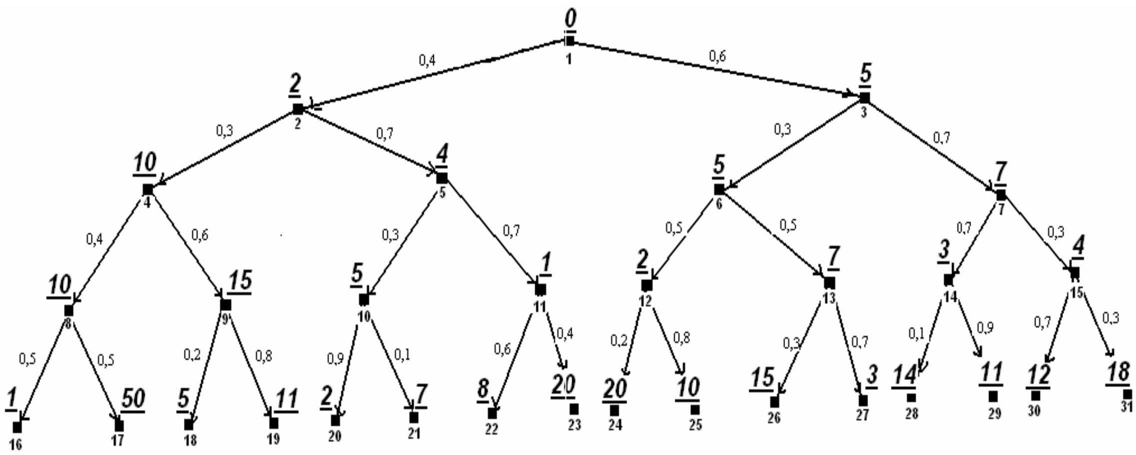


FIGURA 12: Cadeia de Markov em forma de árvore binária.

Esse problema pode ser pensado como um jogo, no qual partindo do estado 1, o jogador atira um dado e começa a percorrer o grafo, ele pode parar em um estado ou continuar jogando, lembrando que ao atingir um vértice final o jogador é obrigado a parar e recebe o retorno desse vértice.

O problema é: qual é a melhor estratégia utilizada que permite o jogador um retorno máximo?

Este problema pode ser resolvido utilizando o procedimento backward (na ordem inversa) para encontrar $v(x)$ (solução da equação de Bellman para o ponto x).

Para todos os vértices finais $v(x) = g(x)$, como $v(x) = \max\{g(x), v(x)\}$, logo $v(15) = \max\{4, (0,3*18 + 0,7* 12)\} = \max\{4, (13,8)\} = 13,8$. Repetindo esse procedimento para todos os vértices, temos a solução da equação de Bellman em cada vértice.

Para os estados onde $v(x) = g(x)$ são estados onde é ótimo parar e receber o retorno deste estado.

O valor de $v(1)$ é interpretado como valor do jogo, ou seja, jogando a partir do estado 1, em média a esperança de ganho é aproximadamente $v(1)$.

Utilizando o programa (em anexo) temos que o valor do jogo é 12.0038. Testes foram realizados com a estratégia "um passo à frente" e com a estratégia de parar onde $v(x) = g(x)$, rodando 10000 vezes a estratégia *um passo à frente* (manda parar se g(x) maior ou igual à média dos vizinhos) à média dos prêmios foi de 7.464667. Já a de parar onde $v(x) = g(x)$, rodando 10000 a média foi 11.983.

Obs: O código do programa de testes está em anexo (Anexo C).

7.2.3 Solução Ótima para o problema da Secretária

Em nossos estados iniciais a informação que nós temos em cada momento é $(y_1, y_2,..., y_k)$, onde y_i é o posto relativo de uma secretária que aparece no momento $i = 1, 2,..., n$. Assim os estados inicias da cadeia de Markov são as tuplas (y_1), (y_1, y_2), ..., $(y_1, y_2, ...,y_n)$. Ainda temos o retorno de cada estado:

$g(y_1,\ldots,y_k) = k/n$ se $y_k = 1$

$g(y_1, ..., y_k) = 0$ se $y_k \neq 1$

Como mostrado na figura a seguir, os demais retornos não informados na figura, valem zero.

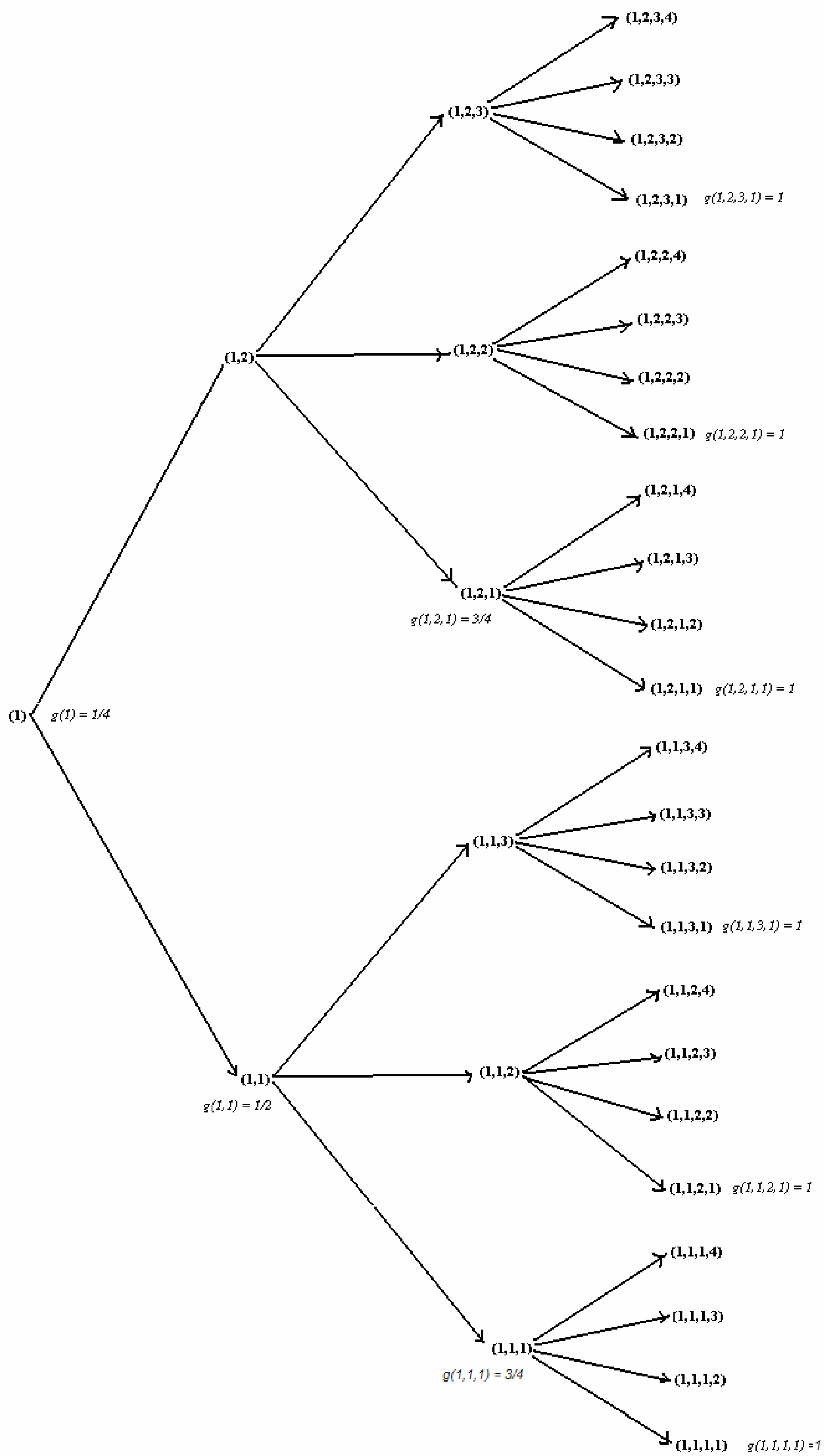


FIGURA 13: Grafo com os estados iniciais do problema da secretária (para 4 secretárias).

Agora vamos introduzir o modelo $M_2 = (X_2, p_2(x, y), g)$ com estados $X_2 = \{(i, k), i = 0,1; k = 1, ..., n\}$, as probabilidades de transições $p_2((j, k-1), (i, k)) = 1/k$ se $i = 1$ e $p_2((j, k-1), (i, k)) = (k-1)/k$ se $i = 0$, e com a função retorno $g(i, k) = k/n$ se $i = 1$ e zero caso contrario. Assim temos o novo grafo que representa o problema da secretária.

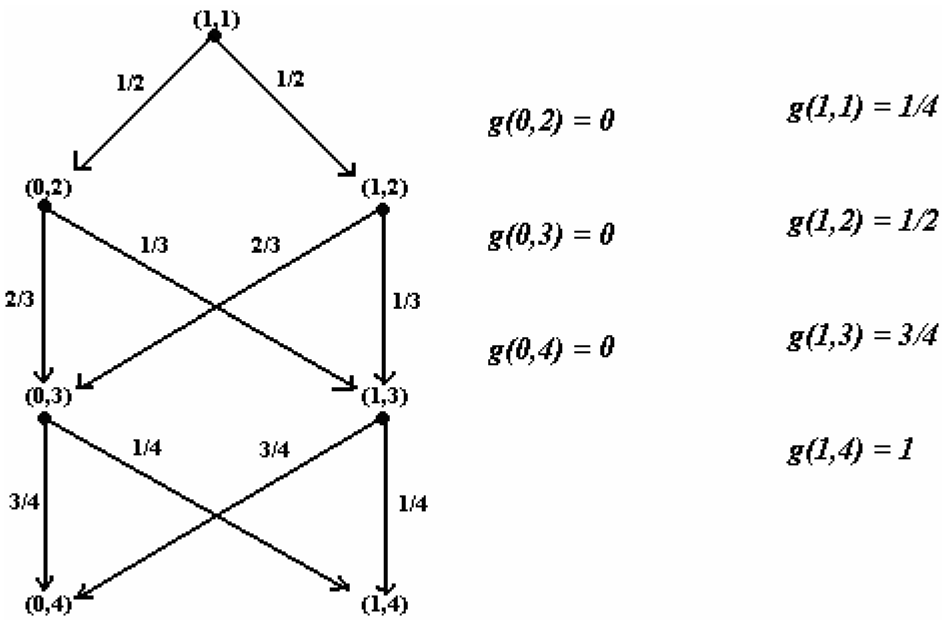


FIGURA 14: Segundo modelo de grafo para o problema da secretária.

Deste modelo um novo modelo M_3 é introduzido, restando os estados com a forma $(1,k) = k$, $k = 1, 2, ..., n$; que é conseguido utilizando o algoritmo de eliminação para eliminar todos os estados de M_2 onde $T_2g(x) > g(x)$. (Sonin 1999).

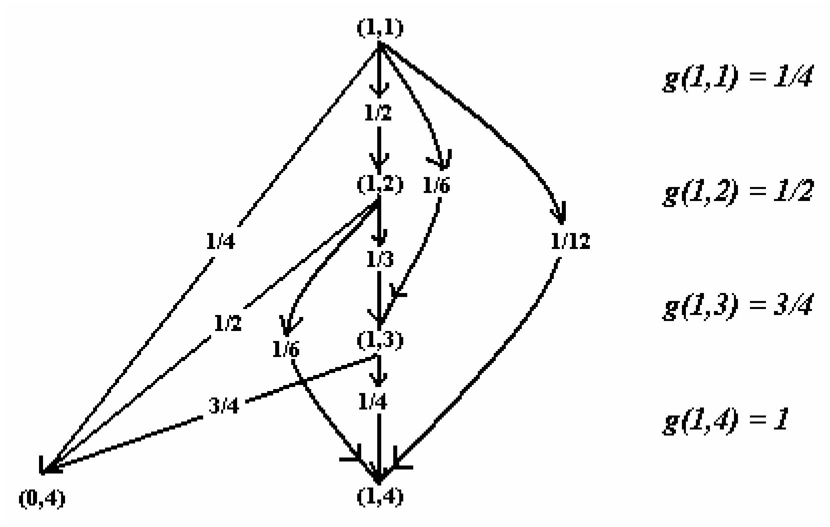


FIGURA 15: Terceiro modelo de grafo para o problema da secretária.

Utilizando o procedimento Backward (na ordem inversa) para encontrar a solução da equação de Bellman, $v(x) = \max \{ g(x), Tv(x) \}$, temos:

$v(1,4) = 1$

$v(1,3) = \max\{ ¾, 1/4 * 1 \} = ¾$

$v(1,2) = \max\{ ½, ((1/3 * ¾)+ (1/6*1)) = \max\{ ½, 5/12 \} = ½$

$v(1,1) = \max\{ ¼, ((1/2 * ½) + (1/6 * ¾) + (1/12 * 1)) = \max\{ ¼, 11/24 \} = 11/24$

Os pontos de parada ótima são os pontos onde $v(x) = g(x)$. Com isso, para um exemplo onde temos 4 secretárias é ótimo parar onde temos um retorno de ½, ¾, e 1.

Comparação entre a os resultados da simulação do programa implementado, com a probabilidade de a melhor estratégia possível obter sucesso:

Tabela 1: Simulação do programa implementado

Número de Teste	Número de Secretárias(n)	Probabilidade de a Estratégia Ótima Obter Sucesso, exposto em Landim (1983). (%)	Porcentagem de Sucesso com a Simulação do Programa Implementado. (%)
10000	20	38,42	38,25
10000	50	37,42	37,51
10000	100	37,10	36,81

1000	500	36,82	37,70
1000	1000	36,81	36,20

8 CONCLUSÕES

O algoritmo de eliminação de estados em cadeias de Markov se revelou eficiente em diminuir a complexidade computacional do problema da secretária, além de se revelar de fácil implementação.

Os resultados dos testes realizados pelo programa implementado aproximam-se dos obtidos pela estratégia ótima Landim (1983).

9 REFERENCIAL BIBLIOGRÁFICO

BISWAS, S. **Applied stochastic processes: a biostatistical and population oriented approach**. New York: Jonh Wiley & Sons, 1995. 427 p.

BOLDRINI J. L., Costa S. I., Figueiredo V.L., Wetzler H. G. **Álgebra linear 3º Edição,** Departamento de Matemática da Universidade Estadual de Campinas – UNICAMP. 1980.

BRIGHENTI, C. R. G**. Um Método de Seleção por Quantil em Problema de Inspeção Seqüencial,** Tese de Mestrado, UFLA, 2003.

DAVIS, M. D. **Teoria dos jogos: uma introdução não técnica**. São Paulo:Cultrix, 1973. 214p.

FERRARI P. **Passeios Aleatórios e Redes Elétricas**. 16º Colóquio Brasileiro de Matemática, 1987.

FREEMAN, P. R. **The secretary problem and its extensions: A review. International Ameriacan Statistics Association, Washington, v. 61, n. 313, p. 35-73, Mar. 1966.**

KOLMAN B. **Introdução à Álgebra Linear com Aplicações**, 6a. Edição, 1999.

LANDIM, C. **Otimização estocástica. In: COLÓQUIO BRASILEIRO DE MATEMÁTICA, 18., 1993. Anais... IMPA, 1993.**

LINDLEY, D.V. **Dynamic programming and decision theory**. Journal of the Royal Statistical Society Serie C – Applied Statistics, London, v. 10, n. 1, p. 39-51, 1961

SONIN I. **The State Reduction and Related Algorithms and Their Applications to the Study of Markov Chains, Graph Theory, and the Optimal Stopping Problem,** Advances in Mathematics 145, 159_188 (1999).

ANEXOS

ANEXO A Código em R, do programa que elimina os vértices 1 e 2 do grafo da sessão 7.1

```
# Elimina os vertices 1 e 2 da matriz 5X5( no arquivo matriz5x5.txt )

remove(list=ls())

numero_vertice <- 5

y <- rep(0,numero_vertice*numero_vertice)

M1 <- matrix(scan('matriz5x5.txt'),ncol=5,byrow=F)

M2 <- matrix(y,ncol=5)

for( x in 3:5 ){
      for( y in 3:5 ){
            M2[x,y] = M1[x,y] + sum( M1[x,1:2] * ( ( 1 - ( M1[1,1:2] +
M1[2,1:2] ) ) ^-1 ) * M1[1:2,y] )
      }
}

print(M1)

print(M2)
```

ANEXO B Código em R, do programa que resolve o problema da parada ótima utilizando a solução direta da equação de Bellman e o método de iteração. Para uma Cadeia de Markov finita e que satisfaz a seguinte propriedade $P(x,y) = 0$ se $y \leq x$

```
tamanho_grafo <- #ENTRAR COM O NUMERO DE ESTADOS DO GRAFO


matriz <- matrix(scan('grafo.r'),ncol=tamanho_grafo,byrow=F)

premios <- scan('premios.r')

Tg <- rep(0,tamanho_grafo)

v <- rep(0,tamanho_grafo)

v <- rep(0,tamanho_grafo)

# Armazena medias de cada vertice no vetor Tg

for( i in 1:tamanho_grafo )
{
  for( j in 1:tamanho_grafo )
    Tg[i] <- (Tg[i] + matriz[i,j] * premios[j])
}

# Equacao de Belman usando backward
v[tamanho_grafo] <- premios[tamanho_grafo]
aux <- 0
for( i in (tamanho_grafo-1):1 )
{
  for( j in (i+1):tamanho_grafo )
  {
     aux <- aux + matriz[i,j]*v[j]
  }
  if(premios[i] >= aux)
    {
      v[i] <- premios[i]
    }
```

```
    else
    {
      v[i] <- aux
    }

    aux <- 0
}

# Usando metodo da intercao

# Armazena medias de cada vertice no vetor Tgx
Tgnx <- rep(0,tamanho_grafo)
for( i in 1:tamanho_grafo )
{
  for( j in 1:tamanho_grafo )
    Tgnx[i] <- (Tgnx[i] + matriz[i,j] * premios[j])
}

# Armazena maior entre tgnx e premio em gnx
gnx <- rep(0,tamanho_grafo)
for( i in 1:tamanho_grafo )
{
  if( Tgnx[i] >= premios[i] )
    gnx[i] <- Tgnx[i]
  else
    gnx[i] <- premios[i]
}

count <- 1
while( count <= 100 )
{
  # Armazena medias de gnx vetor Tgx
  Tgnx <- rep(0,tamanho_grafo)
  for( i in 1:tamanho_grafo )
  {
    for( j in 1:tamanho_grafo )
      Tgnx[i] <- (Tgnx[i] + matriz[i,j] * gnx[j])
  }

  # Armazena maior entre tgnx e premio em gnx
```

```
  for( i in 1:tamanho_grafo )
  {
    if( Tgnx[i] >= premios[i] )
      gnx[i] <- Tgnx[i]
    else
      gnx[i] <- premios[i]
  }
  count <- count + 1
}

print(matriz)
print(premios)
print(gnx)
print(v)
```

ANEXO C Código em R, programa que encontra o valor do jogo, realiza a estratégia um passo à frente e a estratégia de parar onde $v(x) = g(x)$. Para o grafo da figura da sessão 7.3,

```
numVertices <- 31

M1 <- array(0:0, dim=c(numVertices,numVertices+2))

numFolhas <- (numVertices + 1)/2

#Probabilidades de ir para os vertices filhos
M1[1,2] = 0.4
M1[1,3] = 0.6
M1[2,4] = 0.3
M1[2,5] = 0.7
M1[3,6] = 0.3
M1[3,7] = 0.7
M1[4,8] = 0.4
M1[4,9] = 0.6
M1[5,10] = 0.3
M1[5,11] = 0.7
M1[6,12] = 0.5
M1[6,13] = 0.5
M1[7,14] = 0.7
M1[7,15] = 0.3
M1[8,16] = 0.5
M1[8,17] = 0.5
M1[9,18] = 0.2
M1[9,19] = 0.8
M1[10,20] = 0.9
M1[10,21] = 0.1
M1[11,22] = 0.6
M1[11,23] = 0.4
M1[12,24] = 0.2
M1[12,25] = 0.8
M1[13,26] = 0.3
M1[13,27] = 0.7
M1[14,28] = 0.1
M1[14,29] = 0.9
```

```
M1[15,30] = 0.7
M1[15,31] = 0.3

#Premios de cada vertice armazenado na coluna numVertices+1
M1[1,numVertices+1] = 0
M1[2,numVertices+1] = 2
M1[3,numVertices+1] = 5
M1[4,numVertices+1] = 10
M1[5,numVertices+1] = 4
M1[6,numVertices+1] = 5
M1[7,numVertices+1] = 7
M1[8,numVertices+1] = 10
M1[9,numVertices+1] = 15
M1[10,numVertices+1] = 5
M1[11,numVertices+1] = 1
M1[12,numVertices+1] = 2
M1[13,numVertices+1] = 7
M1[14,numVertices+1] = 3
M1[15,numVertices+1] = 4

M1[16,numVertices+1] = 1
M1[17,numVertices+1] = 50
M1[18,numVertices+1] = 5
M1[19,numVertices+1] = 11
M1[20,numVertices+1] = 2
M1[21,numVertices+1] = 7
M1[22,numVertices+1] = 8
M1[23,numVertices+1] = 20
M1[24,numVertices+1] = 20
M1[25,numVertices+1] = 10
M1[26,numVertices+1] = 15
M1[27,numVertices+1] = 3
M1[28,numVertices+1] = 14
M1[29,numVertices+1] = 11
M1[30,numVertices+1] = 12
M1[31,numVertices+1] = 18

#-----------------------------------------------------
#   Valor do jogo na coluna numVertices+2 (BackWard)
#-----------------------------------------------------
```

```
#for(i in (numFolhas-1):1)
i <- numVertices
while( i >= 1 )
{
    M1[i,numVertices+2] = M1[i,numVertices+1]
    i <- i - 1
}
#for(i in (numFolhas-1):1)
i <- numFolhas-1
while( i >= 1 )
{
    if( ((M1[i,i*2] * M1[i*2,numVertices+2])+(M1[i,(i*2)+1] * M1[(i*2)+1,
    numVertices+2])) > M1[i,numVertices+2] )
    {
           M1[i,numVertices+2] <- (M1[i,i*2] * M1[i*2,numVertices+2])+
       (M1[i,(i*2)+1] * M1[(i*2)+1,numVertices+2])
    }
    i <- i - 1
}

precoDoJogo = M1[1,numVertices+2]

#-------------------------------------------------------
# Simulando 3000 vezes com a estrategia um ponto a frente
#-------------------------------------------------------

premios <- rep(0,10000)

for(i in 1:3000) {
  id <- 1
  while( id < numFolhas )
  {
    if( M1[id,numVertices+1] < ((M1[id,id*2]*M1[id*2,numVertices+1]) +
    (M1[id,(id*2)+1]*M1[(id*2)+1,numVertices+1])))
    {
      x <- sample(101, 1, replace = FALSE, prob = NULL) - 1
      if( x <= (M1[id,id*2]*100) )
      {
        id <- id*2
```

```
        tempId <- id
      }
      else
      {
        id <- ((id*2)+1)
        tempId <- id
      }
    }
    else
    {
      premios[i] <- M1[id,numVertices+1]
      tempId <- id
      id <- numFolhas
    }
  }
  premios[i] = M1[tempId,numVertices+1]
}

mediaUpf <- sum(premios)/10000
vetorUpf <- premios

#------------------------------------------------------
# Estrategia depois de saber o preco do jogo pelo BackWard
#------------------------------------------------------

premios <- rep(0,10000)

for(i in 1:3000) {
  id = 1
  while( id < numFolhas )
  {
    if( M1[id,numVertices+1] < M1[id,numVertices+2] )
    {
      x <- sample(101, 1, replace = FALSE, prob = NULL) - 1
      if( x <= (M1[id,id*2]*100) )
      {
        id <- id*2
        tempId <- id
      }
      else
```

```
        {
          id <- ((id*2)+1)
          tempId <- id
        }
      }
      else
      {
        premios[i] <- M1[id,numVertices+1]
        tempId <- id
        id <- numFolhas
      }
    }
    premios[i] <- M1[tempId,numVertices+1]
  }

  mediaBackW <- sum(premios)/10000
  vetorBackW <- prêmios
```

ANEXO D Código em R, do programa que implementação a solução ótima para o problema da secretária,

```
numero_secretarias <- NUMERO_DE_SECRETARIAS # DEVE SER UM INTEIRO


tamanho_grafo <- numero_secretarias*2

#--CRIA MATRIZ M2--------------------------------------------------------
-------
M2 <- array(0:0, dim=c(tamanho_grafo,tamanho_grafo))
g2 <- rep(0,tamanho_grafo/2)
for(i in 1:((tamanho_grafo/2)-1)){
  M2[i,i+1] <- (i/(i+1))
  M2[i,(tamanho_grafo/2)+i+1] <- 1/(i+1)
  g2[(tamanho_grafo/2)+i] <- i/(tamanho_grafo/2)
}
g2[tamanho_grafo] <- 1

for(j in ((tamanho_grafo/2)+1):(tamanho_grafo-1)){
  M2[j,j+1] <- 1/((j-1)-(tamanho_grafo/2)+2)
  M2[j,j-(tamanho_grafo/2)+1] <- (((j-1)-(tamanho_grafo/2)+2)-1)/((j-1)-
  (tamanho_grafo/2)+2)
}
#-----------------------------------------------------------------------
-------

#--CRIA MATRIZ M3--------------------------------------------------------
-------
M3 <- array(0:0, dim=c(tamanho_grafo/2,tamanho_grafo/2))

for( z in 1:((tamanho_grafo/2)-1) ){
  for(l in (z+1):(tamanho_grafo/2))
    M3[z,l] <- z/((l-1)*l)
}
g3 <- rep(0,(tamanho_grafo/2))
for(i in 1:(tamanho_grafo/2))
  g3[i] <- g2[i+(tamanho_grafo/2)]
```

```
#---------------------------------------------------------------------------
-------

#--Equacao de Belman usando backward----------------------------------------
-------
v3 <- g3
aux <- 0
for( i in ((tamanho_grafo/2)-1):1 )
{
  for( j in (i+1):(tamanho_grafo/2) )
  {
     aux <- aux + M3[i,j]*v3[j]
  }
  if(g3[i] >= aux)
    {
      v3[i] <- g3[i]
    }
    else
    {
      v3[i] <- aux
    }

    aux <- 0
}
#---------------------------------------------------------------------------
-------

#--PEGA VALORES DE G3 QUE E IGUAL V3 (ONDE E OTIMO PARAR)----------------
-------
t <- 1
otimo <- rep(0,1)
for(i in 1:(tamanho_grafo/2))
{
  if(g3[i] == v3[i]){
    otimo[t] <- g3[i]
    t <- t+1
  }
}
tamanho_otimo <- (t - 1)
```

```
#-----------------------------------------------------------------------
-------

#--FAZ 10000 TSTES------------------------------------------------------
-------
contador <- 0
for( i in 1:1000 ){

  secretaria <- rep(0,(numero_secretarias))

  x <- sample(numero_secretarias,numero_secretarias,replace = FALSE,prob
= NULL)

  for(i in 1:(numero_secretarias))
    secretaria[i] <- x[i]

  secretaria_contratada <- 0
  for(i in 1:(numero_secretarias))
  {
    ultimo <- 1
    for(j in 1:i)
    {
      if(secretaria[i] > secretaria[j])
        ultimo <- 2
    }
    if( ultimo == 1 ){
      for( k in 1:tamanho_otimo ){
        if( (i/numero_secretarias) == otimo[k])
        {
          secretaria_contratada <- i
          break
        }
        if( secretaria_contratada != 0 )
          break
      }
      if( secretaria_contratada != 0 )
        break
    }
    if( secretaria_contratada != 0 )
```

```
      break
    }
  if(secretaria_contratada == 0)
    secretaria_contratada <- numero_secretarias

  if( secretaria[secretaria_contratada] == 1)
    contador <- contador+1
}

print(contador) #IMPRIME NUMERO DE SUCESSOS EM 10000
```