# 5
# Política de Tempo Igualmente Espaçada por Servidor

Este capítulo investiga uma política de tempo onde as requisições a um mesmo servidor são igualmente espaçadas no tempo. A página do servidor que será revisitada em cada uma destas requisições deve ser escolhida por uma *política de seleção de páginas*. Consideramos neste estudo duas políticas de seleção de páginas, chamadas MERGE e RANDOM. A política MERGE adota a mesma sequência de páginas revisitadas pela política igualmente espaçada por página, motivada pelo fato desta política ser ótima quando não temos a restrição de *politeness*. A política RANDOM sorteia uma página com chance proporcional à sua frequência de revisitação. Políticas semelhantes à política RANDOM foram propostas na literatura (Cho03a, Cof98), e portanto ela é considerada aqui para efeito de comparação.

Uma vantagem da política de tempo igualmente espaçada por servidor é o fato da restrição de *politeness* poder ser definida utilizando apenas as frequências de revisitações das páginas, tornando mais simples a formulação da alocação de recursos. A frequência $f_s$ de requisições a um servidor $s$ é a soma das frequências de revisitação das páginas hospedadas em $s$. Portanto, na política de tempo igualmente espaçada por servidor, o intervalo entre requisições consecutivas ao servidor $s$ tem sempre duração $1/f_s$. Como a restrição de *politeness* impõe um tempo mínimo $P$ entre requisições consecutivas a um mesmo servidor, para atender esta restrição basta que $f_s \leq 1/P$. Esta condição corresponde à restrição (4-4) adicionada na formulação do limite superior UNPOLITE para produzir o limite superior POLITE.

Fatores de aproximação para o *freshness* do repositório são fornecidos neste capítulo para o caso em que empregamos a política de tempo igualmente espaçada por servidor, juntamente com as política de seleção de página MERGE ou RANDOM. Pela Proposição 5.1 apresentada a seguir, temos que o fator de aproximação para o *freshness* das páginas também é um fator de aproximação para o *freshness* do repositório. Portanto, a análise foca no caso mais simples que consiste em encontrar fator de aproximação para o *freshness* das páginas. Esta análise fornece expressões para o fator de aproximação

das políticas MERGE e RANDOM. Utilizando estas expressões, obtemos limite inferior 0,77 para o fator de aproximação da política RANDOM, e apresentamos uma conjectura de que 0,927 é um limite inferior para o fator de aproximação da política MERGE.

**Proposição 5.1** *O fator de aproximação para o* freshness *das páginas é um fator de aproximação para o* freshness *do repositório, independente das importâncias das páginas e da alocação de recursos adotada.*

*Prova.* Fixando uma política de tempo $\mathcal{P}$ e uma alocação de recursos $(f_1, \ldots, f_{|R|})$ para o repositório $R$, o fator de aproximação para o *freshness* deste repositório é definido como $\beta(f_1, \ldots, f_{|R|}) = \sum_{i \in R} w_i A_i(f_i) / \sum_{i \in R} w_i A_i^*(f_i)$, onde $A_i(f_i)$ é o *freshness* da página $i$ fornecido pela política $\mathcal{P}$ quando esta página é revisitada com frequência $f_i$, $A_i^*(f_i)$ é um limite superior para o *freshness* da página $i$ para toda política que revisita a página $i$ com frequência $f_i$, e $w_i$ é a importância da página $i$.

Suponha que para um dado $0 < \alpha \leq 1$, o *freshness* $A_i(f_i)$ da página $i$ fornecido pela política $\mathcal{P}$ é tal que $A_i(f_i) \geq \alpha A_i^*(f_i)$, para toda página $i$ e qualquer frequência de revisitação $f_i \geq 0$ para a página $i$. Neste caso, temos que para qualquer alocação de recursos $(f_1, \ldots, f_{|R|})$ viável, e qualquer atribuição de importâncias às páginas,

$$\beta(f_1, \ldots, f_{|R|}) = \frac{\sum_{i \in R} w_i A_i(f_i)}{\sum_{i \in R} w_i A_i^*(f_i)} \geq \frac{\sum_{i \in R} w_i (\alpha A_i^*(f_i))}{\sum_{i \in R} w_i A_i^*(f_i)} = \alpha. \qquad (5\text{-}1)$$

Note que para determinar um fator de aproximação para o *freshness* do repositório independente da alocação de recursos, devemos considerar uma alocação viável que maximiza o limite superior $\sum_{i \in R} w_i A_i^*(f_i)$. Porém, esta alocação de recursos é um caso particular dentre as alocações consideradas na Equação (5-1). ■

As políticas RANDOM e MERGE são avaliadas experimentalmente através da simulação de 6 anos de operação do *crawler* para manter o nível de atualização do repositório WEBBASE, descrito no Capítulo 3. Os resultados experimentais são apresentados na Seção 5.5. Estes resultados confirmam os limites inferiores para os fatores de aproximação do *freshness* do repositório. Desta simulação podemos concluir que as políticas DELAYED e MERGE fornecem praticamente o mesmo *freshness*, enquanto o *freshness* obtido pela política RANDOM é inferior ao obtido pelas políticas DELAYED e MERGE, principalmente para frequências mais baixas de revisitação.

É importante que a política de revisitação evite picos de utilização do canal de comunicação, conforme discutido no Capítulo 1. Uma análise neste sentido é feita na Seção 5.4. Esta seção fornece um limite superior

**Pré-condições:**

– Execução da alocação de recursos, definindo a frequência de requisições $f_s$ de cada servidor $s$.

**Inicialização:**

– Crie uma *heap* $H$ contendo todos os servidores do repositório, onde a chave $t_s$ de cada servidor $s$ é escolhida uniformemente no intervalo $[0, f_s^{-1})$. O topo da *heap* $H$ contém o servidor com menor chave.

– Inicialize a política de seleção de páginas.

**Operação do *crawler*:**

1. Execute os passos abaixo no início da operação e sempre que uma revisitação for realizada:
   (a) Remova da *heap* $H$ o servidor $s$ com menor chave $t_s$.
   (b) Execute a política de seleção de páginas para determinar a página $p$ do servidor $s$ a ser revisitada.
   (c) Escalone a revisitação da página $p$ para o instante $t_s$.
   (d) Insira o servidor $s$ na *heap* $H$ com chave $t_s + f_s^{-1}$.

Figura 5.1: Pseudo-código da política de tempo igualmente espaçada por servidor.

para a probabilidade de ocorrer mais de $x$ revisitações acima da média em um intervalo de tempo com duração $t$, quando as revisitações/requisições de $n$ elementos são igualmente espaçadas no tempo. Estes elementos podem ser páginas ou servidores, dependendo da política de tempo adotada. Este limite superior obtido é inversamente proporcional à $n$, favorecendo portanto a política de tempo igualmente espaçada por servidor, pois o número de servidores é muito menor que o número de páginas.

Apresentamos a seguir a definição da política de tempo igualmente espaçada por servidor, e um algoritmo eficiente para esta política.

## 5.1 Política de Tempo Igualmente Espaçada por Servidor

A política de tempo igualmente espaçada por servidor estabelece que todos os intervalos entre requisições consecutivas a um mesmo servidor $s$ têm a mesmo duração $f_s^{-1}$, onde $f_s$ é a frequência de requisições ao servidor $s$. Se $f_s \leq 1/P$, onde $P$ é o tempo mínimo permitido entre requisições consecutivas a um mesmo servidor, então a política igualmente espaçada por servidor satisfaz a restrição de *politeness* no servidor $s$.

**Definição 5.2 Política de Tempo Igualmente Espaçada por Servidor:** *para cada servidor $s$, todo intervalo entre requisições consecutivas ao servidor $s$ tem duração $f_s^{-1}$, onde $f_s$ é a soma das frequências de revisitação das páginas hospedadas no servidor $s$. O instante da primeira requisição de cada servidor $s$ é uma variável aleatória uniformemente distribuída no intervalo $[0, f_s^{-1})$, onde o instante 0 corresponde ao início de operação do* crawler*. A página escolhida em cada requisição ao servidor é definida por uma* **Política de Seleção de Páginas**.

A política de tempo igualmente espaçada por servidor pode ser implementada de forma eficiente utilizando uma *heap* de servidores, como ilustra o pseudo-código na Figura 5.1. No topo da *heap* temos o próximo servidor $s$ a receber uma requisição, cuja prioridade é o instante $t_s$ em que esta requisição deve ser realizada. Quando a requisição é feita ao servidor $s$ no instante $t_s$, este servidor retorna para a *heap* com prioridade $t_s + f_s^{-1}$. Além disso, neste instante é necessário executar a política de seleção de páginas para determinar qual página do servidor $s$ será revisitada. A etapa de inicialização constrói uma *heap* de servidores, e portanto tem complexidade da ordem do número de servidores que hospedam páginas do repositório. Devido ao reposicionamento do servidor na *heap*, o escalonamento de cada revisitação é da ordem do logaritmo do número de servidores, mais o tempo de execução da política de seleção de páginas.

O nível de atualização do repositório fornecido pela política de tempo igualmente espaçada por servidor depende da política de seleção de páginas adotada. A seguir, duas políticas de seleção de páginas são avaliadas.

## 5.2 Política de Seleção de Páginas RANDOM

Em cada requisição a um servidor $s$, a política de seleção de páginas RANDOM sorteia uma página $i$ do servidor $s$ com chance proporcional à sua frequência de revisitação.

**Definição 5.3 Política de Seleção de Páginas RANDOM:** *uma página do servidor $s$ é sorteada, onde a probabilidade de uma página $i$ ser escolhida é proporcional a sua frequência de revisitação $f_i$. Ou seja, a probabilidade de sortear a página $i$ do servidor $s$ vale $f_i/f_s$, a razão entre a frequência de revisitação da página $i$ e a frequência de requisições ao servidor $s$.*

A política de seleção de páginas RANDOM pode ser implementada de forma eficiente utilizando o algoritmo proposto em (Wal77) para gerar observações de uma variável aleatória discreta. Desta forma, cada seleção de

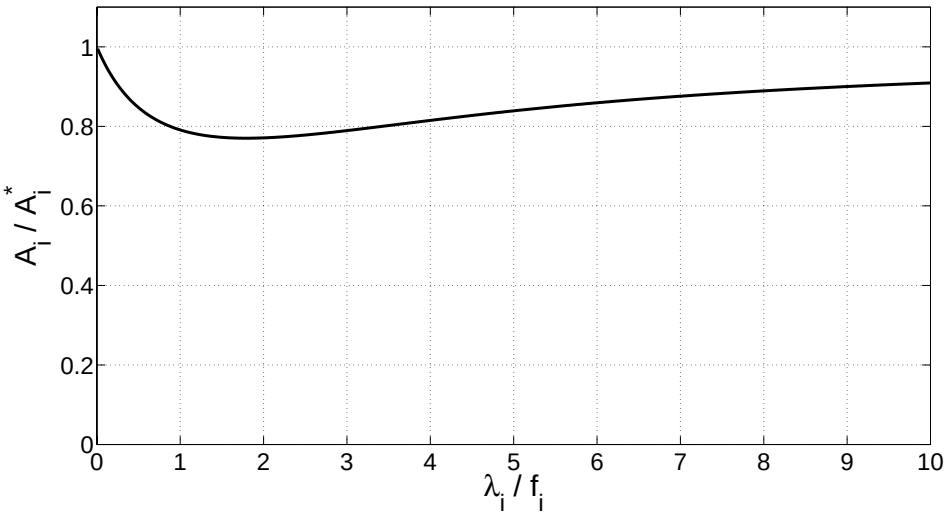


Figura 5.2: Fator de aproximação para o *freshness* de uma página $i$ modificada com taxa $\lambda_i$, e revisitada com frequência $f_i$ pela política RANDOM.

página é feita em $O(1)$. Este algoritmo exige o armazenamento de duas tabelas de *lookup* com tamanho da ordem do número de páginas do servidor. Um pré-processamento para preencher estas tabelas também é necessário, exigindo um tempo de execução quadrático no número de páginas do servidor durante a inicialização da política.



### Fator de Aproximação para uma Página

O Teorema 5.4 fornece o *freshness* que pode ser obtido com a política de seleção de páginas RANDOM, e o Corolário 5.5 fornece um fator de aproximação para este *freshness*.

**Teorema 5.4** *Seja $s$ um servidor que recebe requisições igualmente espaçadas com frequência $f_s$. Se a página revisitada em cada requisição ao servidor $s$ é escolhida de acordo com a política de seleção de páginas RANDOM, então o* freshness *de uma página $i$ do servidor $s$ que é modificada com taxa $\lambda_i$ e é revisitada com frequência $f_i$ é dado por*

$$A_i = \frac{f_i}{\lambda_i}\left(\frac{\exp(\lambda_i/f_s)-1}{f_i/f_s+\exp(\lambda_i/f_s)-1}\right). \tag{5-2}$$

*Da Equação (5-2) podemos derivar o limite inferior*

$$A_i > \frac{f_i}{\lambda_i+f_i}. \tag{5-3}$$

*Prova.* Utilizando a Equação (2-3) podemos calcular o *freshness* fornecido por uma política de revisitação determinando a distribuição das durações dos intervalos entre revisitações. Neste caso, verificamos que um intervalo arbitrário

$U_i$ entre requisição consecutivas da página $i$ tem duração $f_s^{-1}K_i$, onde $K_i$ é uma variável aleatória geométrica com parâmetro $p_i = f_i/f_s$, cuja função geratriz de momentos é dada por $M_{K_i}(x) = p_i/(e^{-x} + p_i - 1)$. Portanto, utilizando a propriedade $M_{aX}(x) = M_X(ax)$ (Mon03),

$$M_{U_i}(x) = M_{f_s^{-1}K_i}(x) = M_{K_i}(f_s^{-1}x) = \frac{p_i}{e^{-f_s^{-1}x} + p_i - 1}.$$

Aplicando $M_{U_i}(x)$ em $-\lambda_i$ obtemos

$$\mathrm{E}[e^{-\lambda_i U_i}] = M_{U_i}(-\lambda_i) = \frac{p_i}{e^{\lambda_i/f_s} + p_i - 1}.$$

Substituindo $\mathrm{E}[e^{-\lambda_i U_i}]$ na Equação (2-3) obtemos o *freshness*

$$A_i = \frac{1 - \mathrm{E}[e^{-\lambda_i U_i}]}{\lambda_i/f_i} = \frac{f_i}{\lambda_i}\left(\frac{\exp(\lambda_i/f_s) - 1}{f_i/f_s + \exp(\lambda_i/f_s) - 1}\right).$$

Como $\exp(x) > x + 1$ para $x > 0$, temos

$$A_i = \frac{f_i}{\lambda_i}\left(\frac{1}{\frac{f_i}{f_s(\exp(\lambda_i/f_s)-1)} + 1}\right) > \frac{f_i}{\lambda_i}\left(\frac{1}{\frac{f_i}{f_s(\lambda_i/f_s+1-1)} + 1}\right) = \frac{f_i}{\lambda_i + f_i}.$$

■

O *freshness* obtido pela política RANDOM é igual ao obtido pela política *purely-random* proposta em (Cho03a). Entretanto, o Teorema 5.4 fornece uma expressão para o *freshness* mais detalhada que a fornecida em (Cho03a). Na análise do *freshness* para a política *purely-random*, Cho e Garcia-Molina (Cho03a) consideram o limite do *freshness* para o número de requisições tendendo ao infinito, ou seja, $\lim_{f_s\to\infty} A_i$. Utilizando a Equação (5-2) como valor de $A_i$, temos que $\lim_{f_s\to\infty} A_i$ é igual ao limite inferior para o *freshness* apresentado na Equação (5-3). Este limite inferior é igual à expressão fornecida em (Cho03a), e reproduzida no Teorema 2.2, para o *freshness* da política *purely-random*. Portanto, o Teorema 5.4 fornece uma expressão mais precisa para o *freshness* de uma página $i$ quando a frequência de revisitação da página $i$ não é insignificante quando comparada com a frequência de requisições ao servidor que hospeda $i$.

**Corolário 5.5** *Se a página revisitada em cada requisição a um servidor s é escolhida de acordo com a política de seleção de páginas RANDOM, então temos o fator de aproximação abaixo para o* freshness *de uma página i do servidor s que se modifica com taxa $\lambda_i$ e é revisitada com frequência $f_i > 0$:*

$$\frac{A_i}{A_i^*} > \frac{\lambda_i/f_i}{(1 + \lambda_i/f_i)(1 - \exp(-\lambda_i/f_i))} > 0,77.$$

*Prova.* O fator de aproximação é obtido através da razão entre o limite inferior para *freshness* $A_i$ da página $i$ fornecido pelo Teorema 5.4 (Equação (5-3)), e o limite superior $A_i^*$ para o *freshness* da página $i$ fornecido pelo Lema 4.2.

$$\begin{aligned}\frac{A_i}{A_i^*} &> \left(\frac{f_i}{\lambda_i + f_i}\right) / \left(\frac{f_i}{\lambda_i}\left(1 - \exp\left(-\frac{\lambda_i}{f_i}\right)\right)\right) \\ &= \frac{\lambda_i/f_i}{(1+\lambda_i/f_i)(1-\exp(-\lambda_i/f_i))}.\end{aligned}$$

A função $f(x) = x/((1+x)(1-\exp(-x)))$ tem apenas um ponto crítico para $x > 0$, que ocorre em $x \approx 1,793$. Fazendo $\lambda_i/f_i = 1,793$ obtemos $A_i/A_i^* > 0,77$. O gráfico do fator de aproximação $A_i/A_i^*$ em função de $\lambda_i/f_i$ é apresentado na Figura 5.2.

■

## 5.3 Política de Seleção de Páginas MERGE

Para determinar a página a ser revisitada em cada requisição ao servidor, a política MERGE utiliza a ordem com que as páginas seriam revisitadas pela política igualmente espaçada por página. Conforme apresentado no Capítulo 4, a política de tempo igualmente espaçada por página é ótima se a violação do *politeness* é permitida. Portanto, a motivação da política MERGE é adaptar uma solução ótima de modo que o *politeness* seja respeitado. Isto é feito tornando os instantes de revisitação igualmente espaçados por servidor, como ilustra a Figura 5.3. Propomos inicialmente esta política em (Sou07), para um modelo onde o tempo de operação do *crawler* é finito.

Na Figura 5.3 ilustra a política de seleção de páginas MERGE. Nesta figura temos os instantes de revisitação das páginas 1 e 2 produzidos pela política de tempo igualmente espaçada por página, onde as páginas 1 e 2 são revisitadas com frequência $f_1$ e $f_2$, respectivamente. Em cada requisição da política igualmente espaçada por servidor, com frequência $f = f_1 + f_2$, a política MERGE seleciona a próxima página que seria revisitada na solução igualmente espaçada por página. Por exemplo, assuma $f_1 = 1/40$ e $f_2 = 1/60$, e a primeira revisitação das páginas 1 e 2 nos instantes 10 e 0, respectivamente. Neste caso, as 3 primeiras revisitações produzidas pela política igualmente espaçada por página ocorrem nos instantes 10, 50 e 90 para a página 1, e nos instantes 0, 60 e 120 para a página 2. Desta forma as páginas são revisitadas na ordem: página 2, página 1, página 1, página 2, página 1, página 2. A política MERGE aproveita esta ordem de revisitação, mas o intervalo entre requisições a este

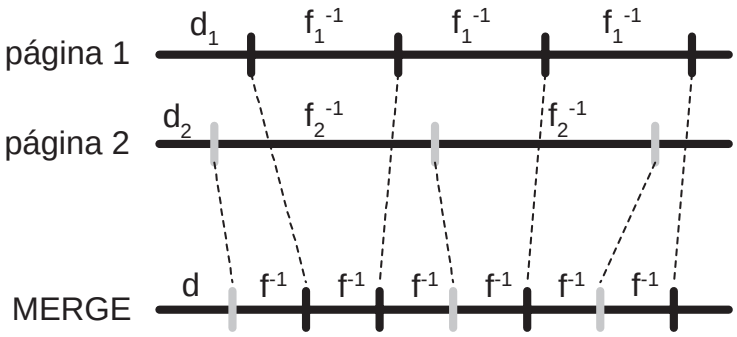


Figura 5.3: Exemplo de aplicação da política de seleção de páginas MERGE. Duas páginas de um mesmo servidor são revisitadas com frequências $f_1$ e $f_2$. A frequência de requisições ao servidor vale $f = f_1 + f_2$. Os deslocamentos $d_1$, $d_2$ e $d$ são uniformemente distribuídos nos intervalos $[0, f_1^{-1})$, $[0, f_2^{-1})$ e $[0, f^{-1})$, respectivamente.

servidor é fixo em $1/(f_1 + f_2) = 24$ unidades de tempo. Ou seja, se a primeira revisitação (página 2) ocorre no instante $d$, então a segunda revisitação (página 1) ocorre no instante $d + 24$, a terceira (página 1) no instante $d + 48$, assim sucessivamente.

O pseudo-código da política de seleção de páginas MERGE é apresentado na Figura 5.4. Na inicialização uma *heap* é criada para cada servidor, contendo as páginas do servidor. A chave de cada página na *heap* é o instante da próxima revisitação da página se a política igualmente espaçada por página estivesse sendo utilizada. Ou seja, toda vez que uma página $i$ é retirada da *heap* com chave $t_i$, ela retorna à *heap* com chave $t_i + f_i^{-1}$, onde $f_i$ é a frequência de revisitação da página $i$. O instante real de revisitação da página é determinado pela política de tempo igualmente espaçada por servidor. A criação das *heaps* na inicialização tem complexidade de tempo da ordem do número de páginas no repositório. A seleção de uma página do servidor $s$ tem complexidade de tempo da ordem do logaritmo do número de páginas do servidor $s$, visto que a página selecionada deve ser reposicionada na *heap* de páginas do servidor $s$.

### Fator de Aproximação para uma Página

O Teorema 5.7, apresentado a seguir, fornece um limite inferior para o *freshness* de uma página $i$ quando o servidor que hospeda $i$ recebe requisições igualmente espaçadas por servidor, e a seleção de páginas é feita através da política MERGE. Um fator de aproximação baseado neste limite inferior é apresentado no Corolário 5.8. A prova do Teorema 5.7 utiliza o Lema 5.6.

**Lema 5.6** *Seja $i$ uma página revisitada com frequência $f_i > 0$ de acordo com a política de tempo igualmente espaçada por página. Então, o número de*

**Pré-condições:**

– Execução da alocação de recursos, definindo a frequência de revisitação $f_i$ de cada página $i$.

**Inicialização:**

1. Para cada servidor $s$,

   (a) Crie uma *heap* $H_s$ contendo todas as páginas do servidor $s$, onde a chave $t_i$ de cada página $i$ é escolhida uniformemente no intervalo $[0, f_i^{-1})$. O topo da *heap* $H_s$ contém a página com menor chave.

**Seleção de página do servidor $s$:**

1. Remova de $H_s$ a página $i$ com menor chave $t_i$.
2. Insira a página $i$ na *heap* $H_s$ com chave $t_i + f_i^{-1}$.
3. Selecione a página $i$ para ser revisitada.

Figura 5.4: Pseudo-código da política de seleção de páginas MERGE.

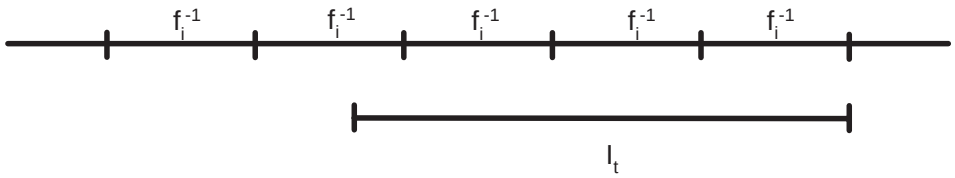


Figura 5.5: Revisitações igualmente espaçadas que ocorrem em um intervalo $I_t$ com duração $t$.

*revisitações da página $i$ que ocorrem em um intervalo arbitrário com duração $t$ é uma variável aleatória $X_i(t) = \lfloor tf_i \rfloor + Y_i(t)$, onde $Y_i(t)$ é uma variável aleatória de Bernoulli com probabilidade de sucesso $\{tf_i\} = tf_i - \lfloor tf_i \rfloor$.*

*Prova.* Seja $I_t$ um intervalo arbitrário com duração $t$ (Figura 5.5). Podemos ter $\lfloor tf_i \rfloor$ ou $\lfloor tf_i \rfloor + 1$ revisitações da página $i$ em $I_t$. Temos $\lfloor tf_i \rfloor + 1$ quando a primeira revisitação dentro de $I_t$ ocorre até $t - f_i^{-1}\lfloor tf_i \rfloor$ unidades de tempo desde o início de $I_t$.

Como a primeira revisitação gerada pela política de tempo igualmente espaçada por página é escolhida uniformemente no intervalo $[0, f_i^{-1})$, temos que a duração entre o início de $I_t$ e a primeira revisitação dentro de $I_t$ também é uniformemente distribuída em $[0, f_i^{-1})$. Portanto, a probabilidade de ocorrer $\lfloor tf_i \rfloor + 1$ revisitações em $I_t$ vale $(t - f_i^{-1}\lfloor tf_i \rfloor)/f_i^{-1} = tf_i - \lfloor tf_i \rfloor$. ■

**Teorema 5.7** *Seja $s$ um servidor que recebe requisições igualmente espaçadas com frequência $f_s > 0$. Se a página revisitada em cada requisição ao servidor $s$ é escolhida de acordo com a política de seleção de páginas MERGE, então*

*temos o seguinte limite inferior para o* freshness *de uma página $i$ do servidor $s$ que se modifica com taxa $\lambda_i > 0$ e é revisitada com frequência $f_i > 0$:*

$$A_i \geq \frac{f_i}{\lambda_i}\left(1 - \exp\left[-\frac{\lambda_i}{f_s} + \left(\exp\left(-\frac{\lambda_i}{f_s}\right) - 1\right)\left(\frac{f_s - f_i}{f_i}\right)\right]\right). \tag{5-4}$$

*Prova.* Considere um intervalo arbitrário $I_k$ entre a $k$-ésima e a $(k+1)$-ésima requisições à página $i$ durante a execução da política igualmente espaçada por servidor com política de seleção de páginas MERGE. A duração $U_{i,k}$ do intervalo arbitrário $I_k$ depende do número de revisitações de cada página $j \neq i$ do servidor $s$ que ocorrem entre a $k$-ésima e a $(k+1)$-ésima revisitações da página $i$, durante a execução da política igualmente espaçada por página (observe na Figura 5.3 o efeito nos intervalos entre revisitações da página 2 provocado pelas revisitações da página 1).

De acordo com o Lema 5.6, o número de revisitações de uma página $j \neq i$ (revisitada de acordo com a política igualmente espaçada por página com frequência $f_j > 0$) que ocorrem em um intervalo arbitrário entre revisitações da página $i$ é uma variável aleatória $X_{i,j} = X_j(f_i^{-1}) = \lfloor f_i^{-1} f_j \rfloor + Y_j(f_i^{-1})$, onde $Y_{i,j} = Y_j(f_i^{-1})$ é uma variável aleatória de Bernoulli com probabilidade de sucesso $\{f_i^{-1} f_j\}$. Portanto,

$$U_{i,k} = f_s^{-1}\left(1 + \sum_{j \neq i} X_{i,j}\right).$$

Como as variáveis aleatórias $X_{i,j}, j \neq i$, são independentes,

$$\begin{aligned} \mathrm{E}[\exp(-\lambda_i U_{i,k})] &= \mathrm{E}\left[\exp\left(-\lambda_i f_s^{-1}\left(1 + \sum_{j \neq i} X_{i,j}\right)\right)\right] \\ &= \exp(-\lambda_i f_s^{-1}) \prod_{j \neq i} \mathrm{E}[\exp(-\lambda_i f_s^{-1} X_{i,j})]. \end{aligned} \tag{5-5}$$

Denotando $r_{i,j} = f_j / f_i$, temos que $X_{i,j} = \lfloor r_{i,j} \rfloor + 1$ com probabilidade $\{r_{i,j}\}$ e $X_{i,j} = \lfloor r_{i,j} \rfloor$ com probabilidade $1 - \{r_{i,j}\}$. Portanto,

$$\begin{aligned} &\mathrm{E}[\exp(-\lambda_i f_s^{-1} X_{i,j})] \\ &= \exp(-\lambda_i f_s^{-1}(\lfloor r_{i,j} \rfloor + 1))\{r_{i,j}\} + \exp(-\lambda_i f_s^{-1} \lfloor r_{i,j} \rfloor)(1 - \{r_{i,j}\}) \\ &= \exp(-\lambda_i f_s^{-1} \lfloor r_{i,j} \rfloor)(1 - \{r_{i,j}\}(1 - \exp(-\lambda_i f_s^{-1}))) \end{aligned} \tag{5-6}$$

Denotando $\exp(-\lambda_i / f_s)$ como $a_{i,s}$, e utilizando o fato de que $1 + x \leq \exp(x)$ para todo $x$, temos que $a_{i,s} \leq \exp(a_{i,s} - 1)$ (ou seja, $x = a_{i,s} - 1$). Elevando ambos os lados de $a_{i,s} \leq \exp(a_{i,s} - 1)$ por $\lfloor r_{i,j} \rfloor$, obtemos $a_{i,s}^{\lfloor r_{i,j} \rfloor} \leq$

$\exp((a_{i,s}-1)\lfloor r_{i,j}\rfloor)$. Utilizando novamente o fato de que $1+x \leq \exp(x)$, obtemos $1-\{r_{i,j}\}(1-a_{i,s}) \leq \exp((a_{i,s}-1)\{r_{i,j}\})$ (ou seja, $x = \{r_{i,j}\}(a_{i,s}-1)$). Portanto, temos o seguinte limite superior para a Equação (5-6):

$$\begin{aligned} \mathrm{E}[\exp(-\lambda_i f_s^{-1} X_{i,j})] &= a_{i,s}^{\lfloor r_{i,j}\rfloor}(1-\{r_{i,j}\}(1-a_{i,s})) \\ &\leq \exp((a_{i,s}-1)\lfloor r_{i,j}\rfloor)\exp((a_{i,s}-1)\{r_{i,j}\}) \\ &= \exp((a_{i,s}-1)r_{i,j}) \end{aligned} \tag{5-7}$$

Substituindo (5-7) em (5-5), e utilizando o fato de que $\sum_{j\neq i} r_{i,j} = (f_s - f_i)/f_i$, obtemos

$$\begin{aligned} \mathrm{E}[\exp(-\lambda_i U_{i,k})] &\leq a_{i,s}\prod_{j\neq i}\exp((a_{i,s}-1)r_{i,j}) = a_{i,s}\exp((a_{i,s}-1)\sum_{j\neq i} r_{i,j}) \\ &= a_{i,s}\exp\left((a_{i,s}-1)\left(\frac{f_s-f_i}{f_i}\right)\right). \end{aligned}$$

Assim, pela Equação (2-3),

$$A_i = \frac{1-\mathrm{E}[\exp(-\lambda_i U_{i,k})]}{\lambda_i/f_i} \geq \frac{f_i}{\lambda_i}\left(1 - a_{i,s}\exp\left((a_{i,s}-1)\left(\frac{f_s-f_i}{f_i}\right)\right)\right).$$

■

**Corolário 5.8** *Seja s um servidor que recebe requisições igualmente espaçadas com frequência $f_s > 0$. Se a página revisitada em cada requisição ao servidor s é escolhida de acordo com a política de seleção de páginas MERGE, então temos o fator de aproximação abaixo para o* freshness *de uma página i do servidor s que se modifica com taxa $\lambda_i > 0$ e é revisitada com frequência $f_i > 0$:*

$$\frac{A_i}{A_i^*} \geq \frac{1-\exp\left[-\frac{\lambda_i}{f_s} + \left(\exp\left(-\frac{\lambda_i}{f_s}\right)-1\right)\left(\frac{f_s-f_i}{f_i}\right)\right]}{1-\exp\left(-\frac{\lambda_i}{f_i}\right)}. \tag{5-8}$$

*Prova.* O fator de aproximação da Equação (5-8) é obtido através da razão entre o limite inferior para o *freshness* $A_i$ da página $i$ fornecido pelo Teorema 5.7 (Equação (5-4)), e o limite superior $A_i^*$ para o *freshness* da página $i$ fornecido pelo Lema 4.2. ■

**Conjectura 5.9** *O fator de aproximação para o* freshness *de uma página é maior que 0,927 quando as requisições são igualmente espaçadas por servidor, e as páginas são selecionadas de acordo com a política MERGE.*

A Conjectura 5.9 baseia-se em uma avaliação numérica da Equação (5-8). Substituindo $f_s$ por $cf_i$ na Equação (5-8), e em seguida substituindo $\lambda_i/f_i$ por $r$, obtemos

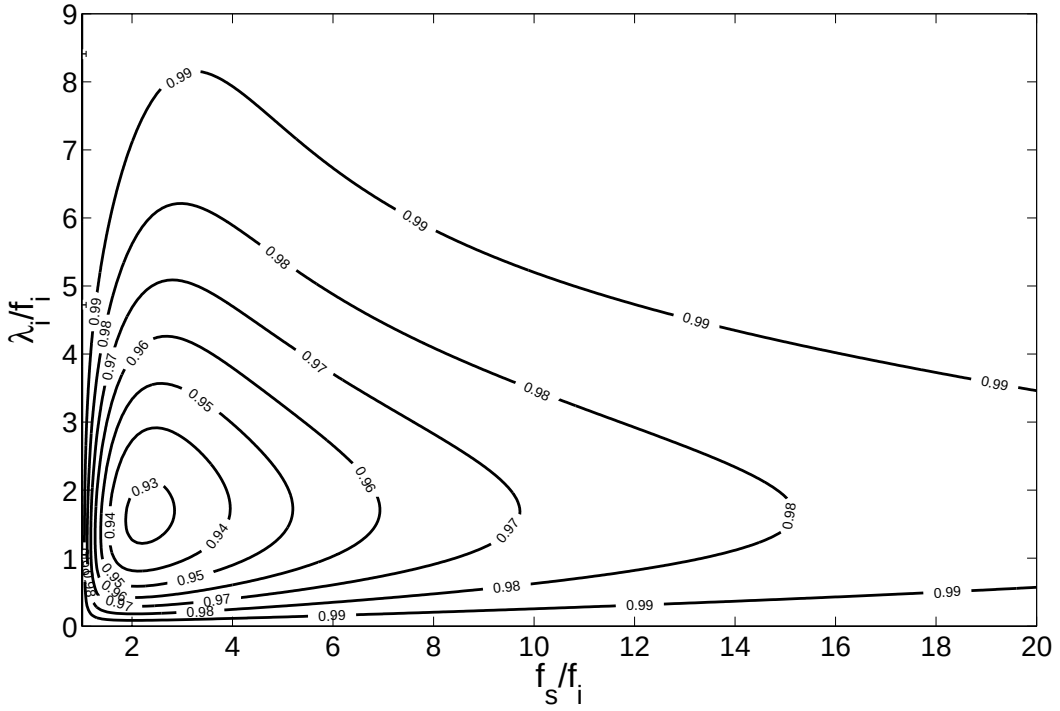


Figura 5.6: Limite inferior fornecido pela Equação (5-9) para o fator de aproximação do *freshness* do repositório quando empregamos a política de seleção de páginas MERGE.

$$\frac{A_i}{A_i^*} \geq \frac{1 - \exp\left[-\frac{r}{c} + \left(\exp\left(-\frac{r}{c}\right) - 1\right)(c-1)\right]}{1 - \exp\left(-r\right)}, \quad \text{para } r > 0 \text{ e } c \geq 1. \quad (5\text{-}9)$$

O gráfico das curvas de nível da Equação (5-9) é apresentado na Figura 5.6. Este gráfico sugere que na região $0 < r \leq 9$ e $1 \leq c \leq 20$ temos apenas um ponto de mínimo. Uma busca local nesta região forneceu um mínimo maior que $0,927$ para o fator de aproximação $A_i/A_i^*$. Para valores maiores de $r$ e $c$, na região $10 \leq r \leq 10000$ e $20 \leq c \leq 10000$, observou-se sempre um aumento do fator de aproximação com o aumento destas variáveis. Entretanto, uma avaliação mais rigorosa da Equação (5-9) deve ser feita para garantir um limite inferior para o fator de aproximação.

## 5.4 Uso Efetivo do Canal de Comunicação

O *crawler* dispõe de um ou mais canais de comunicação por onde é feito o *download* das páginas. A capacidade do canal de comunicação determina a frequência total $f$ de requisições que o *crawler* pode realizar. Se a política de revisitação produz picos de utilização do canal (congestionamento) em determinados instantes, então podem ocorrer atrasos no *download* das páginas ou até mesmo a repetição de requisições devido à perda de conexão com servidores *Web*. Portanto, políticas que distribuem as requisições mais

uniformemente no tempo promovem um uso mais efetivo do canal de comunicação.

O Lema 5.10 fornece um limite superior para a probabilidade de exceder em mais de $x$ unidades o número de revisitações esperadas em um intervalo de tempo com duração $t$, quando realizamos revisitações/requisições igualmente espaçadas de $n$ elementos. Estes elementos são páginas ou servidores, dependendo da política de tempo adotada.

**Lema 5.10** *Considere $n$ páginas/servidores com revisitações/requisições de acordo com a política de tempo igualmente espaçada por páginas/servidor. Seja $f_i$ a frequência de revisitações/requisições de cada página/servidor $i$, e $f = \sum_{i=1}^{n} f_i$ a frequência total de utilização do canal de comunicação. Se $X_t$ é uma variável aleatória que representa o número de utilizações do canal de comunicação que ocorrem em um intervalo de tempo com duração $t$, então*

$$\mathrm{E}[X_t] = tf, \tag{5-10}$$

$$\Pr[X_t > \mathrm{E}[X_t] + x] < e^x \left(\frac{n}{n+x}\right)^{n+x}. \tag{5-11}$$

*Prova.* Pelo Lema 5.6 temos que $X_t = \sum_{i=1}^{n}(\lfloor tf_i \rfloor + Y_i(t))$, onde $Y_i(t)$ é uma variável aleatória de Bernoulli com probabilidade de sucesso $\{tf_i\} = tf_i - \lfloor tf_i \rfloor$. Portanto, pela linearidade do valor esperado (Mon03),

$$\mathrm{E}[X_t] = \sum_{i=1}^{n}(\lfloor tf_i \rfloor + \mathrm{E}[Y_i(t)]) = \sum_{i=1}^{n}(\lfloor tf_i \rfloor + \{tf_i\}) = \sum_{i=1}^{n} tf_i = tf.$$

A Equação (5-11) pode ser obtida pela aplicação do método Chernoff Bound (Mot95), cujos passos são detalhados a seguir. Para simplificar a notação denotamos por $a$ a expressão $\sum_{i=1}^{n}\lfloor tf_i \rfloor$.

Aplicando a desigualdade de Markov (Mot95), para um dado $s > 0$ temos

$$\Pr[X_t > (1+\delta)\mathrm{E}[X_t]] = \Pr[e^{sX_t} > e^{s(1+\delta)\mathrm{E}[X_t]}] < \frac{\mathrm{E}[e^{sX_t}]}{e^{s(1+\delta)\mathrm{E}[X_t]}}. \tag{5-12}$$

Como as variáveis $Y_i(t)$ são independentes, e utilizando o fato de que $1 + x < e^x$ para $x > 0$, obtemos

$$\begin{aligned}\mathrm{E}[e^{sX_t}] &= e^{sa}\mathrm{E}\left[\prod_{i=1}^{n} e^{sY_i(t)}\right] = e^{sa}\prod_{i=1}^{n}\mathrm{E}\left[e^{sY_i(t)}\right] = e^{sa}\prod_{i=1}^{n}(1 + \{tf_i\}(e^s - 1)) \\ &< e^{sa}\prod_{i=1}^{n} e^{\{tf_i\}(e^s-1)} = \exp(sa + (e^s - 1)(\mathrm{E}[X_t] - a))). \end{aligned} \tag{5-13}$$

Substituindo a Equação (5-13) na Equação (5-12) obtemos

$$\Pr[X_t > (1+\delta)\mathrm{E}[X_t]] < \exp(sa + (e^s - 1)(\mathrm{E}[X_t] - a) - s(1+\delta)\mathrm{E}[X_t]). \tag{5-14}$$

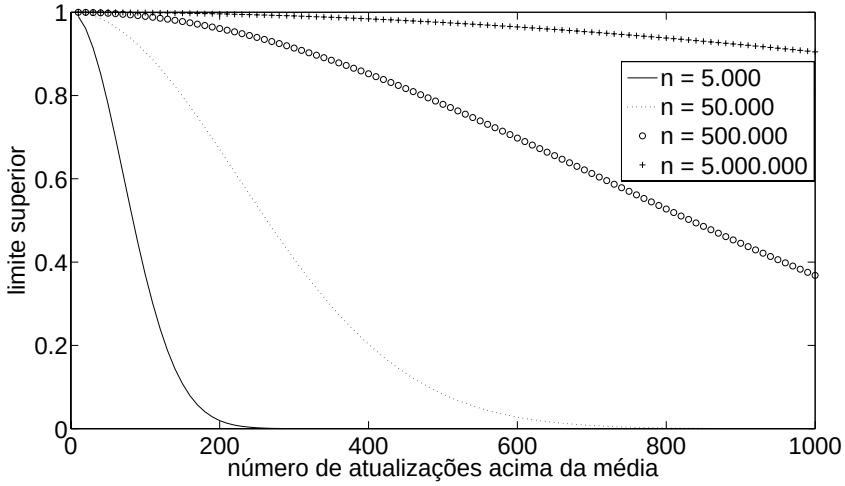


Figura 5.7: Limite superior para a probabilidade de ocorrer pelo menos $x$ requisições acima da quantidade esperada em um intervalo de tempo arbitrário. As requisições são igualmente espaçadas para cada um dos $n$ elementos.

O valor de $s$ que minimiza a Equação (5-14) é dado por

$$s^* = \ln\left(\frac{a-(1+\delta)\mathrm{E}[X_t]}{a-\mathrm{E}[X_t]}\right).$$

Portanto, substituindo $s^*$ na Equação (5-14) obtemos

$$\Pr[X_t > (1+\delta)\mathrm{E}[X_t]] < e^{\delta tf}\left(\frac{\sum_{i=1}^{n}\{tf_i\}}{\sum_{i=1}^{n}\{tf_i\}+\delta tf}\right)^{\sum_{i=1}^{n}\{tf_i\}+\delta tf}. \qquad (5\text{-}15)$$

O limite superior da Equação (5-15) aumenta se substituirmos cada ocorrência de $\sum_{i=1}^{n}\{tf_i\}$ por um mesmo valor $k$ maior que $\sum_{i=1}^{n}\{tf_i\}$. Portanto, como $\{tf_i\} \leq 1$ podemos substituir cada ocorrência de $\sum_{i=1}^{n}\{tf_i\}$ por $n$, e em seguida substituir $\delta tf$ por $x$, resultando assim na Equação (5-11).

■

Na Figura 5.7 temos o gráfico do limite superior fornecido pela Equação (5-11) em função da quantidade $x$ de requisições acima da esperada, para $n \in \{5.000, 50.000, 500.000, 5.000.000\}$. Podemos observar que para valores menores de $n$, o limite superior cai mais rapidamente com o aumento de $x$. O repositório WEBBASE possui cerca de 14,5 milhões de páginas hospedadas em 5.462 servidores. Portanto, o limite superior da Equação (5-11) é pouco informativo para a política de tempo igualmente espaçada por página, mas fornece baixa chance de congestionamento para a política igualmente espaçada por servidor quando o intervalo de tempo considerado não é muito pequeno. Por exemplo, para a política igualmente espaçada por servidor aplicada ao repositório WEBBASE, a probabilidade de ocorrer mais de 160 requisições acima da média é menor que 10% em um intervalo de tempo arbitrário $t > 0$.

Ou seja, se o canal de comunicação permite 100 requisições por segundo, então temos uma probabilidade menor que 10% de um intervalo com duração 16 segundos exceder em mais de 10% o número esperado de requisições neste intervalo.

## 5.5 Resultados Experimentais

A Figura 5.8 apresenta o resultado da simulação de seis anos de operação do *crawler* para revisitar as páginas do repositório WEBBASE. Nesta simulação são consideradas as políticas MERGE, DELAYED e RANDOM, juntamente com a alocação de recursos OPT_POLITE. A Figura 5.8 apresenta também o limite superior POLITE, definido na Seção 4.3. Como a frequência máxima de requisições a um servidor é limitada pelo inverso do tempo mínimo $P$ permitido entre requisições consecutivas, a frequência total $C$ de revisitações realizadas pelo *crawler* está limitada pelo número de servidores vezes $P^{-1}$. Nos gráficos da Figura 5.8, a frequência total $C$ foi fixada em 1%, 10% e 90% desta maior frequência permitida para $C$. As páginas que não foram modificadas durante o período de monitoramento do repositório WEBBASE são mantidas como atualizadas durante toda a simulação. As outras páginas são inicialmente consideradas como desatualizadas.

Como as políticas MERGE e DELAYED fornecem praticamente o mesmo *freshness*, elas são apresentadas com uma única linha nos gráficos da Figura 5.8. Entretanto, ao contrário da política MERGE, não fornecemos garantia de qualidade para a política DELAYED. Além disso, a política MERGE pode ser considerada tão simples em termos de implementação quanto a política DELAYED.

Por outro lado, ao final dos seis anos de operação do *crawler*, a política RANDOM perde 8,3%, 6,5% e 6,2% de *freshness* do repositório quando comparada com as política MERGE/DELAYED, para $C$ igual a 1%, 10% e 90% da frequência máxima, respectivamente.

Os resultados experimentais confirmam o fator de aproximação de 0,77 para a política RANDOM, e a conjectura de que o fator de aproximação para a política MERGE é pelo menos 0,927. Quando computamos a razão entre o *freshness* fornecido pelas políticas MERGE/RANDOM e o limite superior POLITE, obtemos 0,970, 0,975 e 0,977 para a política MERGE, e 0,812, 0,875 e 0,886 para a política RANDOM, para uma frequência total $C$ igual a 1%, 10% e 90% da frequência máxima, respectivamente.

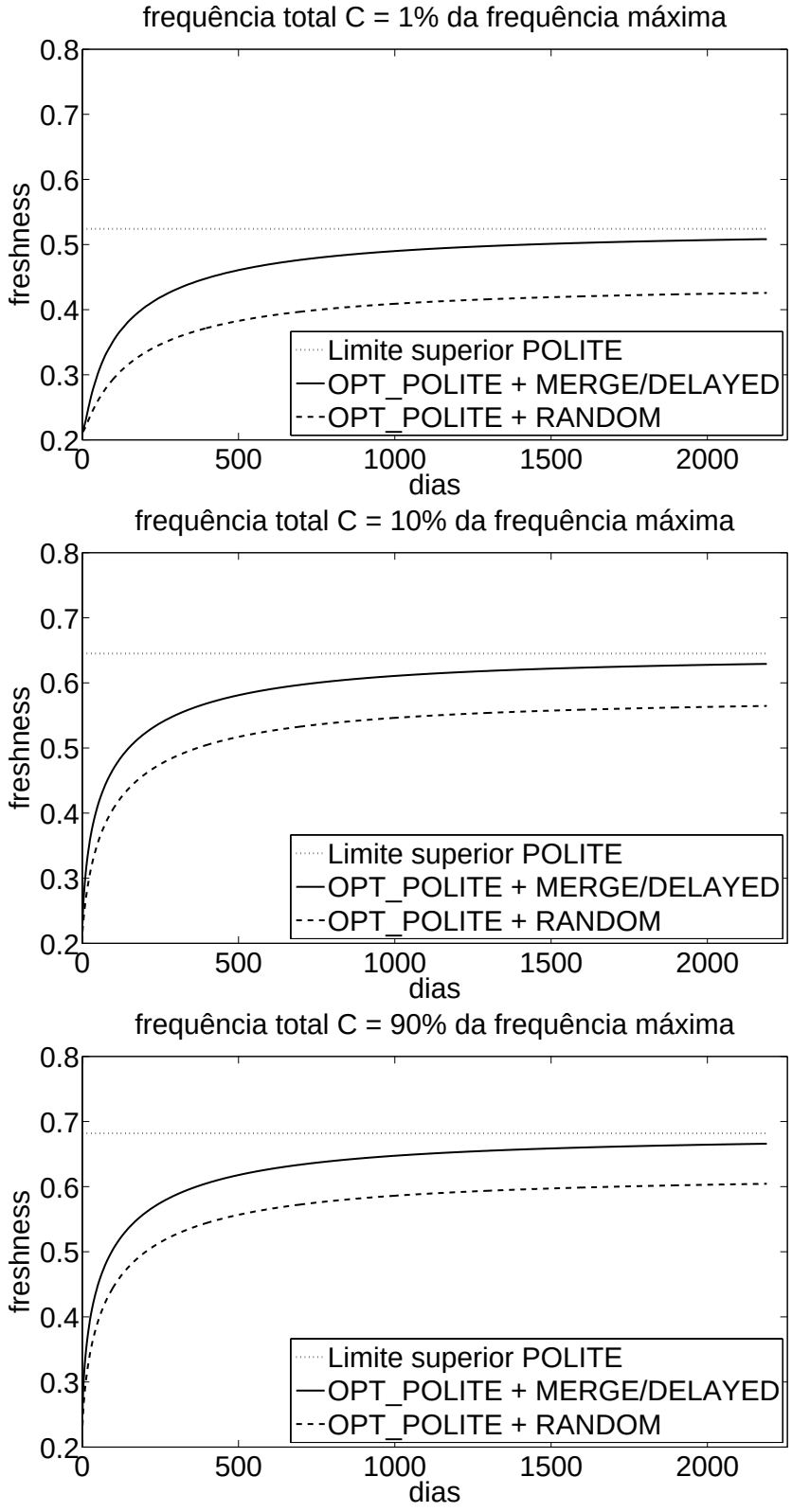


Figura 5.8: *Freshness* do repositório WEBBASE fornecido pela política DELAYED durante 6 anos de operação do *crawler*, para a frequência total $C$ de revisitação igual a 1%, 10% e 90% da frequência máxima permitida pela restrição de *politeness*.