1 OPTIMIZATION ISSUES IN WEB SEARCH ENGINES

Zhen Liu[1] and Philippe Nain[2]

[1]IBM Research
Hawthorne, NY 10532, USA
zhenl@us.ibm.com

[2]INRIA
B.P. 93, 06902, Sophia Antipolis Cedex, France
Philippe.Nain@inria.fr

Abstract: Crawlers are deployed by a Web search engine for collecting information from different Web servers in order to maintain the currency of its data base of Web pages. We present studies on the optimization of Web search engines from different perspectives. We first investigate the number of crawlers to be used by a search engine so as to maximize the currency of the data base without putting an unnecessary load on the network. Both the static setting, where crawlers are always active, and the dynamic setting where, crawlers may be activated/deactivated as a function of the state of the system, are addressed. We then consider the optimal scheduling of the visits of these crawlers to the Web pages assuming these pages are modified at different rates. Finally, we briefly discuss some other optimization issues of Web search engines, including page ranking and system optimization.

Keywords: Web search engines, web crawlers, scheduling, optimal control, queues; Markov decision process.

1.1 INTRODUCTION

The role of World Wide Web as a major information publishing and retrieving mechanism on the Internet is now predominant and continues to grow extremely fast. The amount of information on the Web has long since become too large for manually browsing through any significant portion of its hypertext structure. As a consequence, a number of Web search engines have been developed in the last decade: starting from the pioneering search engines such as Alta Vista, Lycos, Infoseek, Magellan, Excite, to the most successful ones such as Yahoo and Google.

Search engines have become an indispensable utility for Internet users. According to a recent Pew Foundation Internet and Project (January 2005), "Search engines are highly popular among Internet users. Searching the Internet is one of the earliest activities people try when they first start using the Internet, and most users quickly feel comfortable with the act of searching. Users paint a very rosy picture of their online search experiences.", and as of January 2005, "84% of internet users have used search engines. On any given day, 56% of those online use search engines."

Thus, technologies that enhance Web search engines are of high practical interest. These search engines consist of indexing engines for constructing a data base of Web pages, and in many cases **crawlers** for bringing information to the indexing engine. To maintain currency and completeness of the data base, crawlers periodically make recursive traversals of the Web's hypertext structure by accessing pages, then the pages referenced by these pages, and so on. In the literature one finds other colorful terms for crawler, such as wanderer, robot or spider, and the notion of a crawler being 'routed to' or 'visiting' a page. This chapter keeps with the 'crawler' and 'accessing' terminology throughout.

Traditionally, crawlers visit and index the Web pages until the data base reaches certain size. Periodically, this process is repeated through the rebuilding of a brand new data collection in replacement of the old one. Alternatively, the data base can be refreshed or updated incrementally. Such an operational mode is sometimes referred to as incremental crawler, see e.g. Cho and Garcia-Molina (2000b). Throughout this chapter, we consider the latter mode, i.e. the incremental crawler, although most analyses apply to the former as well.

Due to the critical role that these crawlers play in the Web search engines, the optimization issues are topics of a number of research papers. In this chapter we present some of these research problems. Rather than providing comprehensive, but high-level, discussions, we present detailed solutions to some of the technical problems.

More precisely, Section 1.2 considers both the issues of optimizing the number of the crawlers to be deployed when all crawlers are always active (static setting – Section 1.2.1), and of finding an optimal decision rule for the case where crawlers may be activated/deactivated as a function of the state of the system (dynamic setting – Section 1.2.2). Performance of static and dynamic policies are compared in Section 1.2.3. The optimal scheduling of the page visits of these crawlers is studied in Section 1.3. Finally, we provide pointers to some other issues such as page ranking and system optimization (Section 1.4).

A word on the notation in use: $\lfloor x \rfloor$ (respectively $\lceil x \rceil$) denotes the largest (respectively smallest) integer less (respectively greater) than or equal to x. Also for any mappings f and g, the relation $f(x) \overset{x}{\sim} g(x)$ is understood as $\lim_{x\to\infty} f(x)/g(x) = 1$.

1.2 OPTIMIZING THE NUMBER OF CRAWLERS

We first address in Section 1.2.1 the situation where crawlers are always active, regardless of the state of the system, and we determine the optimal number of crawlers to be deployed. Then, we move in Section 1.2.2 to the situation where crawlers may be activated/deactivated as a function of the state of the system, and we find an optimal

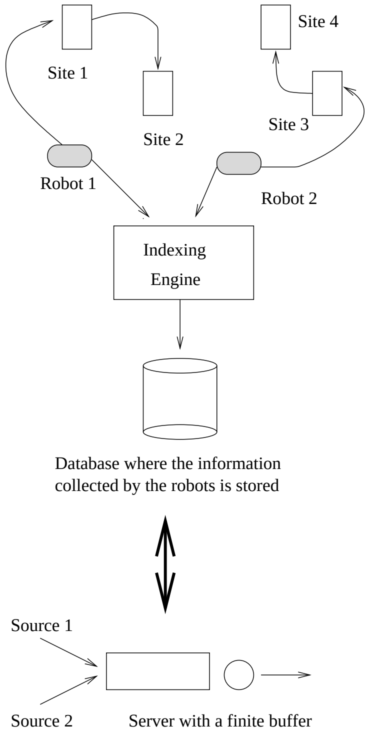

Site 1
Site 2
Site 3
Site 4
Robot 1
Robot 2
Indexing Engine
Database where the information collected by the robots is stored
Source 1
Source 2
Server with a finite buffer

Figure 1.1 Model of search engine with two crawlers

decision rule for the number of active crawlers at any time. In both settings the cost function is a weighted sum of the starvation probability and loss rate.

The results presented in this section are based on the work of Talim et al. (2001b) and Talim et al. (2001a). Practical issues of deploying parallel crawlers are discussed in Cho and Garcia-Molina (2002).

1.2.1 The Static Setting

The search engine is modeled as a single server finite capacity queue. The system capacity is $K \geq 2$ (including the position in the server), see Figure 1.1.

There are $N \geq 1$ crawlers: each crawler brings new pages to the queue according to a Poisson process with rate $\lambda > 0$. These N Poisson processes are assumed to be mutually independent and independent of the indexing (service) times. Hence, new pages are generated according to a Poisson process with intensity λN. An incoming page finding a full queue is lost. Indexing times are assumed to be independent and identically random variables with common distribution $F(x)$. Let $1/\mu$ be the expected indexing time.

The search engine is therefore modeled as the well known M/G/1/K queue (see e.g. Cohen (1982,Chapter III.6)). In this notation we define the cost function as the weighted sum of two terms:

- the fraction of time that the system is empty, hereafter referred to as the *starvation probability*;
- the expected number of times when an arriving crawler finds a full system per unit time, hereafter referred to as the *loss rate*.

Let X (resp. X^*) be the stationary queue-length at arbitrary epochs (resp. stationary queue-length at arrival epochs) in a M/G/1/K queue with arrival rate λN and service rate μ.

With $\rho := N\lambda/\mu > 0$ and for $\gamma > 0$ the cost function is then defined as

$$C(\rho,\gamma,K) := \gamma \,\mathrm{Prob}\,(X=0) + \lambda N \,\mathrm{Prob}\,(X^*=K) \tag{1.1}$$

with $\mathrm{Prob}\,(X=0)$ and $\lambda N\,\mathrm{Prob}\,(X^*=K)$ the starvation probability and the loss rate, respectively. Since $\mathrm{Prob}\,(X^*=i)=\mathrm{Prob}\,(X=i)$ for $i=0,1,\ldots,K$ from the PASTA property Wolff (1982), (1.1) rewrites as

$$C(\rho,\gamma,K) = \gamma \,\mathrm{Prob}\,(X=0) + \rho\mu \,\mathrm{Prob}\,(X=K) \tag{1.2}$$

where λN in (1.1) has been replaced by $\rho\mu$.

Throughout Section 1.2.1 we will assume that indexing times are exponentially distributed. The general case where the indexing times are arbitrarily distributed is more involved, due to the lack of closed-form expressions for the M/G/1/K queue, and is discussed in Talim et al. (2001b).

1.2.1.1 The M/M/1/K Search Engine Model. We assume that the indexing times are exponentially distributed, namely, $F(x)=1-\exp(-\mu x)$. In other words, we model the search engine as an M/M/1/K queue.

In the M/M/1/K queue with traffic intensity ρ the stationary queue-length probabilities at arbitrary epochs are given by Kleinrock (1975):

$$\mathrm{Prob}\,(X=i) = \frac{1-\rho}{1-\rho^{K+1}}\rho^i \tag{1.3}$$

for $i=0,1,\ldots,K$. Therefore,

$$C(\rho,\gamma,K) = \frac{(1-\rho)(\gamma+\mu\rho^{K+1})}{1-\rho^{K+1}}. \tag{1.4}$$

In particular, $C(\rho,\gamma,K)=(\gamma+\mu)/(K+1)$ when $\rho=1$.

Lemma 1 shows the existence of a unique minimum for $C(\rho,\gamma,K)$ considered as a function of ρ. The proof is provided in Talim et al. (2001b).

Lemma 1 *For any $\gamma>0$, $K\geq 2$, the mapping $\rho\to C(\rho,\gamma,K)$ has a unique minimum in $[0,\infty)$, to be denoted $\rho(\gamma,K)$. Furthermore, $0<\rho(\gamma,K)<1$ if $\gamma<\gamma(K)$, $\rho(\gamma,K)=1$ if $\gamma=\gamma(K)$ and $\rho(\gamma,K)>1$ if $\gamma>\gamma(K)$, with $\gamma(K):=\mu(K+2)/K$.* ◇

We now return to the original problem, namely the computation of the number N of crawlers that minimizes the cost function $C(\rho,\gamma,K)$ with $\rho = \lambda N/\mu$. The answer is found in the next result which is a direct corollary of Lemma 1.

Proposition 1 *For any $\gamma > 0$, $K \geq 2$, let $N(\gamma,K)$ be the optimal number of crawlers to use.*

Then,

$$N(\gamma,K) = \arg\min_n C(n\lambda/\mu,\gamma,K) \tag{1.5}$$

with $n \in \{\lfloor \rho(\gamma,K)\mu/\lambda \rfloor, \lceil \rho(\gamma,K)\mu/\lambda \rceil\}$. Furthermore, $N(\gamma,K) \leq \lceil \mu/\lambda \rceil$ if $\gamma < \gamma(K)$, $N(\gamma,K) \in \{\lfloor \mu/\lambda \rfloor, \lceil \mu/\lambda \rceil\}$ if $\gamma = \gamma(K)$, and $N(\gamma,K) \geq \lfloor \mu/\lambda \rfloor$ if $\gamma > \gamma(K)$. $\diamond$

In the next section we investigate the impact of the parameter γ on the optimal number of crawlers.

1.2.1.2 Impact of γ on the Optimal Number of Crawlers.

Recall that the parameter γ is a positive constant that allows us to stress either the probability of starvation or the loss rate. Part of the impact of γ on $\rho(\gamma,K)$, and therefore on $N(\gamma,K)$, the optimal number of crawlers, is captured in the following result.

Proposition 2 *For any $K \geq 2$, the mapping $\gamma \to \rho(\gamma,K)$ is nondecreasing in $(0,\infty)$, with $\lim_{\gamma\to\infty} \rho(\gamma,K) = \infty$.* $\diamond$

Proof. Pick two constants $0 < \gamma_1 < \gamma_2$ and define

$$\begin{aligned} \Delta(\rho,\gamma_1,\gamma_2,K) &:= C(\rho,\gamma_2,K) - C(\rho,\gamma_1,K) \\ &= \frac{1-\rho}{1-\rho^{K+1}}(\gamma_2 - \gamma_1). \end{aligned}$$

We assume that $\rho(\gamma_2,K) < \rho(\gamma_1,K)$ and show that this yields a contradiction.

Under the condition $\gamma_1 < \gamma_2$ the mapping $\rho \to \Delta(\rho,\gamma_1,\gamma_2,K)$ is decreasing in $[0,\infty)$. Therefore,

$$\begin{aligned} 0 &< \Delta(\rho(\gamma_2,K),\gamma_1,\gamma_2,K) - \Delta(\rho(\gamma_1,K),\gamma_1,\gamma_2,K) \\ &= [C(\rho(\gamma_2,K),\gamma_2,K) - C(\rho(\gamma_1,K),\gamma_2,K)] \\ &\quad + [C(\rho(\gamma_1,K),\gamma_1,K) - C(\rho(\gamma_2,K),\gamma_1,K)] \\ &\leq 0, \end{aligned} \tag{1.6}$$

which contradicts the fact that $\rho \to \Delta(\rho,\gamma_1,\gamma_2,K)$ is decreasing in $[0,\infty)$. Therefore $\rho(\gamma_2,K) \geq \rho(\gamma_1,K)$ and the mapping $\gamma \to \rho(\gamma,K)$ is nondecreasing in $[0,\infty)$. We may then define $L := \lim_{\gamma\to\infty} \rho(\gamma,K)$.

From the identity $\partial C(\rho,\gamma,K)/\partial\rho = 0$ for $\rho = \rho(\gamma,K)$ (see Lemma 1) we obtain

$$\begin{aligned} 0 &= \mu\rho(\gamma,K)^{2(K+1)} - (\gamma K + \mu(K+2))\rho(\gamma,K)^{K+1} \\ &\quad + (K+1)(\mu+\gamma)\rho(\gamma,K)^K - \gamma. \end{aligned} \tag{1.7}$$

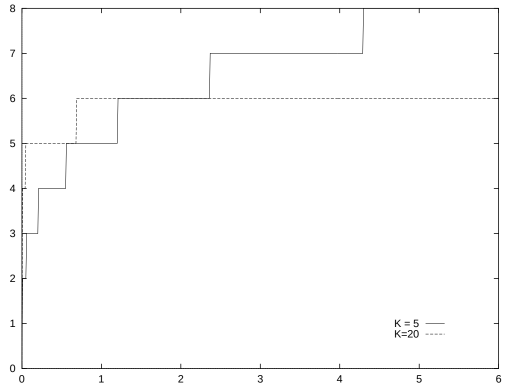


Figure 1.2 $\gamma \to N(\gamma, K)$ for $1/\lambda = 0.6$ and $\mu = 1$

Assume that $L < \infty$. Letting $\gamma \to \infty$ in (1.7) yields

$$\begin{aligned} &\left(KL^{K+1} - (K+1)L^K + 1\right) \lim_{\gamma \to \infty} \gamma \\ &= \mu L^K \left(L^{K+2} - (K+2)\right) L + (K+1)\right). \end{aligned} \quad (1.8)$$

Since $L > 1$ (we have shown in Lemma 1 that $\rho(\gamma, K) > 1$ for $\gamma > \mu(K+2)/K$) it is easily seen that $KL^{K+1} - (K+1)L^K + 1 > 0$, which implies that the l.h.s. of (1.8) is infinite whereas the r.h.s. is finite. Therefore, (1.8) cannot hold if $L < \infty$ and $\lim_{\gamma \to \infty} \rho(\gamma, K) = \infty$. This concludes the proof. ■

Proposition 2 has a simple physical interpretation. As the parameter γ increases the probability of starvation becomes the main quantity to minimize. Hence, the minimization is done by increasing the arrival rate or, equivalently, by increasing the number of crawlers, as shown in Proposition 2. Figure 1.2 provides two numerical examples illustrating the monotonicity of the optimal number of crawlers as a function of γ.

1.2.1.3 Impact of K on the Optimal Number of Crawlers. In this section we examine the behavior of $\rho(\gamma, K)$ as a function of K.

The following results hold (see Talim et al. (2001b)):

Proposition 3

(a) If $0 < \gamma \leq \mu$ then the mapping $K \to \rho(\gamma, K)$ is nondecreasing in $[2, \infty)$;

(b) If $\gamma > \mu$ then there exists an integer $K_0 \geq \lfloor 2u/(\gamma - \lambda) \rfloor$ such that the mapping $K \to \rho(\gamma, K)$ is nondecreasing in $[2, K_0 - 1]$ and non-increasing in $[K_0, \infty)$. ◇

The next proposition examines the limiting behavior of $\rho(\gamma, K)$ as K increases to infinity.

Proposition 4 *For any* $\gamma > 0$,

$$\lim_{K\to\infty} \rho(\gamma, K) = 1. \tag{1.9}$$

$\diamond$

Proof. Let $M := \lim_{K\to\infty} \rho(\gamma, K)$, where the existence of the limit follows from Proposition 3.

Letting now $K \to \infty$ in (1.7) we see that the r.h.s. converges to $-\gamma$ if $M < 1$ and converges to infinity if $M > 1$, thereby showing that necessarily $M = 1$, which concludes the proof. ∎

Proposition 4 shows that the optimal arrival rate converges to the service capacity when the buffer size increases to infinity.

The limiting result (1.9) can be used to derive an approximation for the optimal number of crawlers to be deployed when K is large. Indeed, the relation

$$\lim_{K\to\infty} N(\gamma, K) = \lim_{K\to\infty} \arg \min_{n\in\{\lfloor \mu/\lambda\rfloor, \lceil \mu/\lambda\rceil\}} C(\lambda n/\mu, \gamma, K), \tag{1.10}$$

which follows from (1.5), suggests the following approximation, for large K

$$N(\gamma, K) \overset{K}{\sim} \begin{cases} \lceil \mu/\lambda\rceil & \text{if } C(\rho_+, \gamma, \infty) \le C(\rho_-, \gamma, \infty) \\ \\ \lfloor \mu/\lambda\rfloor & \text{if } C(\rho_+, \gamma, \infty) > C(\rho_-, \gamma, \infty) \end{cases} \tag{1.11}$$

with the notation

$$C(\rho, \gamma, \infty) := \lim_{K\to\infty} C(\rho, \gamma, K),\ \rho_+ := (\lambda/\mu)\lceil \mu/\lambda\rceil \text{ and } \rho_- := (\lambda/\mu)\lfloor \mu/\lambda\rfloor.$$

Since $C(\rho, \gamma, \infty) = \gamma(1-\rho)$ for $\rho \le 1$ and $C(\rho, \gamma, \infty) = -\mu(1-\rho)$ for $\rho \ge 1$ from (1.4), we may rewrite (1.11) as

$$N(\gamma, K) \overset{K}{\sim} \begin{cases} \lceil \mu/\lambda\rceil & \text{if } -\mu(1-\rho_+) \le \gamma(1-\rho_-) \\ \\ \lfloor \mu/\lambda\rfloor & \text{if } -\mu(1-\rho_+) > \gamma(1-\rho_-). \end{cases} \tag{1.12}$$

The mapping $K \to \rho(\gamma, K)$ is displayed in Figure 1.3 for $\gamma < \mu$ and in Figure 1.4 for $\gamma > \mu$. Table 1.1 gives $N(\gamma, K)$ for different values of K and compare these values with the approximation (1.12) (last column in Table 1.1). The approximation (1.12) appears to be fairly sensitive to model parameters; however, in all but one case (1.12) lies within 10% of the exact value as soon as $K \ge 10$. We also observe that the quality of the approximation increases when γ increases (within 10% of the exact value for $\gamma = 2$ for all $K \ge 2$).

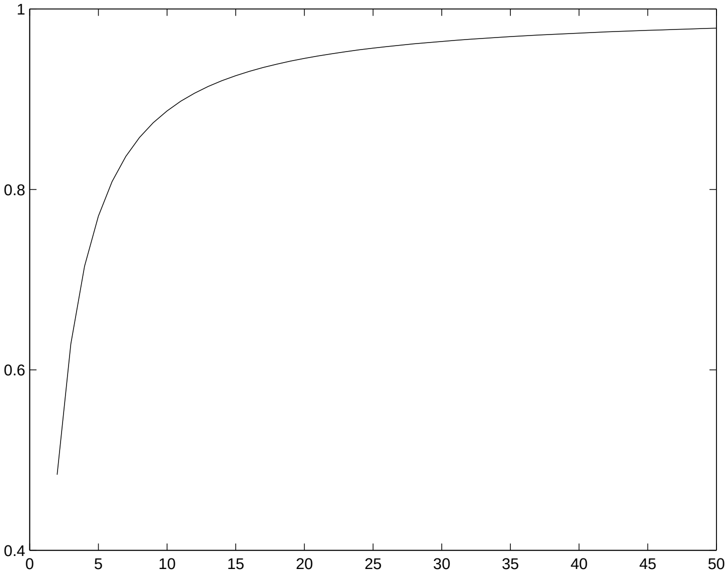


Figure 1.3 $K \rightarrow \rho(\gamma, K)$ for $\gamma = 0.5$ and $\mu = 1$

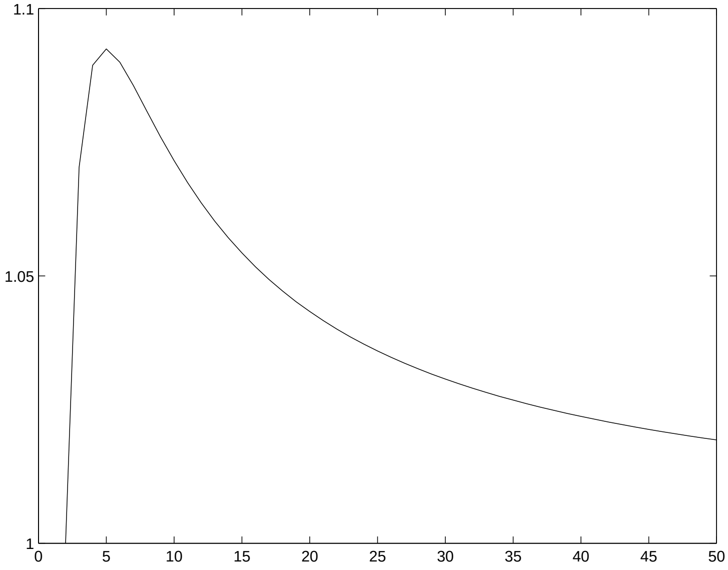


Figure 1.4 $K \rightarrow \rho(\gamma, K)$ for $\gamma = 2$ and $\mu = 1$

Table 1.1 $K \to N(\gamma, K)$ for $\lambda = 0.01$ and $\mu = 1$

$K:$	2	3	4	5	10	20	30	40	50	∞
$\gamma = 0.1$	20	34	44	52	72	85	89	92	94	100
$\gamma = 0.5$	48	63	71	77	89	95	96	97	98	100
$\gamma = 1.2$	77	88	93	96	100	101	101	101	100	100
$\gamma = 1.5$	86	96	100	102	103	102	102	101	101	100
$\gamma = 2$	100	107	109	109	107	104	103	102	102	100

1.2.2 The Dynamic Setting

In this section we assume that the number of active crawlers may vary in time according to the backlog in the queue and to the number of crawlers already active. To address this situation we will cast our model into the Markov Decision Process (MDP) framework (Bertsekas, 1987; Puterman, 1994; Ross, 1983).

The indexing engine is again modeled as a finite-capacity single-server queue. Service times still constitute independent random variables with common negative exponential distribution (with mean $1/\mu$) and the buffer may accommodate at most $K \geq 2$ customers, including the one in service, if any. There are N available crawlers and each of these crawlers, when activated, brings pages to the server according to a Poisson process with rate λ. We assume that these N Poisson processes are mutually independent and further independent of the service time process.

The new feature in this section is that the number of active crawlers may be modified at any arrival and at any departure epoch. When an arrival occurs, the incoming crawler is deactivated at once; the controller may then decide to keep it idle or to reactivate it. When a departure occurs the controller may either decide to activate one additional crawler, if any available, or to do nothing (i.e. the number of active crawlers is not modified).

The objective is to find a policy (to be defined) that minimizes a weighted sum of the stationary starvation probability and the loss rate.

We now introduce the MDP setting in which we will solve this optimization problem. Since the time between transitions is variable we will use the uniformization method (Bertsekas, 1987,Sec. 6.7).

At the n-th decision epoch t_n the state of the MDP is represented by the triple $x_n = (q_n, r_n, s_n) \in \{0,1,\ldots,K\} \times \{0,1,\ldots,N\} \times \{0,1,2\}$, with q_n and r_n the queue-length and the number of active crawlers just *before* the n-th decision epoch, respectively, and s_n the type (arrival, departure, fictitious – see below) of the n-th decision epoch.

The successive decision epochs $\{t_n, n \geq 1\}$ are the jump times of a Poisson process with intensity $\nu := \lambda N + \mu$, independent of the service time process. In this setting, the n-th decision epoch t_n corresponds to an arrival in the original system with probability $\lambda r_n/\nu$ (in which case $s_n = 1$), to a departure with probability μ/ν provided that $q_n > 0$ ($s_n = 0$) and to a fictitious event with the complementary probability $((N - r_n)\lambda + \mu)/\nu$ ($s_n = 2$).

Let $a_n \in \{0, 1\}$ be the action chosen at time t_n. We assume that $a_n = 1$ if the decision is made to activate one additional crawler, if any available, and $a_n = 0$ if the decision is made to keep unchanged the number of active crawlers. By convention we assume that $a_n = 0$ if the n-th decision epoch corresponds to a fictitious event ($s_n = 2$).

From the above definitions we see that states of the form $(\bullet, 0, 1)$ and $(0, \bullet, 0)$ are not feasible, as an arrival cannot occur if all crawlers are inactive and a departure cannot occur if the queue is empty, respectively. Therefore, the state-space for this MDP is

$$\begin{aligned}&\{(q,r,s), 0 \leq q \leq K, 0 \leq r \leq N, s = 0,1,2\}\\ &-\{(0,r,0),(q,0,1), 0 \leq q \leq K, 0 \leq r \leq N\}.\end{aligned}$$

However, this set contains one absorbing state, the "fictitious" state $(0, 0, 2)$. To remove this undesirable state we will only consider policies (see formal definition below) that always choose action $a = 1$ when the system is in state $(1, 0, 0)$ so that $(0, 0, 2)$ can never be reached. This is not a severe restriction since a policy that never activates crawlers when the system is empty is of no interest. In conclusion, the state space for this MDP is

$$\begin{aligned}\mathbf{X} := &\{(q,r,s), 0 \leq q \leq K, 0 \leq r \leq N, s = 0,1,2\}\\ &-\{(0,0,2),(0,r,0),(q,0,1), 0 \leq q \leq K, 0 \leq r \leq N\}\end{aligned}$$

and the set $\mathbf{A}_x$ of allowed actions when the system is in state $x = (q, r, s) \in \mathbf{X}$ is given by

$$\mathbf{A}_x = \begin{cases} \{0\} & \text{if } s = 2 \\ \{1\} & \text{if } (q,r,s) = (1,0,0) \\ \{0,1\} & \text{otherwise.} \end{cases}$$

To complete the definition of the MDP we need to introduce the one-step cost c and the one-step transition probabilities p. Given that the process is in state $x = (q, r, s)$ and that action a is made, the one-step cost is defined as

$$c(x) = \gamma \mathbf{1}(q = 0) + \nu \mathbf{1}(q = K, s = 1), \tag{1.13}$$

independent of a. We will show later on in this section that this choice for the one-step cost will allow us to address, and subsequently to solve, the optimization problem at hand.

For $x \in \mathbf{X}$, the one-step transition probabilities $p_{x,x'}(a)$ are given by

$$p_{x,x'}(a) = \begin{cases} \dfrac{\mu}{\nu}\mathbf{1}(q>1) & \text{if } x' = (q-1, \min\{r+a, N\}, 0) \\[2ex] \dfrac{\lambda r}{\nu} & \text{if } x' = (q-1, \min\{r+a, N\}, 1) \\[2ex] 1 - \dfrac{\mu\mathbf{1}(q>1) - \lambda r}{\nu} & \text{if } x' = (q-1, \min\{r+a, N\}, 2) \end{cases} \tag{1.14}$$

if $s = 0$, $a = 0,1$;

$$p_{x,x'}(a) = \begin{cases} \dfrac{\mu}{\nu} & \text{if } x' = (\min\{q+1, K\}, r+a-1, 0) \\[2ex] \dfrac{\lambda(r+a-1)}{\nu} & \text{if } x' = (\min\{q+1, K\}, r+a-1, 1) \\[2ex] 1 - \dfrac{\mu + \lambda(r+a-1)}{\nu} & \text{if } x' = (\min\{q+1, N\}, r+a-1, 2) \end{cases} \tag{1.15}$$

if $s = 1$, $a = 0,1$;

$$p_{x,x'}(0) = \begin{cases} \dfrac{\mu}{\nu}\mathbf{1}(q>0) & \text{if } x' = (q, r, 0) \\[2ex] \dfrac{\lambda r}{\nu} & \text{if } x' = (q, r, 1) \\[2ex] 1 - \dfrac{\mu\mathbf{1}(q>0) + \lambda r}{\nu} & \text{if } x' = (q, r, 2) \end{cases} \tag{1.16}$$

if $s = 2$. All other transition probabilities are equal to 0.

Without loss of generality we will only consider *pure stationary* policies since it is known that nothing can be gained by considering more general policies (Puterman, 1994,Ch. 8-9). Recall that in the MDP setting a policy π is pure stationary if, at any decision epoch, the action chosen is a non-randomized and time-homogeneous mapping of the current state (Bertsekas, 1987; Puterman, 1994; Ross, 1983). We define an *admissible* stationary policy as any mapping $\pi : \mathbf{X} \to \{0,1\}$ such that $\pi(x) \in \mathbf{A}_x$.

For later use introduce $P(\pi) := [p_{x,x'}(\pi(x))]_{(x,x') \in \mathbf{X} \times \mathbf{X}}$, the transition probability matrix under the stationary policy π.

Let $\mathcal{P}$ be the class of all admissible stationary policies. For any policy $\pi \in \mathcal{P}$ introduce the long-run expected average cost per unit time

$$W_\pi(x) = \lim_{n \to \infty} \frac{1}{n} \mathbf{E}_\pi \left[\sum_{i=1}^{n} c(x_i) \,|\, x_1 = x \right], \qquad x \in \mathbf{X}. \tag{1.17}$$

The existence of the limit in (1.17) is a consequence of the fact that π is stationary and $\mathbf{X}$ is countable (Puterman, 1994,Proposition 8.1.1).

We shall say that a policy $\pi^\star \in \mathcal{P}$ is average cost optimal if

$$W_{\pi^\star}(x) = \inf_{\pi\in\mathcal{P}} W_\pi(x) \qquad \forall x \in \mathbf{X}. \tag{1.18}$$

In order to use results from MDP theory for average cost models we first need to determine to which class (recurrent, unichain, multichain, communicating, etc.) the current MDP belongs to. Consider the following example: $N = 2$ and let π be any stationary policy that selects action 1 in states $(\bullet, r, 1)$ for $r \in \{1,2\}$ and in state $(1,0,0)$, and action 0 otherwise. It is easily seen that this policy induces a MDP with two recurrent classes ($\mathbf{X}\cap\{(\bullet,1,\bullet)\}$ and $\mathbf{X}\cap\{(\bullet,2,\bullet)\}$ and a set of transient states ($\mathbf{X}\cap\{\bullet,0,\bullet\}$). We therefore conclude from this example that the MDP $\{x_n, n \geq 1\}$ is *multichain* (Puterman, 1994,p. 348).

An MDP is *communicating* (Puterman, 1994,p.348) if, for every pair of states $(x,x') \in \mathbf{X}\times\mathbf{X}$, there exists a stationary policy π such that x' is accessible from x, that is, if there exists $n \geq 1$ such that $P^n_{x,x'}(\pi) > 0$, where $P^n_{x,x'}(\pi)$ is the (x,x')-entry of the matrix $P^n(\pi)$.

Lemma 2 *The MDP $(x_n, n \geq 1)$ is communicating.* $\diamond$

The proof of Lemma 2 is given in (Talim et al., 2001a). The next result follows from Lemma 2 and Proposition 4 in (Bertsekas, 1987,Sec. 7.1):

Proposition 5 *There exists a scalar θ and a mapping $h : \mathbf{X} \to \boldsymbol{R}$ such that, for all $x \in \mathbf{X}$,*

$$\theta + h(x) = c(x) + \min_{a\in\mathbf{A}_x} \sum_{x'\in\mathbf{X}} p_{x,x'}(a)\, h(x') \tag{1.19}$$

with $\theta = \inf_{\pi\in\mathcal{P}} W_\pi(x)$ for all $x \in \mathbf{X}$, while if $\pi^\star(x)$ attains the minimum in (1.19) for each $x \in \mathbf{X}$, then the stationary policy $\pi^\star$ is optimal. $\diamond$

The optimal average cost θ and the optimal policy $\pi^\star$ in Proposition 1.17 can be computed by using the following recursive scheme, known as the relative value iteration algorithm.

Proposition 6 *Let $\hat{x}$ be a fixed state in $\mathbf{X}$ and $0 < \tau < 1$ be a fixed number. For $k \geq 0$, $x \in \mathbf{X}$, define the mappings $(h_k, k \geq 0)$ as*

$$h_{k+1}(x) = (1-\tau)h_k(x) + \tau\,(T(h_k)(x) - T(h_k)(\hat{x}))$$

with

$$T(h_k)(x) := c(x) + \min_{a\in\mathbf{A}_x} \sum_{x'\in\mathbf{X}} p_{x,x'}(a)\, h_k(x'),$$

where $h_0(\hat{x}) = 0$ but otherwise h_0 is arbitrary.

Then, the limit $h(x) = \lim_{k\to\infty} h_k(x)$ exists for each $x \in \mathbf{X}$, $\theta = \tau T(h)(\hat{x})$, and the optimal action $\pi^\star(x)$ in state x is given by $\pi^\star(x) \in \operatorname{argmin}_{a\in\mathbf{A}_x} \sum_{x'\in\mathbf{X}} p_{x,x'}(a)\, h(x')$. $\diamond$

Proof. Since the MDP is communicating (cf. Lemma 2) the proof follows from Puterman (1994,Sec. 8.5,9.5.3) (see also Bertsekas (1987,Prop. 4, p. 313)). ∎

We now return to our initial objective, namely, minimizing a weighted sum of the stationary starvation probability and the loss rate. To see why the solution to this problem is given by the solution to the MDP problem formulated in this section, it suffices to show that the average cost (1.17) is a weighted sum of the stationary starvation probability and the loss rate. It should be clear, however, that this result cannot hold for policies that induce an average cost (1.17) that depends on the initial state x as, by definition, the stationary starvation probability and the loss rate are independent of the initial state. We will therefore restrict ourselves to the class $\mathcal{P}_0 \subset \mathcal{P}$ of policies that generate a constant average cost, namely, $\mathcal{P}_0 = \{\pi \in \mathcal{P} : W_\pi(x) = W_\pi(x'), \forall x \in \mathbf{X}\}$.

The set $\mathcal{P}_0$ is non-empty as it is well-known that it contains, among others, all unichain policies (Puterman, 1994,Proposition 8.2.1). Among such policies is the *static* policy π_N that always maintain N crawlers active, namely, $\pi_N(x) = 1$ for all $x = (\bullet, \bullet, s) \in \mathbf{X}$ with $s = 0, 1$ and $\pi_N(x) = 0$ for all $x = (\bullet, \bullet, 2) \in \mathbf{X}$.

We may also note that reducing the search for an optimal policy to policies in $\mathcal{P}_0$ does not yield any loss of generality as it is also known that there always exits an optimal policy with constant average cost in the case of communicating MDP's (Puterman, 1994,Proposition 8.3.2).

Fix $\pi \in \mathcal{P}_0$. Introducing (1.13) into (1.17) yields $W_\pi(x) = \gamma S_\pi(x) + L_\pi(x)$ with

$$\begin{aligned} S_\pi(x) &= \lim_{n\to\infty} \frac{1}{n} \mathbf{E}_\pi \left[\sum_{i=1}^{n} \mathbf{1}(q_i = 0) \,|\, x_1 = x \right] \\ L_\pi(x) &= \nu \lim_{n\to\infty} \frac{1}{n} \mathbf{E}_\pi \left[\sum_{i=1}^{n} \mathbf{1}(q_i = K, s_i = 1) \,|\, x_1 = x \right]. \end{aligned}$$

In the following we will drop the argument x in $S_\pi(x)$ and $L_\pi(x)$ since these quantities do not depend on x from the definition of $\mathcal{P}_0$.

Let us now interpret S_π and L_π. S_π is the stationary probability that the system is empty at decision epochs. Since the decision epochs form a Poisson process, we may conclude from the PASTA property (Wolff, 1982) that S_π is also equal to the stationary probability that the system is empty at *arbitrary epoch* with is nothing but the stationary starvation probability.

Let us now consider L_π. Recall that $\{t_n, n \geq 1\}$, the successive decision instants, is a Poisson process with intensity ν and assume without loss of generality that $t_1 = 0$ a.s. Define $A(t)$ as the total number of customers that have arrived to the queue up to time t, including customers which have been lost, and let $Q(t)$ be the queue length at time t. We assume that the sample paths of the processes $\{A(t), t \geq 0\}$ and $\{Q(t), t \geq 0\}$ are right-continuous with left limit. With these definitions and the identity $\mathbf{E}[t_n] = n/\nu$ we may rewrite L_π as

$$L_\pi = \lim_{n\to\infty} \frac{\mathbf{E}_\pi \left[\int_0^{t_n} \mathbf{1}(Q(t-) = K)\, dA(t) \right]}{\mathbf{E}[t_n]}.$$

In other words, we have shown that L_π is the ratio, as n tends to infinity, of the expected number of losses during the first n decision epochs over the expected occurrence time of the n-th decision epoch.

The interpretation of L_π as a *loss rate* now follows from the identity

$$\begin{aligned} L_\pi &= \lim_{n\to\infty} \frac{\mathbf{E}_\pi\left[\int_0^{t_n} \mathbf{1}(Q(t-)=K)\, dA(t)\right]}{\mathbf{E}[t_n]} \\ &= \lim_{T\to\infty} \frac{1}{T}\mathbf{E}_\pi\left[\int_0^T \mathbf{1}(Q(t-)=K)\, dA(t)\right], \quad \forall\pi\in\mathcal{P}_0, \end{aligned} \tag{1.20}$$

upon noticing that the latter quantity represents the mean number of losses per unit time or the loss rate. The second identity in (1.20) is a direct consequence of the theory of renewal reward processes (Ross, 1983,Theorem 7.5) and of the definition of the set $\mathcal{P}_0$.

In summary, we have shown that for any policy π in $\mathcal{P}_0$ the average cost is

$$W_\pi = \gamma S_\pi + L_\pi,$$

with S_π the starvation probability and L_π the loss rate. ∎

The optimal policy has been computed for different values of the model parameters. Figures 1.5-1.7 display the optimal policy for $N=16$, $K=5$, $\lambda=0.1$, $\mu=1$ and for different values of γ ($\gamma<\gamma(K)=1.4$, $\gamma=\gamma(K)$ and $\gamma>\gamma(K)$). The results were obtained by running the value iteration algorithm given in Proposition 6 with the stopping criterion $\max_{x\in\mathbf{X}} |(h_{k+1}(x)-h_k(x))/h_k(x)| < 10^{-5}$ (254, 255 and 256 iterations were needed to compute the optimal policy displayed in Figures 1.5, 1.6 and 1.7, respectively). We see from these figures that the optimal policy is a *monotone switching curve*, namely, there exist two monotone (decreasing here) integer mappings $f_s : \{0,1,\ldots,N\} \to \{0,1,2,\ldots\}$, $s\in\{0,1\}$, such that $\pi^\star(x) = \mathbf{1}(f_s(r)\geq q)$ for all $x=(q,r,s)\in\mathbf{X}$ with $s=0,1$ (we must also have $f_0(0)\geq 1$ so that $\pi^\star(1,0,0)=1$ as required). We conjecture that the optimal policy always exhibits such a structure but we have not able been to prove it.

s = 0

r	0	1	2	3	4	5	q
16	-	0	0	0	0	0	
15	-	0	0	0	0	0	
14	-	1	0	0	0	0	
13	-	1	0	0	0	0	
12	-	1	1	0	0	0	
11	-	1	1	0	0	0	
10	-	1	1	1	0	0	
9	-	1	1	1	0	0	
8	-	1	1	1	0	0	
7	-	1	1	1	1	0	
6	-	1	1	1	1	0	
5	-	1	1	1	1	1	
4	-	1	1	1	1	1	
3	-	1	1	1	1	1	
2	-	1	1	1	1	1	
1	-	1	1	1	1	1	
0	-	1	1	1	1	1	

s = 1

r	0	1	2	3	4	5	q
16	0	0	0	0	0	0	
15	0	0	0	0	0	0	
14	0	0	0	0	0	0	
13	1	0	0	0	0	0	
12	1	0	0	0	0	0	
11	1	1	0	0	0	0	
10	1	1	0	0	0	0	
9	1	1	0	0	0	0	
8	1	1	1	0	0	0	
7	1	1	1	0	0	0	
6	1	1	1	1	0	0	
5	1	1	1	1	0	0	
4	1	1	1	1	1	1	
3	1	1	1	1	1	1	
2	1	1	1	1	1	1	
1	1	1	1	1	1	1	
0	-	-	-	-	-	-	

Figure 1.5 Optimal policy ($\gamma = 1$, Cost = 0.20907)

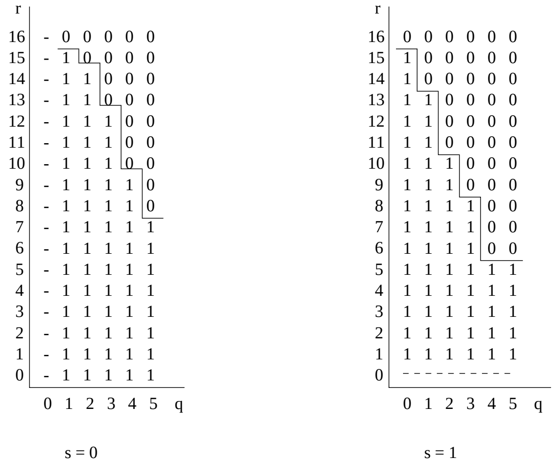
s = 0

r	0	1	2	3	4	5	q
16	-	0	0	0	0	0	
15	-	1	0	0	0	0	
14	-	1	0	0	0	0	
13	-	1	1	0	0	0	
12	-	1	1	0	0	0	
11	-	1	1	1	0	0	
10	-	1	1	1	0	0	
9	-	1	1	1	0	0	
8	-	1	1	1	1	0	
7	-	1	1	1	1	0	
6	-	1	1	1	1	1	
5	-	1	1	1	1	1	
4	-	1	1	1	1	1	
3	-	1	1	1	1	1	
2	-	1	1	1	1	1	
1	-	1	1	1	1	1	
0	-	1	1	1	1	1	

s = 1

r	0	1	2	3	4	5	q
16	0	0	0	0	0	0	
15	0	0	0	0	0	0	
14	1	0	0	0	0	0	
13	1	0	0	0	0	0	
12	1	1	0	0	0	0	
11	1	1	0	0	0	0	
10	1	1	0	0	0	0	
9	1	1	1	0	0	0	
8	1	1	1	0	0	0	
7	1	1	1	1	0	0	
6	1	1	1	1	0	0	
5	1	1	1	1	0	0	
4	1	1	1	1	1	1	
3	1	1	1	1	1	1	
2	1	1	1	1	1	1	
1	1	1	1	1	1	1	
0	-	-	-	-	-	-	

Figure 1.6 Optimal policy ($\gamma = 1.4$, Cost = 0.25924)



r							
16	-	0	0	0	0	0	
15	-	1	0	0	0	0	
14	-	1	1	0	0	0	
13	-	1	1	0	0	0	
12	-	1	1	1	0	0	
11	-	1	1	1	0	0	
10	-	1	1	1	0	0	
9	-	1	1	1	1	0	
8	-	1	1	1	1	0	
7	-	1	1	1	1	1	
6	-	1	1	1	1	1	
5	-	1	1	1	1	1	
4	-	1	1	1	1	1	
3	-	1	1	1	1	1	
2	-	1	1	1	1	1	
1	-	1	1	1	1	1	
0	-	1	1	1	1	1	
	0	1	2	3	4	5	q

s = 0

r							
16	0	0	0	0	0	0	
15	1	0	0	0	0	0	
14	1	0	0	0	0	0	
13	1	1	0	0	0	0	
12	1	1	0	0	0	0	
11	1	1	0	0	0	0	
10	1	1	1	0	0	0	
9	1	1	1	0	0	0	
8	1	1	1	1	0	0	
7	1	1	1	1	0	0	
6	1	1	1	1	0	0	
5	1	1	1	1	1	1	
4	1	1	1	1	1	1	
3	1	1	1	1	1	1	
2	1	1	1	1	1	1	
1	1	1	1	1	1	1	
0	-	-	-	-	-	-	
	0	1	2	3	4	5	q

s = 1

Figure 1.7 Optimal policy ($\gamma = 2$, Cost = 0.32211)

1.2.3 Static Versus Dynamic Policies

In this section we compare static and dynamic policies in the case where the indexing times are exponentially distributed. The results are reported in Tables 1.2 and 1.3. Throughout the experiments $\mu = 1$. For different sets of parameters λ, K, γ, we first computed the optimal number of crawlers N_s (given by Proposition 1) and the average cost C_s (given in (1.4)) in the static setting.

Then, for each set of parameters λ, K, γ, we set the value of the number of available crawlers N to N_s and determined, via the relative value iteration algorithm given in Proposition 6 (with $\tau = 0.99999$ – the closer τ is from 1 the faster the algorithm converges), the optimal average cost C_d (given in (1.18)) as well as the minimum (N_{min}) and the expected ($\overline{N}$) number of crawlers activated by the optimal dynamic policy. These results can be found in Table 1.2.

We stopped the numerical procedure when the relative error between two consecutive iterates was (uniformly) less than 10^{-5}. The number of iterations (N_{iter}) and the relative improvement ($100\% \times (C_s - C_d)/C_d$) are also reported in Table 1.2.

Last, we computed the overall optimal dynamic policy by removing the restriction on the number of available crawlers. The optimal average cost C_d as well as the minimum (N_{min}), expected ($\overline{N}$) and maximum (N_{max}) number crawlers used by the overall optimal dynamic policy are given in Table 1.3.

We observe that substantial gains may be achieved by dynamically controlling the activity of the crawlers. When the number of available crawlers is set to N_s (Table 1.2) the relative improvement w.r.t. to the optimal static policy ranges from 4% to 103% for the considered model parameters; when the restriction on the number of available crawlers is removed then the improvement ranges from 6% to 3226%! The gain appears to be an increasing function of the queue size K and of the arrival rate λ.

Table 1.2 Static vs. dynamic policies (with $\mu = 1$ and $\tau = 0.99999$)

			Static Approach		Dynamic Approach				
λ	K	γ	C_s	N_s	C_d	N_{min}	$\bar{N}$	N_{iter}	Rel. Impr.
0.01	5	0.4	0.17541	73	0.16804	57	70.3	1634	4%
-	-	1.4	0.40000	100	0.38336	86	95.1	1911	4%
-	-	2.4	0.53834	114	0.51746	101	108.4	2051	4%
0.01	10	0.4	0.10207	86	0.09062	60	82.6	1794	13%
-	-	1.2	0.20000	100	0.17534	77	94.1	1939	14%
-	-	2.4	0.28347	110	0.24798	88	102.3	2039	14%
0.01	15	0.4	0.07177	91	0.05891	58	87.7	1860	22%
-	-	1.13	0.13313	100	0.10720	70	94.5	1953	24%
-	-	2.4	0.19192	107	0.15342	78	99.7	2024	25%
0.05	5	0.4	0.17578	15	0.15127	7	13.8	338	16%
-	-	1.4	0.40000	20	0.34733	12	17.7	391	15%
-	-	2.4	0.53841	23	0.46583	15	20.2	422	16%
0.05	10	0.4	0.10220	17	0.08308	5	16.2	369	23%
-	-	1.2	0.20000	20	0.14955	8	18.2	402	34%
-	-	2.4	0.28347	22	0.20541	10	19.4	423	38%
0.05	15	0.4	0.07184	18	0.05514	4	17.4	401	30%
-	-	1.13	0.13313	20	0.09117	6	18.7	426	46%
-	-	2.4	0.19372	21	0.13895	8	19.3	438	39%
0.1	5	0.4	0.17600	7	0.15239	1	6.5	167	15%
-	-	1.4	0.40000	10	0.32198	4	8.6	200	24%
-	-	2.4	0.54067	11	0.44989	5	9.3	211	20%
0.1	10	0.4	0.10403	9	0.06838	0	8.4	204	52%
-	-	1.2	0.20000	10	0.13854	2	9.0	218	44%
-	-	2.4	0.28347	11	0.18585	3	9.6	227	53%
0.1	15	0.4	0.07184	9	0.05326	0	8.7	312	35%
-	-	1.13	0.13313	10	0.08538	1	9.3	359	56%
-	-	2.4	0.19458	11	0.09606	1	9.7	376	103%

Table 1.3 Static vs. dynamic policies (with $\mu = 1$ and $\tau = 0.99999$)

			Static Approach			Dynamic Approach		
λ	K	γ	C_s	N_s	C_d	$N_{\min}$	$\bar{N}$	$N_{\max}$	Rel. Impr.
0.01	5	0.4	0.17541	73	0.16595	58	74.8	82	6%
-	-	1.4	0.40000	100	0.37886	88	99.9	115	6%
-	-	2.4	0.53834	114	0.51179	103	113.0	133	5%
0.01	10	0.4	0.10207	86	0.08124	62	89.3	105	27%
-	-	1.2	0.20000	100	0.15876	78	99.6	123	26%
-	-	2.4	0.28347	110	0.22777	89	107.1	137	24%
0.01	15	0.4	0.07177	91	0.04236	59	94.9	118	69%
-	-	1.13	0.13313	100	0.07812	71	99.8	131	70%
-	-	2.4	0.19192	107	0.11493	79	103.6	143	67%
0.05	5	0.4	0.17578	15	0.13770	7	15.9	20	28%
-	-	1.4	0.40000	20	0.31712	13	19.8	27	26%
-	-	2.4	0.53841	23	0.43292	16	21.9	32	24%
0.05	10	0.4	0.10220	17	0.04128	5	19.0	29	148%
-	-	1.2	0.20000	20	0.12020	8	19.9	33	66%
-	-	2.4	0.28347	22	0.20541	10	20.6	36	38%
0.05	15	0.4	0.07184	18	0.00969	2	19.8	35	641%
-	-	1.13	0.13313	20	0.01818	4	20.0	38	632%
-	-	2.4	0.19372	21	0.02782	6	20.1	41	596%
0.1	5	0.4	0.17600	7	0.11097	2	8.2	12	59%
-	-	1.4	0.40000	10	0.25924	4	9.8	16	54%
-	-	2.4	0.54067	11	0.35805	6	10.7	18	51%
0.1	10	0.4	0.10403	9	0.01937	0	9.7	18	437%
-	-	1.2	0.20000	10	0.03887	1	9.9	20	415%
-	-	2.4	0.28347	11	0.05894	2	10.1	22	381%
0.1	15	0.4	0.07184	9	0.00188	0	10.0	24	3721%
-	-	1.13	0.13313	10	0.00368	0	10.0	25	3518%
-	-	2.4	0.19458	11	0.00585	0	10.0	27	3226%

1.3 OPTIMAL SCHEDULING OF THE CRAWLERS

We now turn to the problems of scheduling a crawler that maintains the currency of existing pages in search-engine data bases. For sake of arguments, we assume that the set of Web pages is fixed. However, as we shall see, our results can be promoted as heuristics that acquire new pages and drop old pages over time. A specific objective will be to find crawler schedules that minimize the *obsolescence* of the data base in some useful sense. For example, assume there are N Web pages, labeled $1, 2, \ldots, N$, which are to be accessed repeatedly by a crawler, the duration of each access being an independent sample from a given distribution. Assume also that the contents of page i are modified at times that follow a Poisson process with parameter μ. A page is considered up-to-date by the indexing engine from the time it is accessed by the crawler until the next time it is modified, at which point it becomes out-of-date until the crawler's next access. Let r_i be the fraction of time page i spends out-of-date. The problem is to find relative page-access frequencies and a sequencing policy that realizes these frequencies such that the objective function $C = \sum_{1 \leq i \leq N} c_i r_i$, is minimized, where the c_i are given weights. Under simplifying but plausible assumptions on the weights, page access times, and the class of allowed policies, we obtain explicit solutions to this problem.

From a theoretical point of view, our problem is closely related to those multiple-queue single-server systems usually called *polling systems* in the queueing literature. Indeed, the crawler can be considered as the server and the pages as the stations in the polling system. The durations of consecutive page accesses correspond to switch-over times and the page modifications correspond to customer arrivals. The service times in this polling system are zero. Our two-stage approach of optimizing crawler schedules (determining access frequencies and then finding a schedule that realizes them) is similar to the approach in Borst (1994), Borst et al. (1994) and Boxma et al. (1993) of optimizing visit sequences in polling systems.

An extensive literature exists on the analysis and control of polling systems. The interested reader is referred to the book of Takagi (1986) for general references; the special issue of the journal *Queueing Systems*, Vol. 11 (1992) on polling models and the recent thesis of Borst (1994) can be consulted for more recent developments. In particular, the polling systems with zero service times were motivated by communication networks such as *teletext* and *videotex* where pages of information are to be broadcast to terminals connected to a computer network (Ammar and Wong, 1987; Dykeman et al., 1986; Liu and Nain, 1992). However, the problem here has not been analyzed. Indeed, in the usual analysis of polling systems with unbounded buffers, interest centers on mean waiting times and mean queue lengths, whereas in our problem, the performance measure of interest, viz., the obsolescence time, corresponds to the maximum waiting time of a customer during a visit cycle of the server. An alternative view of our model identifies it with a polling loss system having unit buffers, in which our obsolescence time becomes the waiting time. With this point of view, our model has potential use in maintenance applications.

The next subsection is devoted to a precise formulation of our model, and a review of some useful concepts in stochastic ordering theory. Section 1.3.2 begins by proving two properties of crawler scheduling policies: *(i)* expected obsolescence times increase as the page-access time increases in the increasing-convex-ordering sense, and *(ii)*, by Schur-convexity results, accesses to any given page should be as evenly spaced as possible. We then derive a tight lower bound on the cost function C assuming that the weights c_i are proportional to the μ_i. These results yield a formula for optimal access frequencies. Our techniques can be extended to general c_i, but explicit formulas are not attainable in general.

To motivate the assumption on weights, note that a useful choice for the c_i is the customer page-access frequency, for in this case the total cost can be regarded as a customer total error rate. The special case where the customer access frequency c_i is proportional to the page-change rate μ is reasonable under this interpretation - the greater the interest (access frequency), the greater the frequency of page modification.

Sections 1.3.3 and 1.3.4 deal with the problem of sequencing page accesses optimally, or near optimally, so as to realize a given set of access frequencies. This material is prefaced by a discussion at the end of Section 1.3.2 which relates our scheduling problem to those that come under the heading of generalized round-robin or template-driven scheduling.

In Section 1.3.3, we introduce randomized page accessing, where each access is determined by an independent and identically distributed (i.i.d.) sample from a distribution $\{f_i\}$. We show how to find that choice for this distribution which minimizes C. In Section 1.3.4, we develop a policy that performs well when N is large. It is based on work of Itai and Rosberg (1984) (in an entirely different setting) and yields a cost within 5% of optimal.

Results presented in this section are based on the work of Coffman Jr. et al. (1998). A related study was conducted in Cho and Garcia-Molina (2000a).

1.3.1 Preliminaries

Let $\{X_k\}$ be the sequence of durations of consecutive page accesses by the crawler, each X_k being distributed independently as a random variable X. For scheduling policy π, let $\pi_n \in \{1,2,\ldots,N\}$ be the scheduling decision for the n-th access, i.e., the index of the n-th page to be accessed by the crawler under π. Define the *inter-access distance* $d^i_j(\pi) = n^i_j(\pi) - n^i_{j-1}(\pi)$, where $n^i_j(\pi)$ is the index of the j-th access of page i, i.e., $n^i_j(\pi) = \inf\{n > n^i_{j-1}(\pi) \mid \pi_n = i\}$, and where $n^i_0(\pi) \equiv 0$. Let $X^i_j = X^i_j(\pi)$ be the j-th *inter-access time* of page i, i.e., the time between the $(j-1)$-st and j-th page-i access completion times. We have $X^i_j = \sum_{k=n^i_{j-1}+1}^{n^i_j} X_k$, so the random variables X^i_j are mutually independent. Note that, if page access times X_k are exponentially distributed, then X^i_j has an Erlang distribution of d^i_j stages.

Hereafter, except in definitions, the policy π will normally be omitted from our notation; in such cases, the policy will always be clear in context.

Let $Z^i_j = Z^i_j(\pi)$ be the time that page i is out-of-date during the j-th inter-access time of page i. Let $m^i_n = m^i_n(\pi)$ be the number of accesses of page i among the first n accesses: $m^i_n = \sum_{k=1}^n \mathbf{1}\{\pi_k = i\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. Hereafter, we consider only stationary scheduling policies in the sense that, for each such policy, the limit

$$f_i = f_i(\pi) = \lim_{n\to\infty} \frac{m^i_n}{n} \tag{1.21}$$

exists and is strictly positive for all i, $1 \le i \le N$. We call f_i the access frequency of page i. We also require that the limits $\lim_{n\to\infty} \sum_{j=1}^n Z^i_j/n$ and $\lim_{n\to\infty} \sum_{j=1}^n E[Z^i_j]/n$

exist and be equal. These last assumptions hold under fairly mild conditions, e.g., when the sequence $\{d^i_j(\pi)\}_j$ is stationary and ergodic (cf. Kingman (1968)).

The *obsolescence rate* $r_i = r_i(\pi)$ of page i is the limiting fraction of time that page i is out of date; precisely, it is defined as

$$r_i = \lim_{n\to\infty} \frac{\sum_{j=1}^{m^i_n} Z^i_j}{\sum_{j=1}^{m^i_n} X^i_j} = \frac{\lim\limits_{n\to\infty} \dfrac{\sum_{j=1}^{m^i_n} Z^i_j}{n}}{\lim\limits_{n\to\infty} \dfrac{\sum_{j=1}^{m^i_n} X^i_j}{n}} = \frac{1}{E[X]} \cdot \lim_{n\to\infty} \frac{\sum_{j=1}^{m^i_n} E[Z^i_j]}{n}. \tag{1.22}$$

In particular, when policy π is cyclic with cycle length K, i.e., when $\pi_{nK+k} = \pi_{(n-1)K+k}$ for all $1 \le k \le K$ and all $n = 1, 2, \ldots$, then

$$r_i = \frac{1}{KE[X]} \sum_{j=1}^{m^i_K} E[Z^i_j], \tag{1.23}$$

where m^i_K is the number of page-i accesses during a cycle. The cost function to be minimized is the weighted sum of the obsolescence rates:

$$C = C(\pi) = \sum_{i=1}^{N} c_i \, r_i, \tag{1.24}$$

where c_i are given positive real numbers and the minimization is to be over all stationary scheduling policies.

A few basics in stochastic ordering conclude this section. For two m-dimensional real vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ majorizes $\mathbf{y}$, written $\mathbf{x} \succ \mathbf{y}$, if $\sum_{i=1}^k x_{[i]} \ge \sum_{i=1}^k y_{[i]}$, for $k = 1, \ldots, m-1$ and $\sum_{i=1}^m x_{[i]} = \sum_{i=1}^m y_{[i]}$, where $x_{[i]}$ is the i^{th} largest component of $\mathbf{x}$. Intuitively, $\mathbf{y}$ is better balanced than $\mathbf{x}$. A function h is said to be *Schur-convex* if $h(\mathbf{x}) \ge h(\mathbf{y})$ whenever $\mathbf{x} \succ \mathbf{y}$. See Marshall and Olkin (1979) for more details about this and related properties.

A random variable Y_1 is said to be no greater than a random variable Y_2 in the convex ordering sense, denoted $Y_1 \le_{cx} Y_2$, if $E[h(Y_1)] \le E[h(Y_2)]$ for all convex functions h, provided the expectations exist. If in this definition 'convex' is replaced everywhere by 'increasing and convex,' then we write $Y_1 \le_{icx} Y_2$. As is easily verified, $Y_1 \le_{cx} Y_2$ implies that Y_1 has the same mean but smaller variance than Y_2. It is also easy to see that $Y_1 \le_{cx} Y_2$ implies $Y_1 \le_{icx} Y_2$. See Stoyan (1933) for equivalent definitions and further properties.

1.3.2 Schur Convexity and a Lower Bound

Recall that a page is considered out-of-date from the time it is modified until the next time it is accessed by the crawler. Thus, if page i is not modified during its j-th inter-access interval, then the obsolescence time is $Z^i_j = 0$. Otherwise, Z^i_j is the time that elapses from the first moment page i is modified during its j-th inter-access interval until the end of that interval. Recall also that the modification (or mutation) epochs of page i follow a Poisson process with parameter μ_i. By the memoryless property of the

Poisson process, the time that elapses from the beginning of page i's j-th inter-access interval to the first subsequent mutation has an exponential distribution with parameter μ_i. Let $R^i_1, R^i_2, \ldots$ be an i.i.d. sequence of such random variables, so that

$$Z^i_j \overset{d}{=} \left(X^i_j - R^i_j\right)^+, \tag{1.25}$$

where x^+ denotes $\max(x,0)$ and $\overset{d}{=}$ denotes equality in distribution.

As an immediate consequence, we obtain

Proposition 7 *If the page access time is decreased in the increasing convex ordering sense, then the obsolescence rate is decreased for all pages under any scheduling policy.*

Proof. Let $\{X'_k\}$ be a sequence of access times distributed independently as X', and define $\{X'^i_j\}_j$ and $\{Z'^i_j\}_j$ as for X_k. Assume that $X' \leq_{icx} X$. Then,

$$Z'^i_j \overset{d}{=} \left(X'^i_j - R^i_j\right)^+ \leq_{icx} \left(X^i_j - R^i_j\right)^+ \overset{d}{=} Z^i_j,$$

and so $E[Z'^i_j] \leq E[Z^i_j]$. Thus,

$$r'_i = \frac{1}{E[X']} \cdot \lim_{n\to\infty} \frac{\sum_{j=1}^{m^i_n} E[Z'^i_j]}{n} \leq \frac{1}{E[X]} \cdot \lim_{n\to\infty} \frac{\sum_{j=1}^{m^i_n} E[Z^i_j]}{n} = r_i,$$

as desired. ∎

Returning to our main problem, where the distribution of page access times is assumed given, we now show that the obsolescence rate is a Schur convex function of the vector of inter-access distances. For this, we need the following calculation which will also be useful for later results. Define $h_i = E[e^{-\mu_i X}]$, the Laplace transform of X evaluated at μ_i.

Lemma 3 *For any page i,*

$$E[Z^i_j] = d^i_j E[X] - \frac{1}{\mu_i}\left(1 - h_i^{d^i_j}\right).$$

Proof. Let G^i_j be the probability distribution of X^i_j. We have from (1.25) that

$$\begin{aligned}
E[Z^i_j] &= \int_0^\infty P(Z^i_j > z)dz \\
&= \int_0^\infty P(X^i_j - R^i_j > z)dz \\
&= \int_0^\infty \int_z^\infty \left(1 - e^{-\mu_i(x-z)}\right) \cdot G^i_j(dx)dz \\
&= \int_0^\infty \int_0^x \left(1 - e^{-\mu_i(x-z)}\right) dz G^i_j(dx) \\
&= \int_0^\infty \left(x - \frac{1 - e^{-\mu_i x}}{\mu_i}\right) G^i_j(dx)
\end{aligned}$$

which yields the lemma. ∎

We can conclude from the above proof that the result of Proposition 7 still holds when the increasing convex ordering is replaced by the weaker Laplace-transform ordering (see Stoyan (1933)). It follows from a result of Schur (cf. Marshall and Olkin (1979,Proposition 3.C.1, page 64)) and Lemma 3 that

Proposition 8 *For any fixed number n of page-i accesses, the expected total obsolescence time of page i, $\sum_{i=1}^{n} E[Z_j^i]$ is a Schur convex function of the distances d_j^i, $j = 1, \ldots, n$.*

Thus, in order to minimize the expected obsolescence time, the accesses to any particular page should be as evenly spaced as possible.

An algorithm that computes a schedule of the crawler that implements a given set of access frequencies in the sense of (1.21) is called an *accessing* policy. In these terms, the scheduling policies proposed in this paper consist of two stages; the first computes a set of access frequencies $\{f_i\}$ and the second is an accessing policy that implements $\{f_i\}$. The even-spacing objective of accessing policies yields a lower bound, as follows.

Proposition 9 *The obsolescence rate under any accessing policy implementing the access frequencies $\{f_i\}$ satisfies for each i,*

$$r_i \geq \frac{1}{E[X]}\left(E[X] - \frac{f_i}{\mu_i} + \frac{f_i}{\mu_i} h_i^{1/f_i}\right).$$

Proof.

$$\begin{aligned}
r_i &= \frac{1}{E[X]} \cdot \lim_{n\to\infty} \frac{\sum_{j=1}^{m_n^i} E[Z_j^i]}{n} \\
&= \frac{1}{E[X]} \cdot \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{m_n^i} \left(d_j^i E[X] - \frac{1}{\mu_i}\left(1 - h_i^{d_j^i}\right)\right) \\
&= \frac{1}{E[X]} \cdot \lim_{n\to\infty} \frac{1}{n} \left(nE[X] - \frac{m_n^i}{\mu_i} + \frac{1}{\mu_i} \sum_{j=1}^{m_n^i} h_i^{d_j^i} \right) \\
&= \frac{1}{E[X]} \left(E[X] - \frac{f_i}{\mu_i} + \frac{1}{\mu_i} \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{m_n^i} h_i^{d_j^i} \right) \\
&\geq \frac{1}{E[X]} \left(E[X] - \frac{f_i}{\mu_i} + \frac{1}{\mu_i} \lim_{n\to\infty} \frac{m_n^i}{n} h_i^{n/m_n^i} \right) \\
&= \frac{1}{E[X]} \left(E[X] - \frac{f_i}{\mu_i} + \frac{f_i}{\mu_i} h_i^{1/f_i} \right),
\end{aligned}$$

where the inequality comes from the Schur convexity of $\sum_{j=1}^{m_n^i} h_i^{d_j^i}$ in the d_j^i's (cf. Proposition 8). ∎

The above lower bound can be achieved only in special cases. For instance, if the frequencies are all equal, the policy that accesses pages $1, 2, \ldots, N$ cyclically yields this optimal obsolescence rate. Another example where we can find a feasible accessing policy achieving the lower bound is when the frequencies are of the form $f_i = 1/2^{k_i}$, where k_i is an integer for every i. We return to the general case after considering the cost-minimization theorem. The proof of the following theorem gives a solution technique applicable to general weights c_i and shows that the technique leads to explicit results in an interesting special case.

Proposition 10 *Assume that the weights in the cost function are proportional to the mutation rates of the pages, i.e., $c_i = c_0\mu_i$ for all $i = 1, 2, \ldots, N$. Then for any scheduling policy,*

$$C = c_0 \cdot \sum_{i=1}^{N} \mu_i r_i \geq c_0 \left(\mu - \frac{1}{E[X]} + \frac{1}{E[X]} \prod_{i=1}^{N} h_i \right) > 0, \tag{1.26}$$

where $\mu = \sum_{i=1}^{N} \mu_i$.

Proof. For the moment, let the c_i be general. Following Proposition 9, we have $C \geq C^*$, where C^* is the solution to the following optimization problem:

$$C^* = \min \sum_{i=1}^{N} c_i \left(1 - \frac{1}{E[X]\mu_i} x_i + \frac{1}{E[X]\mu_i} x_i h_i^{1/x_i} \right) \tag{1.27}$$

subject to $x_i \geq 0$ and

$$\sum_{i=1}^{N} x_i = 1.$$

To solve the above problem, we use Lagrange multipliers and define

$$\begin{aligned} \mathcal{L}(x_1, \ldots, x_N, \lambda) &= \sum_{i=1}^{N} c_i \left(1 - \frac{1}{E[X]\mu_i} x_i + \frac{1}{E[X]\mu_i} x_i h_i^{1/x_i} \right) \\ &\quad + \lambda \left(\sum_{i=1}^{N} x_i - 1 \right). \end{aligned}$$

By the convexity of the function

$$\sum_{i=1}^{N} c_i \left(1 - \frac{1}{\mu_i E[X]} x_i + \frac{1}{\mu_i E[X]} x_i h_i^{1/x_i} \right)$$

in the vector $(x_1, \ldots, x_N)$, the solution satisfies the necessary and sufficient conditions:

$$\frac{\partial \mathcal{L}}{\partial x_i} = -\frac{c_i}{\mu_i E[X]} \left(1 - h_i^{1/x_i} + \frac{\ln h_i}{x_i} h_i^{1/x_i} \right) + \lambda = 0 \tag{1.28}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{i=1}^{N} x_i - 1 = 0. \tag{1.29}$$

Observe that $h_i < 1$, so that $h_i^{1/x_i} < 1$. One can easily check that the function $1 - y + y\ln y$ is strictly decreasing in y for $y < 1$. Thus, under the assumption that c_i is proportional to μ_i, we conclude from (1.28) that all h_i^{1/x_i} are identical and that the minimum is achieved, by (1.29), when

$$x_i = \frac{\ln h_i}{\sum_{i=1}^N \ln h_i} = \frac{\ln(h_i)^{-1}}{\sum_{i=1}^N \ln(h_i)^{-1}}. \tag{1.30}$$

This solution is positive so it is also the solution to the minimization problem in (1.27). Hence,

$$\begin{aligned} C^* &= \sum_{i=1}^N c_0\mu_i - \frac{c_0}{E[X]}\sum_{i=1}^N x_i + \frac{c_0}{E[X]}\sum_{i=1}^N x_i h_i^{1/x_i} \\ &= c_0\left(\mu - \frac{1}{E[X]} + \frac{1}{E[X]}\exp\left\{\sum_{i=1}^N \ln h_i\right\}\right) \\ &= c_0\left(\mu - \frac{1}{E[X]} + \frac{1}{E[X]}\prod_{i=1}^N h_i\right). \end{aligned}$$

Note that

$$\prod_{i=1}^N h_i = E\left[\exp\left\{-\sum_{i=1}^N \mu_i X_i\right\}\right] > 1 - E\left[\sum_{i=1}^N \mu_i X_i\right] = 1 - \mu E[X],$$

so

$$\mu - \frac{1}{E[X]} + \frac{1}{E[X]}\prod_{i=1}^N h_i > 0,$$

and the proof is complete. ∎

When the weights in the cost function are not proportional to the mutation rates of pages, one can still use Lagrange multipliers to solve the optimization problem. As noted earlier, however, we do not have closed-form solutions in general.

It is also worthwhile noticing that the optimal access frequencies (cf. (1.30)) in the above lower bound are not necessarily proportional to the page mutation rates μ_i, a fact that has emerged in the context of other polling systems (see, e.g., Borst et al. (1994) and Boxma et al. (1993)). Rather, they are proportional to $\ln(h_i)^{-1} = \ln\left(E[e^{-\mu_i X}]\right)^{-1}$. Proportionality to the μ_i occurs only when X is a constant. Note also that the magnitude of the difference between $\mu_i E[X]$ and $\ln\left(E[e^{-\mu_i X}]\right)^{-1}$ is large if $Var(X)$ is large (or X is large in the convex ordering sense).

To summarize, the results of this section show that, if the weights in the cost function are proportional to the mutation rates of the pages, then an accessing policy that comes close to the lower bound in Proposition 9 with the f_i nearly proportional to the $\ln(h_i)^{-1}$ will come close to minimizing C.

Finding good accessing policies that realize a given set of access frequencies is the subject of the next two sections. In Section 5, we develop an optimal randomized accessing policy, and in Section 6, we adapt the well-studied golden-ratio policy to our problem, primarily as a candidate for good asymptotic performance; we will see that this policy gives an obsolescence rate within 5% of the lower bound, in the limit of large N.

We remark that this problem is closely related to the design and analysis of polling/ splitting sequences in the context of queueing (and in particular, communication) systems (Andrews et al., 1997; Arian and Levy, 1992; Borst et al., 1994; Boxma et al., 1994; 1993), where algorithms are described as template driven or generalized round robin. With future research in mind, we note that these studies suggest other approaches worth investigating, e.g., extensions of the mathematical programming techniques in Borst et al. (1994) and the algorithms (Arian and Levy, 1992) derived from Hajek's results on regular binary sequences (Hajek, 1985). Although the latter lack the established performance bounds of the golden-ratio policy, simulations in the earlier queueing models show they are superior algorithms. Thus, they make promising candidates for our page-accessing model.

1.3.3 Randomized Accessing and Its Optimal Solution

Let $f_1, f_2, \ldots, f_N$ be given access frequencies. According to the randomized scheduling policy, at each decision point, the crawler chooses to access page i with probability f_i; the decision is made independently of all previous decisions. One can easily see that $\{d^i_j\}_j$, $\{X^i_j\}_j$ and $\{Z^i_j\}_j$ are three sequences of i.i.d. random variables for all i. Moreover, d^i_j has a geometric distribution: $P(d^i_j = n) = f_i(1-f_i)^{n-1}$. Thus, we have

Lemma 4 *For given frequencies* $f_1, f_2, \ldots, f_N$*,*

$$r_i = \frac{1}{E[X]}\left(E[X] - \frac{f_i}{\mu_i} + \frac{f_i}{\mu_i}\cdot\frac{f_i h_i}{1-h_i+f_i h_i}\right).$$

Proof. As d^i_j has a geometric distribution, we obtain

$$E[X^i_j] = \sum_{n=1}^{\infty} f_i(1-f_i)^{n-1} n E[X] = \frac{E[X]}{f_i},$$

and by Lemma 3 we have,

$$\begin{aligned} E[Z^i_j] &= \sum_{n=1}^{\infty} f_i(1-f_i)^{n-1}\left(nE[X] - \frac{1}{\mu_i} + \frac{1}{\mu_i}h_i^n\right) \\ &= \frac{E[X]}{f_i} - \frac{1}{\mu_i} + \frac{1}{\mu_i}\frac{f_i h_i}{1-h_i+f_i h_i}, \end{aligned}$$

so elementary renewal theory and (1.22) imply

$$r_i(\rho) = \frac{E[Z^i_j(\rho)]}{E[X^i_j(\rho)]} = \frac{1}{E[X]}\left(E[X] - \frac{f_i}{\mu_i} + \frac{f_i}{\mu_i}\cdot\frac{f_i h_i}{1-h_i+f_i h_i}\right).$$

■

It is interesting to compare the lower bound of Proposition 9 with the obsolescence rate of the randomized policy. One can see that when f_i is small (close to 0) or large (close to 1), the difference between r_i and the lower bound tends to 0. More precisely, this difference is

$$\frac{h_i}{\mu_i E[X]} f_i^2 + o((f_i)^2)$$

when f_i goes to 0, and is

$$\frac{h_i}{\mu_i E[X]} (1 + \ln h_i)(f_i - 1) + o(1 - f_i)$$

when f_i goes to 1.

We now consider the problem of finding the optimal access frequencies under the randomized policy. First, we have the following lower bound over all frequencies.

Proposition 11 *Assume that the weights in the cost function are proportional to the mutation rates of the pages, i.e., $c_i = c_0 \mu_i$ for all $i = 1, 2, \ldots, N$. Then*

$$C = c_0 \cdot \sum_{i=1}^{N} \mu_i r_i \geq c_0 \left(\mu - \frac{1}{E[X]} \cdot \frac{\sum_{i=1}^{N} (h_i^{-1} - 1)}{1 + \sum_{i=1}^{N} (h_i^{-1} - 1)} \right). \tag{1.31}$$

Moreover, this lower bound is achieved when the access frequencies are proportional to $h_i^{-1} - 1$.

The proof uses again Lagrange multipliers and can be found in Coffman Jr. et al. (1998). Note that if the weights in the cost function are not proportional to the mutation rates of the pages, the bound is still valid, see discussions in Coffman Jr. et al. (1998), provided

$$\min_{1 \leq i \leq N} \sqrt{\frac{c_i}{\mu_i}} \geq \frac{\sum_{i=1}^{N} \sqrt{\frac{c_i}{\mu_i}} \left(h_i^{-1} - 1 \right)}{1 + \sum_{i=1}^{N} \left(h_i^{-1} - 1 \right)}.$$

1.3.4 Asymptotic Optimality and the Golden Ratio Policy

In this section we consider the asymptotic large-N behavior of scheduling policies. A similar study was carried out by Itai and Rosberg (1984) in the context of the control of a multiple-access channel. Some of the results here are analogous to theirs.

We define asymptotically optimal policies with respect to the lower bound in Proposition 10. Hence, we assume throughout this section that the weights in the cost function are proportional to the mutation rates of the pages, i.e., $c_i = c_0 \mu_i$ for all $i = 1, 2, \ldots, N$. We say that a policy π is asymptotically optimal if

$$\lim_{N \to \infty} C(\pi) - C^* = 0.$$

Note first that if the total mutation rate μ tends to zero, then all *cyclic* policies are asymptotically optimal. Indeed, consider an arbitrary cyclic policy with cycle length

K. It follows from (1.23) and Lemma 3 that

$$\begin{aligned} C &= c_0 \cdot \sum_{i=1}^{N} \mu_i r_i \\ &= \frac{c_0}{KE[X]} \cdot \sum_{i=1}^{N} \mu_i \sum_{j=1}^{m_K^i} \left\{ d_j^i E[X] - \frac{1}{\mu_i}\left(1 - h_i^{d_j^i}\right)\right\} \\ &= c_0\mu - \frac{c_0}{KE[X]} \cdot \sum_{i=1}^{N} \sum_{j=1}^{m_K^i} \left(1 - (1 - d_j^i \mu_i E[X] + O(\mu_i^2))\right) \\ &= c_0\mu - \frac{c_0}{KE[X]} \cdot \sum_{i=1}^{N} \sum_{j=1}^{m_K^i} \left(d_j^i \mu_i E[X] + O(\mu_i^2)\right) \\ &= O(\mu^2), \end{aligned}$$

so if $\mu \to 0$, then $C \to 0$.

Thus, we assume that when $N \to \infty$, the total mutation rate μ, as well as the expected access time $E[X]$, is fixed. However, for any i, $1 \le i \le N$, we have $\mu \to 0$ when $N \to \infty$. Under such assumptions, the lower bound C^* in Proposition 10 becomes

$$\begin{aligned} \lim_{N\to\infty} C^* &= c_0\mu - \frac{c_0}{E[X]}\left(1 - \lim_{N\to\infty} \prod_{i=1}^{N} h_i\right) \\ &= c_0\left(\mu - \frac{1}{E[X]} + \frac{1}{E[X]} e^{-\mu E[X]}\right), \end{aligned} \tag{1.32}$$

where we used the facts that $E[e^{-\mu_i X}] = e^{-\mu_i E[X]} + o(\mu_i^\alpha)$ and that $\sum_{i=1}^{N} \mu_i^\alpha \to 0$ for all $1 < \alpha < 2$.

Now consider the following cyclic scheduling policy, called the Golden Ratio policy, and studied in Itai and Rosberg (1984) for the control of a multiple-access channel. The policy is defined in terms of the Fibonacci numbers

$$F_k = \frac{\phi^k - (1-\phi)^k}{\sqrt{5}}, \; k = 0, 1, \ldots,$$

where $\phi = (\sqrt{5}+1)/2$, and where $\phi^{-1} = (\sqrt{5}-1)/2 \simeq 0.6180339887$ is the golden ratio.

For any fixed k, let $\gamma(k,N)$ denote the golden ratio policy with F_k the cycle length, and N the total number of pages. It is assumed that $F_k \ge N$. Let $M_{k,N}^i$ be the number of page-i accesses in each cycle of $\gamma(k,N)$; these numbers satisfy

$$\lfloor f_i F_k \rfloor \le M_{k,N}^i \le \lceil f_i F_k \rceil$$

and $\sum_{i=1}^{N} M_{k,N}^i = F_k$, where f_i are the optimal access frequencies given by (1.30)

$$f_i = \frac{\ln h_i}{\sum_{i=1}^{N} \ln h_i}.$$

Thus,

$$\lim_{k\to\infty} \frac{M^i_{k,N}}{F_k} = f_i.$$

Let $\mathrm{frac}(y) = y - \lfloor y \rfloor$ be the fractional part of y. Define the set $A_k = \{\mathrm{frac}(j\phi^{-1}) \mid j = 0, 1, \ldots, F_k - 1\}$. The s-th access of the crawler is identified with the s-th smallest point of A_k. In the golden ratio policy $\gamma(k, N)$, the points

$$\left\{ \mathrm{frac}(j\phi^{-1}) \mid \sum_{m=1}^{i-1} M^m_{k,N} \le j < \sum_{m=1}^{i} M^m_{k,N} \right\}$$

correspond to the accesses of page i. As an example, let us suppose $N = 4$ and $f_1 = 2/13$, $f_2 = 3/13$, $f_3 = 3/13$ and $f_4 = 5/13$. Let $k = 8$ so that $F_k = 13$. Then, the golden ratio policy $\gamma(k, N)$ defines the access sequence $\{4, 2, 4, 1, 3, 4, 2, 4, 1, 3, 4, 2, 3\}$.

Thus, again from (1.23) and Lemma 3,

$$C(\gamma(k,N)) = c_0\mu - \frac{c_0}{F_k E[X]} \sum_{i=1}^{N} \sum_{m=1}^{M^i_{k,N}} \left(1 - h_i^{d^i_m}\right),$$

where the inter-access distance $d^i_m \in \{F_{j_i}, F_{j_i+1}, F_{j_i+2}\}$, where $j_i = \lceil \ln_\phi f_i \rceil$ (cf. Itai and Rosberg (1984)). Moreover, it can be shown by mimicking the proofs in Itai and Rosberg (1984) that

$$\begin{aligned} C(\gamma(N)) &:= \lim_{k\to\infty} C(\gamma(k,N)) \\ &= c_0\mu - \frac{c_0}{E[X]} \left\{ 1 - \sum_{i=1}^{N} \left[\left(f_i - \phi^{-j_i}\right) h_i^{F_{j_i}} \right.\right. \\ &\qquad \left.\left. + \left(f_i - \phi^{-j_i-1}\right) h_i^{F_{j_i+1}} - \left(f_i - \phi^{-j_i+1}\right) h_i^{F_{j_i+2}} \right] \right\}. \end{aligned}$$

Proposition 12 *Assume for all i that $\mu_i \to 0$ as $N \to \infty$ and that $\sum_{i=1}^{N} \mu_i = \mu > 0$. Then,*

$$\limsup_{N\to\infty} C(\gamma(N)) \le c_0 \left\{ \mu - \frac{1}{E[X]} + \frac{1-\phi^{-1}}{E[X]} e^{-\frac{\mu\phi}{\sqrt{5}} E[X]} + \frac{\phi^{-1}}{E[X]} e^{-\frac{\mu\phi^2}{\sqrt{5}} E[X]} \right\}. \tag{1.33}$$

Proof. By mimicking the proof of Theorem 5.3 in Itai and Rosberg (1984), we can show that

$$C(\gamma(N)) \le c_0\mu - \frac{c_0}{E[X]} \left\{ 1 - \sum_{i=1}^{N} f_i \left[\left(1 - \phi^{-1}\right) t^{\phi}_{i,N} + \phi^{-1} t^{\phi^2}_{i,N} \right] \right\},$$

where $t_{i,N} = h_i^{1/(f_i\sqrt{5})}$. Note that when $\mu_i \to 0$, $h_i = e^{-\mu_i E[X]} + o(\mu_i)$ so that $f_i = \mu_i/\mu + o(\mu_i)$. These imply that $t_{i,N} \to e^{-\mu E[X]/\sqrt{5}}$ when $N \to \infty$. Hence, by noting that $\sum_{i=1}^{N} f_i = 1$, we obtain

$$\limsup_{N\to\infty} C(\gamma(N)) \le c_0\mu - \frac{c_0}{E[X]} \times \left\{ 1 - \left(1-\phi^{-1}\right) e^{-\frac{\mu\phi E[X]}{\sqrt{5}}} - \phi^{-1} e^{-\frac{\mu\phi^2 E[X]}{\sqrt{5}}} \right\}.$$

■

Finally, we compare the right-hand side of (1.33) with (1.32), and obtain the following result. The detailed proof can be found in Coffman Jr. et al. (1998).

O Assume for all i that $\mu_i \to 0$ as $N \to \infty$ and that $\sum_{i=1}^{N} \mu_i = \mu > 0$. Then,

$$\limsup_{N\to\infty} \frac{C(\gamma(N))}{C^*} \le \frac{2\phi^2}{5} = \frac{\sqrt{5}+3}{5} < 1.05. \tag{1.34}$$

1.4 OTHER OPTIMIZATION PROBLEMS

In addition to the problems addressed above, there are a number of other optimization issues that need investigations. Among the most important ones are the page ranking and system implementations.

1.4.1 Page Ranking

Searching contents on the Web without knowing specific URLs is typically through querying search engines using key words. There can be thousands or even millions of Web pages containing the key words of a query. It is therefore crucial for search engines to rank the pages in such a way that the most relevant pages are presented to the users. Search engines have developed various methods to this end. The most influential work so far in this area is that of Brin and Page (1998), the founders of Google search engine, who developed the PageRank technique. A key element in this ranking system is the Markovian representation of the Web. The ranking of a Web page is related to the stationary distribution of the state corresponding to this Web page. There are ramifications since the publication of this seminal paper, as exemplified by the work of Kamvar et al. (2003), which consists in accelerating the computation of PageRank through a novel algorithm, with the reported performance improvement of 25–300%.

Another common line of thoughts is the use of learning algorithms. In Chen et al. (2000) (and a number of other papers by the same authors), building search engines using learning techniques was explored and implemented. Such approach allows online learning of and adaptation to the user behaviors.

Recently, more and more Web pages are generated dynamically. It poses a big problem to the search engines from indexing (and thus ranking) perspective as such dynamic pages are invisible to (or, more accurately, unvisited by) these search engines. Some preliminary work on this issue can be found in Mukherjee (2003), where a probabilistic model together with a ranking algorithm are proposed.

1.4.2 Implementation Issues

Web search engines after all are computer systems. The efficiency of such systems depends quite a lot on the ways they are implemented. It is therefore very important to consider practical issues.

Developing a crawler to crawl the Web looks simple: fetch a Web page using a URL; parse it to extract all referenced URLs, and for those URLs that are not yet seen before, recursively visit these pages. However, due to the big number of available Web pages, these tasks have to be carried out very efficiently. One research effort in this regard is reported in Broder et al. (2003), where the authors propose to use main memory cache to cache the visited pages so as to speed up the operations which determine whether the URLs are previously visited or not.

In the previous sections we provided theoretical investigations on the number of crawlers to be deployed, either statically or dynamically. While the deployment of such parallel crawlers allows search engines to scale up, there is need of coordinating the page visits of these crawlers in order to avoid page visit overlap. Cho and Garcia-Molina (2002) investigate such issues and propose and evaluate in particular tradeoffs between coordination overhead and overlapping degree.

Distributed implementation of Web crawlers increases scalability and resiliency. Boldi et al. (2002) present such an implementation. They use consistent hashing which allows a complete decentralized coordination which yields linear scalability.

1.5 CONCLUSIONS

In this chapter, we have discussed various optimization issues arising in Web search engines. There are still a lot of challenging optimization problems, from both research and system development perspectives. The interested reader is referred to

`http://searchenginewatch.com/`

for more up-to-date discussions on search engines. Additional information can be found in reports Huang (2000) and Boswell (2003).

Acknowledgments

Technical results presented in Sections 1.2 and 1.3 were initially obtained in Talim et al. (2001a), Talim et al. (2001b) and Coffman Jr. et al. (1998). We would like to acknowledge the contributions of the co-authors of these papers, namely, Ed. G. Coffman Jr., J. Talim and R. Weber.

Bibliography

M. H. Ammar and J. W. Wong. On the optimality of cyclic transmission in teletext systems. *IEEE Transactions on Communications*, COM-35(1):68–73, January 1987.

M. Andrews, A. Fernandez, M. Harchol-Balter, T. Leighton, and L. Zhang. Template algorithm for one-hop packet routing. Technical report, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, 1997.

Y. Arian and Y. Levy. Algorithms for generalized round robin routing. *Operations Research Letters*, 12:313–319, 1992.

D. P. Bertsekas. *Dynamic Programming. Deterministic and Stochastic Models*. Prentice-Hall, Inc., Englewood Cliffs, 1987.

P. Boldi, B. Codenotti, M. Santini, and S. Vigna. Ubicrawler: A scalable fully distributed web crawler. In *Proc. AusWeb02. The Eighth Australian World Wide Web Conference*, 2002.

S. Borst. *Polling Systems*. PhD thesis, CWI, The Netherlands, 1994.

S. Borst, O. J. Boxma, J. H. A. Harink, and G. B. Huitema. Optimization of fixed time polling schemes. *Telecommunications Systems*, 3:31–59, 1994.

Dustin Boswell. Distributed high-performance web crawlers: A survey of the state-of-the art, 2003.

O. J. Boxma, H. Levy, and J. A. Weststrate. Efficient visit orders for polling systems. *Performance Evaluation*, 18:103–123, 1993.

O. J. Boxma, H. Levy, and J. A. Weststrate. Efficient visit frequencies for polling tables: Minimization of waiting cost. *Queueing Systems (QUESTA)*, 1994.

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine, 1998.

Andrei Z. Broder, Marc Najork, and Janet Wiener. Efficient url caching for world wide web crawling. In *Proc. 12th WWW Conference (WWWC'03)*. ACM Press, 2003.

Z. Chen, X. Meng, B. Zhu, and R. H. Fowler. Websail: From on-line learning to Web search. In *Web Information Systems Engineering*, pages 206–213, 2000.

J. Cho and H. Garcia-Molina. Synchronizing a database to improve freshness. In *Proc. 2000 ACM SIGMOD Int. Conf. on Management of data (SIGMOD '00)*, pages 117–128, New York, NY, USA, 2000a. ACM Press.

J. Cho and H. Garcia-Molina. Parallel crawlers. In *Proc. 11th WWW Conference (WWWC'02)*. ACM Press, 2002.

Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the Twenty-sixth International Conference on Very Large Databases*, 2000b.

E. G. Coffman Jr., Z. Liu, and R. Weber. Optimal robot scheduling for Web search engines. *Journal of Scheduling*, 1:15–29, 1998.

J. W. Cohen. *The Single Server Queue*. North-Holland Publishing Company, 1982.

H. D. Dykeman, M. H. Ammar, and J. W. Wong. Scheduling algorithms for videotex systems under broadcast delivery. In *Proc. of Int. Communication Conf. (ICC'86)*, pages 1847–1851, 1986.

B. Hajek. Extremal splittings of point processes. *Mathematics of Operations Research*, 10:543–556, 1985.

L. Huang. A survey on web information retrieval technologies, 2000.

Pew Internet and American Life Project. Search engine users, January 2005.

A. Itai and Z. Rosberg. A golden ratio control policy for a multiple-access channel. *IEEE Transactions on Automatic Control*, AC-29(8):712–718, August 1984.

S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Extrapolation methods for accelerating pagerank computations. In *Proc. 12th WWW Conference (WWWC'03)*. ACM Press, 2003.

J. F. C. Kingman. The ergodic theory of subadditive stochastic processes. *Journal of the Royal Statistical Society. Ser. B*, 30:499–510, 1968.

L. Kleinrock. *Queueing Systems, Vol. I*. Wiley & Sons, New York, 1975.

Z. Liu and P. Nain. Optimal scheduling in some multi-queue single-server systems. *IEEE Transactions on Automatic Control*, AC-37(2):247–252, February 1992.

A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Academic Press, New York, 1979.

S. Mukherjee. A probabilistic model for optimal searching of the deep Web, 2003.

M. L. Puterman. *Markov Decision Processes*. Wiley, New York, 1994.

S. M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, New York, 1983.

D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. J. Wiley & Sons, 1933. English translation (D. J. Daley editor).

H. Takagi. *Analysis of Polling Systems*. MIT Press, 1986.

J. Talim, Z. Liu, P. Nain, and E. G. Coffman Jr. Controlling the robots of Web search engines. In *Proc. ACM Sigmetrics - Performance 2001 Conf.*, volume 29 of *Performance Evaluation Review*, pages 236–244, June 2001a.

J. Talim, Z. Liu, P. Nain, and E. G. Coffman Jr. Optimizing the number of robots in Web search engines. *Telecommunication Systems*, 17(1,2):243–264, 2001b.

R. L. Wolff. Poisson arrivals see time averages. *Operations Research*, 30:223–231, 1982.