A Stochastic Checkpoint Optimization Problem

E. G. Coffman, Jr., Leopold Flatto, Paul E. Wright

AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT

We study an abstract moving-server system that models several computer applications, including software debugging and accessing compressed data. In this model, the server moves on the unit interval $[0, 1]$, serving requests where they are encountered. The locations of successive requests are not known in advance, but they are known to be independent samples from a given distribution F on $[0, 1]$. Before serving a request, the server must be moved to a reset point to the left of the request. There is a choice of two reset points, one fixed at 0 and one, called the checkpoint, that can be moved in the course of serving requests. The cost of serving a request is proportional to the distance moved to the request from the chosen reset point. We formulate a stochastic optimization problem whose solution, for a wide class of distributions F, yields a policy for deciding the successive locations of the checkpoint so as to minimize the expected total cost of serving a sequence of requests. Results for both finite and infinite-horizon variations of the problem are presented, along with the properties required of the distribution F.

the checkpoint, minimizes the expected total cost $\sum_{i=1}^{n} E(c_i)$ of serving a sequence of n requests.

Let x_i, s_i denote the respective locations of the checkpoint and the server after serving the i^{th} request at y_i, $1 \leq i \leq n$; x_0, s_0 denote the initial checkpoint and server positions. Figure 1 illustrates the various decisions of policies in $\mathcal{C}$. The first and second stages of the motion undergone by the server at s_i, in order to satisfy the next request at y_{i+1}, are indicated by single and double lines, respectively. In Fig. 1(b) the server at s_{i+1} moves during the first stage to x_{i+1}, which is the nearest reset point to the left of y_{i+2}. During the second stage the server moves the checkpoint to x_{i+2}, where $x_{i+1} \leq x_{i+2} \leq y_{i+2}$, and then moves on to y_{i+2}, which becomes s_{i+2}. We call such a decision conservative, since it incurs a (locally) minimum cost in serving a request. In Fig. 1(a) the first stage of the server motion is nonconservative, since the server resets at 0, even though x_i is the nearer reset point to the left of y_{i+1}. In Fig. 1(c) the second stage of the server motion is nonconservative, since the checkpoint is relocated to the right of y_{i+3}. The two arrowheads in Fig. 1(c) indicate that the server first satisfies the request at y_{i+3}, and then places the checkpoint at x_{i+3}, which becomes s_{i+3}.

The rate r has no effect on our optimization problem, so hereafter we assume $r = 1$ and identify cost with distance. Since the y_i are i.i.d. and the reset motion is cost free, the only state variable of interest is the position x_i of the checkpoint. Hereafter, this position is called the *state*. To simplify terminology we will often say "serve y_i" to mean "serve the request located at y_i."

A little reflection leads one to expect that, among the optimal policies in $\mathcal{C}$, there is one that is conservative, i.e., one that makes only conservative decisions. This is proved in Section 2. The proof requires some effort but the arguments are quite elementary. Section 3 begins by formulating a recurrence that defines the expected total cost $E_n(x)$ incurred by an optimal conservative policy in serving n requests starting in state x. For a broad class of distributions F, the properties of $E_n(x)$ are given in detail. Section 4 uses these properties to define an optimal policy having a simple structure. Under this policy, the sequence of states $x_1, \ldots, x_n$ forms a simple Markov chain with initial state x_0; the transient and stationary regimes of this chain are also described in Section 4. Section 5 studies the class of distributions F to which the optimal policy applies. The remainder of this section briefly discusses background and applications.

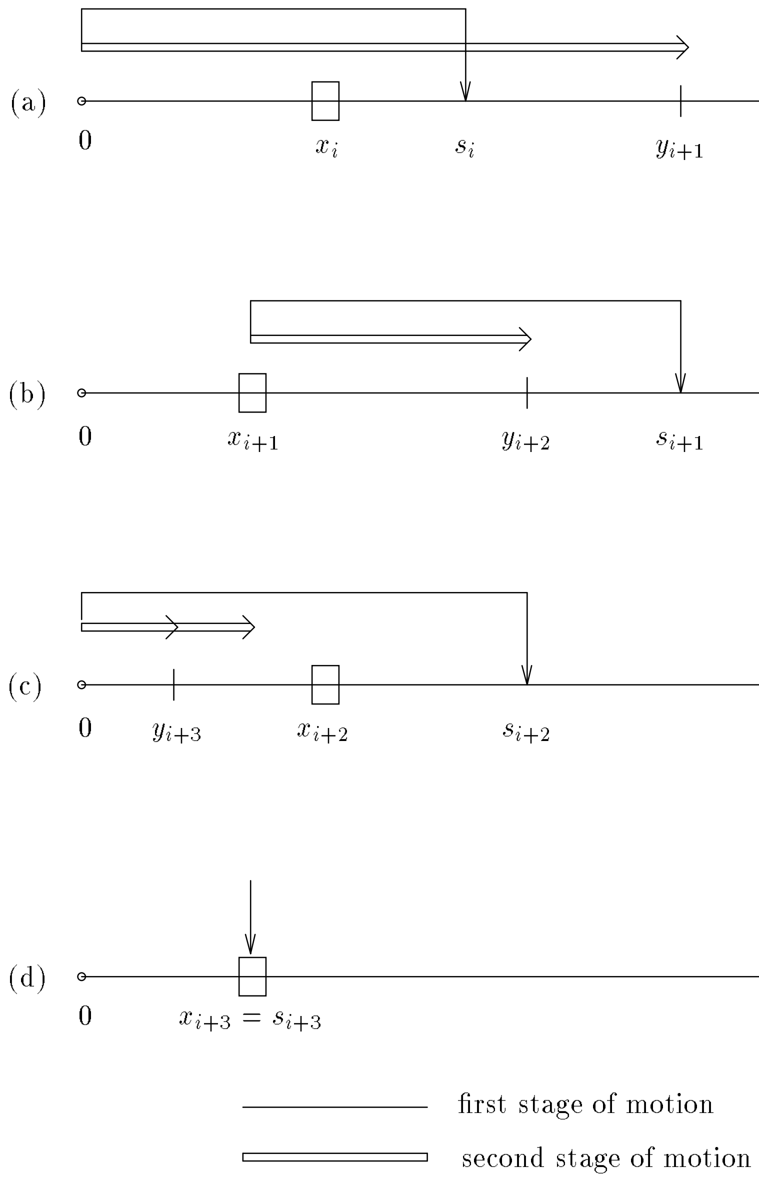

Figure 1 — An Example of decisions under policies in $\mathcal{C}$.

This paper was motivated by recent work of Bern et al. [1], who describe several applications of the moving-server checkpointing model. Here, we give only the flavor of these applications; detailed discussions can be found in [1]. Briefly, in the data base application the interval $[0, 1]$ is taken as a normalized, continuous approximation to a large string of text, say an encyclopedia, in compressed form. To access an item at a known location, say y, in this compressed text, the compression statistics at the time the item was stored must be known to the request server [6]. Without checkpoints this requires the statistics to be recomputed from scratch, from 0 to the point y. Checkpoints storing the statistics generated at various points in the compressed text allow for faster access; the recomputation of statistics needs to be started only at the most recent checkpoint created before the item at y was stored.

In software testing and development, the interval approximates a long sequence of statements or lines of code. Checkpoints are placed at various points in the code in order to save the state of the software system during trial runs. After trial runs, the programmer may wish to track certain state variables beyond some point y in the code. To do so efficiently, the programmer steps through the code starting in the state saved by the last checkpoint encountered before reaching point y.

In general, the model applies to any irreversible process or computation whose states at various times need to be recaptured in order to modify the behavior or structure of the process. Physical simulations are another application of this type brought out by Bern et al. [1]. Note that the models here and in [1] apply to those situations where requests at points earlier than (to the left of) a checkpoint cause the checkpoint to be moved. This applies in those cases where a checkpoint to the right of a request becomes invalid. For example, the program is being modified in the program debugging application, or the compressed text is being modified in the data base application.

Bern et al. consider a more general model in which there are $k > 1$ checkpoints. Their primary goal is a combinatorial analysis of competitiveness, i.e., the best relative performance of on-line algorithms, such as those studied here, versus off-line algorithms that can see all requests in advance. The objective here is a more realistic stochastic analysis of optimal on-line algorithms. As a concession to the greater difficulty of a stochastic model, we limit ourselves to an important special case treated by Bern et al., viz., the case $k = 1$ of only one checkpoint.

A great deal of work has also been published on the use of checkpoints in the design of fault-tolerant systems. The references in [2, 5] provide an excellent gateway to this literature. Most of this research focuses on queueing models defined by given interarrival and running time distributions. System failures occur according to a given probability law. Recovery from failures includes rolling the system back to the most recent checkpoint (error-free state). There are also several papers on checkpointing within the setting of the results presented here, i.e., optimal stochastic scheduling of checkpoints for fault-tolerant computations [2, 3, 4, 7]. However, all of these models concern the placement of checkpoints at fixed locations; the moving checkpoint of the model studied here leads to fundamental changes in the analysis.

2. A Reduction

The following result simplifies the policies that need to be considered.

Theorem 2.1 *There is a conservative policy among the optimal policies in $\mathcal{C}$.*

Proof. We prove that, for any initial state x_0 and sample $y_1, \ldots, y_n$, there is a sequence of conservative decisions whose total cost is minimum over all possible decision sequences of policies in $\mathcal{C}$. It follows that any policy $\mathcal{P}$ may be replaced by a conservative policy $\mathcal{P}'$ whose expected total cost is less than or equal to that of $\mathcal{P}$. In Section 3, we conclude from the Bellman equation that there exists an optimal policy among the conservative ones. By the previous remark this policy is then optimal among all policies.

There are two types of nonconservative decisions that must be considered, one in the first stage of motion and one in the second (see Fig. 1).

Type 1. In serving y_i, $1 \leq i \leq n$, the server is reset to 0 even though the checkpoint is at x_{i-1} with $0 < x_{i-1} \leq y_i$.

Type 2. After serving y_i, the server motion is continued to the right so as to shift the checkpoint to a position $x_i > y_i$.

Suppose policy $\mathcal{P}$ makes a type-2 decision in serving y_i, so that $x_i > y_i$. Let $\mathcal{P}'$ be a policy that makes the same decisions as $\mathcal{P}$ up through the first stage of the i^{th} service, but then proceeds as follows. In the second stage of the i^{th} service $\mathcal{P}'$ makes the shorter move to y_i, where a new checkpoint is established. $\mathcal{P}'$ thus gains an advantage of $(x_i - y_i)$ over $\mathcal{P}$ upon completion of the i^{th} service. If, on the $(i+1)^{\text{st}}$ request, $\mathcal{P}$ resets to 0, then so does $\mathcal{P}'$; $\mathcal{P}'$ mimics $\mathcal{P}$ thereafter and retains the advantage $(x_i - y_i)$. On the other hand, if $\mathcal{P}$ resets to x_i on the $(i+1)^{\text{st}}$ request, then $\mathcal{P}'$ resets to y_i. $\mathcal{P}'$ then begins the second stage of the $(i+1)^{\text{st}}$ service by moving to x_i, thus giving up the advantage $(x_i - y_i)$; $\mathcal{P}'$ mimics $\mathcal{P}$ thereafter. In either case the cost incurred by $\mathcal{P}'$ is at most that incurred by $\mathcal{P}$, and $\mathcal{P}'$ has made one fewer type-2 decision than $\mathcal{P}$. Repetitions of the above argument show that for given x_0, $y_1, \ldots, y_n$ and policy $\mathcal{P}$, there exists a policy $\mathcal{P}'$ that makes no type-2 decision and incurs a cost at most that incurred under $\mathcal{P}$.

Next, let $\mathcal{P}$ be a policy that makes no decisions of type 2 but makes a type-1 decision in serving y_i, i.e., the server resets at 0 even though $0 < x_{i-1} \le y_i$. We may assume that in the second stage of serving y_i, $\mathcal{P}$ establishes a new checkpoint at $x_i < x_{i-1}$; otherwise, the cost under $\mathcal{P}$ can obviously be reduced by an amount x_{i-1} by resetting at x_{i-1} instead of at 0. Let $\mathcal{P}'$ be a policy that is identical to $\mathcal{P}$ up through the $(i-1)^{\text{st}}$ service, but then proceeds as follows. $\mathcal{P}'$ resets to x_{i-1} during the i^{th} service, keeping the checkpoint location at x_{i-1}. The costs of serving y_i under $\mathcal{P}$ and $\mathcal{P}'$ are given respectively by y_i and $(y_i - x_{i-1})$, so $\mathcal{P}'$ gains an advantage of x_{i-1} over $\mathcal{P}$ upon completion of i^{th} service. If, on the $(i+1)^{\text{st}}$ request, $\mathcal{P}$ resets to 0, then so does $\mathcal{P}'$, mimicking $\mathcal{P}$ thereafter and retaining the advantage x_{i-1}. Otherwise, $\mathcal{P}$ resets to x_i, in which case $x_i \le y_{i+1}$. As $\mathcal{P}$ makes no type-2 decision, we also have $x_i \le x_{i+1} \le y_{i+1}$. We now distinguish three possibilities.

(i) $\underline{x_i \le y_{i+1} < x_{i-1}}$. In the $(i+1)^{\text{st}}$ service $\mathcal{P}'$ must reset to 0, thus reducing the advantage to $x_{i-1} - x_i$. $\mathcal{P}'$ then establishes a checkpoint at x_{i+1}, mimics $\mathcal{P}$ thereafter, and retains the advantage $x_{i-1} - x_i$.

(ii) $\underline{x_{i-1} \le y_{i+1} \text{ and } x_{i+1} \ge x_{i-1}}$. In the $(i+1)^{\text{st}}$ service $\mathcal{P}'$ resets to x_{i-1}, establishes a checkpoint at x_{i+1}, and mimics $\mathcal{P}$ thereafter. At the end of the $(i+1)^{\text{st}}$ service, the advantage is increased to $x_{i-1} + (x_{i-1} - x_i) = 2x_{i-1} - x_i$, which is retained thereafter.

(iii) $\underline{x_i \le y_{i+1} \text{ and } x_i \le x_{i+1} < x_{i-1}}$. In the $(i+1)^{\text{st}}$ service $\mathcal{P}'$ resets to x_{i-1} and keeps the checkpoint there, the advantage under $\mathcal{P}'$ being increased to $2x_{i-1} - x_i$ at the end of the $(i+1)^{\text{st}}$ service. The above procedure is then repeated, replacing i by $i+1$.

The resulting policy $\mathcal{P}'$ makes no type-2 decision and one fewer type-1 decision than $\mathcal{P}$. Repetitions of the above argument show that for given $x_0, y_1, \ldots, y_n$ and policy $\mathcal{P}$, there exists a conservative policy $\mathcal{P}'$ that incurs a cost at most that incurred by $\mathcal{P}$. ■

3. The Bellman Equation

For convenience we assume initially that $F(0) = 0$, $F(1) = 1$, and that $F(x)$ is continuous and strictly increasing on $[0,1]$. This section solves a Bellman equation defining the expected total cost $E_n(x)$ incurred by an optimal conservative policy serving n requests and starting in state x. By Theorem 2.1 the expression for $E_n(x)$ will apply to any optimal policy in $\mathcal{C}$.

Lemma 3.1 *For $n \geq 0$ and $0 \leq x \leq 1$*

$$E_{n+1}(x) = \int_0^x \left[y + \inf_{0 \leq z \leq y} E_n(z)\right] dF(y) + \int_x^1 \left[y - x + \inf_{x \leq z \leq y} E_n(x)\right] dF(y) \tag{3.1}$$

where $E_0(x) = 0$, $0 \leq x \leq 1$.

Proof. If the first of the $n+1$ requests is at $y = y_1 < x$, then the conservative server resets to 0 before moving to y. If the checkpoint is repositioned at z, $0 \leq z \leq y$, then the optimal total expected cost of serving the $n+1$ requests can be expressed as $y + E_n(z)$. On the other hand, if $y \geq x$, then the conservative server resets to x, so the optimal total expected cost can be expressed as $y - x + E_n(z)$, where z, $x \leq z \leq y$, is again the point at which the checkpoint has been repositioned. Minimizing over z and averaging over y yields (3.1). ∎

Let $\mu = \int_0^1 udF(u)$ denote the mean of F and define $G(u) = u[1 - F(u)]$. Then by integrating the first terms of the integrands in (3.1), we can write

$$E_{n+1}(x) = \mu - G(x) + \int_0^x \inf_{0 \leq z \leq y} E_n(z)dF(y) + \int_x^1 \inf_{x \leq z \leq y} E_n(z)dF(y) \ . \tag{3.2}$$

To obtain a simple optimal policy, the form of (3.2) suggests that it might be useful to ensure that $E_n(x)$ is unimodal with a unique minimum. To this end, consider the following two conditions on F:

C_1. $G(x)$ is unimodal on $[0, 1]$ with a unique maximum at τ, $0 < \tau < 1$.

C_2. $H(x) = [G(\tau) - G(x)]/F(x)$ is increasing in $[\tau, 1]$.

C_1 does not imply C_2 as shown in Section 5.

Theorem 3.1 *C_1 and C_2 hold jointly if and only if, for $n \geq 1$, $E_n(x)$ is unimodal on $[0, 1]$ with a unique minimum at τ. $E_n(x)$ is given by*

$$E_n(x) = \mu - G(x) + (n-1)\alpha, \quad 0 \leq x \leq \tau \ , \tag{3.3}$$

$$E_n(x) = E_n(\tau) + \frac{G(\tau) - G(x)}{F(x)}\{1 - [1 - F(x)]^n\}, \quad \tau \leq x \leq 1 \ , \tag{3.4}$$

where

$$\alpha = \int_0^\tau [\mu - G(y)]dF(y) + [\mu - G(\tau)][1 - F(\tau)] \ . \tag{3.5}$$

Proof. Suppose that for $n \geq 1$, $E_n(x)$ is unimodal in $[0,1]$ with a unique minimum at τ, $0 < \tau < 1$. We refer to this as the unimodality condition. We first prove, by induction on $n \geq 1$, that this condition implies that the E_n's are given by formulas (3.3), (3.4), and then complete the proof by showing that the unimodality condition is equivalent to C_1, C_2.

Since $E_0(x) = 0$, we obtain from (3.2) that $E_1(x) = \mu - G(x)$, which coincides with (3.3), (3.4) for $n = 1$. Suppose we are given that $E_n(x)$ satisfies (3.3), (3.4) in addition to the unimodality condition. We show that $E_{n+1}(x)$ is given by (3.3), (3.4), with n replaced by $n+1$.

Let $0 \leq x \leq \tau$. Then $\inf_{0 \leq z \leq y} E_n(z) = E_n(y)$, $0 \leq y \leq x$, and for $x \leq y \leq 1$,

$$\inf_{x \leq z \leq y} E_n(z) = \begin{cases} E_n(y) \,, & x \leq y \leq \tau \\ E_n(\tau) \,, & \tau \leq y \leq 1 \,. \end{cases}$$

We conclude from (3.2) that

$$E_{n+1}(x) = \mu - G(x) + \int_0^\tau E_n(y) dF(y) + E_n(\tau)[1 - F(\tau)] \,, \quad 0 \leq x \leq \tau \,. \tag{3.6}$$

Substituting (3.3) into (3.6) both for $E_n(y)$ and $E_n(\tau)$, and taking into account (3.5), we obtain (3.3) with n replaced by $n+1$.

Let $\tau \leq x \leq 1$. We have for $0 \leq y \leq x$

$$\inf_{x \leq z \leq y} E_n(z) = \begin{cases} E_n(y) \,, & 0 \leq y \leq \tau \\ E_n(\tau) \,, & \tau \leq y \leq x \,, \end{cases}$$

and for $x \leq y \leq 1$, $\inf_{x \leq z \leq y} E_n(z) = E_n(x)$. We conclude from (3.2) that

$$E_{n+1}(x) = \mu - G(x) + \int_0^\tau E_n(y) dF(y) + E_n(\tau)[F(x) - F(\tau)] + E_n(x)[1 - F(x)] \,, \quad \tau \leq x \leq 1 \,. \tag{3.7}$$

In (3.7) let $x = \tau$. Subtract the resulting equation from (3.7) to obtain

$$E_{n+1}(x) - E_{n+1}(\tau) = G(\tau) - G(x) + [1 - F(x)][E_n(x) - E_n(\tau)] \,, \quad \tau \leq x \leq 1 \,. \tag{3.8}$$

Substituting for $E_n(x) - E_n(\tau)$ in (3.8) from (3.4), we obtain (3.4) with n replaced by $n+1$.

Next, we show that C_1, C_2 imply the unimodality condition. We proceed by induction. For $n = 1$, $E_1(x) = \mu - G(x)$, so C_1 implies that $E_1(x)$ satisfies the unimodality condition. Suppose that $E_1(x), \ldots, E_{n-1}(x)$ satisfy the unimodality condition. Our earlier argument shows that $E_n(x)$ satisfies (3.3), (3.4). From (3.3) and C_1, we obtain that $E_n(x)$ is decreasing on $[0, \tau]$

with a unique minimum at τ. The function $1 - [1 - F(x)]^n$ is strictly increasing on $[0, 1]$. Hence from (3.4) and C_2, $E_n(x)$ is increasing on $[\tau, 1]$ with a unique minimum at τ. Thus $E_n(x)$ satisfies the unimodality condition.

Finally, we show that the unimodality condition implies C_1, C_2. Since $E_1(x) = \mu - G(x)$, we obtain C_1. Suppose C_2 fails, i.e., $H(x_1) > H(x_2)$ for some $\tau \leq x_1 < x_2 \leq 1$. The unimodality condition implies that $E_n(x)$, $n \geq 1$, satisfies (3.4). Since $\lim_{n\to\infty} \{1 - [1 - F(x)]^n\} = 0$ uniformly on $[\tau, 1]$, we obtain from (3.4) that, for n suffi

If the initial state satisfies $0 \leq x_0 \leq \tau$, then by (4.1) all subsequent states also satisfy $0 \leq x_i \leq \tau$, $i \geq 1$. Otherwise, if $\tau < x_0 \leq 1$, subsequent states remain at x_0 until the first request y_j, $j \geq 1$, such that $0 \leq y_j < x_0$. From that point onward, $0 \leq x_i \leq \tau$, $i \geq j$.

Thus, the transient states of $\{x_i\}$ are those in $(\tau, 1]$ and the recurrent states are those in $[0, \tau]$. The equilibrium measure of $\{x_i\}$ is on $[0, \tau]$ and defined by

$$\begin{aligned} dP(y) &= dF(y), \quad 0 \leq y < \tau \ , \\ P(\tau) &= 1 - F(\tau) \ . \end{aligned} \tag{4.3}$$

It can be seen that C_1 and C_2 were introduced to control the transient behavior of $\mathcal{P}_o$ when started in an initial state $x_0 \in (\tau, 1]$. The result below shows that, if initial states are suitably restricted, then an optimal policy can be defined for a broader class of distributions F. First, consider the following relaxation of C_1.

C_0. For some τ, $0 < \tau < 1$, $G(x)$ has a unique maximum at τ on $[0, 1]$, and $G(x)$ is increasing on $[0, \tau]$.

Corollary 4.2 *If C_0 holds, then for $n \geq 1$, (3.3) holds and $E_n(x)$ has a unique minimum at τ on $[\tau, 1]$.*

Proof. The proof parallels that of Theorem 3.1, proceeding by induction on $n \geq 1$. As $E_1(x) = \mu - G(x)$, the result holds for $n = 1$. Suppose it holds for $n \geq 1$. Formula (3.3) remains valid for $0 \leq x \leq \tau$. For $\tau \leq x \leq 1$, we obtain from (3.2)

$$E_{n+1}(x) = \mu - G(x) + \int_0^\tau E_n(y)dF(y) + E_n(\tau)[F(x) - F(\tau)] + \int_x^1 \inf_{x \leq z \leq y} E_n(z)dF(y) \tag{4.4}$$

which readily implies $E_{n+1}(x) > E_{n+1}(\tau)$, $\tau < x \leq 1$. Then (3.6) and (4.4) imply the corollary for $n + 1$. ■

If $x_0 \in [0, \tau]$, with τ defined by C_0, then by Corollary 4.2 $\mathcal{P}_o$ is completely defined by (4.1). Hence, if C_0 holds and the initial state x_0 is a sample from the equilibrium measure (4.3), then $\mathcal{P}_o$ makes the decisions in (4.1), and $E_n(x_0) = n\alpha$, $n \geq 1$, with α given by (3.5).

5. Examples

First, we verify that there is no redundancy in the conditions C_1 and C_2 of Theorem 3.1.

Theorem 5.1 *There exist distributions for which C_1 holds and C_2 fails. Also, there exist distributions for which C_0 holds but C_1 and C_2 fail.*

Proof. We only prove the first assertion; the second can be proved by a similar approach.

The following definitions simplify the search for examples. Let $p(x)$ be a continuous non-negative function on $[0,1)$ with $\int_0^1 p(x)dx = \infty$. Define $\phi(x) = e^{-\int_0^x p(t)dt}$, $0 \le x \le 1$, so that $F(x) = 1 - \phi(x)$ is a probability distribution on $[0,1]$. On $[0,1]$ we have $G(x) = x\phi(x)$ and on $[0,1)$ $G'(x) = \phi(x)(1 - xp(x))$, so C_1 holds if

$$(5.1) \qquad \begin{aligned} xp(x) &< 1 \ , \quad 0 < x < \tau \\ xp(x) &> 1 \ , \quad \tau < x < 1 \ . \end{aligned}$$

Now differentiating the function $H(x)$ defined by C_2 gives

$$F^2(x)H'(x) = -\phi(x) - x\phi'(x) + \phi^2(x) + \tau\phi(\tau)\phi'(x) \ ,$$

so C_2 holds only if $h(x) = -1 + xp(x) + \phi(x) - \tau\phi(\tau)p(x) \ge 0$, $\tau < x < 1$.

Thus, let us choose a $p(x)$ as defined above so that, for given $0 < \tau < \theta < 1$, $p(x)$ satisfies (5.1), $\theta p(\theta) < 1 + \tau\phi(\tau)/4$, and $\int_0^\theta p(t)dt > -ln(\tau\phi(\tau)/4)$. Then C_1 holds but C_2 fails, since

$$h(\theta) = [\theta p(\theta) - 1] + \phi(\theta) - \tau\phi(\tau)p(\theta) < \frac{\tau\phi(\tau)}{4} + \frac{\tau\phi(\tau)}{4} - \tau\phi(\tau) < 0 \ .$$

Figure 2 sketches a general example, where the condition $\int_0^\theta p(x)dx > -\ln(\tau\phi(\tau)/4)$ is guaranteed by defining $p(x)$ so that it has a sufficiently large hump in (τ, θ). ■

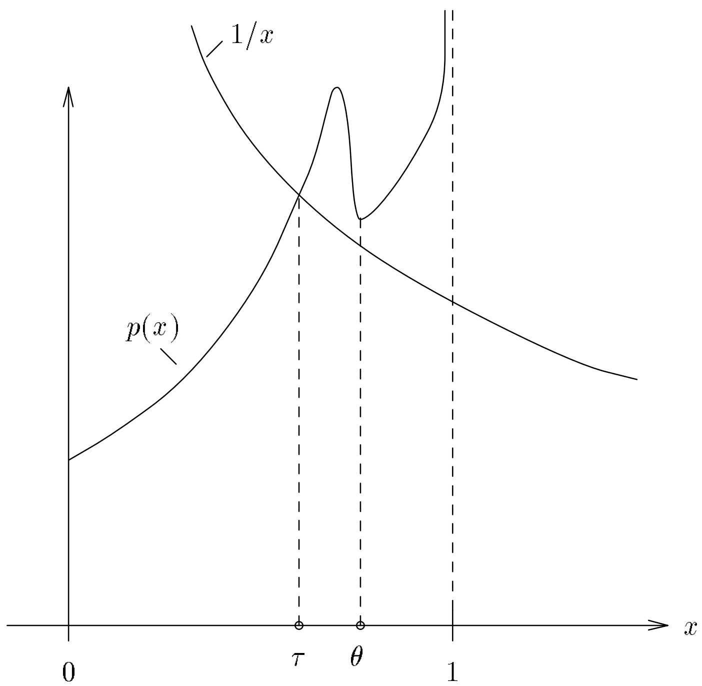

Figure 2 — An example where C_1 holds but C_2 fails for $F(x) = 1 - e^{-\int_0^x p(t)dt}$, $0 \le x \le 1$.

Although C_1 and C_2 were needed to establish the unimodality of $E_n(x)$, they are otherwise somewhat recondite properties of distribution functions. On the other hand, they embrace a wide class of interesting distributions, as the examples below illustrate. Recall our assumptions that $F(0) = 0$, $F(1) = 1$, and $F(x)$ is continuous and strictly increasing in $[0, 1]$.

(i) Consider the convex distributions.

Theorem 5.2 *Suppose F has a nonnegative second derivative, $F''(x) \geq 0$, $0 \leq x \leq 1$. Then C_1 and C_2 hold.*

Proof. From $G(x) = x[1 - F(x)]$ we have $G(0) = G(1) = 0$ and

$$G'(x) = 1 - F(x) - xF'(x), \quad G''(x) = -2F'(x) - xF''(x) \ . \tag{5.2}$$

The assumptions on $F(x)$ readily imply that $F'(x) > 0$ on $(0, 1]$. We conclude from (5.2) that $G''(x) < 0$ for $0 < x \leq 1$, so $G(x)$ is strictly concave in $[0, 1]$. It follows that $G(x)$ is unimodal in $[0, 1]$ with a unique maximum at the solution of

$$G'(\tau) = 1 - F(\tau) - \tau F'(\tau) = 0 \ . \tag{5.3}$$

To show that $H(x)$ is increasing in $\tau \leq x \leq 1$, differentiate and get

$$F^2(x)H'(x) = K(x) \ , \tag{5.4}$$

where

$$K(x) = [x - \tau + \tau F(\tau)]F'(\tau) - F(x)[1 - F(x)] \ . \tag{5.5}$$

Differentiating $K(x)$ gives

$$K'(x) = [x - \tau + \tau F(\tau)]F''(x) + 2F(x)F'(x) \ . \tag{5.6}$$

From (5.3) and (5.5) we get $K(\tau) = 0$, and from (5.6) and $F''(x) \geq 0$ we get $K'(x) > 0$ for $\tau \leq x \leq 1$. Then $K(x)$ and $H'(x)$ are strictly positive in $\tau < x \leq 1$. We conclude that $H(x)$ is increasing in $\tau \leq x \leq 1$. ■

(ii) The convex property in Theorem 5.2 is not necessary for C_1 and C_2. Indeed, C_1 and C_2 also hold for the following useful concave distributions.

Theorem 5.3 *Let $F(x) = x^\alpha$, $0 \leq x \leq 1$, where $0 < \alpha < 1$. Then C_1 and C_2 hold.*

Proof. We have $F''(x) = \alpha(\alpha - 1)x^{\alpha-2} < 0$, $0 < x \leq 1$, so $G''(x) = -(\alpha + \alpha^2)x^{\alpha-1} < 0$, $0 < x \leq 1$. Then $G(x)$ is unimodal with a unique maximum in $[0, 1]$, as in Theorem 5.2.

With $K(x)$ as defined

References

[1] M. Bern, D. H. Greene, A. Raghunathan, and M. Sudan, "On-Line Algorithms for Locating Checkpoints," *Proc. 22nd Ann. ACM Symp. Th. Comput.*, ACM Press, New York (1990), 359–368.

[2] L. B. Boguslavsky, E. G. Coffman, Jr., E. N. Gilbert, and A. Y. Kreinin, "Scheduling Checks and Saves," *ORSA J. Comput.* (to appear).

[3] E. G. Coffman, Jr., L. Flatto, and A. Y. Kreinin, "Scheduling Saves in Fault-Tolerant Computations," AT&T Bell Laboratories, Murray Hill, NJ 07974 (to appear).

[4] E. G. Coffman, Jr. and E. N. Gilbert, "Optimal Strategies for Scheduling Saves and Preventive Maintenance," *IEEE Trans. Reliab.*, **39** (1990), 9–18.

[5] V. G. Kulkarni, V. F. Nicola, and K. S. Trivedi, "Effects of Checkpointing and Queueing on Program Performance," *Stochastic Models*, **6** (1990), 615–648.

[6] J. Storer, *Data Compression*, Computer Science Press, 1988.

[7] S. Toueg and O. Babaoglu, "On the Optimum Checkpoint Selection Problem," *SIAM J. Comput.*, **13** (1984), 630–649.