Scheduling Saves in Fault-Tolerant Computations

E. G. Coffman, Jr.[1], *Leopold Flatto*[1], *A. Y. Kreinin*[2]

Bell Laboratories
[1] AT&T Bell Laboratories, Murray Hill, NJ 07974
[2] Institute of Control Sciences, GSP-312 Moscow U.S.S.R.

1 Introduction

Many computer applications require long computations lasting anywhere from several days to even years. Examples, to name just a few, include cryptography, combinatorial optimization, and asymptotics of stochastic processes. In the scenario studied here, a user starts a program and then leaves the machine unattended; at intervals thereafter the user returns to check on the progress of the computation. On the time scale of the above applications, machine failures or unforeseen shutdowns are sufficiently probable that fault-tolerant programs become necessary. In the method studied here, a program protects itself against undue losses resulting from machine failures by saving its state at certain intervals on a reliable storage device. Failures may occur during saves, but a state successfully saved is perfectly secure thereafter. Then, when the program is restarted after a failure, its new initial state is the last state successfully saved before the failure. Failures (faults, errors, shutdowns) become known to the user only at check times.

As discussed later the ability to determine whether or not a save is successful is a distinctive feature of the system modeled here. For example, in computations of traveling salesman tours, detecting whether a failure occurred during a save may consist simply of determining whether the tour saved is a valid one. To be assured that a successful save exists, one may assume that the program always retains the last two saves; if the most recent one is flawed, then the earlier one may be taken as successful, and hence a valid point at which to restart the program.

Saves are themselves time consuming, so any strategy for scheduling saves must strike a balance between the computing time lost during saves and the computing time that is occasionally lost, because of a failure since the last successful save.

Under a given failure law, the objective here will be to derive a schedule of saves that maximizes the expected amount of work successfully done before a given check time, assuming that the process starts in a recoverable state, and that the process will not complete before the check. The latter assumption approximately models situations where the intervals between checks are small relative to the total computation time. The assumption is exact in those cases where the program never halts voluntarily; the program is halted by the user at some check time when he or she decides that the program has run long enough, e.g., adequate convergence of some process has been observed.

The problem studied here is related to the "checks-and-saves" problem analyzed by Boguslavsky et al. [1]. However, there is one essential difference: in [1] it is assumed that when an error is revealed by a check, the instant the error occurred remains unknown; it is known only that the error occurred since the last check. Thus, there is no point in making a save not immediately preceded by a successful check. Another difference, but one that is not essential, is that in [1] errors could not occur during saves.

Other studies of optimal stochastic checkpointing can be found in [2, 6]. In these models, the "user," which may be a system control program, recognizes failures at the times they occur, as in a computer interrupt system; the response to a failure begins immediately after its occurrence. The large literature on checkpointing is discussed extensively by Trivedi [7], Tantawi and Ruschitzka [5], Goyal et al. [3], and Kulkarni, Nicola, and Trivedi [4]. Most previous research concerns queueing models, with jobs of random sizes arriving at random times.

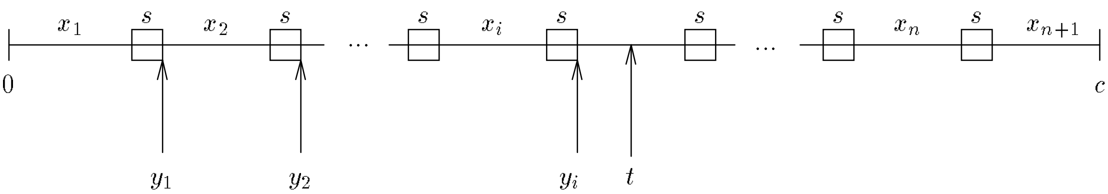
2 Preliminaries

To formalize the problem, suppose that saves are to be scheduled in the interval $[\tau, \tau + c]$, with $\tau + c$ the next check time. As noted earlier, the state at time τ is recoverable and the remaining time of the computation is at least c. For convenience, we put $\tau = 0$ and let $F(t)$, $t \geq 0$, be the failure law, i.e., the probability of a failure in $[0, t]$. Let n be the number of saves to be made in $[0, c]$, each of fixed duration $s > 0$. Then $ns \leq c$ may be assumed. Define x_1 as the computing time before the first save, and x_i, $1 \leq i \leq n$, as the computing time between the $(i-1)$st and ith saves. The computing time between the last save and the progress check is $x_{n+1} = c - ns - x_1 - \cdots - x_n$. Any vector $\mathbf{x}_n = (x_1, \ldots, x_n)$ in the simplex S_n defined by $x_1, \ldots, x_n \geq 0$, $\sum_{i=1}^{n} x_i \leq c - ns$, is called a *strategy*.

With c, s, and F given, $W(\mathbf{x}_n)$ denotes the work done under strategy $\mathbf{x}_n$. The term "work done" refers to the computing successfully done, excluding saves and the time lost, if any, because of a failure. Formally, let $y_j = \sum_{i=1}^{j}(x_i + s)$, $1 \leq j \leq n$, denote the times at which the computation is resumed following saves, and define $y_0 = 0$ and $y_{n+1} = c$. Then conditioned on a failure at time t, the work done is

$$\begin{aligned} W_t(\mathbf{x}_n) &= \sum_{i=1}^{j-1} x_i, \quad y_{j-1} < t \le y_j\,,\ 1 \le j \le n+1 \\ &= \sum_{i=1}^{n+1} x_i, \quad t > y_{n+1} = c\ . \end{aligned}$$

Figure 1 illustrates the definitions.



t = failure time

$\sum_{j=1}^{i} x_j$ = amount saved

Figure 1 – An example

Define the expected value $E(\mathbf{x}_n) = E(W(\mathbf{x}_n)) = \int_0^\infty W_t(\mathbf{x}_n)dF(t)$. In the following sections the main objective is a characterization of the function $E(n) = \sup_{\mathbf{x}_n \in S_n} E(\mathbf{x}_n)$. A strategy $\mathbf{x}_n^*$ is an *optimal n-save strategy* if $E(\mathbf{x}_n^*) = E(n)$; it is called simply an *optimal strategy* if $E(\mathbf{x}_n^*) = \max_{m \ge 0} E(m)$. To compute optimal strategies, characteristics of the functions $E(\mathbf{x}_n)$ and $E(n)$ are needed. Elementary properties following easily from definitions are presented next.

Write $W(\mathbf{x}_n) = \sum_{i=1}^{n+1} \chi_i x_i$, where $\chi_i = 1$ if the computation during x_i is successful, and $\chi_i = 0$ otherwise, $1 \le i \le n+1$. The work during x_i is done if and only if the first failure time satisfies $t > y_i$, so in terms of $G(x) = 1 - F(x)$,

$$E(\mathbf{x}_n) = \sum_{i=1}^{n+1} x_i G(y_i)\ , \tag{2.1}$$

which is conveniently put in the form

$$E(\mathbf{x}_n) = \sum_{i=1}^{n} x_i [G(y_i) - G(y_{n+1})] + (c - ns)G(y_{n+1})\ .$$

Assume hereafter that F has a continuous density f on its support; this assumption is for convenience only. Next, suppose that $E(n)$ is attained at $\mathbf{x}_n$. If $\mathbf{x}_n \in S_n^+$, where S_n^+ is the interior of the simplex S_n, then

$$\frac{\partial E(\mathbf{x}_n)}{\partial x_i} = G(y_i) - G(y_{n+1}) - \sum_{k=i}^{n} x_k f(y_k) = 0, \quad 1 \le i \le n \ . \tag{2.2}$$

Subtracting the $(i+1)^{\text{st}}$ equation from the i^{th} then proves

Lemma 2.1 *If* $\mathbf{x}_n \in S_n^+$ *is an optimal* n*-save strategy, then*

$$G(y_i) - G(y_{i+1}) = x_i f(y_i), \quad 1 \le i \le n \ . \tag{2.3}$$

The following three sections treat the uniform failure law. The normalization $c = 1$ is convenient in this case, so the failure law is $F(t) = \alpha t$, $0 \le t \le 1/\alpha$, with $0 < \alpha \le 1$ to avoid trivialities. A principal result will be that $E(n)$ is unimodal in $0 \le n \le 1/s$. Calculations lead to explicit formulas for $E(n)$ and optimal n-save strategies. For other $F(t)$, the solution to (2.3) is not elementary in general. This is true even for the exponential failure law covered in Section 6. The computation of $E(n)$ is relatively simple, but the functions involved seem difficult to analyze. In particular, the unimodality of $E(n)$ is an open question for this case.

3 Failures Uniform on $[0, 1]$

Recall that $c = 1$ applies to this and the next two sections. Accordingly, this section analyzes the case $F(t) = t$, $0 \le t \le 1$. The next section considers the general case, $F(t) = \alpha t$, $0 \le t \le 1/\alpha$, $0 < \alpha < 1$.

Observe that $\alpha = 1$ implies $G(y_{n+1}) = G(1) = 0$, so with probability 1 a failure occurs before the check at time 1. Later results for general $0 < \alpha < 1$ are easily expressed in terms of those for $\alpha = 1$. We may assume $0 < s < 1$, $1 \le n \le 1/s$.

Equation (2.1) becomes

$$E(\mathbf{x}_n) = \sum_{i=1}^{n} x_i(1 - si - x_1 - \ldots - x_i) = \sum_{i=1}^{n}(1 - si)x_i - \sum_{1 \le i \le j \le n} x_i x_j \ . \tag{3.1}$$

Theorem 3.1 *If $N = N(s) \geq 1$ is the largest integer n such that $n(n+1) \leq 2/s$, i.e., if $N = \left\lfloor \frac{-1+\sqrt{1+8/s}}{2} \right\rfloor$, then*

$$E(n) = E(n,s) = \frac{1}{2}\frac{n}{n+1} - \frac{s}{2}n + \frac{s^2}{24}n(n+1)(n+2), \quad 1 \leq n \leq N \ . \tag{3.2}$$

$E(n)$ is strictly increasing in $1 \leq n \leq N-1$ and strictly decreasing in $N \leq n \leq 1/s$. Also, $E(N-1) \leq E(N)$ with equality if and only if $\frac{-1+\sqrt{1+8/s}}{2}$ is an integer.

Remark. A more complicated formula applies to $E(n)$ for $n > N$. Since it is not needed in the proof, it is omitted.

Proof. We first prove a lemma that describes $E(\mathbf{x}_n)$ as a function extended over all of $\mathbf{R}^n$. We have

$$\sum_{1 \leq i \leq j \leq n} x_i x_j = \frac{1}{2}\sum_{i=1}^{n} x_i^2 + \frac{1}{2}\left(\sum_{i=1}^{n} x_i\right)^2 \geq \frac{1}{2}\sum_{i=1}^{n} x_i^2 \ . \tag{3.3}$$

Then $\sum_{1 \leq i \leq j \leq n} x_i x_j$ is clearly positive definite, so (3.1) yields

Lemma 3.1 *$E(\mathbf{x}_n)$ attains its global maximum at a unique point in $\mathbf{R}^n$.*

We consider separately the cases $n \leq N$ and $n \geq N$.

($n \leq N$). By (3.1) the point at which $E(\mathbf{x}_n)$ assumes its global maximum in $\mathbf{R}^n$ is given by

$$\frac{\partial E(\mathbf{x}_n)}{\partial x_i} = 1 - si - (x_1 + \cdots + x_{i-1} + 2x_i + x_{i+1} + \cdots + x_n) = 0, \quad 1 \leq i \leq n \ . \tag{3.4}$$

Adding the n equations gives $\sum_{i=1}^{n} x_i = \frac{n}{n+1} - \frac{s}{2}n$, whereupon (3.4) yields

$$x_i = \frac{1}{n+1} + s\left(\frac{n}{2} - i\right), \quad 1 \leq i \leq n \tag{3.5}$$

$$x_{n+1} = 1 - ns - x_1 - \cdots - x_n = x_n \ . \tag{3.6}$$

From (3.5), it follows that $x_1 > \cdots > x_n$. Hence the point $(x_1, \ldots, x_n)$ defined by (3.5) is a strategy, i.e., lies in S_n, provided $x_n \geq 0$. This assertion is equivalent to $n(n+1) \leq 2/s$ and hence $n \leq N$.

Using (3.3), rewrite (3.1) as

$$E(\mathbf{x}_n) = \sum_{i=1}^{n} x_i - s\sum_{i=1}^{n} ix_i - \frac{1}{2}\sum_{i=1}^{n} x_i^2 - \frac{1}{2}\left(\sum_{i=1}^{n} x_i\right)^2 . \tag{3.7}$$

Let the x_i be given by (3.5). Computations give

$$\begin{aligned} \sum_{i=1}^{n} ix_i &= \frac{n}{2} - s\frac{n^3+3n^2+2n}{12} , \\ \sum_{i=1}^{n} x_i^2 &= \frac{n}{(n+1)^2} - s\frac{n}{n+1} + s^2\frac{n^3+2n}{12} . \end{aligned} \tag{3.8}$$

For $n \le N$, $E(n) = E(\mathbf{x}_n)$ with $\mathbf{x}_n$ given by (3.5). Then (3.7) and (3.8) yield (3.2). Finally, (3.2) yields

$$E(n) - E(n-1) = \frac{1}{2}\left[\frac{s^2 n(n+1)}{4} + \frac{1}{n(n+1)} - s\right] = \frac{s^2\left[2/s - n(n+1)\right]^2}{n(n+1)} > 0 ,$$

so the theorem for $n \le N$ is proved.

For the case $n \ge N$, the following two lemmas will be useful.

Lemma 3.2 *Let $\mathbf{x}_n$ be an optimal n-step strategy. Then $x_n = x_{n+1}$.*

Remark. If $\mathbf{x}_n \in S_n^+$ then Lemma 3.2 becomes (3.6). Lemma 3.2 states that the result also holds if $\mathbf{x}_n \in \partial S_n$, where ∂S_n denotes the boundary of S_n.

Proof. Write

$$E(\mathbf{x}_n) = \sum_{i=1}^{n-1} x_i(1 - si - x_1 - \cdots - x_n) + x_n(w - x_n) ,$$

where $w = 1 - sn - x_1 - \cdots - x_{n-1}$. With $x_1, \ldots, x_{n-1}$ fixed, x_n varies from 0 to w. But $x_n(w - x_n)$ attains its maximum at $x_n = w/2$, so $x_{n+1} = w - x_n = s/2 = x_n$. ∎

Lemma 3.3 *if $\mathbf{x}_n$ is an optimal n-step strategy, and if $x_i = 0$ for some $1 \le i \le n$, then $x_{i+1} = 0$.*

Remark. Lemmas 3.2 and 3.3 imply that if the expected work done attains its maximum over S_n at $\mathbf{x}_n \in \partial S_n$, then the set of indices i for which $x_i = 0$ is a consecutive string of integers, the first being at most n and the last being $n+1$. In particular, $x_n = x_{n+1} = 0$.

Proof. If $i = n$, then Lemma 3.3 follows from Lemma 3.2. Next, suppose to the contrary that $x_i = 0$, but that $x_{i+1} > 0$ for some $1 \leq i \leq n-1$. Let

$$X(t) = E(x_1, \ldots, x_{i-1}, t,\ x_{i+1} - t,\ x_{i+2}, \ldots, x_n), \quad 0 \leq t \leq x_{i+1} \ .$$

By (2.2), $X'(0) = \frac{\partial E(\mathbf{x}_n)}{\partial x_i} - \frac{\partial E(\mathbf{x}_n)}{\partial x_{i+1}} = s + x_{i+1} > 0$. Hence, for small $t > 0$, $X(t) > E(\mathbf{x}_n)$, which contradicts the n-save optimality of $\mathbf{x}_n$.

$(n \geq N)$. The remainder of the proof shows that $E(n) < E(n-1)$, $n > N$. When $n > N$, the point at which $\frac{\partial E(\mathbf{x}_n)}{\partial x_i} = 0$, $1 \leq i \leq n$, lies outside S_n. Then $E(n)$ is attained at a point $\mathbf{x}_n \in \partial S_n$. By the remark following Lemma 3.3, $x_n = x_{n+1} = 0$. This means that the last two saves are scheduled in $[1-2s, 1]$. The last save cannot save work not already saved, so it can be eliminated. This means that $E(n) = E(\mathbf{x}_n) = E(\mathbf{z}_{n-1})$, where $\mathbf{z}_{n-1} \in S_{n-1}$, $z_i = x_i$, $1 \leq i \leq n-1$, and $z_n = s$.

It remains to prove that $E(\mathbf{z}_{n-1}) < E(n-1)$. If $z_{n-1} = 0$, then $z_{n-1} \neq z_n$ and by Lemma 3.2, $E_{n-1}(\mathbf{z}_{n-1}) < E(n-1)$. Thus suppose $z_{n-1} > 0$, and hence by the remark following Lemma 3.3, $z_i > 0$, $1 \leq i \leq n$, i.e., $\mathbf{z}_{n-1} \in S^+_{n-1}$. If $n > N+1$, then $n-1 > N$ so that $E(n-1)$ is attained on the boundary ∂S_{n-1}. Then again, $E(\mathbf{z}_{n-1}) < E(n-1)$. Finally, if $n = N+1$, then $E(n-1)$ is attained at a point $\mathbf{u}_{n-1}$, where $u_i = \frac{1}{n} + s\left(\frac{n-1}{2} - i\right)$, $1 \leq i \leq n-1$. But $2/s < (N+1)(N+2) = n(n+1)$, which is equivalent to $u_{n-1} < s$. Then $u_n = u_{n-1} \neq z_n$, so $\mathbf{u}_{n-1} \neq \mathbf{z}_{n-1}$ and again $E(\mathbf{z}_{n-1}) < E(n-1)$. ■

The two corollaries below are the main results of this section; they follow immediately from Theorem 3.1.

Corollary 3.1 *The strategy $\mathbf{x}_N$ with the x_i's given by (3.5) and (3.6) is uniquely optimal, unless $\frac{-1\sqrt{1+8/s}}{2}$ is an integer, in which case there are precisely two optimal strategies, x_N and x_{N-1}.*

Next, consider the behavior of an optimal strategy as a function of s. Let $E(s) = E(N, s) = \max_{n \geq 1} E(n, s)$.

Corollary 3.2 *Let N run through the positive integers. Then*

$$E(s) = \frac{1}{2}\frac{N}{N+1} - \frac{s}{2}N + \frac{s^2}{24}N(N+1)(N+2), \; \frac{2}{(N+1)(N+2)} \leq s \leq \frac{2}{N(N+1)} \; . \tag{3.9}$$

$E(s)$ is a continuous, strictly decreasing function in $0 < s \leq 1$ with $E(1) = 0$ and

$$E(s) = \frac{1}{2} - \frac{7}{12}\sqrt{s} + O(s) \quad as \quad s \to 0 \; . \tag{3.10}$$

Proof. Equation (3.9) follows immediately from Theorem 3.1. The intuitive fact that $E(s)$ is continuous and decreasing with $E(1) = 0$ follows easily from (3.9). The asymptotics in (3.10) are also readily derived from (3.9). In particular, $\lim_{s\to 0} E(s) = \frac{1}{2}$ is the mean of the uniform distribution on $[0,1]$. This result can be proved directly without formulas and generalizes to all distributions. ■

4 Failures Uniform on $[0, 1/\alpha]$

The analysis for $0 < \alpha < 1$ is easily expressed in terms of that for $\alpha = 1$. With α a parameter, let $E(n,\alpha)$ denote the maximum-expected work done with n saves. Let $E(n) = E(n,1)$ continue to have the meaning in the last section. Note that the conditional probability distribution of a failure time in $[0,1]$, given that one occurs there, is uniform on $[0,1]$. Then

$$E(n,\alpha) = \alpha E(n) + (1-\alpha)(1-ns), \quad 0 \leq n \leq 1/s \; , \tag{4.1}$$

where $E(0) = 0$, consistent with (3.2). Note that, in contrast to Section 3, the case of $n = 0$ saves makes sense when $\alpha < 1$. As shown below there are values of s such that $E(0,\alpha) = 1 - \alpha = \max_{n\geq 0} E(n,\alpha)$ and the policy of no saves is optimal.

We have

$$E(n,\alpha) - E(n-1,\alpha) = \alpha[E(n) - E(n-1)] - s(1-\alpha), \quad 1 \leq n \leq 1/s \; . \tag{4.2}$$

Theorem 3.1 and (4.2) show that $E(n,\alpha)$ is strictly decreasing for $n \geq N$. For $n \leq N$, (3.2) and (4.2) yield

$$E(n,\alpha) - E(n-1,\alpha) = \frac{\alpha s^2}{8n(n+1)} q(n(n+1)) \;, \tag{4.3}$$

where

$$q(y) = q(y,s,\alpha) = y^2 - \frac{4}{s}\left(\frac{2}{\alpha} - 1\right) y + \frac{4}{s^2} \;. \tag{4.4}$$

Now $q(y)$ has two positive roots y_1, y_2 satisfying $y_1 < 2/s < y_2$, where

$$y_1 = \frac{2}{s} y(\alpha), \quad y(\alpha) = \frac{2}{\alpha} - 1 - \sqrt{\left(\frac{2}{\alpha} - 1\right)^2 - 1} \;. \tag{4.5}$$

By calculus one checks that $y(\alpha)$ increases on $[0,1]$ with $y(0) = 0$ and $y(1) = 1$. Equations (4.2) and (4.3) then give

Theorem 4.1 *Let $0 < s < 1$ and $0 < \alpha < 1$. If $s > y(\alpha)$, then $E(n,\alpha)$ is strictly decreasing in n for $0 \leq n \leq 1/s$. If for some N, $1 \leq N < \infty$, s satisfies*

$$\frac{2y(\alpha)}{(N+1)(N+2)} < s \leq \frac{2y(\alpha)}{N(N+1)} \tag{4.6}$$

then $E(n,\alpha)$ is strictly increasing in $0 \leq n \leq N-1$ and strictly decreasing in $N \leq n \leq 1/s$. Also, $E(N-1,\alpha) \leq E(N,\alpha)$ with equality if and only if equality holds in (4.6).

By Theorem 4.1 and the results of Section 3, a solution to the optimization problem can be described as follows for any $0 < \alpha \leq 1$: If $s > y(\alpha)$, with $y(\alpha)$ given by (4.5), then the policy with no saves is uniquely optimal. Otherwise, we can choose

$$N = \left\lfloor \frac{-1 + \sqrt{1 + 8y(\alpha)/s}}{2} \right\rfloor \geq 1 \tag{4.7}$$

saves; choose the x_i, $1 \leq i \leq N+1$, given by (3.5) and (3.6); and obtain the expected work done, $E(N)$, given by (4.1) and (3.2). This solution is unique unless the expression within the floor brackets of (4.7) is a positive integer, in which case there are exactly two optimal solutions, the other having $N-1$ saves. Numerical examples are given at the end of the next section.

5 Uniform Failure Law and Equal Computing Intervals

It is of interest to compare the unrestricted optimal policy with an optimal policy subject to the constraint that all computing intervals be equal. Assume that $x_1 = x_2 = \cdots = x_{n+1} = x$. Then $(n+1)x + ns = 1$, or $x = (1-ns)/(n+1)$. Let $\tilde{E}(n,\alpha)$, $0 < \alpha \le 1$, be the expected amount of work done by an n-save strategy $\mathbf{x}_n$ with $x_i = x$, $1 \le i \le n+1$. Consider first the case $\alpha = 1$, with $\tilde{E}(n) = \tilde{E}(n,1)$. By (3.1) and the above value of x

$$\tilde{E}(n) = \sum_{i=1}^{n}(1-si)x - \frac{n(n+1)}{2}x^2 = \frac{(1-s)(1-ns)}{2}\frac{n}{n+1} \ . \tag{5.1}$$

Then

$$\tilde{E}(n) - \tilde{E}(n-1) = \frac{1-s}{2}\frac{r(n)}{n(n+1)}, \quad r(n) = 1 + s - sn - sn^2 \ . \tag{5.2}$$

For $0 < \alpha \le 1$, (5.1) and (5.2) give

$$\tilde{E}(n,\alpha) = \alpha\tilde{E}(n) + (1-\alpha)(1-ns) = \left[\frac{\alpha(1-s)}{2}\frac{n}{n+1} + 1 - \alpha\right](1-ns) \tag{5.3}$$

and

$$\tilde{E}(n,\alpha) - \tilde{E}(n-1,\alpha) = \alpha[\tilde{E}(n) - \tilde{E}(n-1)] - (1-\alpha)s = \frac{t(n)}{n(n+1)} \ , \tag{5.4}$$

where

$$t(n) = \alpha\tfrac{1-s}{2}r(n) - (1-\alpha)sn(n+1) = \alpha\tfrac{1-s^2}{2} - \left[(1-\alpha)s + \tfrac{\alpha s(1-s)}{2}\right]n(n+1) \ . \tag{5.5}$$

From (5.1)-(5.5) we obtain

Theorem 5.1 *Let $N = N(s,\alpha)$, $0 < \alpha \le 1$, be the largest non-negative integer such that*

$$N(N+1) \le \frac{\alpha(1-s^2)}{2(1-\alpha)s + \alpha s(1-s)} \ . \tag{5.6}$$

Then $\tilde{E}(n,\alpha)$ is strictly increasing in $0 \le n \le N-1$ and strictly decreasing in $N \le n \le 1/s$. Also, $\tilde{E}(N-1) \le \tilde{E}(N)$ with equality if and only if equality holds in (5.6).

Table 1 shows examples comparing the unrestricted optimal policy with one described by Theorem 5.1, where $x_i = (1 - \tilde{N}s)/(\tilde{N}+1)$, $1 \le i \le \tilde{N}+1$, with $\tilde{N}$ optimal, i.e., $\tilde{E}(\tilde{N}) = \max_{m \ge 0} \tilde{E}(m)$. These examples, and many others that were computed, show that very little is lost by restricting the computing intervals to be equal.

Table 1 – $E(N)$ vs. $\tilde{E}(\tilde{N})$.

	$\alpha = .5$				$\alpha = 1.0$			
	N	$E(N)$	$\tilde{N}$	$\tilde{E}(\tilde{N})$	N	$E(N)$	$\tilde{N}$	$\tilde{E}(\tilde{N})$
$s = .001$	18	.7235	17	.7234	44	.4707	31	.4689
.01	5	.6713	5	.6709	13	.4107	9	.4054
.1	1	.5513	1	.5513	3,4	.2500	2	.2400

Table 2 illustrates the shape of the function $E(n)$ for $1 \le n \le N$. When N is moderately large the function climbs quickly to a region in which $E(n)$ remains nearly flat up to the mode N. These observations suggest that, even if the assumed uniform law is only a moderately good approximation, the results for the optimal equal-computing-interval policy may still be quite good.

Table 2 – $E(n)$, $1 \le n \le N$, for $\alpha = 1.0$.

$s = .1$		$s = .01$	
n	$E(n)$	n	$E(n)$
1	.2025	1	.2450
2	.2433	2	.3234
3	.2500	3	.3603
4	.2500	4	.3805
		5	.3925
		6	.4000
		7	.4046
		8	.4074
		9	.4091
		10	.4100
		11	.4105
		12	.4106
		13	.4107

6 Exponential Failure Law

As a convenient normalization, the failure rate is taken to be 1. The problem instance is then defined by arbitrary c, s subject to $c > s > 0$, and by the failure law $Pr\{\text{failure time} > t) = e^{-t}$,

$0 \le t < \infty$. By (2.1) and (2.2), we have for $1 \le n \le c/s$, $\mathbf{x}_n = (x_1, \ldots, x_n)$,

$$\begin{aligned} E(\mathbf{x}_n) &= \sum_{i=1}^{n} x_i e^{-y_i} + x_{n+1} e^{-c} \\ &= \sum_{i=1}^{n} x_i e^{-y_i} + (c - ns - x_1 - \cdots - x_n) e^{-c} \ , \end{aligned}$$

and hence

$$E(\mathbf{x}_n) = \sum_{i=1}^{n} x_i (e^{-y_i} - e^{-c}) + e^{-c}(c - ns) \ , \tag{6.1}$$

with the partial derivatives,

$$\frac{\partial E(\mathbf{x}_n)}{\partial x_i} = e^{-y_i} - e^{-c} - \sum_{k=1}^{n} x_k e^{-y_k}; \quad 1 \le i \le n \ . \tag{6.2}$$

If we define $\phi(x) = 1 - e^{-(s+x)}$, $\psi(x) = 1 - e^{-x}$, then (6.2) yields

$$\begin{aligned} \frac{\partial E(\mathbf{x}_n)}{\partial x_i} - \frac{\partial E(\mathbf{x}_n)}{\partial x_{i+1}} &= e^{-y_i} - e^{-y_{i+1}} - x_i e^{-y_i} = e^{-y_i}[\phi(x_{i+1}) - x_i], \quad 1 \le i \le n-1, \\ \frac{\partial E(\mathbf{x}_n)}{\partial x_n} &= e^{-y_n} - e^{-c} - x_n e^{-y_n} = e^{-y_n}[\psi(x_{n+1}) - x_n] \ . \end{aligned} \tag{6.3}$$

Let ϕ^k denote the k^{th} composition of ϕ : $\phi^0(x) = x$ and $\phi^k(x) = \phi(\phi^{k-1}(x))$, $k \ge 1$. Define $h_n(x) = x + \sum_{k=0}^{n-1} \phi^k(\psi(x))$. The functions $\phi(x)$ and $\psi(x)$ are strictly increasing in $0 \le x < \infty$, with $\psi(0) = 0$. Hence, $h_n(x)$ is strictly increasing in $0 \le x < \infty$ with $h_n(0) = \phi(0) + \cdots + \phi^{n-1}(0)$, $h_n(\infty) = \infty$.

Lemma 6.1 *$E(\mathbf{x}_n)$ has a critical point $\mathbf{x}_n^* \in S_n$, where the partial derivatives all vanish, if and only if $c \ge ns + h_n(0)$. This point is unique and given by*

$$x_n^* = \psi(x_{n+1}^*), \ x_{n-1}^* = \phi(\psi(x_{n+1}^*)), \ldots, x_1^* = \phi^{n-1}(\psi(x_{n+1}^*)) \ , \tag{6.4}$$

with $x_{n+1}^* \ge 0$ determined by

$$c = ns + h_n(x_{n+1}^*) \ . \tag{6.5}$$

Also, $\mathbf{x}_n^* \in S_n^+$ if and only if $c > ns + h_n(0)$.

Proof. Suppose $\mathbf{x}_n^* \in S_n$ is a critical point of $E(\mathbf{x}_n)$. By (6.3), $x_n^* = \psi(x_{n+1}^*)$, $x_i^* = \phi(x_{i+1}^*)$, $1 \le i \le n-1$, which yields (6.4); (6.5) follows from (6.4) and $c = ns + x_1^* + \cdots + x_{n+1}^*$.

Conversely, let $c \ge ns + h_n(0)$. Since $h_n(x)$ is strictly increasing in $0 \le x < \infty$, there is a unique $x_{n+1}^* \ge 0$ solving (6.5). Then the strategy defined by (6.4) is the unique critical point in S_n. If $c > ns + h_n(0)$, then the $x_1^*, \ldots, x_{n+1}^*$ defined by (6.4) and (6.5) are all strictly positive, so $\mathbf{x}_n^* \in S_n^+$. If $c = ns + h_n(0)$, then $x_{n+1}^* = 0$, so $\mathbf{x}_n^* \in \partial S_n$. ■

Lemma 6.2 *If $E(\mathbf{x}_n)$ attains its maximum at $\mathbf{x}_n^* \in \partial S_n$, then the set of indices $1 \le i \le n+1$ for which $x_i^* = 0$ forms a consecutive string of integers containing n and $n+1$. Furthermore, $c \le ns + h_n(0)$.*

Proof. Since $\mathbf{x}_n^* \in \partial S_n$, we have $x_i^* = 0$ for some i, $1 \le i \le n+1$. Suppose that $x_i^* = 0$, but $x_{i+1}^* > 0$ for some $i < n$. Let

$$X(t) = E(x_1^*, \ldots, x_{i-1}^*, t, x_{i+1}^* - t, x_{i+2}^*, \ldots, x_n^*), \quad 0 \le t \le x_{i+1}^* \ .$$

By (6.3), the derivative at $t = 0$ is

$$X'(0) = \frac{\partial E(\mathbf{x}_n^*)}{\partial x_i} - \frac{\partial E(\mathbf{x}_n^*)}{\partial x_{i+1}} = e^{-y_i^*} \phi(x_{i+1}^*) > 0 \ ,$$

where $y_i^* = \sum_{k=1}^{i} (x_k^* + s)$. Hence, $X(t) > E(\mathbf{x}_n^*)$ for small $t > 0$, contradicting the fact that $E(\mathbf{x}_n)$ attains its maximum at $\mathbf{x}_n^*$. Thus, $x_{i+1}^* = 0$ if $x_i^* = 0$, $i < n$. Similar reasoning shows that $x_{n+1}^* = 0$ must hold if $x_n^* = 0$. Thus, the set of indices for which $x_i^* = 0$ forms a consecutive set of integers containing x_{n+1}^*.

As shown below, the remainder of the lemma will follow once we have verified that

$$x_i^* \le \phi(x_{i+1}^*), \quad 1 \le i \le n-1, \quad x_n^* \le \psi(x_{n+1}^*) \ . \tag{6.6}$$

Suppose to the contrary that $x_i^* > \phi(x_{i+1}^*)$ for some i, $1 \le i \le n-1$. Define

$$X(t) = E(x_1^*, \ldots, x_{i-1}^*, x_i^* - t, x_{i+1}^* + t, x_{i+2}^*, \ldots, x_n^*), \quad 0 \le t \le x_i^* \ .$$

By (6.3),

$$X'(0) = e^{-y_i^*}[x_i^* - \phi(x_{i+1}^*)] > 0 \ .$$

Then $X(t) > E(\mathbf{x}_n^*)$ for small $t > 0$, contradicting the fact that $E(\mathbf{x}_n)$ attains its maximum at $\mathbf{x}_n^*$. We conclude that $x_i^* \le \phi(x_{i+1}^*)$. A similar argument gives $x_n^* \le \psi(x_{n+1}^*)$.

If $x_{n+1}^* = 0$, then $x_n^* = 0$ follows from $x_n^* \le \psi(x_{n+1}^*)$; this completes the proof of the first assertion of the lemma. As ϕ and ψ are increasing in $0 \le x < \infty$, we conclude from (6.6) that

$$x_i^* \le \phi^{n-i}(0), \quad 1 \le i \le n-1 \ . \tag{6.7}$$

Then the second assertion of the lemma is given by

$$c = ns + x_1^* + \cdots + x_{n-1}^* \le ns + \phi(0) + \cdots + \phi^{n-1}(0) = ns + h_n(0) \ . \quad \blacksquare \tag{6.8}$$

The main results of this section follow easily from Lemmas 6.1 and 6.2.

Theorem 6.1 *If $ns + h_n(0) \le c$, then $E(\mathbf{x}_n)$ attains its maximum over S_n at the unique critical point given by (6.4) and (6.5). If $ns + h_n(0) \ge c$, then the maximum can only be attained at a point $\mathbf{x}_n \in \partial S_n$.*

Proof. If the maximum is attained in ∂S_n, then by Lemma 6.2 $ns + h_n(0) \ge c$. Consequently, by Lemma 6.1, $ns + h_n(0) < c$ implies that the maximum of $E(\mathbf{x}_n)$ is attained at the interior point of S_n given by (6.4) and (6.5). If $ns + h_n(0) \ge c$, then again by Lemma 6.1, $E(\mathbf{x}_n)$ has no interior critical point, so the maximum is attained in ∂S_n. For $ns + h_n(0) = c$, the maximum occurs at $(x_1^*, \ldots, x_n^*) \in S_n$; (6.7) and (6.8) show that $(x_1^*, \ldots, x_n^*) = (\phi^n(0), \ldots, \phi(0), 0)$, which is the critical point defined by (6.4) and (6.5). $\blacksquare$

Let $N = N(c, s)$ be the largest integer $n > 0$ such that $ns + h_n(0) \le c$.

Theorem 6.2 *We have*

$$\begin{aligned} E(0) &= ce^{-c} \\ E(n) &= e^{-s-\phi^{n-1}(\psi(x))} + e^{-c}x - e^{-c}, \quad 1 \le n \le N \ , \end{aligned} \tag{6.9}$$

where x is the unique non-negative root of $ns + h_n(x) = c$. The computing intervals are given by $x_{n+1} = x$ and $x_i = \phi^{n-i}(\psi(x))$, $1 \leq i \leq n$. Finally, $E(n)$ is strictly decreasing for $n \geq N$.

Proof. The value of $E(0)$ is clear. For $1 \leq n \leq N$, we have $ns + h_n(0) \leq c$, so by Theorem 6.1, $E(n) = E(\mathbf{x}_n^*)$ where $\mathbf{x}_n^*$ is given by (6.4) and (6.5). Substitution of (6.4) and (6.5) into (6.1) and elementary manipulations yield (6.9) with $x = x_{n+1}^*$.

Now let $n > N$. By Theorem 6.1, $E(\mathbf{x}_n)$ attains its maximum at a point $\mathbf{x}_n^* \in \partial S_n$. Thus, $x_i^* = 0$ for some i, $1 \leq i \leq n+1$. If $1 \leq i \leq n$ then the i^{th} save is superfluous, and if $i = n+1$ the n^{th} save is superfluous. Remove the i^{th} computing interval and let $z_1, \ldots, z_n$ be the resulting computing intervals. Then $E(x_n^*) < E(\mathbf{z}_{n-1})$, so that $E(n) < E(n-1)$. ■

For the equal-computing-interval policy let $x_i = (c - ns)/(n+1)$, $1 \leq i \leq n+1$. Substitution into (6.1) and simplification yields

$$\tilde{E}(n) = \frac{c - ns}{n+1}\left[e^{-c} + \frac{e^{-(c+s)/(n+1)}}{1 - e^{-(c+s)/(n+1)}}(1 - e^{-n(c+s)/(n+1)})\right] . \tag{6.10}$$

Table 3 shows examples comparing $E(n)$ and $\tilde{E}(n)$, as n increases up to the largest value for which $ns + h_n(0) \leq c$.

While $E(n)$ and $\tilde{E}(n)$ can differ substantially, we again see, as in the uniform case, that $\max_{m \geq 0} E(m)$ and $\max_{m \geq 0} \tilde{E}(m)$ are quite close. Note also that the functions $E(n)$ and $\tilde{E}(n)$ are unimodal and quite flat around the mode as in the uniform case. Based on Table 2 and many examples not shown here, it seems reasonable to conjecture that $E(n)$ is in fact unimodal always.

Table 3 – $E(n)$ vs. $\tilde{E}(n)$ for the Exponential Law.
$s = .1$, maxima are in boldface.

$c = 1.0$			$c = 4.0$			$c = 10.0$		
n	$E(n)$	$\tilde{E}(n)$	n	$E(n)$	$\tilde{E}(n)$	n	$E(n)$	$\tilde{E}(n)$
0	.367879	.367879	0	.073263	.073263	0	.000454	.000454
1	**.426560**	**.425173**	1	.386473	.286749	1	.333275	.031951
2	.411336	.410993	2	.504736	.428479	2	.464714	.116752
3	.378701	.374961	3	.559605	.510634	3	.529929	.211038
4	.342015	.329528	4	.586640	.558569	4	.565607	.293649
			5	.599756	.586111	5	.586131	.361197
			6	.605500	**.600722**	6	.598270	.415377
			7	**.607282**	.606676	7	.605565	.458702
			8	.606982	.606570	8	.609988	.493413
			9	.605661	.602063	9	.612683	.521293
			10	.603919	.594264	10	.614328	.543720
						11	.615330	.561751
						12	.615939	.576197
						13	.616307	.587690
						14	.616526	.596726
						15	.616654	.603699
						16	.616727	.608922
						17	.616766	.612652
						18	.616785	.615097
						19	.616792	.616428
						20	**.616793**	**.616789**
						21	.616791	.616299
						22	.616787	.615059
						23	.616782	.613154

Table 4 illustrates optimal computing-interval sequences, and shows how closely they appear to approach the uniform sequence $x_i = (c - Ns)/(N+1)$, $1 \le i \le N+1$, as N becomes large. Note that for fixed s, either the sequence is monotonically increasing or $x_1, \ldots, x_N$ is monotonically decreasing with $x_N < x_{N+1}$. With s fixed, the latter property appears to hold for all c sufficiently large. Indeed, this property can be proved analytically.

Table 4 – Optimal Computing Intervals for Examples in Table 3.

$c = 1.0$		$c = 4.0$		$c = 10.0$	
i	x_i	i	x_i	i	x_i
1	.395945	1	.384901	1	.383179
2	.504089	2	.385973	2	.383177
		3	.387716	3	.383173
		4	.390559	4	.383167
		5	.395213	5	.383157
		6	.402879	6	.383141
		7	.415636	7	.383115
		8	.537231	8	.383073
				9	.383005
				10	.382894
				11	.382715
				12	.382424
				13	.381953
				14	.381191
				15	.379958
				16	.377968
				17	.374764
				18	.369626
				19	.361442
				20	.348542
				21	.428543

7 Final Remarks

In a natural generalization of the problem here, there is a sequence of check intervals with given durations $c_1, c_2, \ldots, c_m$, along with parameters describing the check operation itself, e.g., a time to make a check and, if the check shows that the machine is down, a time to make a repair. The problem of scheduling saves seems to be substantially more difficult, especially when the failure law does not regenerate at the beginning of each check interval. The corresponding problems in [1, 2] also remain open for general failure laws.

With a failure law that regenerates at the beginning of each check interval, the above check-and-save problem would decompose into independent single-interval problems, if one could assume that

(i) the probability of a failure during a check is negligible, as in [1, 2], and

(ii) as part of each check, the current state is recorded.

Then the analysis of preceding sections would apply.

However, if these assumptions do not hold, the earlier results do not apply directly. For, if a failure occurs between a check time and the end of the next save, then it may be necessary to restart in the state of the last successful save of the preceding check interval. This would be inconsistent with the earlier assumption that, if a failure does not occur in $[0, c]$, then all of the computing time, including x_{n+1}, is successfully done. Given a sufficiently simple model of a check operation under assumption (ii), an extension of the earlier model to include the check itself would appear to be a promising line of research. Note, however, that if the computing intervals are small compared to check intervals, then a decomposition into independent check intervals and check operations may well yield a good approximation to optimal design.

Acknowledgements. We gratefully acknowledge helpful discussions with D. S. Johnson and P. W. Shor. In particular, the long computations carried out by Johnson on the traveling salesman problem provided the motivation for the problem studied in this paper.

References

[1] Boguslavsky, L. B., Coffman, E. G., Jr., Gilbert, E. N., and Kreinin, A. Y., "Scheduling Checks and Saves," *ORSA J. Comput.*, **4** (1992), 60–69.

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

[3] A. Goyal, V. Nicola, A. N. Tantawi, K. S. Trivedi, "Reliability of systems with limited repairs," *IEEE Trans. Reliability*, (Special Issue on Fault Tolerant Computing), vol. R-36, 1987, pp. 202–207.

[4] V. G. Kulkarni, V. F. Nicola, K. S. Trivedi, "Effects of checkpointing and queueing on program performance," *Research Rep.* RC 13283, IBM Research, Yorktown Heights, NY 10598 USA.

[5] A. N. Tantawi, M. Ruschitzka, "Performance analysis of checkpointing strategies," *ACM Trans. Computer Systems*, 2, 1984, pp. 123–144.

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

[7] K. S. Trivedi, "Reliability evaluation for fault tolerant systems," in *Mathematical Computer Performance and Reliability*, North-Holland, 1983, pp. 403–414.

Scheduling Saves in Fault-Tolerant Computations

E. G. Coffman, Jr.[1], *Leopold Flatto*[1], *A. Y. Kreinin*[2]

Bell Laboratories
[1] AT&T Bell Laboratories, Murray Hill, NJ 07974
[2] Institute of Control Sciences, GSP-312 Moscow U.S.S.R.

ABSTRACT

Computer users with very long computations run the risk of losing work because of machine failures. Such losses can often be reduced by scheduling saves on secure storage devices of work successfully done. In the model studied here, the user leaves the computation unattended for extended periods of time, after which he or she returns to check whether a machine failure occurs. When a check reveals a failure, the user resets the computation so that it resumes from the point of the last successful save.

Saves are themselves time consuming, so that any strategy for scheduling saves must strike a balance between the computing time lost during saves and the computing time that is occasionally lost, because of a failure since the last successful save.

For a given time to the next check and given constant save times, this paper computes schedules that maximize the expected amount of work successfully done before the next check, under the uniform and exponential failure laws. Explicit formulas are obtained for the uniform law. A recurrence leads to routine numerical calculations for the more difficult system with an exponential failure law.