# Analysis and Algorithms for Restart


Aad P. A. van Moorsel and Katinka Wolter

Aad P. A. van Moorsel
aadvanmoorsel.com
Goßlerstraße 11, 12161 Berlin,
Germany
conferences@aadvanmoorsel.com

Katinka Wolter
Humboldt-Universität Berlin
Institut für Informatik
Unter den Linden 6, 10099 Berlin,
Germany
wolter@informatik.hu-berlin.de



**Abstract**

We analyze and optimize the completion time for a class of jobs whose conditional completion time is not always decreasing with the time invested in the job. For such jobs, restarts may speed up the completion. Examples of such jobs include download of web pages, randomized algorithms, distributed queries and jobs subject to network or other failures. This paper derives computationally attractive expressions for the moments of the completion time of jobs under restarts and provides algorithms that optimize the restart policy. We also identify characteristics of optimal restart times as well as of distributions amenable to restarts.


## 1 Introduction

This work finds its motivation in a very simple problem that every Internet user experiences (and solves), often many times a day: when do I click my browser's reload button if a web page takes too long to download? The trade-off is between waiting a little more to see if the page still comes, or terminating the attempt and try again by clicking the refresh button.

It turns out that downloading a web page is an example of a job which benefits from being retried.[1] Mathematically, for such jobs the completion time conditioned on time elapsed can not be monotonically decreasing, as we will make precise. We will see that this situation arises naturally and increasingly often for Internet applications and for certain scientific jobs, as well as when one considers the failure behavior experienced by jobs.

To analyze and optimize the time at which to restart the job, we use a simple model that lends itself for elegant analysis. The model's core assumptions are that (1) successive tries are statistically independent and identically distributed, and (2) new tries abort previous tries. It should be noted that both above assumptions have been found realistic for the described use case of downloading web pages [13]. For this model we obtain the following results:

[1] We will use retry, restart, reload, refresh and other 're' terms indiscriminately, as suitable and natural for the application at hand.

- An iterative scheme to compute moments of the download time from its lower moments (Algorithm 1).
- Simple upper and lower bounds through geometric distributions (Equation (9)).
- A relation between the hazard rate and the optimal restart time (Theorem 2).
- A demonstration that the cusp point (which minimizes *both* 'reward' and 'risk') identified in [13] does not generally exist (Section 3.1).
- An algorithm that goes backward in time to compute all moments for the case of a finite number of restarts (Algorithm 2).
- An optimization algorithm for the expected completion time with finite number of restarts (Algorithm 3).
- A condition on completion time distributions to be amenable to restart, and a monotonicity relation for the mean completion time as function of the number of restarts (Section 5).

We first discuss in Section 2 the appearance of restarts in systems as well as its relation to other techniques (such as rejuvenation and preventive maintenance), and review related modeling literature. Then we discuss optimization of moments for unbounded number of restarts (Section 3) and finite number of restarts (Section 4), respectively. A discussion about the characteristics of distributions amenable to restarts is given in Sections 2.1 and 5.

# 2 Restart and Its Appearance in Systems

## 2.1 When Does Restart Work?

What characteristics do jobs have that benefit from restarts? In general terms, the completion time when starting new must be less than the completion time when not restarting. We can formalize this by letting the random variable $T$ denote the completion time of a job. Assume we are interested in the mean completion time. Under the assumption of independent identically distributed completion time of successive tries, one would restart at time $\tau$ when:

$$E[T] < E[T - \tau | T > \tau]. \tag{1}$$

The above intuitive reasoning can be made precise, and indeed turns out to be correct. Even stronger, in Section 5 we will show that (1) is a necessary and sufficient condition for *any* number of restarts to be useful as well, a result that is not necessarily obvious at first hand. The question then becomes, what distributions fulfill requirement (1), for at least one value of $\tau$. First, distributions with heavy tails have the required behavior. For such distributions, the tail decreases in polynomial pace, leaving considerable probability mass at high values

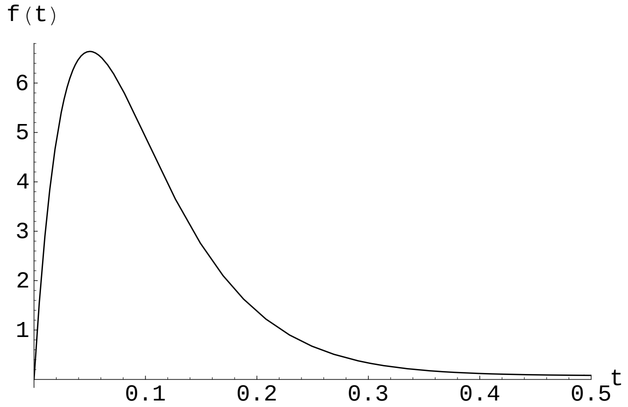


Figure 1: The probability density function of the mixed hyper/hypo-exponential distribution. Optimal restart times are 0.25 for single restart and 0.19 for unbounded number of restarts.

of $T$. Heavy-tailed and similar distributions commonly arise when studying Internet applications, see for instance [11]. However, also distributions with exponentially decaying tails demonstrate the required behavior quite often. Considering the three prototypical cases of exponentially decaying distributions [10], we see the following: for hyper-exponential distributions, condition (1) is *always* true (that is, for any $\tau$), for hypo-exponential distributions including the Erlang distribution (1) is *never* true, and for the exponential distribution (1) becomes an equality, implying restarts do not help, but also do not hurt.

The hyper-exponential distribution, which thus always performs better with restarts, is a distribution of a particular type that seems to be typical for restarts to succeed [16]. These distributions take values from different random variables with different probabilities, that is, with probability $p_1$ it is distributed as $X_1$, with probability $p_2$ distributed as $X_2$, etc.[2] It then is useful to perform a restart when it gets more likely one drew one of the slower distributions, since then a restart provides a chance to draw one of the faster distributions instead. Take for example the mixed hyper/hypo-exponential distribution we will use as example throughout this paper (see Appendix A for the precise mathematical characterization). It draws with probability $p = 0.9$ from an Erlang distribution with two phases and mean 0.1 and with probability 0.1 from an Erlang distribution with two phases and mean 1.0. Figure 1 shows the density of the mixed hyper/hypo-exponential distribution. It turns out that for a single restart, the optimal restart time is about 0.25, while for unbounded number of repeated restarts, the optimal restart time is about 0.19. Both values are indeed not too far above the mean 0.1 of the first Erlang distribution. The expected completion time decreases for a single restart from 0.190 to 0.136, and for unbounded restarts to 0.127, see also Figure 5.

[2]The discussion in [5] about unimodal distributions misses the point, providing a distribution as counter-example for multi-modality, which can be constructed easily as the sum of two random variables. Also note that the example in the introduction of [5] contains mistakes, and that some other analysis in that paper can be better approached as discussed in this paper.

## 2.2 Restart in Systems

The literature on restart in a strict sense is relatively young, about a decade. The first application we know of is scientific computations with random seeds, termed Las Vegas Algorithms [1, 12]. Using such algorithms, one sometimes is unlucky and runs into computations that take long to produce results. A systematic restart policy then often produces results faster.

A follow-up application is that of distributed queries using search algorithms that have a random aspect. This has recently been studied in, e.g., [16], and on-line algorithms have been derived to set the restart time if dependencies between successive tries can be exploited. Note the difference with the black-box restart approach we follow in this paper, in which the only information a restart policy can exploit is the time the current job (query) has been running.

The third area in which restart has been applied is that of web agents [5, 13]. Internet agents carry out varying tasks, using possibly randomized algorithms, over networks with failures and unpredictable delays, and it may therefore be smart to interrupt and restart an agent's job when a task takes too long to complete. The Internet application we opened this paper with (clicking the browser's refresh button) deals with restart as well, and is related to a time-out mechanism in TCP. This has been addressed to some extent in [17], although that paper leaves the interrelation with TCP unexplained. In [11] it is discussed in detail that clicking the reload button 'overrules' the TCP retransmission timer, potentially improving the overall download time.

In a more general sense, however, restart has been around in computing systems since their inception. Time-out schemes that retry an attempt once a threshold has been reached, can be seen as restarts. The above-mentioned TCP retransmission timer is but one example. Results from this paper may therefore be useful for the general issue of setting time-out values, but it should be noted that the modeling assumptions in this paper (independence of subsequent attempts and abortion of the preceding request at retry), although reasonable for Internet tasks [13], may not apply to every time-out problem.

## 2.3 Related Work in Modeling and Analysis

The analysis of restart touches on many areas, from portfolio theory in economics [13], to typical computer science issues such as optimization of rejuvenation and checkpointing policies [4], and mathematical foundations of decision and control theory [12]. There are too many connections to cover them all, but it is clear from the various mistakes or incorrect claims in recent papers on restart that the area needs a derivation of the main results, as provided in this paper.

Of particular interest is the relation with rejuvenation. Rejuvenation is concerned about the 'aging' of a system (e.g., memory leaks), which slows down the processing of jobs. The solution is to halt the job, refresh the system, and then continue the job. Rejuvenation is therefore often analyzed in combination with checkpointing, and often explicitly models the aging aspect of the system. Although such approaches may not suit restart (which is not concerned about aging of the underlying system), one can abstract out this difference, and approach restart as a special case of rejuvenation, namely one in which the system is always rejuvenated back to the original state. Or, in other words, where rejuvenation relies on checkpoints of partial results of a job, to which it returns at a rejuvenation instance, restart has only one checkpoint, namely the begin state at time zero. It is possible that this special case is implicitly included in earlier rejuvenation analysis, but typically the completion time analysis for rejuvenation models a different level of system detail than we do [3, 4, 7].

Also important is the relationship between restart and preventive maintenance, since, like rejuvenation, restart can be considered a preventive maintenance approach. In particular, one can imagine a dual problem of the completion time problem studied in this paper. Namely, maximize the time to failure through preventive maintenance policies, instead of minimize the completion time through restart policies. Resulting schemes that optimize the timing of preventive maintenance are known as age replacement policies, and the policies discussed in this paper and in [12] are in fact age replacement policies. Interesting enough, it is not easy to find results on this dual model (we have only found one in [8], bounding the first moment of time to failure, see Section 3). In general, preventive maintenance is analyzed in terms of cost of preventive versus required maintenance, thus complicating the model, but this is necessary to overcome trivial optimal preventive maintenance solutions.

The checkpointing and rejuvenation models typically aim at obtaining (moments of) the completion time distribution, and we do the same in this paper. An excellent survey of the vast amount of literature related to completion time analysis for checkpointing can be found in [15]. The novelty of the work in this paper does not lie in deriving general expressions for completion time, but in obtaining computationally attractive expressions for a specific case relevant to restart, and exploit these expressions in algorithms that compute and optimize the moments of completion time for unbounded as well as finite number of restarts. Apart from the recent literature on restart [1, 12, 13], existing literature deals with more intricate models than this paper. The expressions derived in this paper are purposely less comprehensive, but come with efficient algorithms and certain specific insights.

## 3 Unbounded Number of Restarts

Let the random variable $T$ represent the completion time of a job without restarts, $f(t)$ its probability density function, and $F(t)$ its distribution. For convenience, but without loss of generality,[3] we assume that $F(t)$ is a continuous probability distribution function defined over the domain $[0, \infty)$, so that $F(t) > 0$ if $t > 0$. Assume $\tau$ is a restart time,[4] and the overhead associated with restarting is $c$ time units for each restart (we also refer to $c$ as the

[3]At the cost of heavier notation, and with a proper discussion for special cases, the results in this paper also apply to distributions defined over finite domains, as well as to defective distributions and distributions with jumps.

[4]At times we also refer to $\tau$ as the as restart *interval*.

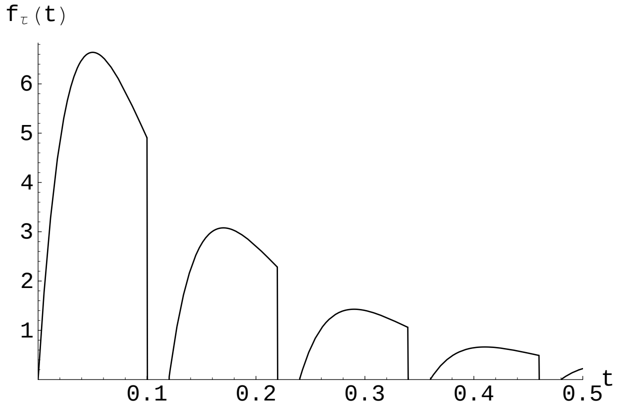

Figure 2: The probability density $f_\tau$ of completion time with unbounded number of restarts (based on hyper/hypo-exponentially distributed completion time without restarts, with restart time $\tau = 0.1$ and cost $c = 0.02$).

'cost' of a restart). We introduce the random variable $T_\tau$ to denote the completion time when an *unbounded* number of restarts is allowed. That is, a restart takes place periodically, every $\tau$ time units, until completion of the job. We write $f_\tau(t)$ and $F_\tau(t)$ for the density and distribution of $T_\tau$, and we are interested in the moments of $T_\tau$, and in the optimal value of the restart time $\tau$ itself.

To formally derive an expression for the moments, we first need an expression for the distribution and density of the completion time with restarts. As we mentioned in the introduction, we assume that a restart preempts the previous attempt, and that the completion times for consecutive attempts are statistically identical and independent. One can then reason about completion of a task in a restart interval as a Bernoulli trial with success probability $F(\tau)$. That is, the completion time with restarts relates to that without restarts as:

$$F_\tau(t) = \begin{cases} 1-(1-F(\tau))^k(1-F(t-k(\tau+c))) & \text{if } k(\tau+c) \le t < k(\tau+c)+\tau \\ 1-(1-F(\tau))^{k+1} & \text{if } k(\tau+c)+\tau \le t < (k+1)(\tau+c) \end{cases} \tag{2}$$

for $k = 0, 1, 2, \ldots$. For the density we obtain, also for any integer value $k = 0, 1, 2, \ldots$:

$$f_\tau(t) = \begin{cases} (1-F(\tau))^k f(t-k(\tau+c)) & \text{if } k(\tau+c) \le t < k(\tau+c)+\tau \\ 0 & \text{if } k(\tau+c)+\tau \le t < (k+1)(\tau+c) \end{cases} \tag{3}$$

It is worth visualizing the density of $T_\tau$, see Figure 2 for a mixed hyper/hypo-exponentially distributed $T$, with parameters as in Appendix A, restart time $\tau = 0.1$, and $c = 0.02$. In what follows, we also need the partial moments $M_n(\tau)$ at $\tau$ of the completion time, which is defined as:

$$M_n(\tau) = \int_0^\tau t^n f(t)dt = \int_0^\tau t^n f_\tau(t)dt. \tag{4}$$

The equality of partial moments of $T$ and $T_\tau$ follows from the fact that their respective densities are identical between 0 and $\tau$ (see (3) for $k = 0$).

In what follows we will exploit the structure of (3) to obtain computationally attractive expressions for the moments of $T_\tau$, and to gain further insight into optimal restart policies. We expected to find in the literature or text books the following rather straightforward derivation of the moments $E[T_\tau^n]$ of completion time $T_\tau$, but did not succeed. The literature provides more generic expressions for completion time (e.g., [15]), but these are computationally not very attractive. Hence, we give the derivation and recursive expressions here, although it is obvious that the authors of for instance [13] and [16] must have worked from some of the same equations.

**Theorem 1.** *The moments $E[T_\tau^n] = \int_0^\infty t^n f_\tau(t)dt$, $n = 1, 2, \ldots,$ of the completion time with unbounded number of restarts, restart interval length $\tau > 0$, and time $c$ consumed by a restart, can be expressed as:*

$$E[T_\tau^n] = \frac{M_n(\tau)}{F(\tau)} + \frac{1 - F(\tau)}{F(\tau)} \sum_{l=0}^{n-1} \binom{n}{l} (\tau + c)^{n-l} E[T_\tau^l], \tag{5}$$

*where $E[T_\tau^0] = 1$.*

*Proof.* The derivation is particularly elegant if one exploits the recursive structure of (3). First, by definition, we have:

$$\begin{aligned} E[T_\tau^n] &= \int_0^\infty t^n f_\tau(t)dt \\ = \int_0^\tau t^n f_\tau(t)dt + \int_{\tau+c}^\infty t^n f_\tau(t)dt &= M_n(\tau) + \int_{\tau+c}^\infty t^n f_\tau(t)dt. \end{aligned} \tag{6}$$

Then, we use that from (3) it follows that for any $t \geq 0$,

$$f_\tau(t + \tau + c) = (1 - F(\tau)) f_\tau(t),$$

and thus:

$$\int_{\tau+c}^\infty t^n f_\tau(t)dt = \int_0^\infty (t + \tau + c)^n f_\tau(t + \tau + c)dt = (1 - F(\tau)) \int_0^\infty (t + \tau + c)^n f_\tau(t)dt. \tag{7}$$

Combining (6) and (7) we obtain:

$$E[T_\tau^n] = M_n(\tau) + (1 - F(\tau)) \int_0^\infty (t + \tau + c)^n f_\tau(t)dt,$$

which we write out as:

$$E[T_\tau^n] = M_n(\tau) + (1 - F(\tau)) \int_0^\infty \sum_{l=0}^n \binom{n}{l} (\tau + c)^{n-l} t^l f_\tau(t)dt =$$
$$M_n(\tau) + (1 - F(\tau)) \sum_{l=0}^n \binom{n}{l} (\tau + c)^{n-l} E[T_\tau^l].$$

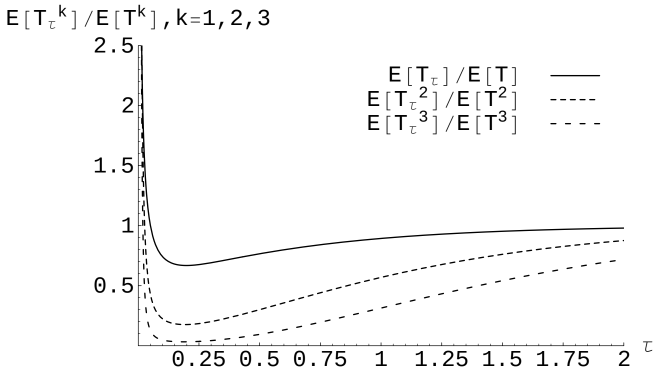


Figure 3: Restart time versus the normalized difference between unbounded restarts and no restarts, for first three moments.

One then solves this equation for $E[T_\tau^n]$, cancelling out the highest moment within the sum, to obtain:

$$E[T_\tau^n] = \frac{M_n(\tau)}{F(\tau)} + \frac{1-F(\tau)}{F(\tau)} \sum_{l=0}^{n-1} \binom{n}{l} (\tau+c)^{n-l} E[T_\tau^l].$$

$\square$

For example, the expected completion time is given by:

$$E[T_\tau] = \frac{M_1(\tau)}{F(\tau)} + \frac{1-F(\tau)}{F(\tau)}(\tau+c). \tag{8}$$

The expression for the variance can be found in [13]. The result for the first moment is indeed as it should be: (8) must account for the interval in which the task completes, as well as for the times the job fails to complete. The first term in (8) is the expected download time conditioned on success within a restart interval. The second term equals interval length $\tau + c$ times the expected value of a modified geometric distribution [10] with parameter $F(\tau)$, since, indeed, in every interval the probability of completion is $F(\tau)$.

Finally, note that by requiring that $\tau > 0$ in Theorem 1, the denominator $F(\tau)$ in (5) is positive, since we assumed continuous distributions defined over $[0, \infty)$. For $\tau \downarrow 0$ and $c = 0$, we can apply l'Hospital's rule, to see that $E[T_\tau] \to f^{-1}(0)$, which tends to infinity for our running example.

Equation (5) directly yields an algorithm to iteratively compute all moments up to some specified value $N$. We reformat it here as an algorithm for completeness of the presentation:

**Algorithm 1 (Computation of all Moments, Unbounded Restarts).**

Set $E[T_\tau^0] = 1$ for chosen $\tau > 0$;
For $n = 1$ to $N$ {
  compute $E[T_\tau^n] = \frac{M_n(\tau)}{F(\tau)} + \frac{1-F(\tau)}{F(\tau)} \sum_{l=0}^{n-1} \binom{n}{l} (\tau+c)^{n-l} E[T_\tau^l]$
}

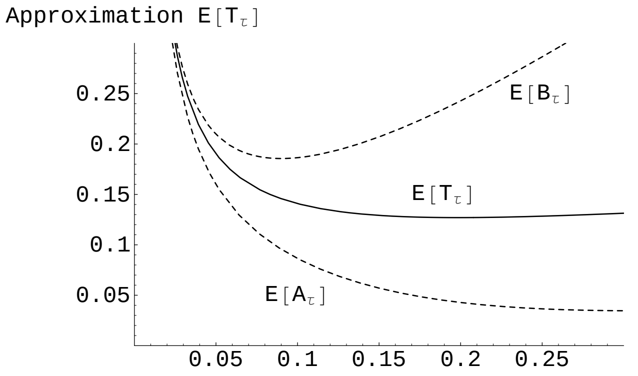


Figure 4: Approximation of expected completion time using geometric distributions.

Using this basic algorithm we obtained the results of Figure 3 through 7. Figure 3 provides the relative gain using restarts for the first three moments. Notice that the gain increases rather dramatically with the order of the moment. Also, notice the wide range of restart times which perform well, which suggests that rough estimates may often suffice to set a restart time. An important engineering rule is to not take the restart time too small, since for many realistic distributions, the completion time will then tend to infinity (the hyper-exponential distribution being one exception). In general, it may be safer to take the restart time too large than too small.

As a corollary of Theorem 1 we state a fundamental result, which was also observed by [6] for failure detectors.

**Corollary 1.** *Under unbounded restarts, the expectation (as well as higher moments) of the completion time $T_\tau$ with restart time $\tau > 0$ (for which $F(\tau) > 0$), is* always *finite, even if the moments of the original completion time are not.*

This is an important observation, stressing the value of restarts for situations in which there is a (strictly) positive probability that a task can fail (thus making the moments of completion time infinite).

**Geometric approximation.** The results obtained above also suggest bounds for the moments by using the (modified) geometric distribution (see also expression (4.2.12) in [8] for the dual result in terms of mean time between failures). To bound the first moment, one replaces the first term in (5), which refers to the interval in which the job completes, by its upper and lower bounds 0 and $\tau$, respectively. This can be generalized to all moments, using two discrete random variables $A_\tau$ and $B_\tau$, with

$$\begin{aligned} A_\tau &= k(\tau + c), \text{ with probability } (1 - F(\tau))^k F(\tau), \quad k = 0, 1, \ldots, \\ B_\tau &= k(\tau + c) + \tau, \text{ with probability } (1 - F(\tau))^k F(\tau), \quad k = 0, 1, \ldots. \end{aligned} \tag{9}$$

Since we know from (2) that

$$k(\tau + c) \le T_\tau \le k(\tau + c) + \tau, \text{ with probability } (1 - F(\tau))^k F(\tau), \quad k = 0, 1, \ldots,$$

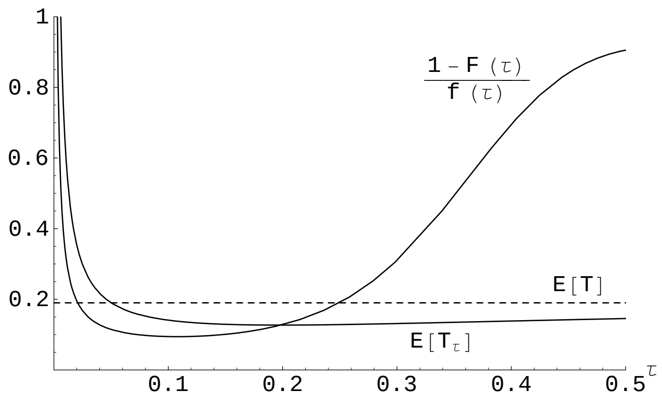


Figure 5: Extrema for the mean completion time are found at restart times $\tau$ for which the inverse hazard rate equals $E[T_\tau]$.

we have that $E[A_\tau^n] \leq E[T_\tau^n] \leq E[B_\tau^n]$, for $n = 1, 2, \ldots$. Note that $A_\tau$ has a modified geometric distribution [10] and that $B_\tau = A_\tau + \tau$. Figure 4 shows $E[T_\tau]$ as well as the bounds for the mixed hyper/hypo-exponential distribution. The bounds, whose summed error equals exactly $\tau$, are excellent approximations as long as the restart time $\tau$ is small relative to the mean completion time $E[T_\tau]$. For the area around the optimal restart time the bounds are not particularly tight. Nevertheless, the geometric approximation may prove very useful for determining a conservative restart time. For instance, for Figure 4, the optimal restart time (that is, the minimum of the curve) for the upper bound is 0.09, and for the lower bound 0.31, while the real optimum lies in between (namely at 0.20). Moreover, Figure 3 shows that for $\tau = 0.31$, the expected completion time is still close to optimal. Using 0.31 as a conservative restart time is also consistent with the above-mentioned engineering rule that it is better to restart too late than too early.

### 3.1 Optimal Restart Times for Unbounded Number of Restarts

We give an implicit relation for the optimal restart time $\tau^*$ for the first moment of $T_\tau$. These implicit expressions provide us with interesting insight in how the hazard rate of a distribution determines the optimal completion time under restarts. It also allows us to refute the claims on the existence of a cusp point in [13].

**Theorem 2.** *The optimal restart time $\tau^* > 0$ that minimizes the expected completion time $E[T_\tau]$ is such that:*

$$\frac{1 - F(\tau^*)}{f(\tau^*)} = E[T_{\tau^*}] + c. \qquad (10)$$

*That is, if $c = 0$, the inverse of the hazard rate at $\tau^*$ equals the expected completion time under unbounded restarts.*

*Proof.* To obtain this result, we equate to zero the derivative with respect to $\tau$ of $E[T_\tau] =$

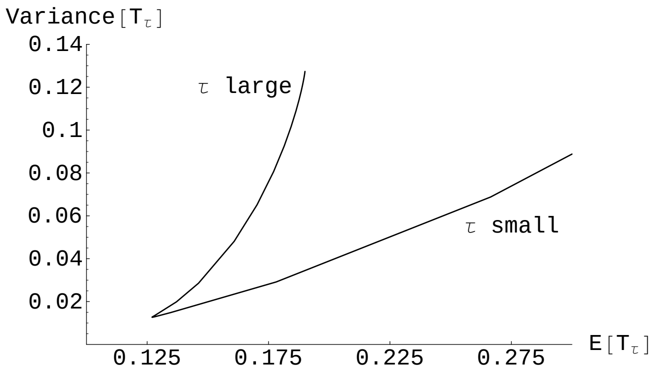


Figure 6: Mean and variance of the completion time, parameterized by restart time $\tau$, as Figure 2 of [13].

$\frac{M_1(\tau)}{F(\tau)} + \frac{1-F(\tau)}{F(\tau)}(\tau + c)$ (the base relation (8)):

$$\frac{d}{d\tau}E[T_\tau] = 0 \iff \frac{\tau f(\tau)F(\tau) - f(\tau)M_1(\tau)}{F^2(\tau)} + \frac{1-F(\tau)}{F(\tau)} - \frac{f(\tau)(\tau + c)}{F^2(\tau)} = 0,$$

which after some manipulation, in which we again apply (8), results in:

$$\frac{d}{d\tau}E[T_\tau] = 0 \iff \frac{1-F(\tau)}{f(\tau)} = E[T_\tau] + c.$$

□

It is important to realize that (10) may hold for many restart values, including $\tau \to \infty$, and not only holds for the global optimum, but also local minima and maxima. For instance, in Figure 5, the inverse hazard rate indeed crosses $E[T_\tau]$ in its minimum, which gives $\tau^* \approx 0.2$, but also meets at point 0, where the maximum of the completion time under restarts is reached. As we remarked in Section 2, there exists a dual problem formulation in terms of maximizing the time to failure, and thus, Theorem 2 not only holds for restart times that minimize completion time, but also for preventive maintenance times that maximize the mean time to failure (which is meaningful only for certain distributions and without cost (i.e., $c = 0$)).

**Cusp Point.** In [13] the authors point to the existence of a 'cusp point', in which both the expected completion time and its variance are minimized. In terms of [13] 'reward' as well as 'risk' are then optimized jointly. Figure 2 in [13] suggests that such a cusp point exists; we have redone this in Figure 6 for our example, and indeed two curves seem to come together in a cusp point, where it reaches the minimum for both mean and variance. (Note that the curve is parameterized over $\tau$, plotting mean versus variance of the completion time for a range of restart times.) However, it turns out that the restart time that minimizes the higher moments of the completion time is typically not identical to $\tau^*$. Since it is easy to see that if the second moment is minimized by a different restart time than $\tau^*$, the variance is also not

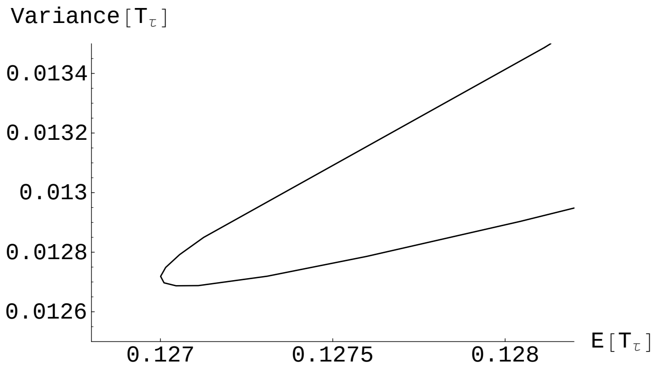


Figure 7: As Figure 6, zoomed in at 'cusp point.'

minimized in $\tau^*$, it follows that the cusp point identified in [13] does not exist–at least, not in general.

One way to show that the cusp point does not exist is to derive for higher moments the counterpart to Theorem 2, so that a relation is established between the hazard rate and the optimal restart time for higher moments. Then it is possible to show that if $\tau^*$ is the restart time that minimizes the completion time for $E[T_\tau], \ldots, E[T_\tau^N]$, then $E[T_{\tau^*}^n] = n!(\frac{1-F(\tau^*)}{f(\tau^*)})^n$, for $n = 1, \ldots, N$. We can certainly construct probability distributions with partial moments $M_n(\tau)$ such that this relation holds when filling in (5), but in general the relation will not hold true. Instead of providing the proof for this negative result, we demonstrate numerically that the cusp point does not exist for our running example. Figure 7 zooms in at the 'cusp point' of Figure 6 and demonstrates that there is no point that minimizes the curve with respect to both the x and y-axis (mean and variance of completion time). In particular, for our example, the minimum expected completion time is for restart time $\tau^* = 0.198$, for the second moment the minimum is for $\tau = 0.192$, and for the variance the minimum is reached at $\tau = 0.188$.

## 4 Finite Number of Restarts

There may be cases in which one is interested in a *finite* number of restarts. For example, in our mixed hyper/hypo-exponential example, too low a restart time is very detrimental for the completion time if there is no bound on the number of restarts. Although this need not generally be the case (the hyper-exponential distribution is a counter example), for many distributions it may be wise to limit the number of restarts, or increase the period between restarts with the restart count. This leads to a situation with finite and non-identical restart intervals, for which we derive an algorithm to compute all moments. Perhaps one would expect that restarts should take place with fixed-length intervals between them, but we will see that this is often not optimal. We provide an algorithm to compute the optimal restart times for

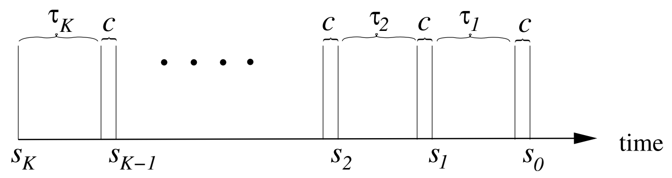


Figure 8: Labeling restarts backward, total of $K$ restarts.

the first moment. We explain why the first moment is considerably simpler to optimize than higher moments, for which we do not know an algorithm with proven convergence.

For our discussion it is convenient to label the restarts 'backward.' Figure 8 shows this. We assume the total number of restarts is $K$, and the restart intervals have length $\tau_K, \tau_{K-1}, \dots, \tau_1$, respectively. The $k$-th interval starts at time $s_k$. So, we get $s_K = 0$, $s_{K-1} = \tau_K + c$, $s_{K-2} = \tau_K + c + \tau_{K-1} + c$, etc., until $s_0 = \sum_{k=1}^K \tau_k + Kc$. The completion time with $K$ restarts is represented by the random variable $T_{\tau_K,\dots,\tau_1}$. The completion time probability distribution $F_{\tau_K,\dots,\tau_1}$ and density $f_{\tau_K,\dots,\tau_1}$ for the scenario with $K$ restarts can be derived in the same way as (2) and (3). If we introduce $\tau_0 = \infty$ for notational purposes we can define the density and distribution function piece-wise over every restart interval:

$$F_{\tau_K,\dots,\tau_1}(t) = \begin{cases} 1 - \prod_{i=k+1}^K (1 - F(\tau_i))(1 - F(t - s_k)) & \text{if } s_k \le t < s_k + \tau, k = 0, 1, \dots, K \\ 1 - \prod_{i=k+1}^K (1 - F(\tau_i)) & \text{if } s_k + \tau_k \le t < s_{k-1}, k = 1, 2, \dots, K \end{cases}$$

$$f_{\tau_K,\dots,\tau_1}(t) = \begin{cases} \prod_{i=k+1}^K (1 - F(\tau_i)) f(t - s_k) & \text{if } s_k \le t < s_k + \tau, k = 0, 1, \dots, K \\ 0 & \text{if } s_k + \tau_k \le t < s_{k-1}, k = 1, 2, \dots, K \end{cases} \tag{11}$$

As for the unbounded case, we express the moments in the following theorem in a manner convenient for computational purposes. This time, we express the moments of the completion time with $K$ restarts in that with one restart less.

**Theorem 3.** *The moments $E[T^n_{\tau_K,\dots,\tau_1}] = \int_0^\infty t^n f_{\tau_K,\dots,\tau_1}(t)dt$, $n = 1, 2, \dots$, of the completion time with $K$ restarts, restart interval lengths $\tau_K, \tau_{K-1}, \dots, \tau_1$, and time $c$ consumed by each restart, can be expressed as:*

$$E[T^n_{\tau_K,\dots,\tau_1}] = M_n(\tau_K) + (1 - F(\tau_K)) \sum_{l=0}^{n} \binom{n}{l} (\tau_K + c)^{n-l} E[T^l_{\tau_{K-1},\dots,\tau_1}], \tag{12}$$

*where $E[T^0_{\tau_{K-1},\dots,\tau_1}] = 1$.*

*Proof.* The derivation is similar to that of Theorem 1. Start from the fact that from (11) it follows that for $t \ge 0$:

$$f_{\tau_K,\dots,\tau_1}(s_{K-1} + t) = (1 - F(\tau_K)) f_{\tau_{K-1},\dots,\tau_1}(t),$$

and then follow the same derivation as in Theorem 1. Only the last step, in which $E[T^n_\tau]$ is solved, has no counterpart in the current proof. □

As an illustration, we get for the first moment:

$$E[T_{\tau_K,\ldots,\tau_1}] = M_1(\tau_K) + (1 - F(\tau_K))(\tau_K + c + E[T_{\tau_{K-1},\ldots,\tau_1}]). \tag{13}$$

The above theorem implies that if $\tau_K, \ldots, \tau_1$ are known beforehand, one can iteratively compute $E[T^N_{\tau_K,\ldots,\tau_1}]$ for any $N > 0$ by going backward in time. That is, starting from the moments $E[T^n_{\tau_1}], n = 1, \ldots, N$, one obtains $E[T^n_{\tau_2,\tau_1}]$, until $E[T^N_{\tau_K,\ldots,\tau_1}]$. The algorithm thus goes as follows:

**Algorithm 2 (Backward Algorithm, first $N$ Moments, $K$ Restarts).**

> `For` $n = 0$ `to` $N$
>
>     `Set` $E[T^n_{\tau_0,\ldots,\tau_1}] = E[T^n]$`;`
>
> `For` $k = 1$ `to` $K$ `{`
>
>     `For` $n = 0$ `to` $N$ `{`
>
>         $E[T^n_{\tau_k,\ldots,\tau_1}] = M_n(\tau_k) + (1 - F(\tau_k)) \sum_{l=0}^{n} \binom{n}{l} (\tau_k + c)^{n-l} E[T^l_{\tau_{k-1},\ldots,\tau_1}]$`;`
>
>     `}`
>
> `}`

A nice feature of the backward algorithm is that it computes moments of subsets $\{\tau_k, \ldots, \tau_1\}$ along the way. One should be careful to interpret those correctly: the moments $E[T^l_{\tau_k,\ldots,\tau_1}]$ are for $s_k = 0$, that is, for the case that completion time starts counting at the $k$-th interval, not before. This feature of the backward algorithm turns out to be its pitfall as well, if we try to use the algorithm for optimization purposes. The issue is that for higher moments optimization of the $k$-th restart time depends on *all* other restart times. Only for the first moment, the optimal value of the $k$-th restart is insensitive to what happens *before* the $k$-th restart (that is, to the restarts we labeled $k + 1, , \ldots, K$). As a consequence, for the first moment, we can optimize the restart intervals concurrently with computing moments using the backward algorithm. For higher moments, it is an open issue if efficient convergent algorithms can be found. The above argument can be made more precise by the theorem we give in Appendix B. This theorem is necessary to show that the following algorithm finds the restart times that optimize the first moment of completion time. The algorithm only requires a single run of $K$ steps; it works backward in time and finds optimal restart times $\tau_1^*, \ldots, \tau_K^*$, in that order.

**Algorithm 3 (Backward Optimization Algorithm, First Moment, $K$ Restarts).**

> `Set` $E[T_{\tau_0^*,\ldots,\tau_1^*}] = E[T]$`;`
>
> `For` $k = 1$ `to` $K$ `{`
>
>     `compute` $\tau_k^*$`, the value of` $\tau_k > 0$ `that`
>
>     `minimizes` $M_1(\tau_k) + (1 - F(\tau_k))(\tau_k + c + E[T_{\tau_{k-1}^*,\ldots,\tau_1^*}])$`;`
>
>     `Set` $E[T_{\tau_k^*,\ldots,\tau_1^*}] = M_1(\tau_k^*) + (1 - F(\tau_k^*))(\tau_k^* + c + E[T_{\tau_{k-1}^*,\ldots,\tau_1^*}])$`;`
>
> `}`

| interval index $k$ | optimal length $\tau_k$ of $k$-th interval |
|---|---|
| 1 | 0.249 |
| 2 | 0.209 |
| 3 | 0.200 |
| 4 | 0.199 |
| 5 | 0.1983 |
| 6 | 0.198265 |
| 7 | 0.198256 |
| 8 | 0.198254 |
| 9 | 0.198254 |
| 10 | 0.198254 |
| unbounded | 0.198254 |

Table 1: Optimal restart intervals for finite and unbounded number of restarts.

As an illustration, we apply the backward optimization algorithm to our mixed hyper/hypo-exponential distribution, with parameters as given in Appendix A, to obtain the optimal restart times given in Table 1. Note that the values in the table imply that, for instance, for $k = 2$, the two restart times will be after 0.209 and $0.209 + 0.249 = 0.458$ time units, respectively. As one sees from Table 1, the restart intervals have different lengths, longer if it is closer to the last restart. Furthermore, the more restarts still follow, the closer the interval length is to the optimum for unbounded restarts, which is 0.198254. This is as expected.

## 5 Characteristics of Probability Distributions and Optimal Restart Policies

The backward algorithm provides us with machinery to further characterize necessary and sufficient conditions for a random variable $T$ to benefit from restarts. We will see that for the mean completion time, the intuitive condition we derived in Section 2 for a single restart is necessary and sufficient for any number of restarts to be useful. We will also show that if a single restart improves the mean completion time, multiple restarts perform even better, and unbounded restarts performs best.

We use the random variable $T_{\tau^K}$ to denote the completion time under $K$ restarts at times $\tau + c, 2(\tau + c), \ldots, K(\tau + c)$, and for technical reasons also use the notation $T_{\tau^0} = T$, for the case without restarts.

**Theorem 4.** *The mean completion time under zero ($E[T]$), $K \geq 1$ ($E[T_{\tau^K}]$) and unbounded restarts ($E[T_\tau]$) interrelate as follows:*

$$E[T_\tau] < \ldots < E[T_{\tau^{K+1}}] < E[T_{\tau^K}] < \ldots < E[T] \iff E[T_\tau] < E[T], \tag{14}$$

*and*

$$E[T+c] < E[T-\tau|T>\tau] \iff E[T_\tau] < E[T]. \tag{15}$$

*Proof.* The first result follows from the backward algorithm, which uses (13) for the first moment. (Note that $T_{\tau^K}$ is in fact identical to $T_{\tau_K,\ldots,\tau_1}$ with $\tau_K = \ldots = \tau_1 = \tau$.) If we introduce

$$C_\tau = M_1(\tau) + (1-F(\tau))(\tau+c),$$

then from (13) we obtain that for any $K \geq 0$ (remembering that $E[T_{\tau^0}] = E[T]$)

$$E[T_{\tau^{K+1}}] = C_\tau + (1-F(\tau))E[T_{\tau^K}], \tag{16}$$

and from (8) that for unbounded restarts:

$$E[T_\tau] = \frac{C_\tau}{F(\tau)}. \tag{17}$$

Combining (16) and (17) it is easy to show that

$$E[T_{\tau^{K+1}}] < E[T_{\tau^K}] \iff \frac{C_\tau}{F(\tau)} < E[T_{\tau^K}] \iff E[T_\tau] < E[T_{\tau^K}].$$

Since this holds for any $K \geq 0$, it follows that $E[T_\tau] < E[T]$, proving (14).

To show that (15) holds, we derive:

$$E[T-\tau|T>\tau] = \frac{\int_\tau^\infty (t-\tau)f(t)dt}{1-F(\tau)}$$
$$= \frac{\int_\tau^\infty tf(t)dt - \tau(1-F(\tau))}{1-F(\tau)} = \frac{E[T]-M_1(\tau)}{1-F(\tau)} - \tau.$$

Then (15) follows using (8):

$$E[T]+c < E[T-\tau|T>\tau] \iff E[T]+c < \frac{E[T]-M_1(\tau)}{1-F(\tau)} - \tau$$
$$\iff (1-F(\tau))E[T] < E[T] - M_1(\tau) - (1-F(\tau))(\tau+c)$$
$$\iff E[T] > \frac{M_1(\tau)}{F(\tau)} + \frac{1-F(\tau)}{F(\tau)}(\tau+c) \iff E[T_\tau] < E[T].$$

□

Note that the above proof shows that the backward algorithm is a fixed-point iteration of the form

$$x_{K+1} = C_\tau + (1-F(\tau))x_K,$$

with initial guess $x_0 = E[T]$ and fixed-point solution $E[T_\tau]$. The consequence of the first result of Theorem 4 is depicted in Figure 9. The straight line is the expected completion time $E[T]$ without restarts, and the curve with the highest maxima and lowest minima is $E[T_\tau]$ for

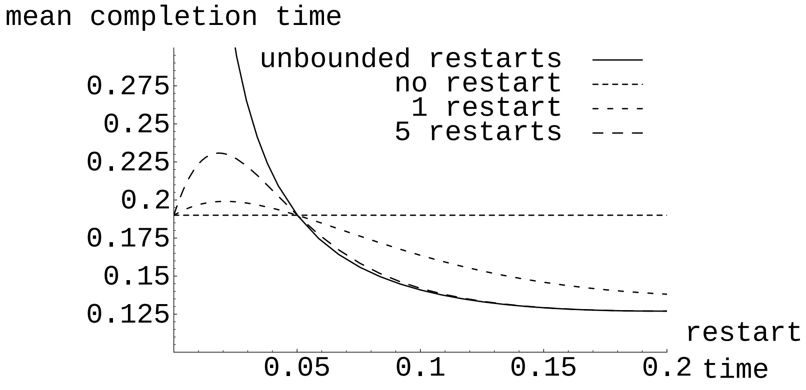


Figure 9: Expected completion time for varying number of restarts.

unbounded restarts. Because of Theorem 4, all curves improve the completion time over the same range of restart times, and the more restarts, the better. Similarly, when the completion time increases with restarts, fewer restarts are less detrimental for the completion time.

Another consequence of Theorem 4 is that if the completion time distribution is such that there exist restart times that improve expected completion time, as well as restart times that increase expected completion time, then there must also exist at least one point in which *all* curves cross, that is, it is immaterial if and how many restarts one executes. Figure 9 shows this, at $\tau \approx 0.05$.

We finally note that the results from Theorem 4 do not extend to higher moments. Figure 10 shows that there exist values for which one or two restarts improve the second moment, but unbounded restarts do not. There also is no point $\tau'$ in which any number of restarts provides the same completion time. The reason higher moments are more problematic is the same as why the backward algorithm does not work for higher moments: whether a restart time improves the completion time's higher moments is sensitive to the starting point of a restart interval. Repeated restarts may therefore not always keep improving the completion time's higher moments. Nevertheless, one can follow a similar conditional argument as in Section 2 to obtain the condition under which restart is beneficial for higher moments, namely $E[(T+c)^n] < E[(T-\tau)^n | T > \tau]$. However, from this we can not conclude anything about the success for multiple or unbounded number of restarts.

## 6 Conclusion

Restart has the potential to speed up tasks such as Internet agent interactions and randomized scientific computations, and has recently come under attention. From studying the literature it is clear that a solid and/or accessible analysis of the restart mechanism has been sorely

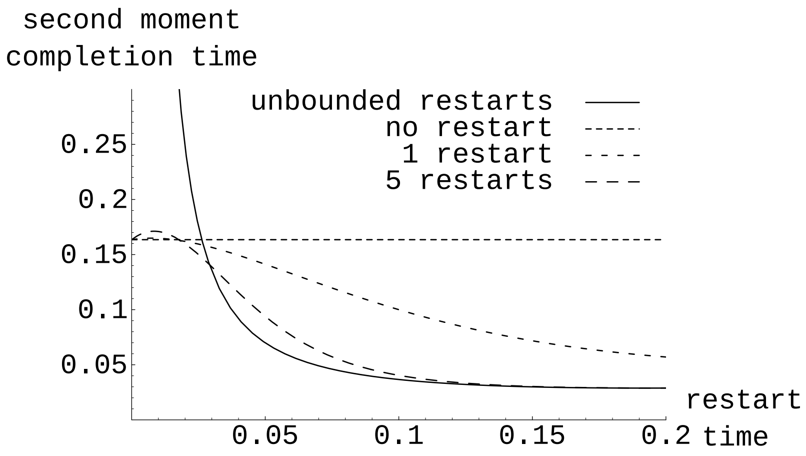


Figure 10: Second moment of completion time for varying number of restarts.

missed. This paper tries to fill this gap by providing expressions and associated algorithms to compute the moments of completion time under restart. It also provides optimization algorithms for the restart policy, as well as various engineering rules for setting restart times. In addition, we identified characteristics of distributions amenable to restarts and of optimal restart policies.

## References


[1] H. Alt, L. Guibas, K. Mehlhorn, R. Karp and A. Wigderson, A Method for Obtaining Randomized Algorithms with Small Tail Probabilities, *Algorithmica,* Vol. 16, Nr. 4/5, pp. 543–547, 1996.

[2] S. Amari and W. Fulton, "Bounds on Optimal Replacement Time of Age Replacement Policy," http://www.relexsoftware.com/resources/papers/Age-Replacement-Bounds.pdf.

[3] A. Bobbio, A. Puliafito, M. Telek and K. Trivedi, "Recent Developments in Non-Markovian Stochastic Petri Nets," *Journal of Systems Circuits and Computers,* 8(1):119-158, 1998.

[4] A. Bobbio, S. Garg, M. Gribaudo, A. Horvath, M. Sereno and M. Telek, "Modeling Software Systems with Rejuvenation, Restoration and Checkpointing through Fluid Stochastic Petri Nets," in *Proceedings of PNPM '99,* Zaragoza, Spain, pp. 82-91, IEEE CS Press, Sept 1999.

[5] P. Chalasani, S. Jha, O. Shehory and K. Sycara, "Query Restart Strategies for Web Agents," in *Proceedings of Agents98,* AAAI Press, 1998.

[6] W. Chen, S. Toueg and M. K. Aguilera, "On the Quality of Service of Failure Detectors," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2000),* pp. 191–200, 2000.

[7] S. Garg, Y. Huang, C. Kintala and K. S. Trivedi, "Minimizing Completion Time of a Program by Checkpointing and Rejuvenation," *ACM SIGMETRICS* 1996, May 1996.

[8] I. Gertsbakh, *Reliability Theory, With Applications to Preventive Maintenance,* Springer Verlag, Berlin, 2000.

[9] D. Gude and K. Schmidt, "Preventive Maintenance of Advanced Manufacturing Systems: A Laboratory Experiment and its Implications for the Human Centered Approach", in *International Journal of Human Factors in Manufacturing,* 3, 335-350, 1993.

[10] B. Haverkort, *Performance of Computer Communication Systems: A Model-Based Approach,* John Wiley, Chichester, 1999.

[11] B. Krishnamurthy and J. Rexford, *Web Protocols and Practice,* Addison Wesley, 2001.

[12] M. Luby, A. Sinclair and D. Zuckerman, "Optimal Speedup of Las Vegas Algorithms," *Israel Symposium on Theory of Computing Systems,* pp. 128–133, 1993.

[13] S. M. Maurer and B. A. Huberman, "Restart strategies and Internet congestion," in *Journal of Economic Dynamics and Control,* vol. 25, pp. 641–654, 2001.

[14] A. van Moorsel and K. Wolter, "Optimization of Failure Detection Retry Times," in *Performability workshop*, Monticello, IL, Oct. 2003.

[15] V. F. Nicola, "Checkpointing and the Modeling of Program Execution Time," in *Trends in Software 3: Software Fault Tolerance*, M. R. Lyu (Ed.), Wiley & Sons, Chichester, UK, Chapter 7, pp. 167–188, 1995.

[16] Y. Ruan, E. Horvitz and H. Kautz, "Restart Policies with Dependence among Runs: A Dynamic Programming Approach," in *Proceedings of the Eight International Conference on Principles and Practice of Constraint Programming,* Ithaca, NY, Sept. 2002.

[17] M. Schroeder and L. Buro, "Does the Restart Method Work? Preliminary Results on Efficiency Improvements for Interactions of Web-Agents," in T. Wagner and O. Rana, editors, *Proceedings of the Workshop on Infrastructure for Agents, MAS, and Scalable MAS at the Conference Autonomous Agents 2001,* Springer Verlag, Montreal, Canada, 2001.

## Appendix A: Running Example

As a running example we use a mixed hyper/hypo-exponential distribution, which, depending on the chosen parameters can be made to be never, always or sometimes amenable to restart. The mixed hyper/hypo-exponential random variable takes with probability $p_i$ a value from an Erlang distribution with $N_i$ phases and parameter $\lambda_i > 0$, $i = 1, 2, \ldots, M$, and $\sum_{i=1}^{M} p_i = 1$. So, we get for the distribution $F_M$ and density $f_M$ (refer, for instance, to [10]):

$$F_M(t) = \sum_{i=1}^{M} p_i (1 - \sum_{j=0}^{N_i - 1} \frac{(\lambda_i t)^j}{j!} e^{-\lambda_i t}),$$

$$f_M(t) = \sum_{i=1}^{M} p_i \lambda_i^{N_i} \frac{t^{N_i - 1}}{(N_i - 1)!} e^{-\lambda_i t}.$$

In the paper we apply the following parameter values: $M = 2$, with $p_1 = 0.9$, $p_2 = 0.1$; $N_1 = N_2 = 2$, with $\lambda_1 = 20$, $\lambda_2 = 2$; and $c = 0$, unless otherwise stated. This mixed distribution has neither monotonically increasing or decreasing hazard rate, see Figure 5, which implies that it depends on the chosen restart time whether restart improves completion time.

## Appendix B: Correctness of the Backward Optimization Algorithm

The following theorem demonstrates that if the backward optimization algorithm (Algorithm 3) computes a restart time $\tau_k^*$ it is optimal, irrespective of the values of $\tau_{k+1}$ to $\tau_K$ computed later in the algorithm.

**Theorem 5.** *Assume a restart strategy with $K$ restarts at times $\tau_K, \tau_{K-1}, \ldots, \tau_1$. For the first moment $E[T_{\tau_K,\ldots,\tau_1}]$, for any value of $k$ between 1 and $K$, the restart time $\tau_k^*$, that minimizes $E[T_{\tau_k,\ldots,\tau_1}]$, also minimizes $E[T_{\tau_K,\ldots,\tau_1}]$ (and is thus independent of $\tau_K, \tau_{K-1}, \ldots, \tau_{k+1}$). For higher moments this is not generally true.*

*Proof.* $E[T_{\tau_K,\ldots,\tau_1}]$ and $E[T_{\tau_k,\ldots,\tau_1}]$ relate as follows:

$$\begin{aligned} E[T_{\tau_K,\ldots,\tau_1}] &= \\ \int_0^{s_k} t f_{\tau_K,\ldots,\tau_1}(t)dt + \prod_{i=k+1}^{K}(1-F(\tau_i)) \int_{s_k}^{\infty} t f_{\tau_k,\ldots,\tau_1}(t-s_k)dt &= \\ \int_0^{s_k} t f_{\tau_K,\ldots,\tau_1}(t)dt + \prod_{i=k+1}^{K}(1-F(\tau_i))E[s_k + T_{\tau_k,\ldots,\tau_1}]. & \end{aligned} \tag{18}$$

Since $E[s_k + T_{\tau_k,\ldots,\tau_1}] = s_k + E[T_{\tau_k,\ldots,\tau_1}]$, and since the integral between 0 and $s_k$ is independent of $\tau_k$, we see immediately that the value of $s_k$ is immaterial for the optimal restart time $\tau_k^*$:

$$\frac{d}{d\tau_k}E[T_{\tau_K,\ldots,\tau_1}] = 0 \iff \frac{d}{d\tau_k}E[s_k + T_{\tau_k,\ldots,\tau_1}] = 0 \iff \frac{d}{d\tau_k}E[T_{\tau_k,\ldots,\tau_1}] = 0.$$

Hence, $\tau_k^*$ is an extreme for $E[T_{\tau_K,\ldots,\tau_1}]$ if and only if it is an extreme for $E[T_{\tau_k,\ldots,\tau_1}]$. Even stronger, by inspection of (18) it follows that if $\tau_k^*$ is an element of the global minimum $\{\tau_k^*, \ldots, \tau_1^*\}$ for $E[T_{\tau_k,\ldots,\tau_1}]$ (with $k$ restarts) then no other value of $\tau_k$ can result in a better expected overall completion time $E[T_{\tau_K,\ldots,\tau_1}]$ (with $K$ restarts). Thus, $\tau_k^*$ must then also be an element of the global optimum $\{\tau_K^*, \ldots, \tau_1^*\}$ for $E[T_{\tau_K,\ldots,\tau_1}]$.

For higher moments, we can not remove the dependence on $s_k$, and the backward optimization algorithm does therefore not extend to higher moments. As an example, $E[(s_k + T_{\tau_k,\ldots,\tau_1})^2]$ leaves a term with $s_k \frac{d}{d\tau_k}E[T_{\tau_K,\ldots,\tau_1}]$ in the derivative. Hence, restart times $\tau_k, \ldots, \tau_1$ that optimize $E[T^2_{\tau_k,\ldots,\tau_1}]$ may not optimize $E[(s_k + T_{\tau_k,\ldots,\tau_1})^2]$ and/or $E[T^2_{\tau_K,\ldots,\tau_1}]$. □