Chapter 5
Meeting Deadlines Through Restart

Retrying tasks is an obvious thing to do if one suspects a task has failed. However, also if a task has not failed, it may be faster to restart it than to let it continue. Whether restart is indeed faster depends on the completion time distribution of tasks, and on the correlation between the completion times of consecutive tries. As in the previous chapter also in this chapter we assume that the completion times of consecutive tries are independent and identically distributed, an assumption that has been shown to be not unreasonable for Internet applications [128]. Furthermore, we analyse algorithms that are tailored to lognormal distributions, which we (and others) have found to be representative for various Internet applications [86, 128].

Our metric of interest is the probability that a pre-determined deadline is met, and we want to find the restart times that maximise this metric. Note that the metric of meeting deadlines corresponds to points in the completion time distribution, a metric often harder to obtain than moments of completion time which we analysed in the previous chapter. The material in this chapter has been published previously in [178, 164, 177].

We derive two very efficient algorithms to determine the optimal time for restart. The ‘equihazard’ algorithm finds all restart intervals with equal hazard rates, which corresponds to all local extrema for the probability of making the deadline. We applied the algorithm to lognormal distributed completion times. It turns out that among the equihazard restart intervals in all considered cases equidistant points are optimal. Therefore, a practical engineering approach is to only consider equidistant points, which we do in our second algorithm. The equihazard algorithm finds each local extremum in logarithmic time, the equidistant algorithm takes a constant time to do the same, and finds the globally optimal solution in a few iterations. Hence, these algorithms are excellent candidates for online deployment in potential future adaptive restart implementations.

5.1 A Model for the Probability of Meeting a Deadline Under Restart

To analyse and optimise the time at which to restart a job, we start from a simple model that lends itself to elegant analysis. As in the previous chapter we assume
that the restart of a task terminates the previous attempt and that successive tries are statistically independent and identically distributed. This is for instance the case when we click the reload button in a web browser: the connection with the server is terminated and a new download attempt is tried. In mathematical terms, the problem formulation is as follows. Let the random variable \( T \) denote the completion time of a job, with probability distribution \( F(t), t \in [0, \infty) \), and let \( d \) denote the deadline we set out to meet. Obviously, without restart, the probability that the deadline is met is \( F(d) \). Assume \( \tau \) is the restart time, and the random variable \( T_\tau \) denotes the completion time when an unbounded number of retries is allowed. That is, a retry takes place periodically, every \( \tau \) time units, until completion of the job or until the deadline has passed, which ever comes first. We write \( f_\tau(t) \) and \( F_\tau(t) \) for the density and distribution of \( T_\tau \), respectively, and we are interested in the probability \( F_\tau(d) \) that the deadline is met.

One can intuitively reason about the completion time distribution with restarts as Bernoulli trials. At each interval between restarts there is a probability \( F(\tau) \) that the completion ‘succeeds.’ Hence, if the deadline \( d \) is a multiple of the restart time \( \tau \), we can relate the probability of missing the deadline without and with restart through:

\[
1 - F_\tau(d) = \left(1 - F(\tau)\right)^\frac{d}{\tau}.
\]  

If the restart intervals are not identical we denote their length by \( \tau_1, \ldots, \tau_n \), assuming \( n \) intervals. If furthermore \( d \) is not exactly reached by the last interval the remaining time after the last interval until the deadline is \( d - \sum_{i=1}^{n} \tau_i \). If we furthermore introduce a penalty, or cost \( c \) associated with restart the probability of missing the deadline without and with restart relate similarly to the probability distribution function defined in (4.3):

\[
1 - F_\tau(d) = \begin{cases}
\prod_{i=1}^{k} (1 - F(\tau_i)) \cdot (1 - F(d - \sum_{i=1}^{k} (\tau_i + c))) & \text{if } \sum_{i=1}^{k} (\tau_i + c) \leq d \leq \sum_{i=1}^{k+1} \tau_i + kc \\
\prod_{i=1}^{k+1} (1 - F(\tau_i)) & \text{if } \sum_{i=1}^{k+1} \tau_i + kc \leq d < \sum_{i=1}^{k+1} (\tau_i + c).
\end{cases}
\]  

(5.2)

For the sake of an easier treatment we will in the following assume that \( c = 0 \) and \( \tau_1 = \ldots = \tau_n = \tau \), and that the deadline is an integer multiple of the restart time \( d = n\tau \).

For a single retry during the finite interval \([0, d)\), when the retry is at time \( \tau \), \( \tau < d \), then the probability of completion before \( d \) is:

\[
F_\tau(d) = 1 - (1 - F(\tau))(1 - F(d - \tau)).
\]  

(5.3)

By equating the derivative with respect to \( \tau \) to zero, we obtain for the extrema of \( F_\tau(d) \) that:
The function
\[ h(t) = \frac{f(t)}{1 - F(t)} \]
is known as the hazard rate, and is key throughout our analysis and algorithms. The above result shows that minima and maxima for the probability that a deadline is met with restarts are found at *equihazard* restart intervals. Moreover, the *equidistant* restart intervals \( \tau = \frac{d}{2} \) are a special case of equihazard intervals, and form thus also a local extremum.

For multiple retries before the deadline similar mathematics can be applied. This time we take derivatives with respect to each restart interval \( \tau_i, i = 1, \ldots, N \). (Note, the restarts take place at times \( \tau_1, \tau_1 + \tau_2, \ldots, \sum_{n=1}^{N} \tau_n \), and we assume without loss of generality that \( \sum_{n=1}^{N} \tau_n = d \).) Then we obtain that an optimum with respect to all retry intervals \( \tau_1, \ldots, \tau_N \) is found when:
\[ \frac{f(\tau_1)}{1 - F(\tau_1)} = \frac{f(\tau_2)}{1 - F(\tau_2)} = \ldots = \frac{f(\tau_N)}{1 - F(\tau_N)}. \tag{5.5} \]

Again, the extrema are at equihazard intervals, with as special case the equidistant restart intervals \( \tau_n = \frac{d}{N} \).

### 5.2 Algorithms for Optimal Restart Times

Very often, completion times for Internet tasks have a distribution function that can be closely fit by a lognormal distribution [86, 128]. Since the Internet is one of our anticipated application fields we chose in this section the lognormal distribution with parameters we fit to the data in [128]. The density function and the hazard rate of a lognormal distribution are shown in the Appendix in Sect. B.2.6.

The lognormal shape of the hazard function can be exploited by optimisation algorithms, since it has at most two points with the same hazard function value. This allows us to quickly identify all potential solutions of the optimisation problem. The following algorithm finds the two restart interval lengths \( \tau_a \) and \( \tau_b \) for which holds:
\[ h(\tau_a) = h(\tau_b), \tag{5.6} \]
\[ n_a \tau_a + n_b \tau_b = d, \tag{5.7} \]
where \( n_a \) and \( n_b \) denote the number of intervals of each length. The parameters \( n_a \) and \( n_b \) are input to the algorithm, and to find the optimal restart strategy, one needs to call the algorithm for all relevant combinations of \( n_a \) and \( n_b \), and then select from
all the equihazard solutions the one that optimises the probability of meeting the deadline.

**Algorithm 5 (Equihazard Restart Intervals)**

Input \( n_a \) and \( n_b \);
\[ \text{top} = d/n_b; \quad \text{bottom} = d/(n_a + n_b); \]
\[ \tau_b = \text{top}; \quad \tau_a = \frac{d-n_b\tau_b}{n_a}; \]
Repeat {
\[ \text{top} = (\text{top}+\text{bottom})/2; \]
\[ \tau_b = \text{top}; \]
\[ \tau_a = \frac{d-n_b\tau_b}{n_a}; \] (so interval lengths sum to \( d \))
\[ \text{If}( \text{SignChanged}(h(\tau_b) - h(\tau_a)) \} \{
\quad \text{bottom} = \text{top};
\quad \text{top} = \text{PreviousValue}(\text{top});
\}
Until (\text{top-bottom} \approx 0)

To explain the working of Algorithm 5, first note that one solution to (5.7) is the equidistant restart strategy \( \tau_a = \tau_b = \frac{d}{N} \). The algorithm will end up with that solution, unless there exists a second solution. For this solution, it cannot be that \( \tau_a \) and \( \tau_b \) are both smaller or both larger than \( \frac{d}{N} \), since then the intervals would not sum to \( d \). Therefore, we can choose \( \tau_b > \frac{d}{N} \) and \( \tau_a < \frac{d}{N} \). Furthermore, it also must hold that \( \tau_b \leq \frac{d}{n_b} \). The algorithm utilises these facts to initialise an interval between \( \text{bottom} \) and \( \text{top} \) in which \( \tau_b \) lies, and then breaks the interval in two at every iteration, until \( \text{top} \approx \text{bottom} \). At every iteration, it sets \( \tau_b \) to the guess \( \text{top} \) and computes the belonging \( \tau_a = \frac{d-n_b\tau_b}{n_a} \). It then tests if the sign of \( h(\tau_b) - h(\tau_a) \) changes, to decide if \( \tau_b \) lies in the upper or lower half. This test works correctly thanks to the particular shape of the lognormal hazard function. Note that since the algorithm divides the considered interval in two in every iteration, it takes logarithmic time to find the optimum for every pair \( n_a, n_b \) for which the algorithm is run.

We applied Algorithm 5 to the lognormal distribution with parameters \( \mu = -2.3 \) and \( \sigma = 0.97 \), and deadline \( d = 0.7 \). The parameters fit data collected in [128], but are otherwise arbitrary. Figure 5.1 shows typical behaviour if one considers a single restart. The equidistant restart (at \( \tau = 0.35 \)) is optimal, while the other equihazard points turn out to be minima (\( \tau = 0.013 \) or \( \tau = 0.687 \)). The improvement in probability of making the deadline is from 0.977 to 0.990. Table 5.1 shows results for increasing number of restarts, displaying all sets of equihazard intervals that are extrema. We see from the table that for this example equidistant hazard rates always outperform the other equihazard points, and that the optimum is for three equidistant restarts (and thus four intervals). We also see from the table that if we restart too
5.2 Algorithms for Optimal Restart Times

It turns out that equidistant restarts are optimal in all experiments with lognormal distributions. Although we can construct examples in which for instance two non-equidistant points outperform equidistant points, for the lognormal distribution this only seems to be possible if no restart performs even better. Unfortunately, we have no proof for this phenomenon, but it gives us ground to use an algorithm that limits its search for optima to equidistant points, which can be done even faster than Algorithm 5 for equihazard points. In the following algorithm we increase the number of equidistant restart points (starting from 0), consider the probability of making the deadline for that number of restarts and stop as soon as we see no more improvement. This is a very advantageous stopping criterion since one needs not to set an arbitrary maximum on the number of restart points. We do not give the derivation of the correctness of this stopping criterion here, but instead close the discussion with the algorithm.

![Fig. 5.1 Probability of meeting deadline for one restart (d = 0.7, \( \mu = -2.3, \sigma = 0.97 \))](image)

deadline probability for single restart

<table>
<thead>
<tr>
<th># restarts</th>
<th>Equihazard intervals</th>
<th>( P(T_{[\tau]} &lt; d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>—</td>
<td>0.978</td>
</tr>
<tr>
<td>1</td>
<td>0.35, 0.35</td>
<td>0.990</td>
</tr>
<tr>
<td>1</td>
<td>0.013, 0.687</td>
<td>0.977</td>
</tr>
<tr>
<td>2</td>
<td>0.23, 0.23, 0.23</td>
<td>0.993</td>
</tr>
<tr>
<td>2</td>
<td>0.019, 0.34, 0.34</td>
<td>0.990</td>
</tr>
<tr>
<td>2</td>
<td>0.013, 0.013, 0.674</td>
<td>0.976</td>
</tr>
<tr>
<td>3</td>
<td>0.175, 0.175, 0.175, 0.175</td>
<td>0.99374</td>
</tr>
<tr>
<td>3</td>
<td>0.024, 0.225, 0.225, 0.225</td>
<td>0.993</td>
</tr>
<tr>
<td>3</td>
<td>0.019, 0.019, 0.331, 0.331</td>
<td>0.989</td>
</tr>
<tr>
<td>3</td>
<td>0.013, 0.013, 0.013, 0.660</td>
<td>0.976</td>
</tr>
<tr>
<td>4</td>
<td>0.14, 0.14, 0.14, 0.14, 0.14</td>
<td>0.99366</td>
</tr>
</tbody>
</table>

Table 5.1 Equihazard restart intervals and associated probability of meeting the deadline (\( d = 0.7, \mu = -2.3, \sigma = 0.97 \))
Algorithm 6 (Equidistant Restart Intervals)

\[
\begin{align*}
n &= 1; \ \text{prob}[1] = F(d); \\
&\text{Do}
\begin{align*}
&\quad n++; \\
&\quad \text{prob}[n] = 1 - (1 - F(d/n))^n;
\end{align*}
\text{Until} \ (\text{prob}[n] < \text{prob}[n-1]) \\
&\text{Return}(d/(n - 1))
\end{align*}
\]

Where other mechanisms like early fault detection add an enormous computation overhead to increase availability [133] we are able to increase the probability of completion before a deadline with a very easy mechanism by a factor of two (‘half a nine’). Further study is needed not only to see how the algorithms generalise to other completion time distributions but also how the impact of restarts depends on the remoteness of the deadline.

5.3 An Engineering Rule to Approximate the Optimal Restart Time

The algorithms for finding the optimal restart time to maximise the probability of meeting a deadline work well for theoretical distributions of a certain shape. However, for empirical data we might want a more rough approximation of (5.1) to be able to implement online methods for finding those optimal restart times, as we will show in the following section. And indeed, an approximation of (5.1) helps us to find a simple rule which we use in an online fashion.

As in Sect. 5.1 we again need some definitions: Let the random variable \( T \) denote the completion time of a job, with probability distribution \( F(t), t \in [0, \infty) \). Assume \( \tau \) is a restart time, and the random variable \( T_\tau \) denotes the completion time when an unbounded number of retries is allowed. We write \( f_\tau(t) \) and \( F_\tau(t) \) for the density and distribution of \( T_\tau \). We also use the hazard rate

\[
h(t) = \frac{f(t)}{1 - F(t)}, \tag{5.8}
\]

New in this section is the use of the cumulative hazard

\[
H(t) = \int_{s=0}^{t} h(s) \, ds,
\]

which is closely related to the distribution function in that

\[
1 - F(t) = e^{-H(t)} \tag{5.9}
\]
5.3 An Engineering Rule to Approximate the Optimal Restart Time

or

\[
H(t) = -\log(1 - F(t)).
\] (5.10)

Restart at time \( \tau \) is beneficial only if the probability \( F_\tau(d) \) of making the deadline \( d \) under restart is greater than the probability of making the deadline without restart, i.e.

\[
F_\tau(d) > F(d).
\] (5.11)

We again start from (5.1) which for completeness we repeat here

\[
1 - F_\tau(t) = (1 - F(\tau))^\frac{d}{\tau}.
\] (5.12)

If there would be a restart time \( \tau \) that maximises the completion probability \( F_\tau(t) \) for all values of \( t \), this would be the ideal restart time and be ‘stochastically’ optimal. However, except for pathological cases, such a restart time does not exist. Equation (5.12) is correct only for values of \( d \) and \( \tau \) such that \( d \) is an integer multiple of \( \tau \). But if we ignore this fact, or simply accept (5.12) as an approximation, we can find the optimal restart time in a straightforward way. Surprisingly, it turns out that the approximation gives us a restart time independent of the deadline \( d \), which is optimal in the limit \( d \to \infty \). That is, it optimises the tail of the completion time distribution under restarts, and is therefore beneficial for many other metrics as well, such as higher moments of the completion time.

**Theorem 5.1** The following statements about our approximation \( \tau^* \) are equivalent.

1. \( \tau^* \) is an extremum (in \( \tau \)) of

\[
(1 - F(\tau))^\frac{d}{\tau},
\] (5.13)

for any deadline \( d \);

2. \( \tau^* \) is the point where

\[
\tau^* \cdot h(\tau^*) = -\log (1 - F(\tau^*));
\] (5.14)

3. \( \tau^* \) is a point where

\[
\tau^* \cdot h(\tau^*) = H(\tau^*);
\] (5.15)

4. \( \tau^* \) is an extremum of

\[
-\frac{\log(1 - F(\tau))}{\tau};
\] (5.16)
5. \( \tau^* \) is an extremum of

\[
(1 - F(\tau))^\frac{1}{\tau}.
\]  

(5.17)

Proof We use

\[
\frac{d}{dx}(g(x))^x = (g(x))^x \left( x \frac{d}{dx} \frac{g(x)}{g(x)} + \log(g(x)) \right).
\]

If the first item is true, then \( \tau^* \) is an extremum when the derivative of \( (1 - F(\tau))^\frac{d}{\tau} \) equates to 0:

\[
\frac{d}{d\tau}(1 - F(\tau))^\frac{d}{\tau} = (1 - F(\tau))^\frac{d}{\tau} \left( \frac{f(\tau)}{1 - F(\tau)} + \log(1 - F(\tau)) \right) = 0.
\]

Irrespective of the value of \( d \), Statement 2 then follows immediately:

\[
\frac{f(\tau^*)}{1 - F(\tau^*)} = \frac{-\log(1 - F(\tau^*))}{\tau^*},
\]

which can be rewritten using (5.8) into

\[
\tau^* \cdot h(\tau^*) = -\log(1 - F(\tau^*)),
\]

and thus Statement 2 holds if and only if Statement 1 holds.

The equivalence between Statements 2 and 3 follows using the relation (5.10).

Statement 4 follows from taking the derivative of \( \frac{-\log(1 - F(\tau))}{\tau} \) and equate it to 0:

\[
\frac{d}{d\tau} - \log(1 - F(\tau)) = \frac{1}{1 - F(\tau)}(-f(\tau))\tau - \log(1 - F(\tau)) = 0
\]

\[
\iff \frac{f(\tau)}{1 - F(\tau)} = \frac{\log(1 - F(\tau))}{\tau}.
\]

So, Statement 4 holds if and only if statement 1 holds.

Statement 5 then follows from taking exponential power of the expression in Statement 4

\[
\exp\left( -\log(1 - F(\tau)) \cdot \frac{1}{\tau} \right) = (1 - F(\tau))^\frac{1}{\tau}.
\]

Note that, alternatively, Statements 4 and 5 can be obtained from properly manipulating the result in Statement 1 for \( d = 1 \).

The heuristics behind this approximation is that (1) the optimal restart times correspond to equihazard points; (2) equidistant restart times are equihazard points
and often (albeit not always) optimal. The mathematical trick then is to relax the restriction that the number of restarts must be integer valued. In doing so, one obtains a continuous function, for which one can take derivatives and get relations for its extremes. If one carries this out, it turns out that the optimal restart time is independent of the time $d$ one wants to optimise $(1 - F(\tau))^\frac{d}{\tau}$ for. Moreover, if $\frac{d}{\tau^*}$ takes an integer value, restart time $\tau^*$ is an equidistant restart strategy, and thus a local extremum. From this reasoning it also follows that the rule gets closer to the optimum for the tail of $F(\tau)$, since then $\frac{d}{\tau^*}$ is close to an integer value. This results in the following claim, which is stated without proof:

**Theorem 5.2** For $d \to \infty$, the approximation error in the restart interval lengths converges to zero:

$$ d - n^* \cdot \tau^* \to 0. $$

(5.18)

where $n^*$ is the maximum number of intervals of length $\tau^*$ that can be accommodated in the interval $[0, d]$.

$$ n^* = \left\lfloor \frac{d}{\tau^*} \right\rfloor. $$

Item 3 of Theorem 5.1 can be interpreted in the following way: the surface under the hazard rate curve up to point $\tau^*$ equals the rectangle defined by x- and y-value of $h(\tau^*)$ as illustrated in Fig. 5.2. We will refer to (5.15) as the *rectangle equals surface rule*. This very illustrative and simple rule is used later in a pragmatic algorithm for an empirical hazard rate to find an empirical optimal restart time that maximises the probability of completion, the probability of making an infinite deadline.

![Fig. 5.2 Illustration of the surface = rectangle rule, with optimal restart time 0.18 (hyper/hypoeexponential distribution with parameters as indicated in Appendix B.2.4)](image)
The quality of the approximation of the equivalent formulas in Theorem 5.1 is evaluated by formulating an engineering rule based on (5.16) (Item 4 of Theorem 5.1) and using this rule in experiments. The rule is defined as follows.

**Algorithm 7** Set the restart time at \( \tau^* \) with \( \tau^* \) the optimum of \( \frac{-\log(1-F(d))}{d} \).

It should be noted that if the hazard rate is monotonously increasing, no value of \( \tau \) exists that satisfies Theorem 5.1. In this case restart will not help increasing the probability of completion. Whereas if the hazard rate is monotonously decreasing the rectangle equals surface rule holds only for \( \tau = 0 \), which means immediate restarts. Most other distributions will have a single maximum in their hazard rates, after which the hazard rate then decreases. Although, of course, distributions can be constructed that have several local maxima also in their hazard rates. In both latter cases after some point a value of \( \tau \) exists, such that (5.15) holds. Only then restart can be applied successfully.

Figure 5.3 shows a variety of optimal restart times. The example is the lognormal distribution we already used in the previous chapter, with parameters \( \mu = -2.3 \) and \( \sigma = 0.97 \). The straight lines are the approximation, and the optimal restart times to minimise the first and second moment of the completion time. The sea-saw line is the optimal restart time for points on the distribution. That is, the x-axis gives the point \( t \) on the distribution, the y-axis the restart time \( \tau \) that maximises \( F_\tau(t) \). We see that the approximation gets closer to optimal as \( t \) increases; in other words, the approximation works best for the tail of the completion time distribution. This fact suggests that the approximation may be better for higher moments of completion time, since these are more sensitive to the tail of the distribution. This seems indeed to be the case, since the restart time that minimises the first moment is farther off our approximation than the restart time that minimises the second moment.

Figure 5.4 shows the completion time distribution for the various restart regimes. The solid curve gives \( F_\tau(t) \) with \( t \) on the x-axis and \( \tau \) equal to our approximation only for the interval \( t \in [0.0, 0.5] \). The dotted curve uses the restart time at point

![Fig. 5.3 Optimal restart times: approximation (lowest), optimal for first moment (highest), optimal for second moment (middle), optimal for points in distribution (sea-saw)]
that optimises $F_{\tau}(t)$. This is the theoretical optimum. The dashed curve uses the restart time that minimises the first moment of completion time. We see that both the approximation and the optimum restart time for the first moment are very close to the completion time distribution using the theoretically optimal $\tau$. Figure 5.5 shows on a logscale the difference between the completion time distributions $F_{\tau}(t)$ over a larger time horizon when computing $\tau$ using the approximation and the optimal completion time, as well as the difference between the completion time with restart optimised for the first moment and the theoretical optimum. We see that the approximation is exact at the spikes, which appear with distance equal to $\tau^*$. We also see that if $t$ increases, the optimal restart time for the first moment never reaches the theoretical optimum, and in fact slowly diverges from the approximation.

Table 5.2 shows how three different restart times perform with respect to moments and quantiles of the completion time distribution under restarts. One can see that the differences are minor, our approximation performing best for the 90% and higher quantiles. (The value in the table under quantiles is the point $t$ at which $F_{\tau}(t) = 0.9$.

**Fig. 5.4** $F_{\tau}(t)$ using optimal restart time at $t$ (dotted), approximation (solid line), and using the optimal restart time for first moment (dashed)

**Fig. 5.5** Logarithm of differences in completion time distribution $F_{\tau}(t)$ when using $\tau$ as computed for optimal restart times and approximation (solid line), using $\tau$ computed for optimal restart time and optimised for first moment (dashed)
(0.99, 0.999), smaller values are thus better.) The first moment is slightly lower using the average optimal restart time and the value of $\tau$ that optimises the second moment.

### 5.4 Towards Online Restart for Self-Management of Systems

The objective in this section is to automate restart, building on the above work. We want to explore online decision taking to see whether restart will be beneficial and when to apply it. We simulate an online procedure by using increasingly more data from measurements taken earlier [128], but the applied methods can easily be included in a software module like the proxy server in [128] to be executed in real-time.

In Sect. 3.2 we have already seen that the shape of the hazard rate of a probability distribution indicates whether restart is beneficial. For empirical data the correct theoretical distribution is unknown and the hazard rate therefore needs to be estimated based on observations. Estimating the hazard rate is not a straightforward task, since it needs numerical computation of the derivative of the cumulative hazard rate. We derive and implement our simple rule based on the hazard rate that allows us to find the optimal restart time to maximise the probability of making a deadline. The rule approximates the optimal restart time independent of the exact value of the deadline, and is asymptotically exact (when the deadline increases). Moreover, the rule is very simple, making it a likely candidate for runtime deployment. One has to bear in mind that not in all cases the optimal restart time does exist. Restart is applicable to a system if (and only if) the rule finds an optimal restart time. So, our simple rule actually serves a two-fold purpose: it enables us to decide whether restart will be beneficial in the given situation, and if so, it provides us with the optimal restart time.

We apply the rule to data sets we collected for HTTP, thus mimicking the online execution of the algorithm. We explore how much data is required to arrive at reasonable estimates of the optimal restart time. We also study the effect of failed HTTP requests by artificially introducing failures in the data sets. Based on these explorations we provide engineering insights useful for runtime deployment of our algorithm.

<table>
<thead>
<tr>
<th>Restart time</th>
<th>$E[T_1]$</th>
<th>$E[T_2]$</th>
<th>90% quant.</th>
<th>99% quant.</th>
<th>99.9% quant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximation</td>
<td>0.162</td>
<td>0.14600</td>
<td>0.0399</td>
<td>0.317</td>
<td>0.635</td>
</tr>
<tr>
<td>2nd Moment opt.</td>
<td>0.177</td>
<td>0.14573</td>
<td>0.0398</td>
<td>0.318</td>
<td>0.639</td>
</tr>
<tr>
<td>Average optimal</td>
<td>0.189</td>
<td>0.14568</td>
<td>0.0399</td>
<td>0.319</td>
<td>0.645</td>
</tr>
</tbody>
</table>
5.4 Towards Online Restart for Self-Management of Systems

5.4.1 Estimating the Hazard Rate

It follows from the surface equals rectangle rule (5.15) that an estimate \( \hat{h}(t) \) of the hazard rate curve is needed to determine the optimal restart time. We will in this section provide the main steps of how to estimate the hazard rate and implement the rule (5.15) in an algorithm. Some details are shifted to Appendix B.2.6. We use the theory on survival analysis in [80]. The hazard rate \( h(t) \) cannot be estimated directly from a given data set. Instead, first the cumulative hazard rate \( H(t) \) is estimated and then the hazard rate itself is computed as a numerical derivative.

Let us consider a sample of \( n \) individuals, that is \( n \) completions in our study. We sample the completion times and if we order them, we obtain a data set of \( D \) distinct times \( t_1 \leq t_2 \leq \ldots \leq t_D \) where at time \( t_i \) there are \( d_i \) events, that is \( d_i \) completions take time \( t_i \). The random variable \( Y_i \) counts the number of jobs that need more or equal to \( t_i \) time units to complete. We can write \( Y_i \) as

\[
Y_i = n - \sum_{j=1}^{i-1} d_j
\]

All observations that have not completed at the end of the regarded time period, usually time \( t_D \), are called right censored. There are \( Y_n - d_n \) right censored observations. The experimental data we use falls in that category, since Internet transactions commonly use TCP, which aborts (censors) transactions if they do not succeed within a given time.

The hazard rate estimator \( \hat{h}(t) \) is the derivative of the cumulative hazard rate estimator \( \hat{H}(t) \), which is defined in Appendix C.1. It is estimated as the slope of the cumulative hazard rate. Better estimates are obtained when using a kernel function to smooth the numerical derivative of the cumulative hazard rate. The smoothing is done over a window of size \( 2b \). A bad estimate of the hazard rate will yield a bad estimate of the optimal restart time and the optimised metric is very sensitive to whether the restart time is chosen too short. Therefore obtaining a good estimate of the hazard rate is important.

Let the magnitude of the jumps in \( \hat{H}(t) \) and in the estimator of its variance \( \hat{V}[\hat{H}(t)] \) at the jump instants \( t_i \) be \( \Delta \hat{H}(t_i) = \hat{H}(t_i) - \hat{H}(t_{i-1}) \) and \( \Delta \hat{V}[\hat{H}(t_i)] = \hat{V}[\hat{H}(t_i)] - \hat{V}[\hat{H}(t_{i-1})] \). Note that \( \Delta \hat{H}(t_i) \) is a crude estimator for \( \hat{h}(t_i) \).

The kernel-smoothed hazard rate estimator is defined separately for the first and last points, for which \( t - b < 0 \) or \( t + b > t_D \). For inner points with \( b \leq t \leq t_D - b \) the kernel-smoothed estimator of \( h(t) \) is given by

\[
\hat{h}(t) = b^{-1} \sum_{i=1}^{D} K \left( \frac{t - t_i}{b} \right) \Delta \hat{H}(t_i).
\]  

(5.19)

The variance of \( \hat{h}(t) \) is needed for the confidence interval and is estimated as
\[
\sigma^2[\hat{h}(t)] = b^{-2} \sum_{i=1}^{D} K \left( \frac{t - t_i}{b} \right)^2 \Delta \hat{V}[\hat{H}(t_i)].
\]

The function \( K(.) \) is the Epanechnikov kernel defined in Appendix C.2.

A \((1 - \alpha) \cdot 100\%\) point wise confidence interval around \( \hat{h}(t) \) is constructed as

\[
\left[ \hat{h}(t) \exp \left( -\frac{z_{1-\alpha/2} \sigma(\hat{h}(t))}{\hat{h}(t)} \right), \hat{h}(t) \exp \left[ \frac{z_{1-\alpha/2} \sigma(\hat{h}(t))}{\hat{h}(t)} \right] \right].
\]

where \( z_{1-\alpha/2} \) is the \((1 - \alpha/2)\) quantile of the standard normal distribution.

The choice of the right bandwidth \( b \) is a delicate matter, but is important since the shape of the hazard rate curve greatly depends on the chosen bandwidth (see Fig. 5.7) and hence a badly chosen bandwidth will have a serious effect on the optimal restart time. One way to pick a good bandwidth is to use a cross-validation technique of determining the bandwidth that minimises some measure of how well the estimator performs. One such measure is the mean integrated squared error (MISE) of \( \hat{h} \) over the range \( \tau_{\text{min}} \) to \( \tau_{\text{max}} \). The mean integrated squared error can be found in Appendix C. To find the value of \( b \) which minimises the MISE we find \( b \) which minimises the function

\[
g(b) = \sum_{i=1}^{M-1} \left( \frac{t_{i+1} - t_i}{2} \right) (\hat{h}^2(t_i) + \hat{h}^2(t_{i+1})) - 2b^{-1} \sum_{i \neq j} K \left( \frac{t_i - t_j}{b} \right) \Delta \hat{H}(t_i) \Delta \hat{H}(t_j).
\]

Then \( g(b) \) is evaluated for different values of \( b \). Each evaluation of \( g(b) \) requires the computation of the estimator of the hazard rate. The optimal bandwidth can be determined only in a trial-and-error procedure. We found in our experiments that the optimal bandwidth is related with the size of the data set and the variance of the data. We use the standard deviation to determine a starting value and then do a simple step-wise increase of the bandwidth until \( g(b) \) takes on its minimal value. In case the hazard rate is increasing in the first steps, we decrease \( b \) and start again, since then we are obviously beyond the minimum already. In our experiments and in the literature we always found a global minimum, never any local minima. Advanced hill-climbing algorithms can be applied to find the minimum more quickly and more accurately than we do here.

Once the best estimate of the hazard rate is found we need to determine the point \( i^* \) that satisfies the rectangle equals surface rule (5.15). The following simple algorithm determines the optimal restart time \( \tau^* \) by testing all observed points \( t_i, i = 1, \ldots, n \) as potential candidates.
Algorithm 8 (Optimal restart time)

Input $\hat{h}$, $\hat{H}$ and $t$;
$i = 1$;  \#($t = t_1, \ldots, t_n$)
While((i < n) and ($t_i \cdot \hat{h}(t_i) > \hat{H}(t_i)$) ) {
    $i$ ++;
}
return $t_i$;

This algorithm returns in the positive case the smallest observed value that is greater than the estimated optimal restart time $\tau^*$. In many cases, however, the studied data set does not contain observations large enough to be equal or greater than the optimal restart time. Then we extrapolate the estimated hazard rate to find the point where the rectangle equals the surface under the curve. Assuming we have a data set of $n$ observations $t_i$, $i = 1, \ldots, n$, at first the slope of the estimated hazard rate at the end of the curve is determined as the difference quotient

$$\text{slope} = \frac{\hat{h}(t_n) - \hat{h}(t_{n-1})}{t_n - t_{n-1}}.$$

Then $t_\tau = t_n + \Delta t$ is determined such that for $t_\tau$ Eq. (5.15) holds.

$$
(t_n + \Delta t) \cdot (\hat{h}(t_n) + \text{slope} \cdot \Delta t) = \hat{H}(t_n) \cdot \text{slope} \cdot \Delta t \cdot t_n
\iff
\Delta t = \frac{\hat{H}(t_n) - t \cdot \hat{h}(t_n)}{\hat{h}(t_n) - 2 \text{slope} t_n - \hat{H}(t_n) - \text{slope}}.
$$

The computational complexity of the algorithm depends in first place on the number of iterations needed to find the optimal bandwidth for the hazard rate estimator. In our experiments we used a heuristic based on the standard deviation of the data set that gave us the optimal bandwidth often in less than 5 iterations, but sometimes took up to 20 iterations.

The second important parameter is the number of observations considered. Each iteration on the bandwidth requires the computation of the estimated hazard rate, which in turn needs traversing all observations and uses for each point a window of size $2b$. Complexity of the hazard rate estimator is therefore at most $O(n^2)$. Improving on the heuristic for the bandwidth, so that in all cases only few iterations are needed is certainly worth while.

### 5.4.2 Experiments

We have implemented the algorithm to estimate the hazard rate and determine the optimal restart time as defined by (5.15) in Theorem 5.1. The implementation is
done in Mathematica and has been applied to the HTTP connection setup data studied in [128]. This data in fact consists of the time needed for TCP’s three-way handshake to set up a connection between two hosts.

In our experiments we investigate various issues. One is the uncertainty introduced by small sample sizes. The available data sets consist of approximately one thousand observations for each URL, that is thousand connection setup times to the same Internet address. We use these data sets and take subsets of first one hundred then two hundred observations etc. as indicated in the caption of the figure and in the table. We do not use data of different URLs in one experiment since we found that very often different URLs have different distributions or at least distribution parameters. Furthermore, the application we have in mind is web transactions between two hosts.

The data we study is Data Set ‘28’ consisting of roughly 1,000 connection setup times to http://nuevamayoria.com, measured in seconds. This data set shows characteristics such as a lower bound on all observation and a pattern of variation which we found in many other data sets as well, even though usually not with the same parameters. The chosen data set is therefore to be seen as one typical representative of a large number of potential candidates. The considered connection setup times are shown in Fig. 5.6. The largest observation in this data set is 0.399678 s.

For each of the mentioned subsamples the optimal smoothing factor, or bandwidth, is computed by evaluating (5.22) several times, finding the minimum in a simple search. Figure 5.7 shows estimates of the hazard rate for different values of the bandwidth. Parameter $b_1$ is too large, whereas $b_2$ is too small, $b_3$ is the one that minimises the error and is therefore the optimal bandwidth. One can see that too large a bandwidth leads to an extremely smooth curve, whereas too small a bandwidth produces over-emphasised peaks. From the figure one might conclude that rather too large a bandwidth should be chosen than one that is too small, but more experiments are needed for a statement of this kind. Using the optimal bandwidth,
the hazard rate and its 95% confidence interval are estimated according to (5.19) and (5.21). Finally, for each estimated hazard rate the optimal restart time \( \tau^* \) is computed using Algorithm 8. In some cases, the algorithm finds the optimal restart time, since the data set includes still an observation greater than the optimal restart time. If the data set has no observation large enough to be greater than the optimal restart time, we extrapolate according to (5.24). The optimal restart times are drawn as vertical bars in the plots in Figs. 5.8 and 5.9.

Note that in Fig. 5.8 although it looks like all optimal restart times are extrapolated in fact none of them is. The extrapolated optimal restart times are indicated by an asterisk in Table 5.3.

The hazard rate curve has no value at the point of the largest observation, since for the numerical derivation always two data points are needed. Furthermore, because of the limited amount of data in the tail, it is not surprising that the confidence interval at the last observations grows rapidly.

Table 5.3 shows some characteristics obtained in the program runs for Data Set 28. Each block of the table belongs to a subset of size \( n \) with corresponding standard deviation. The standard deviation changes as more observations come into consideration. For each subsample three different cases are studied. In the first one only the \( n \) observations are used and the failure probability equals either zero, or the relative fraction of observations that are greater than 3.0. This threshold is the first retransmission timeout of TCP and hence observations greater 3.0 are (somewhat arbitrarily) censored and retried. We treat them as censored observations and all censored observations contribute to the failure probability. Data Set ‘28’ does not have any such censored observations, but many other data sets do. The second group consists of the \( n \) observations plus 2\( n \) censored ones and has therefore failure prob-
Fig. 5.8 Estimated hazard rates and confidence intervals for the estimates for increasing sample size (top row $n = 100$ and $n = 200$, middle row $n = 400$ and $n = 600$, bottom row $n = 800$) and failure probability 0.0

Fig. 5.9 Estimated hazard rates and confidence intervals for sample size $n = 1,000$, failure probability 0.0 (left) and 0.8 (right)
Table 5.3 Optimal restart time ($\tau^*$) and optimal bandwidth (bw) for different subsample sizes of Data Set 28 and different failure probabilities

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\text{StdDev} = 0.0121551$</th>
<th>$n = 200$, $\text{StdDev} = 0.0117341$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>Failure prob. bw</td>
<td>$\tau^*$</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>0.0</td>
<td>0.006758 0.389027</td>
<td>0.0</td>
</tr>
<tr>
<td>0.666667</td>
<td>0.001779 0.597251*</td>
<td>0.666667</td>
</tr>
<tr>
<td>0.8</td>
<td>0.001779 0.554513*</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n = 300$, $\text{StdDev} = 0.0106746$</th>
<th>$n = 400$, $\text{StdDev} = 0.010383$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>Failure prob. bw</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>0.011742 0.389027</td>
</tr>
<tr>
<td>0.666667</td>
<td>0.001272 0.333271</td>
</tr>
<tr>
<td>0.8</td>
<td>0.001156 0.333271</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n = 500$, $\text{StdDev} = 0.00997916$</th>
<th>$n = 600$, $\text{StdDev} = 0.00941125$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>Failure prob. bw</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>0.010977 0.399678</td>
</tr>
<tr>
<td>0.666667</td>
<td>0.001081 0.333271</td>
</tr>
<tr>
<td>0.8</td>
<td>0.001081 0.333271</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n = 700$, $\text{StdDev} = 0.00895504$</th>
<th>$n = 800$, $\text{StdDev} = 0.00851243$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>Failure prob. bw</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>0.009850 0.309209</td>
</tr>
<tr>
<td>0.666667</td>
<td>0.000970 0.333271</td>
</tr>
<tr>
<td>0.8</td>
<td>0.000970 0.333271</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n = 900$, $\text{StdDev} = 0.00816283$</th>
<th>$n = 1000$, $\text{StdDev} = 0.00784583$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>Failure prob. bw</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>0.0</td>
<td>0.009877 0.308456</td>
</tr>
<tr>
<td>0.6667</td>
<td>0.000884 0.332014</td>
</tr>
<tr>
<td>0.8</td>
<td>0.000884 0.332014</td>
</tr>
</tbody>
</table>

ability 2/3, or a little higher if there are additional censored observations present in the data set. Analogously, the third group has $n + 4n$ observations and a failure probability of $4n/5n = 0.8$ (or more if there are censored observations in the data set).

When looking at the results for failure probability zero, also plotted in Fig. 5.8 for $n = 100$, 200, 400, 600, 800 we see that the small data sets lead to an overestimated optimal restart time (if we assume that the full 1,000 observations give us a correct estimate), and the ‘correct’ value is overestimated by less than 5%.

Such high failure probability can be interpreted as an interruption of the connection. It should be noted that a failure probability of 0.1 or less does not show in the results at all. Looking at the results for the different sample sizes in the group with high failure probability, we also find that with the small samples the optimal restart time is overestimated.

The impact of the failure probability within a group of fixed sample size has been investigated as well. The failure probability is increased by subsequently adding more failed (and hence censored) observations and then estimates for the hazard rate and optimal restart time are computed. The failed attempts of course increase the sample size. We notice (as can be seen in Table 5.3) that the bandwidth used for estimating the hazard rate decreases for increasing failure rate, while the sample standard deviation is computed only from non-failed observations and hence does
not change with changing failure probability. We found in [163] that for theoretical
distributions the optimal restart time decreases with increasing failure probability.

Typically, our experiments agree with this property, which, however, is not true
for some subsets of Data Set ‘28’.

An additional purpose of the experiments was to find out whether we can relate
the optimal bandwidth to any characteristic of the data set. In the literature no
strategy is pointed out that helps in finding the optimal bandwidth quickly. In our
implementation we set the standard deviation as a starting value for the search. If we
have no censored observations (failure probability zero) we always find the optimal
bandwidth within less than five iterations. If the data set has many censored obser-
vations the optimal bandwidth roughly by factor 5 and we need more iterations to
find that value, since our heuristic has a starting value far too large in that case.

Figure 5.9 compares two hazard rates using another, larger data set, the first has
zero failure rate and the second has failure rate 0.8. It can be seen that the high
number of added censored observations leads to a much narrower hazard rate, with
lower optimal restart time. Note that this figure is based on a different data set than
the ones above, which has a larger sample size than the data set used before.

In summary, we have provided an algorithm that gives us an optimal restart time
to maximise the probability of meeting a deadline only if restart will indeed help
maximising that metric. So if the algorithm returns an optimal restart time we can
be sure that restart will help. We found a heuristic based on the variance of the data
that helps in finding the bandwidth parameter needed for the hazard rate estimator
fast. We observe that small data sets usually lead to an overestimated optimal restart
time. But we saw earlier (in [163]) that an overestimated restart time does much less
harm to the metric of interest than an underestimated one and we therefore willingly
accept overestimates. The whole restart process is automated to an extent that allows
us to propose it for self-management of systems.

The runtime of the algorithm depends on the considered number of observations
and on the number of iterations needed to find a good bandwidth for the hazard rate
estimation. We found that for our smaller data sets with up to 400 observations less
than 5 iterations are needed and the algorithm is very fast. We did not evaluate CPU
time and the Mathematica implementation is not runtime optimised, but a suggestion
for an optimal restart time in the above setting can be provided within a few seconds.
If, however, the data set grows large, e.g. more than 800 observations, each
iteration on the bandwidth takes in the order of some 1 or 2 min. The polynomial
complexity becomes relevant and the method is no longer applicable in an online
algorithm.

A good heuristic for choosing the optimal bandwidth is a key part in the whole
process. The better the first guess, the less iterations are needed and the faster the
optimal restart time is obtained. We cannot compare our heuristic to others since in
the literature nothing but pure ‘trial and error’ is proposed. But we can say, that for
small data sets and failure probability zero the optimal restart time is obtained very
fast since the heuristic provides a good first estimate of the bandwidth.

In our experience the smallest data sets were usually sufficient for a reasonably
good estimate of the optimal restart time. The optimal restart time will always be
Towards Online Restart for Self-Management of Systems

placed at the end of the bulk of the observations and some few hundred observations are enough to get a notion of ‘bulk’ and ‘end of the bulk’. If we consider that some web pages consist of up to 200 objects a data set of 100 samples is neither difficult to obtain nor unrealistic. In Internet transactions some hundred samples are easily accumulated. Furthermore, small samples seem to overestimate the optimal restart time, which does the maximised metric much less harm than underestimation.

One may argue that if everybody applies restart networks become more congested and response times will drop further. And in fact restart changes the TCP timeout – for selected applications. In our measurements we found that less than 0.5% of all connection setup attempts fail. Our method tries to detect failures faster than the TCP timeout and to restart failed attempts, since for slow connections restart typically does not lead to improved response time, whereas for failed connections in many cases it does. Failed attempts, however, are so rare that restarting those does not impose significant extra load on a network, while potentially speeding them up significantly. Obviously, if the failed attempts target at a server that is out of operation restart cannot improve the completion time. Restart can only help in the presence of transient failures. When sending requests to a permanently failed system the timeout values will increase, thus avoiding heavy load on the network connecting the failed server.
Stochastic Models for Fault Tolerance
Restart, Rejuvenation and Checkpointing
Wolter, K.
2010, XVI, 269 p., Hardcover
ISBN: 978-3-642-11256-0