Fitting Heavy-Tailed HTTP Traces with the New Stratified EM-Algorithm

Ramin Sadre   Boudewijn R. Haverkort
Design and Analysis of Communication Systems, University of Twente
Enschede, The Netherlands
{r.sadre,brh}@cs.utwente.nl

***Abstract*— A typical step in the model-based evaluation of communication systems is to fit measured data to analytically tractable distributions. Due to the increased speed of today's networks, even basic measurements, such as logging the requests at a Web server, can quickly generate large data traces with millions of entries. Employing complex fitting algorithms on such traces can take a significant amount of time. In this paper, we focus on the Expectation Maximization-based fitting of heavy-tailed distributed data to hyper-exponential distributions. We present a data aggregation algorithm which accelerates the fitting by several orders of magnitude. The employed aggregation algorithm has been derived from a sampling stratification technique and adapts dynamically to the distribution of the data. We illustrate the performance of the algorithm by applying it to empirical and artificial data traces.**

I. Introduction

The evaluation of the performance of communication networks (expressed in terms like blocking probability, response time, etc.) heavily relies on the availability of correct models of the traffic between the communicating entities. In the past, simple "Markovian" distributions, such as Poisson distributions, were often used to model various aspects of the underlying traffic processes, but extensive traffic measurements have shown the presence of properties such as self-similarity and long-range dependency in network traffic [1]. For World Wide Web traffic, it has been argued [2] that these can be explained by the heavy-tailedness of many of the involved distributions. In an heavy-tailed distribution (HTD), the complementary cumulative distribution function F^c decays more slowly than exponentially, i.e., $e^{\theta x}F^c(x) \to \infty$ as $x \to \infty$ for all $\theta > 0$. For a random variable X, distributed according to some HTD, we typically have $P[X > x] \sim x^{-\alpha}$ for $x \to \infty$ and $0 < \alpha < 2$.[1] As a consequence, models based on Poisson modeling generally fail to predict important performance measures for such traffic [3] and various efforts have been performed to develop more appropriate traffic models.

Typically, the first step in developing a traffic model is to fit the empirical distribution found in measurement data to an explicit heavy-tailed distribution. For this purpose, distributions generated by Markovian processes have been of special interest [4]–[14] because they can be easily incorporated in Markov-chain based models and, hence, analyzed with existing tools, in contrast to classical HTDs, such as Weibull and Pareto distributions. Hyper-exponential distributions (HEDs) and their extensions have been shown to well approximate empirical HTDs. Whereas Weibull and Pareto distributions only have two free parameters, the typically used HEDs have more than 10 free parameters and, hence, require more complex fitting algorithms. Especially Expectation-Maximization (EM) based algorithms have been successfully employed for the direct fitting of HEDs to measurement data.

In previous publications [15]–[19], we focused on data traces obtained from measurements at web proxy servers. With the increasing speed of the Internet, such traces may comprise several million of queries even for short periods of measurement. Although the standard EM-algorithm has a complexity which is "only" linear to the number of data points, its iterative nature generally does not allow to predict its overall runtime for a given data set. Hence, techniques to speed-up the algorithm have been of particular interest. In this paper, we present a new EM-based fitting technique which is significantly faster than the standard EM-algorithm by using a data stratification and aggregation algorithm. That algorithm has been derived from a sampling approach and adapts dynamically to the distribution of the data.

The paper is organized as follows. In Section II we give a brief overview on related works in the area of EM-based algorithms and trace sampling. The standard EM-algorithm, including its specialization to HEDs, is presented in Section III. Our new EM-algorithm with stratification is introduced in Section IV. We evaluate its performance in Section V by applying it to several empirical and artificial WWW document size distributions.

II. Related work

The stratified EM-algorithm that we present here is based on the EM-algorithm for HEDs (see Section III), introduced in [15], [16] and later used in [17]–[19] to design new caching and scheduling algorithms for Web proxy servers. Two other EM-based fitting methods that increase the efficiency and the accuracy of the EM-algorithm have recently been developed by other authors: D&C-EM and G-FIT.

In the Divide-and-Conquer-EM algorithm (D&C-EM) [11], the data is first partitioned and the EM-algorithm is then used to fit an HED to each partition. The thus obtained HEDs are then composed to a final HED for the overall trace. To increase the efficiency, the partitions are selected so that they exhibit a

[1]Note that "$x \to \infty$" should be read as "for very large x" in case of measurements.

lower variability than the variability of the entire trace. This leads to a faster convergence of the EM-algorithm. Speed-ups of 10 to 1000 in comparison to the original EM-algorithm, depending on the number of phases and the characteristics of the data set, have been reported. In [12], the authors extend the approach to a mixture of an Erlang and a hyper-exponential distribution which allows to approximate data sets with non-monotonically decreasing probability density function.

In [13], [20], an EM-algorithm, denoted as G-FIT, has been presented for fitting mixtures of Erlang distributions, so-called hyper-Erlang distributions, to trace data. In [10], G-FIT is extended by an aggregation algorithm that reduces the size of the data set before the EM-algorithm actually is applied. Two types of aggregation are distinguished. In so-called *uniform trace aggregation*, the range of possible data values is divided into uniform intervals. Then, for each interval the average of the data points inside the interval is determined. The thus-obtained averages of each interval are then used as input data for a modified version of G-FIT. For heavy-tailed distributions, the uniform partitioning does not perform well. Due to the heavy-tailedness, the data set covers a large range of values. One would be forced to use a large number of intervals in order to approximate the distribution well. For this reason, the authors also propose a *logarithmic trace aggregation* where the intervals have equal width on the logarithmic scale. The experiments in [10] show that the reduction of a heavy-tailed distributed data trace with more than 10^6 elements to an aggregate trace with some hundred elements yields accurate results.

Our stratified EM-algorithm has been derived from a sampling approach. Sampling techniques have been used before to analyze large network traces. The estimation of flow distributions by means of packet sampling [21], [22] addresses specific problems that are not present when sampling Web queries. For example, important information about the flows, such as connection initialization and finalization packets, may be erased by the sampling operation. Sampling techniques in the context of Web data are studied in [23]. The authors compare systematic and Poisson sampling of web queries, however, without a deeper analysis of the consequences of the sampling on the statistical properties of the obtained data. In [24], it is observed that Web traces can give very good estimates of the expected future volume of network traffic if the long-range dependent nature of the data is respected.

III. EM-FITTING WITH HYPER-EXPONENTIAL DISTRIBUTIONS

The Expectation Maximization method (EM) is a well-known algorithm to fit measurements to distributions [4], [25]–[27]. The EM-algorithm operates in an iterative fashion directly on a set of measurement values. Below we give a brief summary of [15] and [16]. We outline the method in general in Section III-A, and then give in Section III-B its specialization to the case where the distribution function to fit to is a hyper-exponential distribution (HED).

A. General approach

Given N measurement data points $x_1, \ldots, x_N$, we search the parameters $\mathbf{c} = (c_1, \ldots, c_I)$ and $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_I})$ of the density function

$$p(x|(\mathbf{c}, \boldsymbol{\theta})) = \sum_{i=1}^{I} c_i \cdot p(x|\boldsymbol{\theta_i}), \tag{1}$$

so that it "best" fits the density of the measurement data. The density in Equation (1) is a convex combination of basic density functions $p(x|\boldsymbol{\theta_i})$ parameterized by $\boldsymbol{\theta_i}$ with weights $c_i \geq 0$ and $\sum_{i=1}^{I} c_i = 1$.

Let $\alpha = (\mathbf{c}, \boldsymbol{\theta})$ and $\alpha' = (\mathbf{c'}, \boldsymbol{\theta'})$ be two sets of parameters for the density p. The EM-algorithm defines a new probability mass function

$$\delta(i|x_n, \alpha) = \frac{c_i \cdot p(x_n|\boldsymbol{\theta_i})}{p(x_n|\alpha)},$$

as well as the function

$$Q(\alpha, \alpha') = \sum_{n=1}^{N} \sum_{i=1}^{I} \delta(i|x_n, \alpha) \cdot \log(c'_i \cdot p(x_n|\boldsymbol{\theta'_i})),$$

which provides a quality criterion for α and α': it says how much better the density function $p(x|\alpha')$ fits the measurement data than the density function $p(x|\alpha)$.

The EM algorithm proceeds iteratively: starting from an initial parameter set α, it computes a new parameter set α' such that $Q(\alpha, \alpha')$ is maximized. This α' is used as starting point for the next iteration. The algorithm stops when $\alpha \approx \alpha'$ (see below). To find the next value α', that is, to optimize Q, one has to take derivatives to subsequently solve the (possibly non-linear) equation system:

$$\frac{\partial Q}{\partial \alpha'} = 0 \Rightarrow \frac{\partial Q}{\partial \boldsymbol{\theta'_1}} = 0, \cdots, \frac{\partial Q}{\partial \boldsymbol{\theta'_I}} = 0. \tag{2}$$

Using Lagrange multipliers (with auxiliary condition $\sum_{i=1}^{I} c_i = 1$), the new weights are given by $c'_i = \frac{1}{N} \sum_{n=1}^{N} \delta(i|x_n, \alpha)$.

B. Specialization to HEDs

In general, the system (2) is difficult to solve. However, in case of hyper-exponentials as basic densities this becomes easily feasible [15], [16]. An HED can be interpreted as a probabilistic choice between I exponential distributions. With (initial) probability c_i the i-th negative exponential distribution (with rate λ_i) is chosen. Such an I-phase HED has density

$$f(x) = \sum_{i=1}^{I} c_i \lambda_i e^{-\lambda_i x}.$$

Hence, the basic densities in the EM-algorithm are $p(x|\lambda_i) = \lambda_i e^{-\lambda_i x}$. Note that, for $I \rightarrow \infty$, one can represent *any* distribution with squared coefficient of variation at least 1 and with completely monotone probability density function arbitrary close by hyper-exponentials [8]. However, it has been shown that with values of I up to 20 [8], HEDs can approximate HTDs, such as the Weibull distribution, for large ranges of x.

1) Select an appropriate number of distributions I and initial parameters $\alpha = (c_1, \ldots, c_I, \lambda_1, \ldots, \lambda_I)$, as well as a positive required accuracy ε.
2) Compute for $i := 1$ to I:
   a) $\delta(i|x_n, \alpha) = \frac{c_i \cdot p(x_n|\lambda_i)}{p(x_n|\alpha)}$,
   b) $c'_i = \frac{1}{N} \sum_{n=1}^{N} \delta(i|x_n, \alpha)$,
   c) $\lambda'_i = \frac{\sum_{n=1}^{N} \delta(i|x_n,\alpha)}{\sum_{n=1}^{N} \delta(i|x_n,\alpha) \cdot x_n}$.
3) Return to step 2 with $c_i := c'_i$ and $\lambda_i := \lambda'_i$ until the difference between c_i and c'_i and/or the difference between λ_i and λ'_i, for all i, is smaller than the accuracy boundary ε.

Fig. 1. Standard EM-algorithm for HEDs (from [15], [16])

System (2) gives us, after some intermediate steps, that

$$\lambda'_i = \frac{\sum_{n=1}^{N} \delta(i|x_n, \alpha)}{\sum_{n=1}^{N} \delta(i|x_n, \alpha) \cdot x_n}.$$

The EM-algorithm specialized for HEDs now takes the form shown in Figure 1. The complexity of each iteration is $O(N \cdot I)$, with N the number of data points and I the number of phases.

A problem with the EM-algorithm is the fact that it is difficult to predict the number of iterations needed to reach a given precision of the result [4]. However, in our experiments, good results generally have been obtained within $10-30$ iterations. Additionally, it should be noted that even for a case study with N well over 20 million (see below) and $I = 5$, one iteration takes less than 10 seconds on a standard personal computer. The number of required iterations is heavily influenced by the choice of the initial values. The knowledge about the shape of the distribution function can be used to choose initial values that are near to the (expected) final results of the algorithm. In [15], [16], the c_i and λ_i are initialized with exponentially decreasing (with i) values. In [10], the authors derive initial values from the measurement data.

IV. EM-ALGORITHM WITH STRATIFICATION

We have seen that the complexity of the EM-algorithm directly depends on the number of observations to process per iteration. A common method to reduce this number is the method of *sampling* [28]. The idea is to apply the algorithm only on a selected set of observations that are considered to be representative for the whole data trace. Usually, a random process is employed to select the observations out of the whole trace (*random sampling*) because a *systematic sampling*, for example by selecting every n-th observation, would be too vulnerable to periodicities in the trace. Often, a Poisson process is used as random process because it yields independent selections (*Poisson sampling*) [23].

When random sampling is applied on a heavy-tail distributed data set it quickly shows that the obtained results exhibit a large variance. This is caused by the nature of the heavy-tailedness: such a data set comprises many small and only few, but very large values. Using random sampling, there is a high probability to accidentally "overlook" some of the values located in the tail of the distribution. *Stratified sampling* is a general approach to reduce the variance of the result [28]. To achieve this, the population is divided into strata and each stratum is sampled separately such that even rare values have a high probability to be selected.

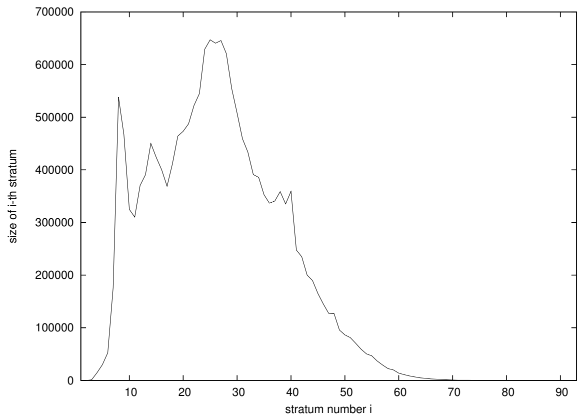

Fig. 2. Size of the strata for the RWTH$^+$ trace

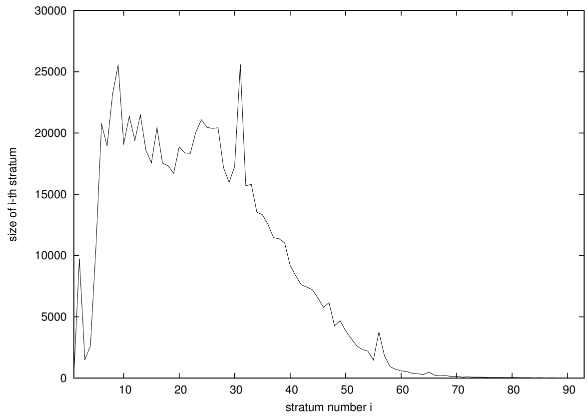

Fig. 3. Size of the strata for the IRCache trace

We use the following algorithm to build the strata [29]. It operates on the sorted list of observations. Starting with the smallest value, the observations are added to the first stratum until the squared coefficient of variation of the data in the stratum reaches a threshold c^2_{max}. Then, a new, empty stratum is created and the algorithm is applied to the remaining observations. As a consequence, the regions of the distribution with low variability are assigned to large strata, whereas the regions with high variability are divided into many small strata. By selecting a fixed number of observations from each stratum, this approach makes that the regions with high variability are over-represented in the sampling result.

We have applied the stratification algorithm on several heavy-tailed data traces (they are discussed in detail in Section V). Figure 2 and 3 show the size (number of elements) of the generated strata for the two most recent traces, the RWTH$^+$ trace and the IRCache trace, with $c^2_{max} = 0.001$. It shows that, even for such a low threshold c^2_{max}, only a small number of strata is generated. This is true for all traces discussed in Section V and is caused by the fact that, although the distribution is heavy-tailed, most of the data is located in

1a) Build the strata with threshold c^2_{max}.
1b) Compute for each stratum i, $i = 1, \ldots, N^*$, the average $\widehat{x}_i$ and weight w_i.
1c) Select an appropriate number of distributions I and initial parameters $\alpha = (c_1, \ldots, c_I, \lambda_1, \ldots, \lambda_I)$, as well as a positive required accuracy ε.
2) Compute for $i := 1$ to I:
   a) $\delta(i|\widehat{x}_n, \alpha) = \frac{c_i \cdot p(\widehat{x}_n|\lambda_i)}{p(\widehat{x}_n|\alpha)}$,
   b) $c'_i = \sum_{n=1}^{N^*} w_n \delta(i|\widehat{x}_n, \alpha)$,
   c) $\lambda'_i = \frac{\sum_{n=1}^{N^*} w_n \delta(i|\widehat{x}_n, \alpha)}{\sum_{n=1}^{N^*} w_n \delta(i|\widehat{x}_n, \alpha) \cdot \widehat{x}_n}$.
3) Return to step 2 with $c_i := c'_i$ and $\lambda_i := \lambda'_i$ until the difference between c_i and c'_i and/or the difference between λ_i and λ'_i, for all i, is smaller than the accuracy boundary ε.

Fig. 4. EM-algorithm with stratification for low threshold c^2_{max}

the head of the distribution. The specific shape of the strata size plots can be explained by comparing them with the histogram of the respective trace data. We will discuss this for the two traces in more detail in Section V.

An important simplification of the algorithm follows from the observed strata sizes. The threshold $c^2_{max} = 0.001$ means that the data contained in each stratum obeys a more or less deterministic distribution. For such strata, the random sampling within the stratum actually is not required to select representative observations. Instead, it is sufficient to represent a stratum simply by the average of all values contained in the stratum. In this way, we obtain a small sequence of average values $\widehat{x}_1, \ldots, \widehat{x}_{N^*}$ which can be used as input data for the EM-algorithm. Since the strata have different sizes, the EM-algorithm has to weight each average $\widehat{x}_i$ by the weight $w_i = s_i/N$, where s_i is the size of stratum i and N is the total number of observations [10]. For HEDs, the stratified EM-algorithm now takes the form shown in Figure 4.

This means that the EM-algorithm with stratification (denoted as *sEM* in the following) becomes identical to the aggregation-based G-FIT algorithm [10] for low threshold c^2_{max}. However, in the sEM-algorithm the number of strata and their size are determined dynamically.

V. Application and Validation

In order to evaluate our fitting approaches, we applied them to four data traces. Using these, we made comparisons between:

1) first-order statistics of the obtained HEDs and those of the original data traces;
2) performance results for an M|G|1|K queue where the original data traces and the fitted distributions are used as service time distribution.

The data traces are described in detail in Section V-A. The obtained HEDs are discussed in Section V-B. In Section V-C, we present the results for the M|G|1|K queue.

A. Statistics of the data traces

Some important object size statistics of the studied data traces are summarized in Table I. The object sizes are measured in bytes. The squared coefficient of variation is denoted as SCV. We discuss each trace in the following.

TABLE I
Object size statistics for the data traces

trace	#entries	min	max	mean	median	SCV
RWTH$^+$	$20.7 \cdot 10^6$	17	$228 \cdot 10^6$	12303.2	2416	316.2
NASA	$3.1 \cdot 10^6$	3	$7 \cdot 10^6$	20744.9	4142	13.39
IRCache	$0.74 \cdot 10^6$	189	$43 \cdot 10^6$	13094.8	2710	200.1
Weibull	10^6	10^{-4}	$29 \cdot 10^6$	10000	1185	10

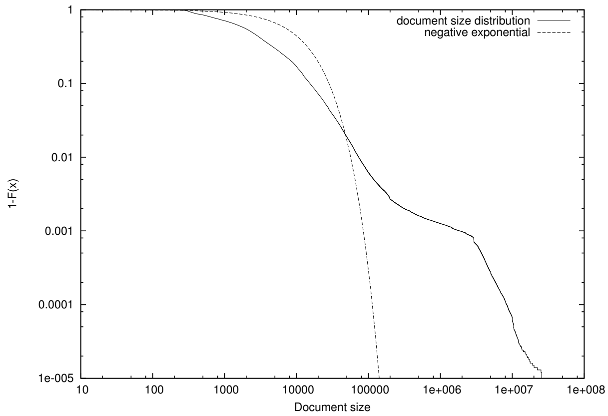

Fig. 5. Complementary object size distribution (RWTH$^+$ trace)

1) RWTH$^+$ trace: Early 2000, we collected the access logs of the RWTH Aachen proxy server. The log comprises the description of over 20 million HTTP requests made over a period of about 25 days. A subset of this trace, reduced to requests to cacheable objects, has been used in earlier publications [15]–[19]. We studied the sizes of the objects requested by the clients. Figure 5 shows the complementary distribution of the object sizes as log-log plot. Obviously, the object-size distribution function decays much slower than a negative exponential distribution and is clearly heavy tailed. This is confirmed by the observation that the median of the distribution is much smaller than the mean.

Figure 6 shows the object size frequencies using logarithmic bin sizes. The figure also depicts the frequencies for specific object types. We have grouped the objects into three categories: HTML documents, images (GIF, JPEG, etc.), and objects of other types. A direct correspondence between the strata sizes (see Figure 2) and the histograms can be observed. It shows that the peaks in the strata size plot are mainly caused by the image objects.

2) NASA trace: The NASA trace was first presented and evaluated in 1996 by Arlitt and Williamson [30]. It consists of about 3.1 million requests collected at the web server of the Kennedy Space Center. As in the RWTH$^+$ trace, the size distribution of the requested objects in the NASA trace is clearly heavy tailed, yielding a high SCV and a mean much larger than the median. Figure 7 shows the complementary distribution of the object sizes as log-log plot. Again, we observe that the object-size distribution function decays much slower than a negative exponential distribution.

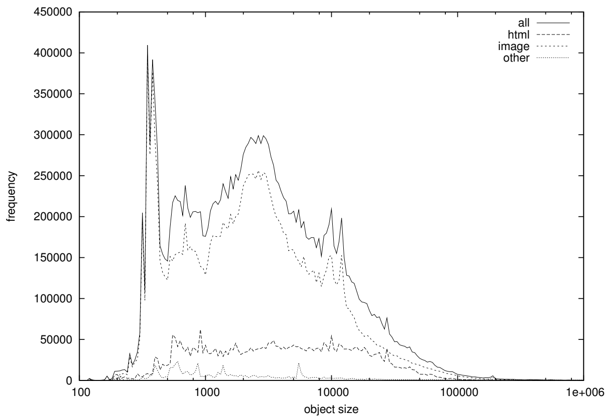

Fig. 6. Object size (bytes) frequencies by object type for the RWTH$^+$ trace (logarithmic bin size)

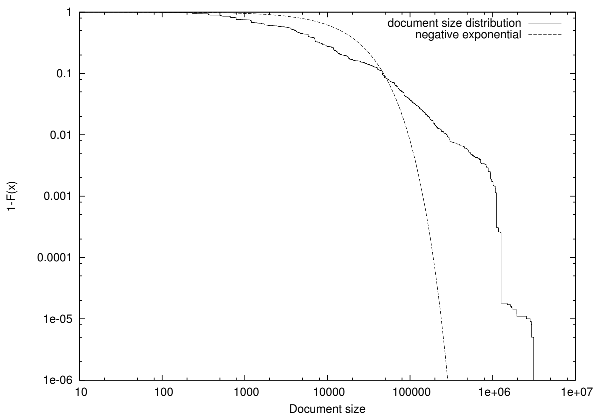

Fig. 7. Complementary object size distribution (NASA trace)

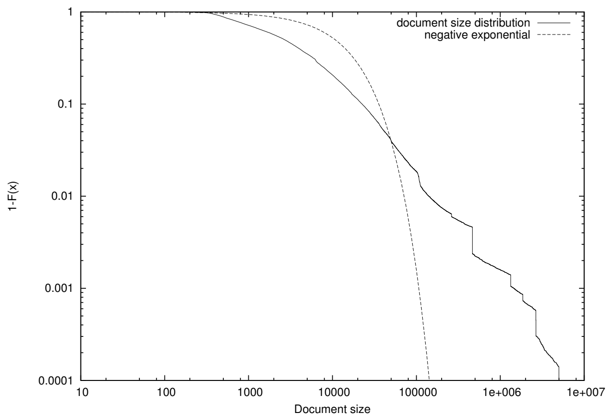

Fig. 8. Complementary object size distribution (IRCache trace)

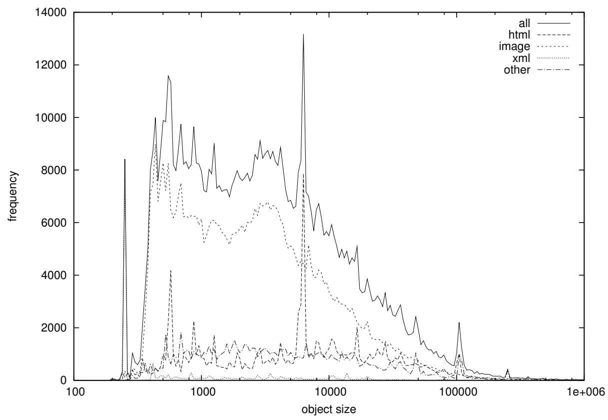

Fig. 9. Object size (bytes) frequencies by object type for the IRCache trace (logarithmic bin size)

3) IRCache trace: The IRCache trace was collected at the New York Squid proxy server of the IRCache system [31] from June 15th, 2007 to June 21th, 2007. We studied the sizes of the objects requested by the GET method via HTTP. Figure 8 shows the complementary distribution of the object sizes as log-log plot. The distribution function clearly is heavy-tailed.

Figure 9 shows the object size frequencies for the trace using logarithmic bin sizes. The correspondence between histogram and strata sizes and the dominance of image objects, as observed for the RWTH$^+$ trace, is also true for the IRCache trace (the strata sizes are shown in Figure 3). Interestingly, the more recent IRCache trace exhibits some differences to the RWTH$^+$ trace. Firstly, we observe a significant peak for objects of size 250. This is caused by XML-files, a file type which does not show any significant presence in the RWTH$^+$ trace. Secondly, there is a peak for HTML documents at 6000–7000 bytes. These documents are mostly Google search result pages. Finally, the most important difference is the higher fraction of objects of other types in the IRCache trace. The dominant type in this category is Javascript: About 6.6% of all requests in the IRCache trace concern Javascript files. In the RWTH$^+$ trace, that proportion was only 1.1%.

4) Weibull trace: The Weibull trace has been artificially generated using a Weibull distribution with mean 10000 and SCV 10. Figure 10 shows the complementary distribution of the Weibull distribution as log-log plot.

B. Matching HTDs

We have fitted different distributions to the data traces. For the empirical traces, a Weibull and Pareto distribution has been fitted by matching the first and second moment of the distribution. HEDs with 5, 10 and 20 phases have been fitted as well, however, the HEDs with five phases only provided satisfying results for the NASA trace. For the other three traces, the HEDs with five phases yield worse results than the larger HEDs and hence are not discussed in the following. The resulting HEDs are denoted as EM:H_5, EM:H_{10}, EM:H_{20} for the EM-algorithm, respectively, sEM:H_5, sEM:H_{10}, and sEM:H_{20} for the sEM-algorithm. The statistics of the fitted distributions are discussed in the following. We also discuss the speed-ups obtained by sEM in comparison to EM; that information is summarized in Table II.

1) RWTH$^+$ trace: The results for the RWTH$^+$ trace are shown in Table III. We have also included the statistics of the trace, the relative errors of the SCV and the skewness (always relative to the respective value of the trace), and the number of iterations required by EM and sEM to reach a fixed precision. We observe that the Weibull and the Pareto distribution fail to match the skewness (where available) and

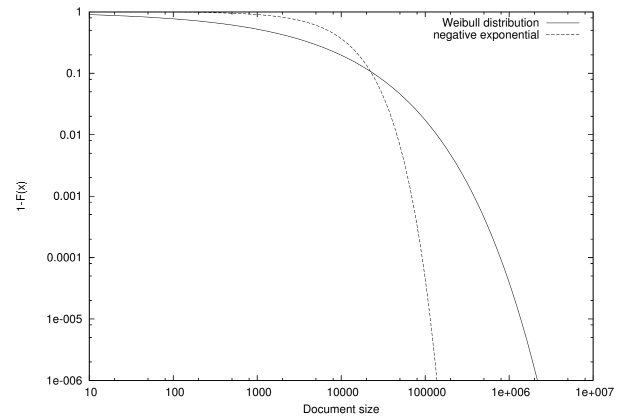

Fig. 10. Complementary Weibull distribution

TABLE II
RUN-TIME CHARACTERISTICS OF THE EM AND SEM-ALGORITHM

trace	size EM	iterations	size sEM	iterations	speed-up
RWTH$^+$	$20.7 \cdot 10^6$	4	123	4	168571
NASA	$3.1 \cdot 10^6$	78	83	78	37340
IRCache	$0.74 \cdot 10^6$	4	107	4	6916
Weibull	10^6	107	1259	109	780

TABLE III
COMPARISON OF THE STATISTICS OF THE MEASUREMENT DATA AND FITTED DISTRIBUTIONS (RWTH$^+$ TRACE)

measure	trace	Weibull	Pareto	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
E$[X]$	12303.2	12303.2	12303.2	12303.2	12303.2	12303.2	12303.2
SCV	316.2	316.2	316.2	328.4	328.4	328.1	328.1
rel. error	–	0%	0%	3.9%	3.9%	3.8%	3.8%
skewness	153.8	10.46	n.d.	188.8	188.8	188.7	188.7
rel. error	–	-93.2%	n.d.	22.8%	22.8%	22.7%	22.7%
median	2416	84	8704	2437	2437	2438	2438
iterations	–	–	–	4	4	4	4

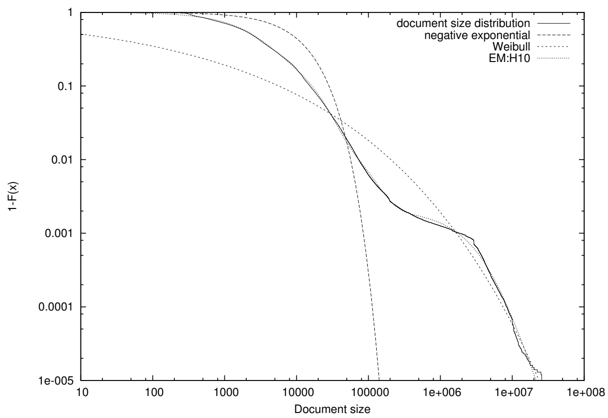

Fig. 11. Complementary log-log plot of the empirical distribution and three fitted distributions (RWTH$^+$ trace)

the median of the data trace. The HEDs match the SCV and the median very well but show a larger error for the skewness. This illustrates the fact that EM-based algorithms try to find an optimal *global* solution instead of focusing on specific moments of the distribution. No significant difference can be observed between the HEDs with 10 phases and those with 20 phases. Table III furthermore shows that EM and sEM yield nearly identical results with the same number of iterations. The fact that sEM has to process less data per iteration (123 entries instead of $20.7 \cdot 10^6$ in the original trace) results in a speed-up of approximately 168000. However, this does not include the time to sort the data, as required by sEM.

Figure 11 illustrates the accuracy of the fitted HEDs. It shows the empirical document size distribution and the fitted exponential, Weibull, and, respectively, EM:H_{10} distribution. The HED almost completely overlaps the empirical distribution, whereas the Weibull distribution only fits the end of the tail. Note that HEDs always have a completely monotone probability density function and, hence, they are not able to model well the *head* of the histogram of the object sizes (see Figure 6). However, although a clear difference between the HED and the empirical distribution is visible for very small objects in Figure 12 (which presents the same as Figure 11, however just for object sizes up to 10000 bytes), the effect of a heavy-tailed distribution, for example when used in queueing processes, is mostly governed by its waist and tail. Consequently, all fitted HEDs provide very convincing results in Section V-C.

2) NASA trace: The results for the NASA trace are shown in Table IV. The Weibull and the Pareto distribution fail to match the skewness and the median. Concerning the HEDs, we observe similar results as for the RWTH$^+$ trace: a good fit of the SCV but a larger error for the skewness. Again, no significant differences can be observed between the fitted HEDs. The sEM-algorithm has to process 83 entries per iteration ($3.1 \cdot 10^6$ in the original trace) which results in a speed-up of approximately 3700.

3) IRCache trace: The results for the IRCache trace are shown in Table V. Again, the Weibull and the Pareto distribution fail to match the skewness and the median. The results for the HEDs are very good. This time we also observe a very good fit of the skewness. For the HEDs with 10 and 20 phases, no significant differences can be observed between the HEDs. The sEM-algorithm has to process 107 entries per iteration ($0.74 \cdot 10^6$ in the original trace) which results in a speed-up of about 6900.

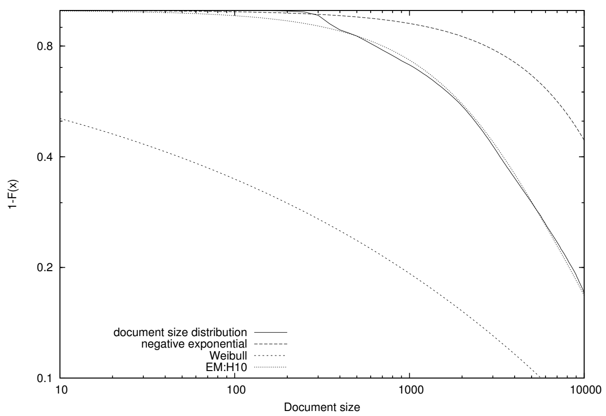

Fig. 12. Complementary log-log plot of the empirical distribution and three fitted distributions for small object sizes (RWTH$^+$ trace)

TABLE IV

COMPARISON OF THE STATISTICS OF THE MEASUREMENT DATA AND FITTED DISTRIBUTIONS (NASA TRACE)

measure	trace	Weibull	Pareto	EM:H_5	EM:H_{10}	sEM:H_5	sEM:H_{10}
E$[X]$	20744.9	20744.9	20744.9	20744.9	20744.9	20744.9	20744.9
SCV	13.39	13.39	13.39	14.43	14.43	14.42	14.42
rel. error	–	0%	0%	7.8%	7.8%	7.7%	7.7%
skewness	10.64	14.60	n.d.	12.73	12.73	12.73	12.73
rel. error	–	37.2%	n.d.	19.6%	19.6%	19.6%	19.6%
median	4142	1762	15008	3718	3718	3720	3720
iterations	–	–	–	78	78	78	78

TABLE V

COMPARISON OF THE STATISTICS OF THE MEASUREMENT DATA AND FITTED DISTRIBUTIONS (IRCACHE TRACE)

measure	trace	Weibull	Pareto	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
E$[X]$	13094.8	13094.8	13094.8	13094.8	13094.8	13094.8	13094.8
SCV	200.1	200.1	200.1	194.6	194.6	194.2	194.2
rel. error	–	0%	0%	-2.7%	-2.7%	-2.9%	-2.9%
skewness	115.1	6.96	n.d.	115.2	115.2	114.7	114.7
rel. error	–	-94.0%	n.d.	0.1%	0.1%	0.3%	0.3%
median	2710	174	9267	2740	2740	2741	2741
iterations	–	–	–	4	4	4	4

4) Weibull trace: The results for the Weibull trace are shown in Table VI. The fitted HEDs match very well the statistics of the trace. Only for the skewness, we observe better results for the HEDs fitted by the sEM-algorithm. The sEM-algorithm has to process 1259 elements per iteration. This, in comparison with the other two traces, large number is caused by the artificial nature of the trace. Unlike a true measurement trace, it contains many unique values over a wide range. The speed-up is approximately 780.

C. Embedding HTDs in queueing models

We have used the fitted distributions as service-time distribution in an M|G|1|100 queue. We have studied the mean E$[N]$ and the squared coefficient of variation c_N^2 of the queue length distribution for two different offered loads (0.7 and 0.9). For the measurement data and for the Weibull distribution, the results were computed using a trace-driven and a stochastic discrete-event simulation. For the fitted HEDs, the results were obtained by numerical analysis of the M|HED|1|100 queue using the FiFiQueues tool [32]–[34]. The Pareto distribution has not been evaluated here due to the obvious mismatch to the data traces, as discussed in the previous section.

1) RWTH$^+$ trace: The results for the RWTH$^+$ trace are shown in Table VII. All HEDs provide good (identical) results, whereas the Weibull distribution yields quite large errors.

TABLE VI

COMPARISON OF THE STATISTICS OF THE MEASUREMENT DATA AND FITTED DISTRIBUTIONS (WEIBULL TRACE)

measure	trace	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
E$[X]$	10000	10000	10000	10000	10000
SCV	10	9.99	9.99	10	10
rel. error	–	-0.1%	-0.1%	0%	0%
skewness	11.36	10.19	10.19	11.08	11.08
rel. error	–	-10.3%	-10.3%	-2.5%	-2.5%
median	1185	1164	1164	1141	1141
iterations	–	107	107	109	109

TABLE VII

QUEUE LENGTH MEAN AND SCV (RWTH$^+$ TRACE)

	measure	trace	Weibull	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
load=0.7	E$[N]$	24.22	27.67	25.12	25.12	25.12	25.12
	rel. error	–	14.2%	3.7%	3.7%	3.7%	3.7%
	c_N^2	2.41	1.78	2.39	2.39	2.39	2.39
	rel. error	–	-26.1%	-0.8%	-0.8%	-0.8%	-0.8%
load=0.9	E$[N]$	32.16	38.28	33.18	33.18	33.18	33.18
	rel. error	–	19.0%	3.2%	3.2%	3.2%	3.2%
	c_N^2	1.66	1.11	1.60	1.60	1.60	1.60
	rel. error	–	-33.1%	-3.6%	-3.6%	-3.6%	-3.6%

TABLE VIII

QUEUE LENGTH MEAN AND SCV (NASA TRACE)

	measure	trace	Weibull	EM:H_5	EM:H_{10}	sEM:H_5	sEM:H_{10}
load=0.7	E$[N]$	12.02	11.44	12.48	12.48	12.48	12.48
	rel. error	–	-4.8%	3.8%	3.8%	3.8%	3.8%
	c_N^2	2.20	2.34	2.32	2.32	2.32	2.32
	rel. error	–	6.4%	5.5%	5.5%	5.5%	5.5%
load=0.9	E$[N]$	32.57	31.16	32.23	32.23	32.23	32.23
	rel. error	–	-4.3%	-1.0%	-1.0%	-1.0%	-1.0%
	c_N^2	0.87	0.89	0.91	0.91	0.91	0.91
	rel. error	–	2.3%	4.6%	4.6%	4.6%	4.6%

2) NASA trace: Table VIII shows the results for the NASA trace. All distributions provide good results. Since the HEDs do not match the SCV of the NASA trace very well, larger (but still small) errors can be observed for them in comparison to the RWTH$^+$ trace.

3) IRCache trace: The results for the IRCache trace are shown in Table IX. All results provided by the HEDs are satisfying whereas the Weibull distribution yields large errors.

4) Weibull trace: The results for the Weibull trace are shown in Table X. All HEDs provide accurate (and nearly identical) results.

TABLE IX

QUEUE LENGTH MEAN AND SCV (IRCACHE TRACE)

	measure	trace	Weibull	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
load=0.7	E$[N]$	17.76	26.08	18.64	18.64	18.64	18.64
	rel. error	–	46.8%	5.0%	5.0%	5.0%	5.0%
	c_N^2	3.07	1.81	3.03	3.03	3.03	3.03
	rel. error	–	-41.0%	-1.3%	-1.3%	-1.3%	-1.3%
load=0.9	E$[N]$	27.62	37.61	28.75	28.75	28.75	28.75
	rel. error	–	36.2%	4.1%	4.1%	4.1%	4.1%
	c_N^2	1.68	1.06	1.64	1.64	1.64	1.64
	rel. error	–	-36.9%	-2.4%	-2.4%	-2.4%	-2.4%

TABLE X

QUEUE LENGTH MEAN AND SCV (WEIBULL TRACE)

	measure	trace	EM:H_{10}	EM:H_{20}	sEM:H_{10}	sEM:H_{20}
load=0.7	E$[N]$	9.40	9.52	9.52	9.49	9.49
	rel. error	–	1.3%	1.3%	1.0%	1.0%
	c_N^2	2.39	2.33	2.33	2.41	2.41
	rel. error	–	-2.5%	-2.5%	0.8%	0.8%
load=0.9	E$[N]$	29.27	29.64	29.64	29.34	29.34
	rel. error	–	1.3%	1.3%	0.2%	0.2%
	c_N^2	0.93	0.92	0.92	0.93	0.93
	rel. error	–	-1.1%	-1.1%	0.0%	0.0%

VI. Conclusion

In [15], [16], an algorithm for fitting HEDs to heavy-tailed distributed measurement data was presented. The approach was based on the EM-algorithm and the thus-obtained HEDs matched the first and second moment as well as higher moments and shape characteristics of the original distribution very well.

Here, we have presented a *major* extension of the algorithm that applies sampling and stratification techniques to the data in order to increase the efficiency of the method. We have applied the new method to various large empirical HTTP traces and observed that it yields equally good results, while exhibiting a speed-up of several orders of magnitude. We used both traces that were used in the past, as well as a very recent new trace (June 2007). We have furthermore shown that the extension is, in fact, a generalization of the aggregation algorithm presented in [10] when applied to heavy-tail distributed measurement data; but it has the advantage to dynamically adapt some of its parameters that were assumed fixed in [10]. For the future, we plan to apply the new method on non-heavy-tail distributed data and with other target, i.e., non-hyper-exponential, distributions.

Acknowledgment

We thank Duane Wessels for the access to the IRCache traces.

References

[1] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. ACM SIGCOMM '93*, ser. Computer Communications Review, vol. 23, Oct. 1993, pp. 183–193.

[2] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[3] V. Paxson and S. Floid, "Wide area traffic: The failure of Poisson modelling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[4] S. Asmussen and O. Nerman, "Fitting Phase-Type Distributions via the EM Algorithm," in *Symposium i Anvendt Statistik*, 1991, pp. 335–346.

[5] A. Horvath and M. Telek, "Approximating Heavy-Tailed Behaviour with Phase-Type Distributions," in *Advances in Algorithmic Methods for Stochastic Models*, G. Latouche and P. Taylor, Eds. Notable Publications, Inc., 2000, pp. 191–213.

[6] R. El Abdouni Khayari, R. Sadre, and B. Haverkort, "A validation of the pseudo self-similar traffic model." in *2002 International Conference on Dependable Systems and Networks (DSN 2002)*. IEEE Computer Society, 2002, pp. 727–734.

[7] R. El Abdouni Khayari, R. Sadre, B. Haverkort, and A. Ost, "The pseudo-self-similar traffic model: application and validation," *Performance Evaluation*, vol. 56, no. 1-4, pp. 3–22, 2004.

[8] A. Feldmann and W. Whitt, "Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models," *Performance Evaluation*, vol. 31, pp. 245–258, 1998.

[9] B. Friis, "Modelling Long-Range Dependent and Heavy-Tailed Phenomena by Matrix Analytic Methods," in *Advances in Algorithmic Methods for Stochastic Models*, G. Latouche and P. Taylor, Eds. Notable Publications, Inc., 2000, pp. 265–278.

[10] A. Panchenko and A. Thümmler, "Efficient phase-type fitting with aggregated traffic traces," *Performance Evaluation*, vol. 64, no. 7–8, pp. 629–645, August 2007.

[11] A. Riska, V. Diev, and E. Smirni, "Efficient fitting of long-tailed data sets into hyperexponential distributions," *Internet Performance Symposium, IEEE GlobeCom 2002*, vol. 3, pp. 2513–2517, 2002.

[12] ——, "An EM-based technique for approximating long-tailed data sets with PH distributions," *Performance Evaluation*, vol. 55, no. 1–2, pp. 147–164, 2004.

[13] A. Thümmler, P. Buchholz, and M. Telek, "A Novel Approach for Phase-Type Fitting with the EM Algorithm," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, pp. 245–258, 2005.

[14] P. Buchholz, "An EM-algorithm for MAP fitting from real traffic data," in *Proc. 13th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, vol. 2794 of LNCS, 2003, p. 218/236.

[15] R. El Abdouni Khayari, R. Sadre, and B. Haverkort, "Fitting World-Wide Web request traces with the EM-algorithm," in *Proc. of SPIE 4523 (Internet Performance and Control of Network Systems)*, 2001, pp. 211–220.

[16] ——, "Fitting World-Wide Web request traces with the EM-algorithm," *Performance Evaluation*, vol. 52, no. 2-3, pp. 175–191, 2003.

[17] ——, "A class-based least-recently used caching algorithm for WWW proxies," in *Proceedings of the 2th Polish-German Teletraffic Symposium, Gdansk, Poland*, September 2002.

[18] R. El Abdouni Khayari, R. Sadre, B. Haverkort, and N. Zoschke, "Weighted fair queueing scheduling for World Wide Web proxy servers," in *Proc. of SPIE 4865 (Internet Performance and Control of Network Systems III)*, 2002, pp. 120–131.

[19] B. Haverkort, R. El Abdouni Khayari, and R. Sadre, "A class-based least-recently used caching algorithm for World-Wide Web proxies," in *Computer Performance Evaluations, Modelling Techniques and Tools. 13th International Conference, TOOLS 2003*, ser. Lecture Notes in Computer Science, vol. 2794. Springer, 2003, pp. 273–290.

[20] A. Thümmler, P. Buchholz, and M. Telek, "A novel approach for fitting probability distributions to real trace data with the EM algorithm," in *International Conference on Dependable Systems and Networks, 2005. DSN 2005. Proceedings*. IEEE CS Press 2005, 2005, pp. 712–721.

[21] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM Press, 2002, pp. 159–171.

[22] ——, "Estimating flow distributions from sampled flow statistics," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 933–946, 2005.

[23] H. Ozmutlu, A. Spink, and S. Ozmutlu, "Analysis of large data logs: an application of poisson sampling on excite web queries," *Information Processing and Management*, vol. 38, no. 4, pp. 473–490, 2002.

[24] J. Judge, H. W. P. Beadle, and J. Chicharo, "Sampling HTTP response packets for prediction of Web traffic volume statistics," in *Proceedings of the Globecom'98 conference*, 1998.

[25] J. Dahmen, K. Beulen, and H. Ney, "A Mixture Density Based Approach for Object Recognition for Image Retrieval," *6th International RIAO Conference on Content-Based Multimedia Information Access*, pp. 1632–1647, 2000.

[26] T. Ryden, "Statistical Estimation for Markov-modulated Poisson Processes and Markovian Arrival Processes," in *Advances in Algorithmic Methods for Stochastic Models*, G. Latouche and P. Taylor, Eds. Notable Publications, Inc., 2000, pp. 329–350.

[27] E. Schukat-Talamazzini, *Automatische Spracherkennung — Grundlagen, Statistische Modelle und Effiziente Algorithmen*. Friedrich Vieweg & Sons, 1995.

[28] P. Levy and S. Lemeshow, *Sampling of Populations: Methods and Applications*. Wiley, 1999.

[29] R. Sadre, "Decomposition-based analysis of queueing networks," Ph.D. dissertation, University of Twente, 2006.

[30] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 631–645, 1997.

[31] "IRCache www.ircache.net," 2007.

[32] R. Sadre and B. Haverkort, "FiFiQueues: fixed-point analysis of queueing networks with finite-buffer stations," in *MMB (Kurzvorträge)*, vol. 99-16. Universität Trier, 1999, pp. 77–80.

[33] ——, "FiFiQueues: fixed-point analysis of queueing networks with finite-buffer stations," in *Computer Performance Evaluation. Modelling Techniques and Tools: 11th International Conference, TOOLS 2000*, ser. Lecture Notes in Computer Science, vol. 1786. Springer, 2000, pp. 324–327.

[34] R. Sadre, B. Haverkort, and A. Ost, "An efficient and accurate decomposition method for open finite- and infinite-buffer queueing networks," in *Proc. 3rd Int. Workshop on Numerical Solution of Markov Chains*, W. Stewart and B. Plateau, Eds. Zaragosa University Press, 1999, pp. 1–20.