Application-oriented Evaluation of Measurement Estimation

Adam Wierzbicki[1] and Lars Burgstahler[2]

[1] Polish-Japanese Institute of Information Technology
Chair of Computer Networks, Warsaw, Poland
adamw@icm.edu.pl
[2] Institute for Communication Networks and Computer Engineering
University of Stuttgart, Germany
burgstahler@ikr.uni-stuttgart.de

Abstract. The design and operation of telecommunication networks often requires measurement-based decisions. Examples of such decisions are QoS-based routing, where the measurement of the utilized link capacity influences the path selection, or cache location that uses measurements of average TCP throughput. However, the variability of measurements of network conditions makes it difficult to use them for such decisions. In QoS routing, decisions need to be stable in order to avoid packet reordering; in cache location, a small change in the measurements can lead to a completely different location decision. Therefore, estimation algorithms are used to smooth the measurements. The quality of decisions based on estimations depends on how close the estimation is to real conditions, and on how variable it is. There is a trade-off between these two objectives that makes it difficult to choose an appropriate estimation method. In this paper, an approach that uses multi-criteria analysis to evaluate the quality of an estimation algorithm is introduced. It is shown how different applications require different criteria and how the preferences of these criteria can be set by the algorithm designer. Some example estimation algorithms are evaluated on synthetic and real traffic traces using the proposed approach.

1 Introduction

Measurements of network conditions can be of great use in many areas of telecommunications. For some functionality, such as congestion control, such measurements are essential. On the other hand, it is not always possible to use the measurements without any further processing (or estimation). The TCP protocol relies on measurements, yet some of the most important improvements to the protocol were made when better methods of measurement estimation were utilized. TCP congestion control has a single objective: to adapt as fast as possible to the available bandwidth. In other applications, the use of measurements can have more than one objective.

An example application is QoS-based routing, e.g. in a DiffServ network where routers measure the available capacity on the links, possibly separately for each service class. With this information the router then can select e.g. the shortest path for high priority real time traffic and the widest path — which could be longer — for low priority bulk traffic (i.e. different routing algorithms would be used). An alternative usage of the information is load balancing for increased utilization of a network if different paths are available between a source-destination node pair.

However, load-based routing should not lead to frequent switching between different paths. Otherwise, packets would arrive out-of-order and provoke retransmission, decreasing thereby the performance of the network on the transport layer. Hence, the router needs to use an estimation that has two objectives: efficient path selection and routing stability.

Another example is the location of Internet caches and design of Content Delivery Networks (CDN). These locations can be planned in order to reduce the average delays of users. To do so, network management measures the average available TCP throughput on the paths from the clients to the potential cache locations and from there to the origin servers [1]. However, available TCP throughput is extremely variable, and the location decisions are very sensitive to small changes in the input data. Again, it is important to stabilize the measurements before they can be used to make decisions.

An estimation algorithm can simplify decision making by reducing the variability of the measurements. However, the quality of such a decision depends strongly on the quality of the estimation. There can be different requirements that estimation algorithms must meet, and they can sometimes be contradictory. Stability and accuracy are examples of such requirements. A stable estimation makes decisions very easy, but the estimates may be far too inaccurate to allow for a correct decision.

For applications that have different (and contradictory) requirements that the estimation algorithms must meet, we propose to use *multi-criteria analysis* to evaluate the quality of the algorithms. Multi-criteria analysis allows for an objective evaluation of different estimation algorithms that will take into account all diverse application requirements. As different applications (e.g., cache location, bandwidth measurement for routing) have different requirements, different criteria of algorithm performance (e.g., stability, accuracy, delay) can be selected. The criteria can be combined in a way that takes into account the preferences of the algorithm designer. The subject of this paper is the development of evaluation methods for the design of estimation algorithms that must satisfy diverse requirements, and the use of these to evaluate estimation algorithms on traces of network conditions (real and synthetic).

The remainder of this paper is organized as follows: Section 2 gives an overview on some estimation algorithms we have evaluated so far. Section 3 focuses on multi-criteria approaches for the evaluation of these algorithms. Section 5 describes the scenario we used for the evaluation, i.e. the traffic and the parameterization of the algorithms. Further, the selected criteria for two different applications are presented. In Sect. 6 we show and discuss the results and Sect. 7 gives a conclusion and a further outlook.

2 Estimation algorithms

Although estimation algorithms change the shape of the measurement, the main purpose is to adapt the measurement to the needs of the application, i.e. calculating reasonable and usable values or to forecast values. The result of an estimation is one single new value y_t based on a series of measured values x_t and sometimes older estimates y_{t-i}. In contrast, smoothing only focuses on the form of a curve and not the meaning of the measurement. A smoothing algorithm modifies the data set to make it smooth and nearly continuous and to remove or to diminish outlying points. Estimation and smoothing usually add extra delay to the measurement.

2.1 Median filter

The *median* is the value in the center of an ordered list. Usually the median is used as a *running median*, where a window of the size $L = 2N+1$ (N is the order of the median) is shifted continuously on the measurement. The values within the window are ordered and the $(N+1)^{st}$ value is the median. It is assigned to the time when the value in the middle of the window was measured. Since the median can only be determined, when the last value of the window is measured, a delay of N samples is always introduced.

A median filter has three important characteristics: First, for a given N, all peaks with a length $l_p \leq N$ are completely removed. Peaks with a larger length will not be altered, not even by multiple application of the same filter [2]. Secondly, median filters preserve discontinuities, provided the discontinuity exceeds some minimum duration. This minimum duration refers directly to the first characteristic, i.e., it has to last for at least $N+1$ samples. Thirdly, median filters follow polynomial curves rather tight. This is important when polynomial (e.g., linear) smoothing is applied before the estimation [3].

2.2 Exponential moving average

The general *exponential moving average (EMA)* of order N is described as follows:

$$y_t = \alpha_0 \cdot y_{t-N} + \alpha_1 \cdot y_{t-N+1} \cdots + \alpha_{N-1} \cdot x_t, \quad \text{with} \sum_{k=0}^{N} \alpha_k = 1 \tag{1}$$

However, in most cases the most basic form, a 1^{st}-order EMA, is used. The formula then is simplified to $y_t = (1-\alpha) \cdot y_{t-1} + \alpha \cdot x_t$. EMAs behave like a low pass, i.e., they filter high frequencies. In our case, a high frequency corresponds to a high variability within a short time. The responsiveness of the EMA is controlled by the weights $\alpha_k \, (0 \leq k < N)$. In the simple case of a 1^{st}-order EMA, a high α leads to a fast reaction to changing measurement, but the estimation follows the measurement too close. A low α leads to a more stable behavior but also to a larger delay and thus the accuracy of the estimation gets worse.

The main disadvantage of the basic EMA is its disability to adapt to a changed characteristic of the measured values. However, there are a number of dynamic EMAs where the weight is calculated dynamically depending on e.g., the time distance between the measured values or the change of the values themselves. An overview of dynamic EMAs can be found in [4].

2.3 Discrete intervals with hysteresis

Any estimation filter can be stabilized further by a simple estimation algorithm that frequently reduces the standard deviation. The algorithm divides the range of possible measurement results into *intervals* (not necessarily of the same size). In its simplest form, the algorithm replaces the estimate with the midpoint of the interval in which the median lies. Therefore, the width of the intervals determines the stability and error of the algorithm: by taking a wide enough interval (from min. to max. measurement), the estimate could be reduced to a straight line (a constant estimate). The desired width of the interval has to be determined by observation of the values of the estimate.

However, the described simple form of the algorithm is not sufficient. If the estimate lies close to the border of two intervals, even very small variations can change the interval and thus the value of the estimate. In such a case, the algorithm would increase the standard deviation. To avoid this behavior, *hysteresis* can be used. To change the interval, the estimate has to cross the border by a sufficient value. The value can be specified by a proportion of the width of the interval. For example, if the hysteresis parameter is equal to 0.3, and the estimate increases, the estimate has to exceed the lower border of an interval of width h by at least $0.3h$ in order to change the interval. A similar rule is applied when the estimate decreases.

Discrete intervals with hysteresis can be used for every estimation algorithm. For our evaluations, we used this combination for a median filter (*m9*) as well as for exponential moving averages with a weight $\alpha = 0.3$ and $\alpha = 0.7$ (*ema3DF* and *ema7DF*).

2.4 Smoothing

In our evaluation *hanning* was used as proposed for signal processing in [3]. An apodization function is used on a window of size L to calculate the smoothed value which is associated to the time in the middle of the window, thus the function adds a delay of N samples ($L = 2N + 1$). The apodization function for the hanning window is described by

$$\mathrm{Hn}(i) = \frac{1}{2}\left[1 - \cos\left(\frac{2i\pi}{L-1}\right)\right], \quad \text{with } 0 \leq i < L-1 \tag{2}$$

and therefore the smoothed value m_t for the time t results to

$$m_t = \sum_{k=0}^{L} \mathrm{Hn}(k) \cdot x_k \tag{3}$$

where x_k is the k^{th} measured value in the smoothing window.

2.5 Filter chains

The concatenation of a smoothing and an evaluation algorithm can already be considered as a *filter chain*. Sometimes it can also be useful to chain several estimation algorithms. As described in [2], applying the same median filter multiple times on a measurement can help to remove variabilities completely up to a certain degree. Such a stable estimate is called a *root* to this filter. By chaining different filters, their different characteristics can be combined.

3 Evaluation methods

The two main desirable characteristics of the estimations of measured data are: *stability* and a *good fit*. The reason for using estimations is to decrease the variability of the original measurement. On the other hand, the least variable estimate is a constant estimate, which is also unsatisfactory. Therefore, it is required that an estimate should fit the original measurement fairly well.

As was mentioned before, there exists a tradeoff between these two objectives. However, the two characteristics cannot be measured on a common scale. It is entirely unclear how much of the fit can be sacrificed to decrease the variability by some amount. To answer these questions, it would be useful to have a scale of comparison of the two characteristics. Yet, if one estimation has a better fit than another, but a worse variability, which of the two should be chosen, and consequently, which estimation algorithm is better?

These questions have been the subject of *multi-criteria decision analysis and support*. This area of operations research deals with the process of decision making and optimization in the case when there are many conflicting and incomparable objectives (called *criteria*). The methods of ranking outcomes (in our case, estimations) that have been developed by multi-criteria decision analysis could therefore be used for the evaluation of estimation algorithms that require stability and good fit.

Note, that the described evaluation methods are not to be used in a running system (e.g. a router). They evaluate the quality of an estimation algorithm off using traffic traces. Whether the estimation algorithm itself influences the performance of the system (e.g. in a router) or not (location of the CDN) is out of the scope of this paper. However, if the complexity of an estimation algorithm is important (e.g. for a router), it can be used as an additional characteristic. Yet we will not discuss this characteristic in the following.

3.1 Criteria for evaluation of estimation algorithms

The first step of multi-criteria decision analysis is to identify the criteria that should be used to evaluate outcomes. Below, seven criteria for the evaluation of an estimate will be introduced. Each criterion will be expressed by a formula that uses the following notation: x_t is the measurement made at time t; y_t is the estimate available at time t (not using any measurement that has been made later than t, which means, that the estimate is generally delayed); N is the number of measurements; $\overline{x}$, $\overline{y}$ are the mean values of the measurement and the estimate, and s_x, s_y are the standard deviations of the measurement and the estimate respectively.

Consider first the variability of an estimate. The simplest (and most intuitive) criterion of this characteristic is

1. the *standard deviation* of the estimate, s_y.

The quality of the fit of an estimate is harder to evaluate, since there is no single intuitive candidate for a measure of this characteristic. Several different criteria could be used:

2. the *mean absolute error* (MAE) of the estimate:

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t}} |\mathrm{y_t} - \mathrm{x_t}|. \tag{4}$$

3. the R^2 *measure* of the estimate:

$$R^2 = 1 - \frac{\sum_t (y_t - x_t)^2}{\sum_t (x_t - \bar{x})^2} \quad (\text{and } 0 \text{ if } s_x = 0). \tag{5}$$

4. the *correlation coefficient* (CC) of the estimate and the measurement:

$$\mathrm{CC} = \frac{1}{\mathrm{N} \cdot \mathrm{s_x} \cdot \mathrm{s_y}} \sum_{\mathrm{t}} (\mathrm{y_t} - \overline{\mathrm{y}})(\mathrm{x_t} - \overline{\mathrm{x}}) \tag{6}$$

5. the Pearson correlation coefficient (PCC) of the estimate and the measurement

All of these criteria are related to the “global” fit of the estimate to the measurement. However, in practical applications, it is often more important how well, in the worst case, the estimate performs as a forecast of the measurement, than how well it fits on the average. To express this type of fit, a measure of forecast quality of the estimate could be used. Such measures take as their parameters the desired length of a forecast. Since we can update estimates (and forecasts) continuously by taking new measurements, the length of the desired forecast is not dictated by a measurement period, but should be determined by the application. As an example, let the desired length of the forecast be three observation periods, $h = 3$. Since the application requires that the forecast be good even in the worst case, and since the measures of forecast quality are minimized, the criterion should be the maximum of a forecast quality measure. Two such measures have been taken into consideration, both independent of the scale of the measurement and estimate:

6. the *maximum of the mean absolute percentage error of the forecast* (max. MAPEF), taken over any time period where the forecast can be compared to the measurement:

$$\mathrm{MAPEF_{max}} = \mathrm{max_t} \frac{1}{(\mathrm{h}+1)} \sum_{\mathrm{i=t}}^{\mathrm{t+h}} |\frac{\mathrm{y_t} - \mathrm{x_t}}{\mathrm{x_t}}|. \tag{7}$$

7. the *maximum of the Theil inequality coefficient* (max. TIC) of the forecast, taken over any time period where the forecast can be compared to the measurement:

$$\mathrm{TIC_{max}} = \mathrm{max_t} \frac{\sqrt{\sum_{\mathrm{i=t}}^{\mathrm{t+h}} (\mathrm{y_t} - \mathrm{x_t})^2}}{\sqrt{\sum_{\mathrm{i=t}}^{\mathrm{t+h}} \mathrm{x_t^2}} + \sqrt{\sum_{\mathrm{i=t}}^{\mathrm{t+h}} \mathrm{y_t^2}}}. \tag{8}$$

Of these seven criteria, only one is used for the evaluation of estimate variability, while all others are different measures of estimate fit. To evaluate an estimation algorithm, not all criteria have to be used. Instead we can limit ourselves to two criteria, one of variability and the other one of fit.

3.2 Choosing good estimations using multi-criteria methods

Given a set of estimation algorithms, a measurement, and a set of criteria Q_i for evaluation, it should be possible to select algorithms that produce good estimations. The concept of *Pareto-optimality* (in multi-criteria decision analysis; game theory uses the term *Nash equilibrium*) can be used for that purpose. For simplification, let all criteria be minimized. An estimation y' is Pareto-optimal if no other estimation y'' has values of all criteria smaller than the values of the same criteria for estimation y'. Several of the estimations for a given measurement can satisfy this condition. However, an estimate that is not Pareto-optimal need no longer be taken into consideration.

To choose one estimation from the Pareto-optimal estimations it is necessary to use an *objective function* (OF) that combines all criteria. Such an objective function must express the preferences of the algorithm designer, who should be able to specify what levels of the criteria he requires. To do so, the range of variability of any criterion should be known. This is also a requirement for scaling the criteria, so that they may be aggregated without bias.

To establish a common scale of comparison of all criteria, it is necessary to find "best" and "worst" values of any criterion. These values can be estimated in the following way: the "best" value of the standard deviation of an estimate is clearly 0 (achieved by a constant estimate). The "worst" value is the standard deviation of the original measurement. For the measures of fit, the situation differs. The "best" value is usually 0 (with the exception of R^2, hence, $1 - R^2$ will be used instead). The "worst" value can be 1 for some of the measures, such as the correlation coefficient or the Theil inequality coefficient, or a larger value. For the other fit criteria, the "worst" value can be obtained by considering a constant estimate equal to the mean or median of the measurement. Let the "best" and "worst" values of the criteria be denoted by Q_i^u and Q_i^n, respectively, where $i \in 1,\ldots,k$, and k is the number of all criteria (in our case, 6). Recall that for minimized criteria, $Q_i^u \leq Q_i^n$.

The algorithm designer can specify for each criterion a level that would satisfy him completely (an aspiration level, Q_i^a), and a level that should be achieved at least (a reservation level, Q_i^r). For minimized criteria, $Q_i^u \leq Q_i^a < Q_i^r \leq Q_i^n$ such an approach follows the multi-criteria methodology called the *reference-point method* [5]. These levels could be specified in absolute terms or in terms of relative distance between the "best" and "worst" values. Once the levels are specified, each of the criteria can be scaled using a so-called *achievement function*:

$$\sigma_i = \begin{cases} 1 + \alpha \cdot \frac{Q_i^a - Q_i}{Q_i^a - Q_i^u}, & Q_i^u \leq Q_i < Q_i^a \\ \frac{Q_i^r - Q_i}{Q_i^r - Q_i^a}, & Q_i^a \leq Q_i < Q_i^r \\ \beta \cdot \frac{Q_i^r - Q_i}{Q_i^n - Q_i^r}, & Q_i^r \leq Q_i \leq Q_i^n \end{cases} \tag{9}$$

The achievement function is piecewise linear, and the coefficients α and β can be obtained from the reservation and aspiration levels. The slope of the achievement function in the interval $[Q_i^a, Q_i^r]$ is known (it is equal to $\frac{1}{|Q_i^a - Q_i^r|}$). β can be chosen to be twice that slope, and α to be half the slope.

Now, the objective function of the reference-point method can be expressed using the achievement function:

$$Q^{RP} = \min_i \sigma_i(Q_i, Q_i^a, Q_i^r) + \epsilon \sum_i \sigma_i(Q_i, Q_i^a, Q_i^r) \tag{10}$$

The parameter ϵ is a small, positive number, usually 0.01. If ϵ was too large, the reference-point method would become similar to a simple sum. The value of the objective function (10) can be negative (since the achievement functions can be negative). By changing the parameters Q_i^a and Q_i^r, any solution can be chosen as the optimal solution — in other words, the objective function is sufficiently flexible to express any preferences of the algorithm designer.

4 Preferences of algorithm designers

Two applications have been considered for the evaluation of estimation algorithms. The first is the use of estimations for the design of CDNs, which requires stable estimations of available throughput. The second is the use of estimations for QoS-based routing with the purpose of load-balancing. This application requires more accurate estimations than the first one.

The preferences of the algorithm designer for the first application value stability over accuracy. However, it is clearly unreasonable to use an algorithm that provides inaccurate estimates, if the accuracy can be improved without sacrificing stability. The first criterion is standard deviation: $Q_1 = s_y$. For the expression of accuracy, the algorithm designer chose $Q_2 = 1 - \text{CC}$. The aspiration and reservation levels were chosen with respect to the relative distance between the "best" and "worst" values. Let the distance between the "best" and "worst" values for criterion i be $\delta_i = |Q_i^n - Q_i^u|$. The aspiration and reservation levels for Q_1 were chosen as $Q_1^a = Q_1^u + 0.2\delta_1$, and $Q_1^r = Q_1^u + 0.95\delta_i$. The aspiration and reservation levels for Q_2 were chosen as $Q_2^a = Q_2^u + 0.5\delta_2$, and $Q_2^r = Q_2^u + 0.99\delta_2$.

The preferences of the algorithm designer for the second application value accuracy over stability. The first criterion is standard deviation: $Q_1 = s_y$. For the expression of accuracy, the algorithm designer chose $Q_2 = 1 - \mathrm{CC}$ and $Q_3 = \mathrm{MAPEF}_{\max}$. The aspiration and reservation level was also chosen with respect to "best" and "worst" values. For Q_1, they were chosen as $Q_1^a = Q_1^u + 0.8\delta_1$ and $Q_1^r = Q_1^u + 0.99\delta_1$. For Q_2, they were chosen as $Q_2^a = Q_2^u + 0.1\delta_2$ and $Q_2^r = Q_2^u + 0.25\delta_2$. For Q_3, they were chosen as $Q_3^a = Q_3^u + 0.2\delta_3$ and $Q_3^r = Q_3^u + 0.3\delta_3$.

The selection of appropriate aspiration and reservation levels by the algorithm designers was an iterative process. The algorithm designers started with some initial values, then explored the available solutions through the ranking produced by the objective function, and next could modify their values and iterate. During this process, the algorithm designers were able to learn about the expression of their preferences using aspiration and reservation levels.

5 Time series for evaluation

5.1 The synthetic traces

The synthetic traffic was generated with the help of the IKR simulation library (IKRSimLib). A variable number of traffic generators was used to generate self-similar traffic for different traces. The link's bandwidth was limited so that overload was possible. The burst length was Pareto-distributed ($\alpha = 1.6$, min. burst length 3750 byte). The resulting traffic was measured during 4000sec with a granularity of 1 sample per 5 seconds (i.e., there are 800 samples). The trace corresponds to what a router has to process to calculate link state information based on by-passing traffic. The traces will be referred to as *synthetic trace* synt−1 (6 generators), synt−2 (8 generators) and synt−3 (16 generators).

Table 1. Parameterization of the filter chains

Name	Type	Parameters		
		EMA	Hanning	Median
emaX	EMA	$\alpha = 0.X$		
emaXm3	EMA/Median	$\alpha = 0.X$		1^{st}-order
emaXDF	EMA, Disc. Intervals w. Hysteresis	$\alpha = 0.X$		
m9	Median			4^{th}-order
m9DF	Median, Disc. Intervals w. Hysteresis			4^{th}-order
hXmY	Hanning/Median		$w = X$	$(\frac{Y-1}{2})$-order
h3m3m3	Hanning/Median/Median		$w = 3$	1^{st}-order
m3h3m3	Median/Hanning/Median		$w = 3$	1^{st}-order
c	Mean	of whole measurement sample		

5.2 Traces of measurements of Internet conditions

The real traffic traces are measurements made by the NIMI infrastructure [6] (courtesy of Vern Paxson and Andrew Adams). The traces contained measurements of the available TCP throughput, which is measured using the treno tool [7]. The measurements were made daily during March and April, 2001. Due to some irregularities in the measurement time, the traces do not contain measurements made at an exact point in the day; however, the measurements were chosen in such a way that all measurements were made within working hours (using local time at the starting point of the measurement). The traces will be referred to by the start-, and end-point of the measurement, e.g. *Grouse-Cern*.

5.3 Evaluated estimation algorithms

To evaluate the estimation algorithms described in Sect. 2, different combinations were used on all of the traces. Table 1 shows the parameterization of the different filter chains.

6 Results

The estimation algorithms were evaluated on all available traces using two sets of preferences for the two applications described above. Naturally, the outcomes of the two evaluations were different; this difference may be used to illustrate the tradeoffs introduced by certain algorithms. First, the results of the evaluation of estimation algorithms for the first application (CDN design) will be shown.

The first stage of multi-criteria analysis is the selection of Pareto-optimal solutions. This process can be demonstrated on the example of estimations obtained by the considered algorithms on one of the traces (for example, synt$-$3). Figure 1 shows all estimations on a plane with Q_1 plotted on the x axis and Q_2 on the y axis. Of the 14 estimations, only 5 are Pareto-optimal: c, $ema3DF$, $ema3$, $ema7DF$ and $ema7$.

Another way of evaluating the estimations is to rank them using the objective function. Such a ranking can be shown for our example on Fig. 2. On the figure, the standard deviation increases from right to left. It can be seen that a reduction of the standard deviation always increases the objective function, under the chosen preferences.

This method has the advantage that it chooses one Pareto-optimal solution (the best in the ranking) out of all. The objective function of the reference point method always chooses a Pareto-optimal solution as the best one; however, it is possible, that non-Pareto-optimal solutions rank higher than Pareto-optimal ones. On the figure, it can be seen that the objective function indeed selects the Pareto-optimal solution c as best, but the third solution in the ranking, $m9DF$, is not Pareto-optimal. Such a situation occurred infrequently on the evaluated traces (for 10 traces, it occurred in 3 cases), but it is necessary to remove such suboptimal solutions from the ranking. By such a procedure it is possible to obtain a ranking of Pareto-optimal solutions, which for our example is shown on Table 2.

From the ranking it can be seen that the constant estimate c, equal to the mean of all values in the measurement sample, was always chosen as the best estimation. This can be explained by the preferences of the algorithm designer, who valued stability over accuracy. The constant estimate has a standard deviation of 0 (equal to q_1^U) and a correlation of 0 that results in $Q_2 = 1$ (halfway between q_2^U and Q_2^N). To beat it under the specified preferences, a more accurate estimation algorithm would have to be 100% accurate ($Q_2 = 0$) and have a standard deviation that is at least twice smaller than the standard deviation of the measurement, which is impossible. However, the constant estimate is not a practical estimation algorithm, since it would take very long to obtain the necessary data; also, it is not robust to trends or structural changes in the measurement. This would have been apparent if the algorithm designer would have chosen a different criterion of accuracy, for example MAE.

The ranking also shows that the second-best choice is usually an algorithm that was combined with the discrete intervals with hysteresis. The reason for this can be explained on Fig. 1. The result of combining the algorithms $ema3$ and $ema7$ is a reduction of the standard deviation at the expense of Q_2. This is not always the case (sometimes the standard deviation increases, if the resulting estimation consistently over-, or underestimates the measurement), but usually the discrete intervals with hysteresis allow to trade off accuracy for stability, which was preferred by the algorithm designer.

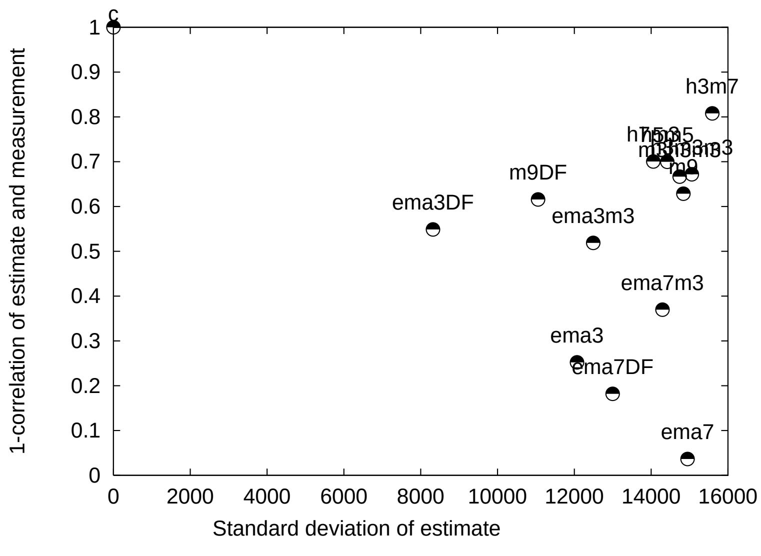

Fig. 1. Selection of Pareto optimal solutions using two criteria

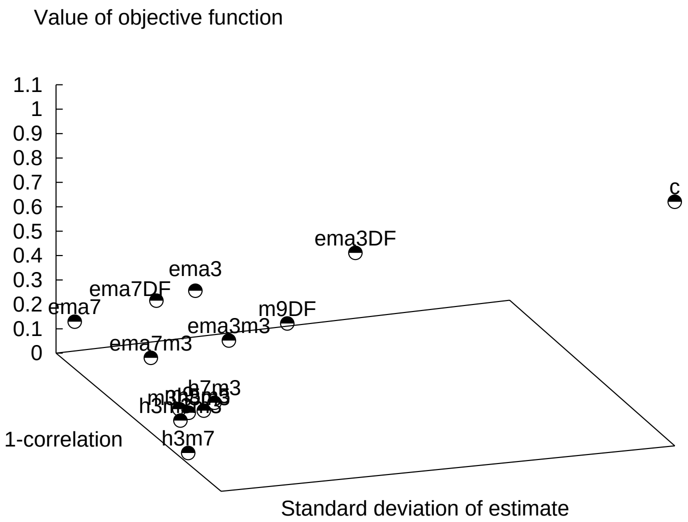

Fig. 2. Selection solutions using the objective function

The evaluation of estimation algorithms for the second application (routing) used one more criterion ($\mathrm{MAPE}_{\mathrm{max}}$) and different aspiration and reservation values that represented preferences that valued accuracy more than stability. The ranking of results using the objective function was produced in the same way as for the first application.

The results shown in Table 2 are quite different than for the first application. In all cases, the $ema7$ algorithm scored first; for the first application, this algorithm produced too variable estimations, and never appeared in the first three positions of the ranking. For most other traces, the $ema7_{DF}$ algorithm is second, and $ema3$ is third. $ema3$ is on third place of the ranking for the first application for three traces (it was in the first five positions of the ranking for most traces). On the other hand, the constant estimate c appears only once in the first three positions of the ranking for the second application. Clearly, this algorithm is too insensitive for an application that requires accuracy.

Estimation using discrete intervals clearly decreases the performance of the estimations for the second application, which supports the conclusion that this algorithm trades off accuracy for stability.

7 Conclusion

In this paper, we first presented an evaluation approach for estimation algorithms that uses multi-criteria analysis. The approach was used on two examples with different requirements of estimation algorithms. It was shown how multi-criteria evaluation can be adapted to the preferences of an algorithm designer, and how the results of the analysis can be used to obtain insight into algorithm operation.

Table 2. Overview of the three best estimation algorithms for each trace (OF-based)

	CDN				Routing				
Measure-ment	Estimation algorithm	s_y	1-CC	Value of OF	Estimation algorithm	s_y	1-CC	max MAPE	Value of OF
synt-1	c	0	1.00	1.00	$ema7$	19683	0.04	0.42	1.0000
	$h3m7$	12385	0.94	0.55	$ema7m3$	18047	0.38	1.21	-0.3000
	$ema3m3$	13999	0.53	0.46	c	0	1.00	1.04	-0.3200
synt-2	c	0	1.00	1.00	$ema7$	19982	0.04	0.33	1.0000
	$h3m7$	13798	0.82	0.49	$ema7DF$	18960	0.06	0.38	-0.0200
	$ema3DF$	14327	0.33	0.46	$ema7m3$	18218	0.38	0.74	-0.7100
synt-3	c	0	0.00	1.00	$ema7$	14948	0.04	0.11	1.0100
	$ema3DF$	8320	0.55	0.63	$ema7DF$	12998	0.18	0.17	0.0036
	$ema3$	12069	0.25	0.34	$ema7m3$	14295	0.37	0.35	0.0010
info to tahoe	c	0	1.00	1.00	$ema7$	475	0.03	4.41	0.0024
	$m9DF$	0	1.00	1.00	$ema7DF$	491	0.04	7.43	0.0023
	$ema3m3$	234	0.71	0.72	$ema3$	276	0.23	10.97	0.0018
grouse to cern	c	0	1.00	1.00	$ema7$	1980	0.03	1.21	0.0028
	$ema3DF$	1647	0.31	0.27	$ema7DF$	1859	0.06	1.39	0.0023
	$ema3$	1658	0.22	0.27	$ema3$	1658	0.22	4.81	0.0015
grouse to nimi1	c	0	1.00	1.00	$ema7$	1365	0.06	2.02	0.0023
	$m9DF$	0	1.00	1.00	$ema7DF$	1505	0.16	1.85	0.0018
	$ema3$	1174	0.47	0.32	$ema3$	1174	0.47	3.57	0.0003
grouse to tahoe	c	0	1.00	1.00	$ema7$	2000	0.04	0.49	0.0028
	$m9DF$	959	0.53	0.70	$ema7DF$	2138	0.11	0.65	0.0024
	$ema3DF$	1720	0.32	0.25	$ema3$	2026	0.22	0.97	0.0017
grouse to tracer	c	0	1.00	1.00	$ema7$	158	0.13	0.66	0.0027
	$m9DF$	0	1.00	1.00	$ema3$	203	0.74	1.49	-0.1500
	$ema3DF$	135	1.10	0.36	$ema7DF$	235	0.40	0.69	-0.1800
grouse to stanford	c	0	1.00	1.00	$ema7$	1231	0.05	1.02	0.0026
	$m9DF$	493	1.04	0.84	$ema7DF$	1187	0.10	1.26	0.0023
	$ema3DF$	806	0.49	0.57	$ema3$	840	0.34	1.87	0.0010
grouse to pendragon	c	0	1.00	1.00	$ema7$	1200	0.09	1.02	0.0023
	$m9DF$	0	1.00	1.00	$ema7DF$	788	0.11	1.22	0.0023
	$ema7DF$	788	0.11	0.56	$ema3$	1390	0.49	2.11	0.0002

The question of the best algorithm for either of the two applications remains open, since we have not evaluated a sufficient number of algorithms, and it is possible to use more advanced estimation techniques. However, the chosen evaluation methodology should be useful for a further exploration of this topic. Our approach allows also to preselect estimation algorithms that satisfy certain requirements.

The adaptation of the multi-criteria methodology for a particular application is made by the selection of appropriate aspiration and reservation levels. This process cannot be thought of as a single-step process, but should be regarded as an iterative procedure. The algorithm designer can learn how to express his preferences using aspiration and reservation levels, and he can modify his preferences over time, as he grows more acquainted with the tradeoffs and characteristics of the problem.

It is not always possible to specify the criteria for estimation algorithm evaluation using closed-form, analytical formulas, as in this paper. An example could be the algorithm that TCP uses for RTT estimation. It was designed with the single objective that the protocol should use only the available bandwidth. Introducing reduced delay as an additional objective could require another estimation algorithm. The criteria used to evaluate algorithm performance in this case would require a simulation of the TCP flow control algorithm that uses a particular estimation method. However, regardless of the method of criteria calculation, the evaluation scheme and the usage of the objective function remains the same as shown above.

References

1. Wierzbicki, A.: Models for internet cache location. In: 7th International Web Caching Workshop. (2002)
2. Gallagher, N.C., Wise, G.L.: A theoretical analysis of the properties of median filters. IEEE Transactions on Acoustics, Speech, and Signal Processing **ASSP-29** (1981) 1136–1141
3. Rabiner, L.R., Sambur, M.R., Schmidt, C.E.: Applications of a nonlinear smoothing algorithm to speech processing. IEEE Transactions on Acoustics, Speech, and Signal Processing **ASSP-23** (1975) 552–557
4. Burgstahler, L., Neubauer, M.: New modifications of the exponential moving average algorithm for bandwidth estimation. In: Proceedings of the 15^{th} ITC Specialist Seminar, Würzburg, Germany. (2002) 210–219
5. Wierzbicki, A.P., Makowski, M., Wessels, J., eds.: Model-Based Decision Support Methodology with Environmental Applications. Kluwer Academic Publishers (2000)
6. National Laboratory for Applied Network Research (NLANR): National internet measurement infrastructure. http://www.ncne.nlanr.net/nimi/ (2002)
7. Pittsburgh Supercomputing Center, Carnegie Mellon University: About the psc treno server. http://www.psc.edu/networking/treno_info.html (2000)