# Tracking the Best Quantizer

András György, *Member, IEEE*, Tamás Linder, *Senior Member, IEEE*, and Gábor Lugosi, *Member, IEEE*

***Abstract*—An algorithm is presented for online prediction that allows to track the best expert efficiently even when the number of experts is exponentially large, provided that the set of experts has a certain additive structure. As an example, we work out the case where each expert is represented by a path in a directed graph and the loss of each expert is the sum of the weights over the edges in the path. These results are then used to construct universal limited-delay schemes for lossy coding of individual sequences. In particular, we consider the problem of tracking the best scalar quantizer that is adaptively matched to the source sequence with piecewise different behavior. A randomized algorithm is presented which can perform, on any source sequence, asymptotically as well as the best scalar quantization algorithm that is matched to the sequence and is allowed to change the employed quantizer for a given number of times. The complexity of the algorithm is quadratic in the sequence length, but at the price of some deterioration in performance, the complexity can be made linear. Analogous results are obtained for sequential multiresolution and multiple description scalar quantization of individual sequences.**

***Index Terms*—Algorithmic efficiency, individual sequences, lossy source coding, multiple description quantization, multiresolution coding, nonstationary sources, scalar quantization, sequential coding, sequential prediction.**

## I. Introduction

In this paper, we consider limited-delay lossy coding schemes for individual sequences. Our goal is to provide a universal coding method which can dynamically adapt to the changes in the source behavior, with particular emphasis on the situation where the behavior of the source can change a given number of times (which is a function of the sequence length). We concentrate on low-complexity methods that perform uniformly well with respect to a given reference coder class on every individual (deterministic) sequence. In this individual-sequence setting no probabilistic assumptions are made on the source sequence, which provides a natural model for situations where very little is known about the source to be encoded.

Consider the widely used model for fixed-rate lossy source coding at rate $R$ where an infinite sequence of real-valued source symbols $x_1, x_2, \ldots$ is transformed into a sequence of channel symbols $b_1, b_2, \ldots$ taking values from the finite channel alphabet $\{1, 2, \ldots, M\}$, $M = 2^R$. These channel symbols are losslessly transmitted and then used to produce the reproduction sequence $\hat{x}_1, \hat{x}_2, \ldots$. The scheme is said to have delay $\delta$ if the reproduction symbol $\hat{x}_n$ can be decoded at most $\delta$ time instants after $x_n$ was available at the encoder. A general model for this situation is that each channel symbol $b_n$ depends only on the source symbols $x_1, \ldots, x_{n+\delta}$, and the reproduction $\hat{x}_n$ for the source symbol $x_n$ depends only on the channel symbols $b_1, \ldots, b_n$. Thus, the encoder produces $b_n$ as soon as $x_{n+\delta}$ is available, and the decoder can produce $\hat{x}_n$ when $b_n$ is received.

The performance of a scheme is measured with respect to a reference class of coding schemes, and the goal is to perform, on any source sequence, asymptotically as well as the best scheme in the reference class. Thus, the performance is measured by the distortion redundancy defined as the maximum, over all source sequences of length $n$, of the difference of the normalized cumulative distortion of our scheme and the normalized cumulative distortion of the best scheme in the reference class.

In the initial study of zero-delay coding for individual sequences [1], the reference class was the class of all scalar quantizers, and a coding scheme was provided (using common randomization at the encoder and the decoder) whose distortion redundancy was $O(n^{-1/5} \log n)$ for bounded sequences of length $n$. The results in [1] were improved and generalized by Weissman and Merhav [2] who constructed schemes that can compete with any finite set of limited-delay finite-memory coding schemes without requiring that the decoder have access to the randomization sequence. The resulting scheme has distortion redundancy $O(n^{-1/3} \log^{2/3} N)$, where $N$ is the size of the reference coder class. To our knowledge, this is the best known redundancy bound for this problem. In the special case where the reference class is the (infinite) set of scalar quantizers, an $O(n^{-1/3} \log n)$ distortion redundancy can be achieved by approximating the reference class by an appropriately chosen finite set of quantizers. The coding schemes of [1] and [2] are based on the theory of prediction using expert advice. The basic theoretical results were established by Hannan [3] and Blackwell [4] in the 1950's and brought to the center of attention in learning theory in the 1990's by Vovk [5], Littlestone and Warmuth [6], Cesa-Bianchi *et al.* [7]; see also Cesa-Bianchi and Lugosi [8] for a comprehensive treatment. These results show that it is possible to construct algorithms for online prediction that predict an arbitrary sequence of outcomes almost as well as the best of $N$ experts in the sense that the cumulative loss of the predictor is at most as large as that of

Manuscript received September 15, 2006; revised November 8, 2007. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the NATO Science Fellowship of Canada, the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, the Hungarian Scientific Research Fund (OTKA F60787), the Mobile Innovation Center of Hungary, Spanish Ministry of Science and Technology Grant MTM2006–05650 and by the PASCAL Network of Excellence under EC Grant 506778. The material in this paper was presented in part at the 2005 IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005 and at the IEEE Conference on Decision and Control and European Control Conference, Sevilla, Spain, December 2005.

A. György is with the Machine Learning Research Group, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary, H-1111 (e-mail: gya@szit.bme.hu).

T. Linder is with the Department of Mathematics and Statistics, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: linder@mast.queensu.ca).

G. Lugosi is with ICREA and the Department of Economics, Pompeu Fabra University, 08005 Barcelona, Spain (e-mail: lugosi@upf.es).

Communicated by W. Szpankowski, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2008.917651

0018-9448/$25.00 © 2008 IEEE

the best expert plus a term proportional to $\sqrt{\ln N/n}$ for any bounded loss function, where $n$ is the number of rounds in the prediction game. The logarithmic dependence on the number of experts makes it possible to obtain meaningful bounds even when the pool of experts is very large.

Unfortunately, the basic prediction algorithms, such as the exponentially weighted average predictor, which was applied both in [1] and [2], have computational complexity that is proportional to the number of experts and are therefore infeasible when this number is very large. Thus, although the coding schemes of [1] and [2] have the attractive property of performing uniformly well on individual sequences, they are computationally inefficient. For example, for the reference class of scalar quantizers, these methods use about $n^{c\,2^R}$ quantizers as "experts" where $c = 1/5$ for the scheme in [1] and $c = 1/3$ for the scheme in [2] and $R$ is the rate of the scheme, resulting in a computational complexity that is polynomial in $n$ with degree that is proportional to $M = 2^R$. This complexity comes from the fact that, in order to approximate the performance of the best scalar quantizer, these methods have to calculate and store the cumulative distortion of each of the approximately $n^{c\,2^R}$ quantizers. Clearly, even for moderate values of the encoding rate, this complexity becomes prohibitive.

For more general finite reference classes, the method of [2] has to maintain a weight for each of the $N$ reference codes. This results in a computational complexity of order $nN$, which only allows the use of small reference classes. When the reference class is an infinite set of codes, the method is applied to a finite approximation of the reference class, which can result in a prohibitively large $N$ if the approximation is to be close.

Fortunately, in many applications the set of experts has a certain structure that may be exploited in the construction of efficient prediction algorithms. Examples of structured classes of experts for which efficient algorithms have been constructed include prunings of decision trees (Helmbold and Schapire [9], Pereira and Singer [10]), and planar decision graphs (Mohri [11] and Takimoto and Warmuth [12], [13]). These algorithms are all based on efficient implementations of the exponentially weighted average predictor. Using a similar approach which exploits the special structure of scalar quantizers, in [14] we provided an efficient implementation of the algorithm of [2] for the reference class of scalar quantizers. In this algorithm, the encoding complexity is reduced to $O(n^{4/3})$ while maintaining the $O(n^{-1/3}\log n)$ distortion redundancy. Moreover, the complexity can be made linear in the sequence length at the price of increasing the distortion redundancy to $O(n^{-1/4}\sqrt{\log n})$.

In the prediction context, a different approach was taken by Kalai and Vempala [15] who considered Hannan's original predictor [3] and showed that it may be used to obtain efficient algorithms for a large class of problems that they call "geometric experts." Based on this method, a zero-delay quantization algorithm with linear encoding complexity was given in [16] which is conceptually simpler than the coding method of [14], and has only a slightly larger distortion redundancy $O(n^{-1/4}\log n)$. Recently, Matloub and Weissman [17] extended the general coding scheme of [2] and the method of efficient implementation of [16] to zero-delay joint source–channel coding of individual sequences over stochastic channels by showing that, under general conditions on the channel noise, the problem can be traced back to the source coding problem by replacing the distortion measure with its expectation over the channel noise.

As suggested in [2], it is an interesting open problem to find an algorithm of low complexity that is able to approximate the performance of the best scheme from a larger reference class. However, it seems that to date no low-complexity algorithms have been devised which work for more powerful reference classes than the class of scalar quantizers.

In this paper, we consider a more general reference class in which each reference scheme partitions the input sequence into contiguous segments and for each segment a different delay-$\delta$ code from a finite base reference class $\mathcal{F}$ may be employed. If a combined scheme can change the applied code $m$ times for an input sequence of length $n$, then the number of such schemes is $\sum_{j=0}^{m}\binom{n}{j}|\mathcal{F}|(|\mathcal{F}|-1)^j$. If one has to maintain a weight for each reference code, the implementation is infeasible even for a very small $\mathcal{F}$, as the straightforward implementation requires $O((n+m)^m|\mathcal{F}|^m)$ computations. However, as we will show in this paper, the structure of the reference codes provides a possibility to overcome this problem.

Similar problems have been investigated earlier in the contexts of universal prediction and universal lossless compression of piecewise stationary memoryless sources. The latter problem has been studied by Willems [18], Shamir and Merhav [19], and Shamir and Costello [20]. The efficient sequential algorithms in these papers are based on a two-stage mixture procedure, where mixture estimates are used for the source parameters in each segment, as well as over the possible (or most likely) segmentations of the observed sequence.

The corresponding prediction problem for individual sequences, known as the problem of *tracking the best expert*, is perhaps the best known example of a structured reference class. In this problem, a small number of "base" experts is given and the goal of the predictor is to predict as well as the best "meta" expert that is formed by certain allowable sequences of base experts. A sequence is allowable if it consists of at most $m+1$ blocks such that in each block the meta expert predicts according to a fixed base expert. If there are $N$ base experts and the length of the prediction game is $n$, then the total number of meta experts is $\sum_{j=0}^{m}\binom{n}{j}N(N-1)^j$. For this problem, Herbster and Warmuth [21] exhibited computationally efficient algorithms that predict almost as well as the best of the meta experts and have regret bounds that depend on the *logarithm* of the number of the (meta) experts. See also Auer and Warmuth [22], Vovk [23], Bousquet and Warmuth [24], and Herbster and Warmuth [25] for various extensions and powerful variants of the problem. However, these methods become computationally too expensive if the "base" reference class is very large.

In this paper, we develop efficient algorithms to track the best expert in the case when the class of "base" experts is already very large, but has a certain structure. Thus, in a sense, we consider a combination of the two types of structured experts described above. Our approach is based on a suitable modification of the original tracking algorithm of Herbster and Warmuth [21] that can handle large, structured expert classes, and results in an implementation of the tracking predictor that uses a similar mixture over the possible segmentations as the algorithms of [18] and [19] (however, extending their methods in many directions). This modification is described in Section II. In Section III, we use the modified tracking predictor algorithm combined with the coding method of Weissman and Merhav [2] to obtain codes that can track any finite class of limited-delay finite-memory codes efficiently. The proposed method has computational complexity of order $n|\mathcal{F}|$, significantly less than the $O((n+m)^m|\mathcal{F}|^m)$ complexity of the algorithm in [2] when applied to this problem, and has basically the same distortion redundancy. In Section IV, we illustrate our new prediction method on a problem in which a base expert is associated with a path in a directed graph and the loss of a base expert is the sum of the weights over the path (which may change in every round of the prediction game). The special structure of the experts allows efficient implementation of tracking. This graph representation of the experts is used in Section V to obtain efficient coding algorithms to track the best scalar quantizer (i.e., to code asymptotically as well as the best combined coding scheme from scalar quantizers). Finally, in Section VI, we consider two network quantization versions of this problem: tracking the best multiple description scalar quantizer and tracking the best multiresolution scalar quantizer (among quantizers with interval cells). The encoding and decoding complexity of each of these algorithms can be made linear in the sequence length at the price of some performance deterioration.

We note here that all the algorithms we present are horizon-dependent; i.e., they depend on the length of the encoded sequence. However, it is a standard exercise (see, e.g., [8]) to convert such an algorithm into a horizon-independent one by applying it to blocks of inputs with exponentially increasing lengths. The resulting truly sequential scheme will perform essentially as well as the original algorithm.

## II. Tracking the Best Expert: A Variation

In this section, we present a modification of a prediction algorithm by Herbster and Warmuth [21] for tracking the best expert. This modification will facilitate efficient implementation if the number of experts is very large.

The online decision problem we consider is described as follows. Suppose we want to use a sequential decision scheme to make predictions concerning the outcomes of a sequence $y_1, y_2, \ldots$ taking values in a set $\mathcal{Y}$. We assume that the (randomized) predictor has access to a sequence of independent random variables $U_1, U_2, \ldots$ that are uniformly distributed over the interval $[0,1]$. At each time instant $t = 1, 2, \ldots$, the predictor observes $U_t$, and based on $U_t$ and the past input values $y^{t-1} = (y_1, \ldots, y_{t-1})$ produces an "action" $\hat{y}_t \in \hat{\mathcal{Y}}$, where $\hat{\mathcal{Y}}$ is the set of predictor actions that may not be the same as $\mathcal{Y}$. Then, the predictor can observe the next input symbol $y_t$ and calculate its loss $\ell(y_t, \hat{y}_t)$ with respect to some bounded loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to [0, B]$, where $B > 0$. Formally, the prediction game is defined in Fig. 1.

**Parameters:** number $N$ of base experts, outcome space $\mathcal{Y}$, action space $\hat{\mathcal{Y}}$, loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to [0, B]$, number $T$ of rounds.
For each round $t = 1, \ldots, T$,

(1) each (base) expert forms its prediction $\hat{y}_t^{(i)} \in \hat{\mathcal{Y}}$, $i = 1, \ldots, N$;

(2) the predictor observes the predictions of the base experts and the random variable $U_t$, and chooses an estimate $\hat{y}_t \in \hat{\mathcal{Y}}$;

(3) the environment reveals the next outcome $y_t \in \mathcal{Y}$.

Fig. 1. The prediction game.

The *cumulative loss* of the sequential scheme at time $T$ is given by

$$L_T = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t).$$

One of the most popular algorithms for the online prediction game described above is the *exponentially weighted average* predictor (see [5]–[7]) which is defined as follows: let $\eta > 0$ be a parameter and to each $i = 1, \ldots, N$ assign the initial weight $w_{1,i} = 1/N$. At time instants $t = 1, 2, \ldots, T$, let $v_t^{(i)} = w_{t,i}/W_t$ where $W_t = \sum_{i=1}^{N} w_{t,i}$ and predict $\hat{y}_t$ randomly according to the distribution $\mathbb{P}\{\hat{y}_t = \hat{y}_t^{(i)}\} = v_t^{(i)}$. After observing $y_t$, update the weights by $w_{t+1,i} = w_{t,i} e^{-\eta \ell(y_t, \hat{y}_t^{(i)})}$. This yields

$$v_t^{(i)} = \frac{e^{-\eta \sum_{t'=1}^{t-1} \ell(y_t, \hat{y}_t^{(i)})}}{\sum_{j=1}^{N} e^{-\eta \sum_{t'=1}^{t-1} \ell(y_t, \hat{y}_t^{(i)})}}$$

that is, $v_t^{(i)}$ is proportional to the "exponential" cumulative performance of expert $i$ up to time $t-1$. It is well known that the expected cumulative regret of the exponentially weighted average predictor may be bounded, for all possible sequences generated by the environment, by

$$\mathbb{E}\left(L_T - \min_{i=1,\ldots,N} \sum_{t=1}^{T} \ell\left(y_t, \hat{y}_t^{(i)}\right)\right) \le B\left(\frac{\ln N}{\eta} + \frac{n\eta}{8}\right)$$

where the expectation is understood with respect to the randomization sequence $U_1, \ldots, U_T$ of the predictor. In particular, if $\eta = B\sqrt{8 \ln N/T}$ is chosen to optimize the upper bound, then the bound becomes $B\sqrt{(T/2)\ln N}$. (For various versions and more discussion on the performance of this algorithm, we refer the reader to [8].)

The exponentially weighted average algorithm is thus guaranteed to perform, on the average, almost as well as the expert with the smallest cumulative loss. A more ambitious goal of the predictor is to achieve a cumulative loss (almost) as small as the best tracking of the $N$ base experts. More precisely, to describe the loss the predictor is compared to, consider the following "$m$-partition" prediction scheme: The sequence of examples $y_1, \ldots, y_T$ is partitioned into $m+1$ contiguous segments, and on each segment the scheme assigns exactly one of the $N$ base experts. Formally, an $m$-partition $\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})$ of the first $T$ samples is given by an $m$-tuple $\boldsymbol{t} = (t_1, \ldots, t_m)$ such that $t_0 = 0 < t_1 < \cdots < t_m < T = t_{m+1}$, and an $(m+1)$-vector $\boldsymbol{e} = (e_0, \ldots, e_m)$ where $e_i \in \{1, \ldots, N\}$. At

**Algorithm 1** *Fix the positive numbers $\eta$ and $\alpha < 1$, and initialize weights $w_{1,i}^s = 1/N$ for $i = 1, \ldots, N$. At time instants $t = 1, 2, \ldots, T$ let $v_t^{(i)} = w_{t,i}^s / W_t$ where $W_t = \sum_{i=1}^N w_{t,i}^s$, and predict $\hat{y}_t$ randomly according to the distribution*

$$\mathbb{P}\left\{\hat{y}_t = \hat{y}_t^{(i)}\right\} = v_t^{(i)}. \tag{1}$$

*After observing $y_t$,* for all $i = 1, \ldots, N$, *let*

$$w_{t,i}^m = w_{t,i}^s e^{-\eta \ell\left(y_t, \hat{y}_t^{(i)}\right)} \tag{2}$$

*and*

$$w_{t+1,i}^s = \frac{\alpha W_{t+1}}{N} + (1-\alpha) w_{t,i}^m \tag{3}$$

*where* $W_{t+1} = \sum_{i=1}^N w_{t,i}^m$.

Fig. 2. The modified fixed-share tracking algorithm.

each time instant $t$, $t_i < t \le t_{i+1}$, expert $e_i$ is used to predict $y_t$. The cumulative loss of a partition $\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})$ is

$$\begin{aligned} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) &= \sum_{i=0}^{m} \sum_{t=t_i+1}^{t_{i+1}} \ell\left(y_t, \hat{y}_t^{(e_i)}\right) \\ &= \sum_{i=0}^{m} L((t_i, t_{i+1}], e_i) \end{aligned}$$

where for any time interval $I$, $L(I, i) = \sum_{t \in I} \ell(y_t, \hat{y}_t^{(i)})$ denotes the cumulative loss of expert $i$ in $I$. Here and later in the paper we adopt the convention that in case the summation is over an empty index set, the sum is defined to be zero (e.g., for $a > b$, $L([a, b], i) = 0$).

The goal of the predictor is to perform nearly as well as the best partition, that is, to keep the normalized regret

$$\frac{1}{T}(L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})))$$

as small as possible (with high probability) for all possible outcome sequences. A slightly different goal is to keep the normalized expected regret

$$\frac{1}{T}\mathbb{E}(L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})))$$

as small as possible, where the expectation is taken with respect to the randomizing sequence $U^T = (U_1, \ldots, U_T)$.

Herbster and Warmuth [21] constructed a so-called "fixed-share" share update algorithm for the tracking prediction problem. We present a slightly modified version of this algorithm in Fig. 2. While this modification was also introduced by Bousquet and Warmuth [24], the performance bounds provided there are insufficient for our purposes.

Observe that $\sum_{i=1}^N w_{t+1,i}^s = \sum_{i=1}^N w_{t,i}^m = W_{t+1}$ in the algorithm; thus, there is no ambiguity in the definition of $W_{t+1}$. Note that (3) is slightly changed compared to the original algorithm of [21].

The following theorem bounds the loss of the algorithm. The proof follows the lines of the proof in [21] (with standard modifications necessary to handle nonconvex action spaces and hence randomized prediction), and therefore it is deferred to the Appendix.

*Theorem 1:* For all positive integers $m, T$ with $T \ge m + 1$, real numbers $0 < \alpha < 1$, $\eta > 0$, and $0 < p < 1$, and for any sequence $y_1, \ldots, y_T$ and loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to [0, B]$, with probability at least $1 - p$, the regret of Algorithm 1 can be bounded as

$$\begin{aligned} &L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) \\ &\le \frac{1}{\eta} \ln\left(\frac{N^{m+1}}{\alpha^m (1-\alpha)^{T-m-1}}\right) + \frac{T\eta B^2}{8} + B\sqrt{\frac{T \ln(1/p)}{2}} \end{aligned} \tag{4}$$

and the expected regret can be bounded as

$$\begin{aligned} &\mathbb{E}\left(L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e}))\right) \\ &\qquad \le \frac{1}{\eta} \ln\left(\frac{N^{m+1}}{\alpha^m (1-\alpha)^{T-m-1}}\right) + \frac{T\eta B^2}{8}. \end{aligned} \tag{5}$$

In particular, if $\alpha = \frac{m}{T-1} < 1$ and $\eta$ is chosen to minimize the above bound as

$$\eta = \sqrt{\frac{8 \ln\left(\frac{N^{m+1}}{\alpha^m (1-\alpha)^{T-m-1}}\right)}{TB^2}} \tag{6}$$

we have

$$\begin{aligned} &L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) \\ &\quad \le T^{1/2} \frac{B}{\sqrt{2}} \sqrt{(m+1)\ln N + m \ln \frac{T-1}{m} + m} \\ &\quad + B\sqrt{\frac{T \ln(1/p)}{2}} \end{aligned} \tag{7}$$

and

$$\begin{aligned} &\mathbb{E}\left(L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e}))\right) \\ &\quad \le T^{1/2} \frac{B}{\sqrt{2}} \sqrt{(m+1)\ln N + m \ln \frac{T-1}{m} + m}. \end{aligned} \tag{8}$$

*Remarks:*

i) If the number of experts $N$ is proportional to $T^\gamma$ for some $\gamma > 0$, then the bound in (8) is of order $\sqrt{(mT)}\ln T$, and so the normalized expected regret is

$$\frac{1}{T}\mathbb{E}(L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e}))) = O(\sqrt{(m/T)\ln T}).$$

That is, the rate of convergence is the same (up to a constant factor) as if we competed with the best static expert on a segment of average length $T/m$.

ii) Note that to achieve the optimal convergence rate above, the value of $\alpha$ has to be set based on the (*a priori* unknown) number of switches $m$. To avoid this problem, Vovk [23] gave an elegant solution using randomization over $\alpha$ that only causes a slight performance loss. Earlier, Willems [18] used a similar mixture method to estimate switching probabilities in the probabilistic setting. Although it would be possible to introduce such mixtures over $\alpha$ into our algorithms, for the sake of simplicity we only consider fixed values of $\alpha$ throughout the paper.

### A. *Implementation of Algorithm 1*

If the number of experts $N$ is large, for example, $N = T^\gamma$ for some large $\gamma > 1$, then the implementation of Algorithm 1 may become computationally prohibitive. The main message of this section is the nontrivial observation that if the standard exponentially weighted prediction algorithm can be efficiently implemented, then one can also efficiently implement Algorithm 1. The main step toward demonstrating this is the following alternative expression for the weights in Algorithm 1.

*Lemma 1:* For any $t = 2, \dots, T$, the probability $v_t^{(i)}$ and the corresponding normalization factor $W_t$ in Algorithm 1 can be obtained as

$$v_t^{(i)} = \frac{(1-\alpha)^{t-1}}{NW_t} e^{-\eta L([1,t-1],i)} + \frac{\alpha}{NW_t} \sum_{t'=2}^{t-1} (1-\alpha)^{t-t'} W_{t'} e^{-\eta L([t',t-1],i)} + \frac{\alpha}{N} \tag{9}$$

$$W_t = \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} Z_{t',t-1} + \frac{(1-\alpha)^{t-2}}{N} Z_{1,t-1} \tag{10}$$

where $Z_{t',t-1} = \sum_{i=1}^N e^{-\eta L([t',t-1],i)}$ is the sum of the (unnormalized) weights assigned to the experts by the exponentially weighted prediction method for the input samples $(y_{t'}, \dots, y_{t-1})$.

*Proof:* The expressions in the lemma follow directly from the recursive definition of the weights $\{w_{t,i}^s\}$. First we show that for $t = 1, \dots, T$

$$w_{t,i}^m = \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t-t'} W_{t'} e^{-\eta L([t',t],i)} + \frac{(1-\alpha)^{t-1}}{N} e^{-\eta L([1,t],i)} \tag{11}$$

$$w_{t+1,i}^s = \frac{\alpha}{N} W_{t+1} + \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t],i)} + \frac{(1-\alpha)^t}{N} e^{-\eta L([1,t],i)}. \tag{12}$$

Clearly, for a given $t$, (11) implies (12) by the definition (3). Since $w_{1,i}^s = 1/N$ for every expert $i$, (11) and (12) hold for $t = 1$ and $t = 2$ (for $t = 1$ the summations are 0 in both equations). Now assume that they hold for some $t \geq 2$. We show that then (11) holds for $t+1$. By definition

$$\begin{aligned} w_{t+1,i}^m &= w_{t+1,i}^s e^{-\eta \ell\left(y_{t+1}, \hat{y}_{t+1}^{(i)}\right)} \\ &= \frac{\alpha}{N} W_{t+1} e^{-\eta \ell\left(y_{t+1}, \hat{y}_{t+1}^{(i)}\right)} \\ &\quad + \frac{\alpha}{N} \sum_{t'=2}^{t} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t+1],i)} \\ &\quad + \frac{(1-\alpha)^t}{N} e^{-\eta L([1,t+1],i)} \\ &= \frac{\alpha}{N} \sum_{t'=2}^{t+1} (1-\alpha)^{t+1-t'} W_{t'} e^{-\eta L([t',t+1],i)} \\ &\quad + \frac{(1-\alpha)^t}{N} e^{-\eta L([1,t+1],i)} \end{aligned}$$

thus, (11) and (12) hold for all $t = 1, \dots, T$. Now (9) follows from (12) by normalization for $t = 2, \dots, T+1$. Finally, (10) can easily be proved from (11), as for any $t = 2, \dots, T$

$$\begin{aligned} W_t &= \sum_{i=1}^N w_{t-1,i}^m \\ &= \sum_{i=1}^N \left( \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} e^{-\eta L([t',t-1],i)} \right. \\ &\quad \left. + \frac{(1-\alpha)^{t-2}}{N} e^{-\eta L([1,t-1],i)} \right) \\ &= \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} \sum_{i=1}^N e^{-\eta L([t',t-1],i)} \\ &\quad + \frac{(1-\alpha)^{t-2}}{N} \sum_{i=1}^N e^{-\eta L([1,t-1],i)} \\ &= \frac{\alpha}{N} \sum_{t'=2}^{t-1} (1-\alpha)^{t-1-t'} W_{t'} Z_{t',t-1} \\ &\quad + \frac{(1-\alpha)^{t-2}}{N} Z_{1,t-1}. \end{aligned}$$

□

Examining formula (9), one can see that the $t'$th term in the summation (including the first and last individual terms for $t' = 1$ and $t' = t$, respectively) is some multiple of $e^{-\eta L([t',t-1],i)}$. Recall that the normalized version of $e^{-\eta L([t',t-1],i)}$ is the weight assigned to expert $i$ by the exponentially weighted prediction method for the last $t - t'$ input samples $(y_{t'}, \dots, y_{t-1})$ (the last term in the summation corresponds to the case where no previous samples of the sequence are taken into consideration). Therefore, for $t \geq 2$, the random choice of a predictor (1) can be performed in two steps. First, we choose a random time $\tau_t$, which specifies how many of the most recent samples we are going to use for the prediction. Then we choose the predictor according to the exponentially weighted prediction for these samples. Thus, $\mathbb{P}\{\tau_t = t'\}$ is the sum of the $t'$th terms with respect to the index $i$ in the expressions for $v_t^{(i)}$, and given $\tau_t = t'$, the probability that $\hat{y}_t = \hat{y}_t^{(i)}$ is just the probability assigned to expert $i$ using the exponentially weighted average prediction based on the

**Algorithm 2** *For* $t = 1$, *choose* $\hat{y}_1$ *uniformly from the set* $\{\hat{y}_1^{(1)}, \dots, \hat{y}_1^{(N)}\}$. *For* $t \geq 2$, *choose* $\tau_t$ *randomly according to the distribution*

$$\mathbb{P}\{\tau_t = t'\} = \begin{cases} \frac{(1-\alpha)^{t-1} Z_{1,t-1}}{NW_t}, & \text{for } t' = 1 \\ \frac{\alpha(1-\alpha)^{t-t'} W_{t'} Z_{t',t-1}}{NW_t}, & \text{for } t' = 2, \dots, t \end{cases} \tag{13}$$

*where we define* $Z_{t,t-1} = N$. *Given* $\tau_t = t'$, *choose* $\hat{y}_t$ *randomly according to the conditional probabilities*

$$\mathbb{P}\left\{ \hat{y}_t = \hat{y}_t^{(i)} \,\middle|\, \tau_t = t' \right\} = \begin{cases} \frac{e^{-\eta L([t',t-1],i)}}{Z_{t',t-1}}, & \text{for } t' = 1, \dots, t-1 \\ \frac{1}{N}, & \text{for } t' = t. \end{cases} \tag{14}$$

Fig. 3. Efficient implementation of the modified fixed-share tracking algorithm.

samples $(y_{t'}, \dots, y_{t-1})$. Hence, we obtain the following algorithm shown in Fig. 3.

We note here that the algorithm is somewhat similar to that of Willems [18]. His second, so-called "linear-complexity coding method" for the lossless compression of a probabilistic source with piecewise independent and identical distribution is a mixture code with $t$ component codes corresponding to the hypotheses that the last change in the source statistics occurred at time $t'$ for $t' = 1, \dots, t$. The conditional probability assigned for the $t$th sample by such a component code depends only on the last $t - t'$ samples of the source sequence, similarly to our Algorithm 2.

The discussion preceding Algorithm 2 shows that it provides an alternative implementation of Algorithm 1.

*Theorem 2:* Algorithms 1 and 2 are equivalent in the sense that the predictor sequences generated by the two randomized algorithms have the same distribution. In particular, the distribution of the sequence $(\hat{y}_1, \dots, \hat{y}_T)$ generated by Algorithm 2 satisfies $\mathbb{P}\{\hat{y}_1 = \hat{y}_1^{(i)}\} = v_1^{(i)}$ and

$$\mathbb{P}_{t-1}\{\hat{y}_t = \hat{y}_t^{(i)}\} = v_t^{(i)} \tag{15}$$

for all $t = 2, \dots, T$ and $i = 1, \dots, N$, where $\mathbb{P}_{t-1}$ denotes conditional probability given the input sequence $y_1, \dots, y_{t-1}$ and expert predictions $\{\hat{y}_1^{(i)}\}_{i=1}^N, \dots, \{\hat{y}_{t-1}^{(i)}\}_{i=1}^N$ up to time $t-1$, and the $v_t^{(i)}$ are the normalized weights generated by Algorithm 1.

In some special, but important, problems efficient algorithms are known to implement the exponentially weighted average prediction for the samples $(y_{t'}, \dots, y_{t-1})$ for any $t' < t$. Generally, as a byproduct, these algorithms can also compute the corresponding probabilities $\mathbb{P}\{\hat{y}_t = \hat{y}_t^{(i)} | \tau_t = t'\}$ and normalization factors $Z_{t',t-1}$ efficiently. Then $W_t$ can be obtained via the recursion formula (10), and so Algorithm 2 can be implemented efficiently.

In the following sections, we apply the prediction method of Algorithm 1 to obtain efficient adaptive quantization schemes.

## III. Tracking the Best Finite-Delay Finite-Memory Source Code

In this section, we consider the problem of coding an individual sequence with a fixed-rate limited-delay and finite-memory source coding scheme as defined by Weissman and Merhav [2]. Our goal is to construct an algorithm that performs as well as the best combined coding scheme which is allowed, several times during the coding procedure, to choose a new code from a finite reference class of limited-delay finite-memory source codes.

A fixed-rate delay-$\delta$ sequential source code of rate $R = \log M$ is defined by an encoder–decoder pair connected via a discrete noiseless channel of capacity $R$. (Here $\delta$ is a nonnegative integer, $M$ is a positive integer, and $\log$ denotes base-2 logarithm.) The input to the encoder is a sequence $x_1, x_2, \dots$ taking values in some source alphabet $\mathcal{X}$. At each time instant $i = 1, 2, \dots$, the encoder observes $x_i$ and based on the source sequence $x^{i+\delta} = (x_1, \dots, x_{i+\delta})$, the encoder produces a channel symbol $b_i \in \{1, 2, \dots, M\}$ which is then noiselessly transmitted to the decoder. After receiving $b_i$, the decoder outputs the reproduction $\hat{x}_i$ (taking value in a reproduction alphabet $\hat{\mathcal{X}}$) based on the channel symbols $b^i = (b_1, \dots, b_i)$ received so far.

Formally, the code is given by a sequence of encoder–decoder functions $(f, g) = \{f_i, g_i\}_{i=1}^{\infty}$, where

$$f_i : \mathcal{X}^{i+\delta} \to \{1, 2, \dots, M\}$$

and

$$g_i : \{1, 2, \dots, M\}^i \to \hat{\mathcal{X}}$$

so that $b_i = f_i(x^{i+\delta})$ and $\hat{x}_i = g_i(b^i)$, $i = 1, 2, \dots$. Note that the total delay of the encoding and decoding process is $\delta$. Although we require the decoder to operate with zero delay, this requirement introduces no loss in generality, as any finite-delay coding system with $\delta_1$ encoding and $\delta_2$ decoding delay can be equivalently represented in this way with $\delta_1 + \delta_2$ encoding and zero decoding delay [2].

The *normalized cumulative distortion* of the sequential scheme after reproducing the first $n$ symbols is given by

$$\frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

where $d : \mathcal{X} \times \hat{\mathcal{X}} \to [0, 1]$ is some distortion measure. (All results may be extended trivially to arbitrary bounded distortion measures.)

The decoder $\{g_i\}_{i=1}^{\infty}$ is said to be of finite memory $s \geq 0$ if $g_i(b^i) = g_i(\hat{b}^i)$ for all $i$ and $b^i, \hat{b}^i \in \{1, \dots, M\}^i$ such that $b_{i-s}^i = \hat{b}_{i-s}^i$, where $b_{i-s}^i = (b_{i-s}, b_{i-s+1}, \dots, b_i)$ and $\hat{b}_{i-s}^i = (\hat{b}_{i-s}, \hat{b}_{i-s+1}, \dots, \hat{b}_i)$. In order to emphasize that the output depends only on $b_{i-s}^i$, sometimes we will write $g_i(b_{i-s}^i)$

instead of $g_i(b^i)$ for such decoders. Let $\mathcal{F}^\delta$ denote the collection of all delay-$\delta$ sequential source codes of rate $R$, and let $\mathcal{F}_s^\delta$ denote the class of codes in $\mathcal{F}^\delta$ with memory $s$.[1]

Let $\mathcal{F} \subset \mathcal{F}_s^\delta$ be a finite class of reference codes. Our goal is to construct a delay-$\delta$ scheme which, for every sequence $x^n$, performs "nearly" as well as the best coding scheme that employs codes from $\mathcal{F}$ and is allowed to change the code $m$ times. Formally, a code in this class $\mathcal{F}_{m,n}$ against which our scheme competes is given by integers $1 \le i_1 < i_2 < \cdots < i_m < n$ and codes $\{(f_i^{(j)}, g_i^{(j)})\}_{i=1}^\infty$, $j = 0, \ldots, m$ such that $\{(f_i^{(j)}, g_i^{(j)}\}_{i=1}^\infty \in \mathcal{F}$ for all $j$, and $b_i = f_i^{(j)}(x^{i+\delta})$, $\hat{x}_i = g_i^{(j)}(b_{i-s}^i)$ for $i_j < i \le i_{j+1}$, where $i_0 = 0$ and $i_{m+1} = n$.

The minimum normalized cumulative distortion achievable by schemes in $\mathcal{F}_{m,n}$ for $n$ reproduction values is

$$D^*_{\mathcal{F},m,n}(\boldsymbol{x}) = \frac{1}{n} \min_{1\le i_1 < \cdots < i_m < n} \sum_{j=0}^{m} \min_{(f,g)\in\mathcal{F}} \sum_{i=i_j+1}^{i_{j+1}} d\Big(x_i, g_i(f_{i-s}(x^{i+\delta-s}), \ldots, f_i(x^{i+\delta}))\Big) \tag{16}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots)$ denotes the entire input sequence.

Note that the minimum distortion in (16) is calculated under the idealized assumption that at each time instant $i = i_j + 1, \ldots, i_{j+1}$ in the $j$th time segment, the decoder $g_i^{(j)}$ has access to the channel symbols $b_{i-s}^i$ generated by the $j$th code, so that it can output $\hat{x}_i = g_i(f_{i-s}^{(j)}(x^{i+\delta-s}), \ldots, f_i^{(j)}(x^{i+\delta}))$. However, in the real system, the channel symbols $b_{i_j-s}, \ldots, b_{i_j}$ are generated by code $j-1$ (i.e., $b_i = f_i^{(j-1)}(x^{i+\delta-s})$ for $i = i_j - s, \ldots, i_j$), and so the scheme cannot decode the first $s$ symbols at the beginning of the time segment. Since our goal is to compete with the best scheme in $\mathcal{F}_{m,n}$, the idealized definition of $D^*_{\mathcal{F},m,n}(\boldsymbol{x})$ is in fact a pessimistic assumption on our part.

In the rest of this section, we construct a general scheme for tracking a finite set of limited-delay finite-memory source codes. Low-complexity implementations for various scalar quantization scenarios will be discussed in the subsequent sections. The following general method is a combination of the coding scheme of Weissman and Merhav [2] and our modification of the prediction scheme of Herbster and Warmuth [21] described in Section II.

The scheme works as follows. Divide the source sequence $x^n$ into nonoverlapping blocks of length $l$ (for simplicity assume that $l$ divides $n$). At the beginning of the $k$th block, that is, at time instants $t = (k-1)l + 1$, $k = 1, \ldots, n/l$, a coding scheme $(f^{(k)}, g^{(k)}) = \{f_i^{(k)}, g_i^{(k)}\}_{i=1}^\infty$ is chosen randomly from the finite reference class $\mathcal{F} \subset \mathcal{F}_s^\delta$. The exact distribution for the random choice of $(f^{(k)}, g^{(k)})$ will be specified later based on the results in Section II (see (20) and (21)). The encoder uses the first $\lceil \frac{1}{R}\log|\mathcal{F}|\rceil$ time instants of the block to describe the selected coding scheme $(f^{(k)}, g^{(k)})$ to the receiver ($\lceil x \rceil$ denotes the smallest integer greater than or equal to $x$). More precisely, for time instants

$$i = (k-1)l + 1, \ldots, (k-1)l + \left\lceil \frac{1}{R}\log|\mathcal{F}| \right\rceil$$

an index uniquely identifying $(f^{(k)}, g^{(k)})$ is transmitted. In the rest of the block, that is, for time instants

$$i = (k-1)l + \left\lceil \frac{1}{R}\log|\mathcal{F}| \right\rceil + 1, \ldots, kl$$

the encoder uses $f_i^{(k)}$ to produce and transmit $b_i = f_i^{(k)}(x^{i+\delta})$ to the receiver. In the first

$$h = \left\lceil \frac{1}{R}\log|\mathcal{F}| \right\rceil + s$$

time instants of the $k$th block, that is, while the index of the coding scheme $(f^{(k)}, g^{(k)})$ is communicated and the first $s$ correct channel symbols are received, the decoder emits an arbitrary reproduction symbol $\hat{x}_i = \hat{x}$ with distortion at most

$$\hat{d} = \sup_{x\in\mathcal{X}} d(x, \hat{x}) \le 1.$$

In the remainder of the block, the decoder uses $g_i^{(k)}$ to decode the transmitted channel symbols as

$$\hat{x}_i = g_i^{(k)}(b^i) = g_i^{(k)}\left(b_{i-s}^i\right)$$

where $b_{i-s}^i = (b_{i-s}, b_{i-s+1}, \ldots, b_i)$ (recall that the decoder $g^{(k)}$ has finite memory $s$).

Now except for the distortion induced by communicating the quantizer index and the first $s$ correct code symbols at the beginning of each block, the above scheme can easily be fit in the sequential decision framework. We want to make a sequence of decisions concerning the sequence $\{y_k\}$ defined by $y_k = (x_{(k-1)l+h+1}, \ldots, x_{kl})$ for $k = 1, \ldots, n/l$. We consider any $(f,g) \in \mathcal{F}$ an expert whose prediction is

$$\hat{y}_k^{(f,g)} = (\hat{x}_{(k-1)l+h+1}^{(f,g)}, \ldots, \hat{x}_{kl}^{(f,g)})$$

where

$$\hat{x}_i^{(f,g)} = g_i(f_{i-s}(x^{i-s+\delta}), \ldots, f_i(x^{i+\delta})).$$

Thus, $(f,g)$ incurs loss $\ell(y_k, \hat{y}_k^{(f,g)})$, where the loss $\ell : \mathcal{X}^{l-h} \times \mathcal{X}^{l-h}$ is defined by

$$\ell(y, \hat{y}) = \sum_{j=1}^{l-h} d(x(j), \hat{x}(j)) \tag{17}$$

for any $y = (x(1), \ldots, x(l-h))$ and $\hat{y} = (\hat{x}(1), \ldots, \hat{x}(l-h))$. Then

$$\sum_{i=1}^{n} d(x_i, \hat{x}_i) \le \sum_{k=1}^{n/l} \ell\left(y_k, \hat{y}_k^{(f,g)}\right) + \frac{nh\hat{d}}{l} \tag{18}$$

where the second term comes from the fact that in each block the distortion at each of the first $h$ time instants is at most $\hat{d}$.

Using the notation of Section II, we have $N = |\mathcal{F}|$, $T = n/l$, and $B = l - h$. For any $(f,g) \in \mathcal{F}$ and all integers $1 \le k' \le k \le n/l$, let

$$L([k',k],(f,g)) = \sum_{j=k'}^{k} \sum_{i=(j-1)l+h+1}^{jl} d\left(x_i, \hat{x}_i^{(f,g)}\right). \tag{19}$$

Choose $\eta > 0$ and $0 < \alpha < 1$, and define

$$Z_{k',k} = \sum_{(f,g)\in\mathcal{F}} e^{-\eta L([k',k],(f,g))}.$$

[1] In [2], the codes in $\mathcal{F}^\delta$ and $\mathcal{F}_s^\delta$ were allowed to use randomization. Since the applications we consider in Sections V and VI are for nonrandomized reference classes, we use a slightly less general definition. However, all results in this section remain valid for randomized reference classes.

Let $W_1 = 1$, and for $k = 2, \ldots, n/l$

$$W_{k+1} = \frac{\alpha}{|\mathcal{F}|} \sum_{k'=2}^{k} (1-\alpha)^{k-k'} W_{k'} Z_{k',k} + \frac{(1-\alpha)^{k-1}}{|\mathcal{F}|} Z_{1,k}.$$

Finally, using $\{Z_{k',k}\}$ and $\{W_k\}$, define the probability distribution of $(f^{(k)}, g^{(k)})$, according to Algorithm 1, as

$$\mathbb{P}\{\tau_k = k'\} = \begin{cases} \frac{(1-\alpha)^{k-1} Z_{1,k-1}}{|\mathcal{F}| W_k}, & \text{for } k' = 1 \\ \frac{\alpha(1-\alpha)^{k-k'} W_{k'} Z_{k',k-1}}{|\mathcal{F}| W_k}, & \text{for } k' = 2, \ldots, k \end{cases} \tag{20}$$

and

$$\mathbb{P}\left\{\left(f^{(k)}, g^{(k)}\right) = (f,g) | \tau_k = k'\right\} = \begin{cases} \frac{e^{-\eta L([k',k-1],(f,g))}}{Z_{k',k-1}}, & \text{for } k' = 1, \ldots, k-1 \\ \frac{1}{|\mathcal{F}|}, & \text{for } ; k' = k. \end{cases} \tag{21}$$

From Theorem 1 we obtain the following performance bound for the above scheme.

*Theorem 3:* Let $\mathcal{F} \subset \mathcal{F}_s^\delta$ be a finite class of delay-$\delta$ memory-$s$ codes. Assume that $m, n$, $l$, $M$, and $s$ are positive integers such that $h = \lceil \log |\mathcal{F}| / \log M \rceil + s \le l, n/l \ge m+1$, and $l$ divides $n$, and let $0 < \alpha < 1, \eta > 0$. Then the difference of the normalized cumulative distortion of the constructed randomized, delay-$\delta$ coding scheme and that of the idealized scheme in (16) can be bounded for any sequence $\boldsymbol{x} \in \mathcal{X}^\infty$ as

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)\right] - D^*_{\mathcal{F},m,n}(\boldsymbol{x}) \\ \le \frac{h\hat{d}}{l} + \frac{1}{\eta n} \ln\left(\frac{|\mathcal{F}|^{m+1}}{\alpha^m (1-\alpha)^{n/l-m-1}}\right) \\ + \frac{\eta(l-h)^2}{8l} + \frac{m(l-1)}{n}. \end{aligned} \tag{22}$$

*Proof:* The proof follows from applying Theorem 1 to the transformed "prediction problem" described in (17)–(21). The last term on the right-hand side of (22) is due to the fact that the idealized scheme achieving $D^*_{\mathcal{F},m,n}(\boldsymbol{x})$ can switch its base code not only at the segment boundaries but also inside the segments. Thus, the minimum loss of any algorithm that is restricted to changes at the segment boundaries may exceed $D^*_{\mathcal{F},m,n}$ by at most $l-1$ for each occasion the change in the optimal idealized scheme occurs inside the segment. □

*Remark:* To optimize the bound in Theorem 3, first we choose $\eta$ optimally according to (6) as

$$\eta = \sqrt{\frac{8l}{n(l-h)^2} \ln\left(\frac{|\mathcal{F}|^{m+1}}{\alpha^m (1-\alpha)^{n/l-m-1}}\right)}.$$

Assuming $m < n/l - 1$ and letting $\alpha = m/(n/l-1)$, similarly to the derivation of (8) in the proof of Theorem 1, we obtain that the distortion redundancy can be bounded as

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)\right] - D^*_{\mathcal{F},m,n}(\boldsymbol{x}) \\ \le C_1 \frac{\log |\mathcal{F}|}{l} + C_2 \sqrt{\frac{lm}{n} \log \frac{n|\mathcal{F}|}{lm}} + \frac{m(l-1)}{n} \end{aligned} \tag{23}$$

where $C_1$ and $C_2$ are positive constants. From here it is easy to see that the best possible rate for the normalized distortion redundancy is achieved by setting $l = c_1(n \log^2 |\mathcal{F}|/m)^{1/3}$ for some positive constant $c_1$, yielding

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)\right] - D^*_{\mathcal{F},m,n}(\boldsymbol{x}) \\ = O\left(\left(\frac{m}{n}\right)^{\frac{1}{3}} \left(\log^{1/3} |\mathcal{F}| + \sqrt{\log \frac{n|\mathcal{F}|}{m}}\right)\right). \end{aligned}$$

A straightforward implementation of the above general coding scheme can be done via Algorithm 1, but this is efficient only if $\mathcal{F}$ is quite small, which severely limits the best achievable performance. If $\mathcal{F}$ is a large class of codes, Algorithm 1 provides an efficient implementation if the codes in $\mathcal{F}$ posses a certain structure. In the remainder of the paper, we will show this to be the case if $\mathcal{F}$ is a set of scalar quantizers, scalar multiple description, and multiresolution quantizers, respectively.

Tracking the best (traditional) scalar quantizer can be efficiently implemented by combining Algorithm 1 with the efficient implementation of the exponentially weighted prediction tailored to zero-delay quantization in [14]. To provide a framework for efficient implementations that work for traditional, as well as multiresolution and multiple description scalar quantization, we first present a general, efficient method for tracking the minimum-weight path in an acyclic weighted directed graph. We then demonstrate that this model provides a unified approach for the above mentioned three fixed-rate scalar quantization problems. The idea of posing optimal scalar quantizer design in terms of dynamic programming or as a problem of finding a minimum-weight path in an acyclic weighted directed graph is well known; see, e.g., [26], [27]. For network quantization, a similar approach is taken by Muresan and Effros [28] who consider (offline) design of entropy-constrained multiple description and multiresolution scalar quantizers. However, instead of an offline design we consider an online problem, and the differences in the details necessitate a detailed description of these models.

## IV. Minimum-Weight Path in a Directed Graph

In this section, we consider the problem of tracking the minimum-weight path in an acyclic weighted directed graph. The method presented here is a combination of the efficient implementation (Algorithm 2) of the tracking algorithm of Herbster and Warmuth [21], and the weight pushing algorithm of [11]–[13] which enables efficient computation of the constants $\{Z_{k',k}\}$. The slightly different problem of tracking the minimum-weight path of a given length was considered in [29].

Consider an acyclic directed graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertices and edges, respectively. Given a fixed pair of vertices $s$ and $u$, let $\mathcal{R}$ denote the set of all directed paths from $s$ to $u$, and assume that $\mathcal{R}$ is not empty. We also assume that for all $z \ne u$, $z \in \mathcal{V}$, there is an edge starting from $z$. (Otherwise, vertex $z$ is of no use in finding a path from $s$ to $u$, and all such vertices can be removed iteratively from the graph at the beginning of the algorithm in $O(|\mathcal{V}|) + O(|\mathcal{E}|)$ time.) Finally, we assume that the vertices are labeled by the integers

$1, 2, \ldots, |\mathcal{V}|$ such that $s = 1, u = |\mathcal{V}|$, and if $z_1 < z_2$, then there is no edge from $z_2$ to $z_1$ (such an ordered labeling can be found in $O(|\mathcal{E}|)$ time since the graph is acyclic). At time $t = 1, 2, \ldots$, the predictor picks a path $\hat{y}_t \in \mathcal{R}$. The cost of this path is the sum of the weights $\delta_t(a)$ on the edges $a$ of the path (the weights are assumed to be nonnegative real numbers), which are revealed for each $a \in \mathcal{E}$ only after the path has been chosen. To use our previous definition for prediction in Section II, we may define $y_t = \{\delta_t(a)\}_{a\in\mathcal{E}}$, and the loss function

$$\ell(y_t, \hat{y}_t) = \sum_{a\in\hat{y}_t} \delta_t(a)$$

for each pair $(y_t, \hat{y}_t)$. The cumulative loss at time $T$ is given by

$$L_T = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t).$$

Our goal is to perform as well as the best sequence of paths in which paths are allowed to change $m$ times during the time interval $t = 1, \ldots, T$. As in the prediction context, such a combination is given by an $m$-partition $\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})$, where $\boldsymbol{t} = (t_1, \ldots, t_m)$ such that $t_0 = 0 < t_1 < \cdots < t_m < t_{m+1} = T$, and $\boldsymbol{e} = (e_0, \ldots, e_m)$, where $e_i \in \mathcal{R}$ (that is, expert $e \in \mathcal{R}$ predicts $\hat{y}_t^{(e)} = e$). The cumulative loss of a partition $\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})$ is

$$\begin{aligned} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) &= \sum_{i=0}^{m} \sum_{t=t_i+1}^{t_{i+1}} \ell(y_t, e_i) \\ &= \sum_{i=0}^{m} \sum_{t=t_i+1}^{t_{i+1}} \sum_{a\in e_i} \delta_t(a). \end{aligned}$$

Now Algorithms 1 and 2 can be used to choose the path $\hat{y}_t$ randomly at each time instant $t = 1, \ldots, T$, and the regret

$$L_T - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e}))$$

can be bounded by Theorem 1. In this setup, with the aid of the weight pushing algorithm [11]–[13], we can compute efficiently a path based on the exponentially weighted prediction method and the constants $Z_{t',t}$, and thus prove the following theorem.

*Theorem 4:* For the minimum-weight path problem described in this section, Algorithm 2 can be implemented in $O(T^2|\mathcal{E}|)$ time. Moreover, let $N$ denote the number of different paths from vertex $s$ to vertex $u$, and assume that $\alpha = \frac{m}{T-1} < 1$, $\delta_t(a) < B/(|\mathcal{V}|-1)$ for all $t$ and edges $a \in \mathcal{E}$, and $\eta$ is chosen according to (6). Then, for any $p \in (0, 1)$, the regret of the algorithm can be bounded from above, with probability at least $1 - p$, as

$$\begin{aligned} L_T &- \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) \\ &\le T^{1/2}\frac{B}{\sqrt{2}}\sqrt{(m+1)\ln N + m\ln\frac{T-1}{m} + m} \\ &\quad + B\sqrt{\frac{T\ln(1/p)}{2}}. \end{aligned}$$

The expected regret of the algorithm can be bounded as

$$\begin{aligned} \mathbb{E}L_T &- \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T, m, \boldsymbol{t}, \boldsymbol{e})) \\ &\le T^{1/2}\frac{B}{\sqrt{2}}\sqrt{(m+1)\ln N + m\ln\frac{T-1}{m} + m}. \end{aligned}$$

*Proof:* The performance bound in the theorem follows trivially from the optimized bound (7) in Theorem 1. All we need to show is that the algorithm can be implemented in $O(T^2|\mathcal{E}|)$ time. To do this, we first revisit the weight pushing algorithm [11]–[13] via a modification of the algorithm of [14] for choosing a path $\hat{y}_t$ randomly based on $(y_{t'}, y_{t'+1}, \ldots, y_{t-1})$. That is, based on the weights $\{\delta_i(a)\}_{a\in\mathcal{E}}, i \in [t', t-1]$, we have to choose a path $\hat{y}_t$ according to the probabilities

$$\mathbb{P}\{\hat{y}_t = r\} = \frac{e^{-\eta\sum_{a\in r}\Delta_{t',t-1}(a)}}{\sum_{r'\in\mathcal{R}} e^{-\eta\sum_{a\in r'}\Delta_{t',t-1}(a)}} \tag{24}$$

where $\Delta_{t',t-1}(a) = \sum_{i=t'}^{t-1}\delta_i(a)$, and compute

$$Z_{t',t-1} = \sum_{r\in\mathcal{R}} e^{-\eta\sum_{a\in r}\Delta_{t',t-1}(a)}.$$

Using the constants $Z_{1,t-1}, \ldots, Z_{t-1,t-1}$ and $W_1, \ldots, W_{t-1}$, we can compute $W_t$, and perform the random choice of $\tau_t$ via Algorithm 2. In what follows we show how these steps can be done efficiently.

For any $z \in \mathcal{V}$, let $\mathcal{R}_z$ denote the set of paths from $z$ to $u$ (we define $\mathcal{R}_u = \emptyset$), and let $G_{t',t}(z)$ denote the sum of the exponential cumulative losses in the interval $[t', t]$ of all paths in $\mathcal{R}_z$. Formally, if $\mathcal{R}_z$ is empty then we define $G_{t',t}(z) = 1$, otherwise

$$G_{t',t}(z) = \sum_{r\in\mathcal{R}_z} e^{-\eta\sum_{a\in r}\Delta_{t',t}(a)}. \tag{25}$$

Then $Z_{t',t} = G_{t',t}(s)$, and $G_{t',t}(z)$ can be computed recursively for $z = u-1, u-2, \ldots, s = 1$, as $G_{t',t}(u) = 1$

$$G_{t',t}(z) = \sum_{\hat{z}:(z,\hat{z})\in\mathcal{E}} e^{-\eta\Delta_{t',t}((z,\hat{z}))} G_{t',t}(\hat{z}). \tag{26}$$

Note that since $\hat{z} > z$ if $(z, \hat{z}) \in \mathcal{E}$, $G_{t',t}(\hat{z})$ is already available when it is needed in the above formula. In the recursion, each edge is taken into consideration exactly once. Therefore, calculating $G_{t',t}(z)$ for all $z \in \mathcal{V}$ requires $O(|\mathcal{E}|)$ computations for any fixed $1 \le t' \le t$, provided the cumulative weights $\Delta_{t',t}(a)$ are known for all edges $a \in \mathcal{E}$. Now for a given $t$, as $t'$ is decreased from $t$ to 1, if we store the cumulative weights $\Delta_{t',t}(a)$ for each edge $a$, then only $O(|\mathcal{E}|)$ computations are needed to update the cumulative weights at the edges for each $t'$. Therefore, for a given $t$, calculating $G_{t',t}(z)$ for all $z \in \mathcal{V}$ and $1 \le t' \le t$ requires $O(t|\mathcal{E}|)$ computations.

The function $G_{t',t-1}$ offers an efficient way of drawing $\hat{y}_t$ randomly for a given $\tau_t = t'$: For any $z \in \mathcal{V} \setminus \{u\}$, let $\mathcal{E}_z = \{\hat{z} : (z, \hat{z}) \in \mathcal{E}\}$ and

$$p_{t',t}(\hat{z}|z) = e^{-\eta\Delta_{t',t-1}((z,\hat{z}))}\frac{G_{t',t-1}(\hat{z})}{G_{t',t-1}(z)}, \quad \hat{z} \in \mathcal{E}_z.$$

For fixed $z$, $p_{t',t}(\hat{z}|z)$ is a probability distribution on $\mathcal{E}_z$ since by (25)

$$\sum_{\hat{z}\in\mathcal{E}_z} p_{t',t}(\hat{z}|z) = 1.$$

Denote the $k$th vertex along a path $r \in \mathcal{R}$ by $z_{r,k}$ for $k = 0, 1, \ldots, |r|$, where $|r|$ is the length of the path $r$ ($z_{r,0} = s$ and $z_{r,|r|} = u$). Then

$$\begin{aligned}\prod_{k=1}^{|r|} & p_{t',t}(z_{k,r}|z_{k-1,r}) \\ &= \prod_{k=1}^{|r|} e^{-\eta\Delta_{t',t-1}((z_{k-1,r},z_{k,r}))} \frac{G_{t',t-1}(z_{k,r})}{G_{t',t-1}(z_{k-1,r})} \\ &= e^{-\eta\sum_{k=1}^{|r|}\Delta_{t',t-1}((z_{k-1,r},z_{k,r}))} \frac{G_{t',t-1}(u)}{G_{t',t-1}(s)} \\ &= \mathbb{P}\{\hat{y}_t = r\} \qquad (27)\end{aligned}$$

by (24) since $G_{t',t-1}(u) = 1$ and $G_{t',t-1}(s) = \sum_{r'\in\mathcal{R}} e^{-\eta\sum_{a\in r'}\Delta_{t',t}(a)}$. Thus, $\hat{y}_t$ can be drawn randomly in a sequential manner: Starting from $z_{\hat{y}_t,0} = s$, in each step $k = 1, 2, \ldots$ choose $z = z_{\hat{y}_t,k}$ randomly from $\mathcal{E}_{z_{\hat{y}_t,k-1}}$ with probability $p_{t',t}(z|z_{\hat{y}_t,k-1})$. The procedure stops when $z_{\hat{y}_t,k} = u$. Thus, $\hat{y}_t$ can be computed in $O(|\mathcal{V}|)$ steps if $\tau_t = t'$ and the functions $\Delta_{t',t-1}$ and $G_{t',t-1}$ are given, as any path from $s$ to $r$ is of length at most $|\mathcal{V}| - 1$.

It remains to show that $\tau_t$ can be chosen efficiently. As we have seen before, $G_{t',t}(z)$ can be computed in $O(t|\mathcal{E}|)$ time for all $z$ and $1 \le t' \le t$; hence, finding $Z_{t',t-1} = G_{t',t-1}(s)$ requires $O(t|\mathcal{E}|)$ computations. Then, given $W_{t'}$ for $t' = 1, \ldots, t-1$, $W_t$ can be computed by (10) in $O(t)$ steps, and so for all $t' = 1, \ldots, t-1$, the computational time of $W_t$ and $Z_{t',t-1}$ is $O(t|\mathcal{E}|) + O(t) = O(t|\mathcal{E}|)$. Therefore, $\tau_t$ can be chosen randomly according to (13) in the same computational time. Finally, as we have seen in the preceding paragraph, given $\tau_t = t'$ and the function $G_{t',t-1}$, $\hat{y}_t$ can be computed in $O(|\mathcal{V}|)$ steps. Thus, the overall time complexity of computing $\hat{y}_t$ for a given $t$ (using values computed up to time $t-1$) is $O(t|\mathcal{E}|) + O(|\mathcal{V}|) = O(t|\mathcal{E}|)$ (as $|\mathcal{E}| \ge |\mathcal{V}| - 1$). Thus, Algorithm 2 can be performed in $O(T^2|\mathcal{E}|)$ time. □

## V. Online Scalar Quantization

In this section, we apply the results of Sections III and IV to construct efficient zero-delay sequential source codes. Our goal is to find efficiently implementable zero-delay coding schemes that perform asymptotically as well as the best scalar quantization scheme which is allowed to change the employed quantizer a certain number of times.

We assume that the source and reproduction symbols belong to the interval $[0, 1]$. Then a zero-delay scheme using encoder randomization is given formally by the encoder–decoder functions $\{f_i, g_i\}_{i=1}^{\infty}$, where

$$f_i : [0,1]^i \times [0,1]^i \to \{1, 2, \ldots, M\}$$

and

$$g_i : \{1, 2, \ldots, M\}^i \to [0,1]$$

so that $b_i = f_i(x^i, U^i)$ and $\hat{x}_i = g_i(b^i)$, $i = 1, 2, \ldots$. Recall that $\{U_i\}$ is the randomization sequence, and note that there is no delay in the encoding and decoding process; i.e., $\delta = 0$ in the terminology of Section III.

We also assume that the distortion is measured by some bounded nondecreasing difference distortion measure of the form

$$d(x, \hat{x}) = \rho(|x - \hat{x}|) \qquad (28)$$

where $\rho : [0,1] \to [0,1]$ is assumed to satisfy the Lipschitz condition

$$|\rho(x) - \rho(\hat{x})| \le c_\rho|x - \hat{x}|, \quad \text{for all } x, \hat{x} \in [0,1] \qquad (29)$$

for some constant $c_\rho > 0$. (For the squared error distortion $\rho(x) = x^2$, we have $c_\rho = 2$.) The base set of reference codes we use is the set of scalar quantizers. Formally, an $M$-level scalar quantizer $Q$ is a measurable mapping $\mathbb{R} \to C$, where the *codebook* $C$ is a finite subset of $\mathbb{R}$ with cardinality $|C| = M$. The elements of $C$ are called the *code points*. The instantaneous distortion of $Q$ for input $x$ is $\rho(|x - Q(x)|)$. A quantizer $Q$ is called a nearest neighbor quantizer if for all $x$

$$|Q(x) - x| = \min_{\hat{x}\in C} |x - \hat{x}|.$$

As $\rho$ is nondecreasing, it is immediate from the definition that if $Q$ is a nearest neighbor quantizer and $\hat{Q}$ has the same codebook as $Q$, then $\rho(|Q(x) - x|) \le \rho(|\hat{Q}(x) - x|)$ for all $x$. For this reason, we only consider nearest neighbor quantizers. Also, since we consider sequences with components in $[0, 1]$, we can assume without loss of generality that the domain of definition of $Q$ is $[0, 1]$ and that all its code points are in $[0, 1]$.

Let $\mathcal{Q}$ denote the collection of all $M$-level nearest neighbor quantizers. For any sequence $x^n$, we want our scheme to perform asymptotically as well as the best coding scheme which employs $M$-level scalar quantizers and is allowed to change quantizers $m$ times. Formally, a code in this class $\mathcal{Q}_{m,n}$ is given by the integers $1 \le i_1 < i_2 < \cdots < i_m < n$ and $M$-level scalar quantizers $q_0, \ldots, q_m \in \mathcal{Q}$ such that $x_i$ is encoded to $q_j(x_i)$ if $i_j < i \le i_{j+1}$, where $i_0 = 0$ and $i_{m+1} = n$. The minimum normalized cumulative distortion achievable by such schemes is

$$\begin{aligned}D^*_{\mathcal{Q},m,n}&(x^n) \\ &= \frac{1}{n} \min_{1\le i_1<i_2<\ldots<i_m<n} \sum_{j=0}^{m} \min_{q\in\mathcal{Q}} \sum_{i=i_j+1}^{i_{j+1}} \rho(|x_i - q(x_i)|).\end{aligned}$$

Note that to find the best scheme achieving this minimum one has to know the entire sequence $x^n$ in advance. Moreover, unlike in (16), the minimum $D^*_{\mathcal{Q},m,n}(x^n)$ is indeed achievable by realizable coding schemes since we now deal with the zero delay case (however, the optimal scheme will in general be different for each source sequence $x^n$).

The expected *distortion redundancy* of a scheme (with respect to the class $\mathcal{Q}_{m,n}$) is the quantity

$$\sup_{x^n}\left(\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i - \hat{x}_i|)\right] - D^*_{\mathcal{Q},m,n}(x^n)\right) \qquad (30)$$

where the supremum is over all individual sequences of length $n$ with components in $[0,1]$ (recall that the expectation is taken over the randomizing sequence).

We could immediately apply the coding scheme of Section III if the set $\mathcal{Q}$ were finite. Since this is not the case, we approximate $\mathcal{Q}$ with $\mathcal{Q}_K$, the set of all $M$-level nearest neighbor quantizers whose code points all belong to the finite grid

$$C^{(K)} = \{1/(2K), 3/(2K), \ldots, (2K-1)/(2K)\}. \tag{31}$$

By the Lipschitz condition on $\rho$, for any $q \in \mathcal{Q}$ there is a $q' \in \mathcal{Q}_K$ such that

$$\sup_{x\in[0,1]} |\rho(|x - q(x)|) - \rho(|x - q'(x)|)| \le c_\rho/(2K). \tag{32}$$

In particular, for the squared error distortion the difference is at most $1/K$.

The next theorem shows that a slightly modified version of the coding scheme of Section III applied to the base reference class $\mathcal{Q}_K$ (which has delay $\delta = 0$ and decoder memory $s = 0$) can perform as well as the best coding scheme that uses scalar quantization and is allowed to change its quantizer $m$ times for $n$ source samples. Moreover, the proposed scheme can be implemented efficiently.

*Theorem 5:* Assume that $m, n, l, K, M$ are positive integers such that $M \le K$, $l \ge \lceil \log\binom{K}{M}/\log M \rceil$, $n/l > m$, and $l$ divides $n$, and let $0 < \alpha < 1$, $\eta > 0$. Then there is a coding scheme with zero delay and rate $R = \log M$ whose normalized cumulative distortion can be bounded for any sequence $x^n \in [0,1]^n$ as

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i - \hat{x}_i|)\right] &- D^*_{\mathcal{Q},m,n}(x^n)\\ &\le \frac{\rho(1/2)}{l}\left\lceil \frac{1}{R}\log\binom{K}{M}\right\rceil\\ &\quad + \frac{1}{\eta n}\ln\left(\frac{\binom{K}{M}^{m+1}}{\alpha^m(1-\alpha)^{n/l-m-1}}\right) + \frac{\eta l}{8} + \frac{c_\rho}{K} + \frac{m(l-1)}{n}.\end{aligned} \tag{33}$$

Moreover, the algorithm can be implemented with

$$O(MK^2n^2/l^2) + O(K^3n/l) + O(n)$$

computational complexity.

*Remark:* Assuming $m < n/l - 1$, let $\alpha = m/(n/l - 1)$. Then, choosing $\eta$ optimally (based on (6)), similarly to (23) we obtain that the distortion redundancy can be bounded as

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i - \hat{x}_i|)\right] &- D^*_{\mathcal{Q},m,n}(x^n)\\ &\le \frac{C_1\log K}{l} + C_2\sqrt{\frac{lm}{n}\log\frac{n}{lm}} + \frac{1}{K} + \frac{ml}{n}\end{aligned}$$

where $C_1$ and $C_2$ are appropriate positive constants. From here the best possible rate achievable is $O((m/n)^{1/3}\log(n/m))$, when $l = c_1(n/m)^{1/3}$ and $K = c_2(n/m)^{1/3}$ (where $c_1$ and $c_2$ are arbitrary positive constants), which requires $O(Mn^2)$ computations.

In a practical implementation it is desirable that the computational complexity per unit time remains a constant as the length of the input sequence increases. In our case, such an implementation is possible if the total computational complexity is linear in $n$. This may be achieved by setting $l = c_1n^{2/3}$ and $K = c_2n^{1/6}$. Then the computational complexity of the algorithm is $O(Mn)$. However, with this choice, the normalized distortion redundancy deteriorates to $O(m^{1/2}\log n/n^{1/6})$ (here we require $m = o(n^{1/3}/\log^2 n)$ in order to ensure that the distortion redundancy converges to zero).

*Proof of Theorem 5:* Let $q_{2K} : [0,1] \to C^{(2K)}$ be a $2K$-level uniform quantizer in $[0,1]$ (that is, a nearest neighbor quantizer with codebook $C^{(2K)}$), and let $\bar{x}_i = q_{2K}(x_i)$ be the uniformly quantized version of $x_i$. The algorithm of Section III is modified so that when choosing the quantizer $Q^{(k)}$ from $\mathcal{Q}_K$, the cumulative distortions in (19) are computed with respect to the sequence $\{\bar{x}_i\}$ instead of $\{x_i\}$. This "pre-quantization" step is necessary to reduce the computational complexity of the algorithm and only results in a slight increase of the distortion if $K$ is judiciously chosen. The latter claim can be seen as follows: Without loss of generality, we can assume that in each quantizer (including $q_{2K}$) each decision threshold is quantized to the smaller nearest code point (that is, the quantization cells are right-closed intervals). Then $q(x) = q(\bar{x})$ for any quantizer $q \in \mathcal{Q}_K$ and $x \in [0,1]$, where $\bar{x} = q_{2K}(x)$. Therefore, assuming the same realization of $\{U_n\}$ is used, the output sequence $\hat{x}^n$ is the same in the following two situations: 1) the original algorithm of Section III is applied to the input $\bar{x}^n$; 2) the modified version of the algorithm above is applied to the input $x^n$. Moreover

$$|(x - q(x)) - (\bar{x} - q(\bar{x}))| = |x - \bar{x}| \le 1/(4K)$$

implying

$$\rho(|x - q(x)|) - \rho(|\bar{x} - q(\bar{x})|) \le c_\rho/(4K).$$

Thus, the difference of the normalized cumulative distortion of the two algorithms can be bounded as

$$\left|\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i - \hat{x}_i|)\right] - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|\bar{x}_i - \hat{x}_i|)\right]\right| \le \frac{c_\rho}{4K}. \tag{34}$$

Similarly, the difference of the minimum distortions achievable by changing quantizers $m$ times can be bounded as

$$|D^*_{\mathcal{Q},m,n}(x^n) - D^*_{\mathcal{Q},m,n}(\bar{x}^n)| \le \frac{c_\rho}{4K}. \tag{35}$$

This implies that the normalized expected distortion redundancy of our modified algorithm is no more than the redundancy of the original algorithm applied to $\bar{x}^n$ plus $c_\rho/(2K)$. The latter redundancy can easily be bounded by Theorem 3 with $|\mathcal{F}| = |\mathcal{Q}_K| = \binom{K}{M}$, $\hat{x} = 1/2$, and $\hat{d} = \rho(1/2)$ as

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|\bar{x}_i - \hat{x}_i|)\right] - D^*_{\mathcal{Q},m,n}(\bar{x}^n)\\
&\quad\le \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|\bar{x}_i - \hat{x}_i|)\right] - D^*_{\mathcal{Q}_K,m,n}(\bar{x}^n) + \frac{c_\rho}{2K}\\
&\quad\le \frac{\rho(1/2)}{l}\left\lceil\frac{1}{R}\log\binom{K}{M}\right\rceil + \frac{1}{\eta n}\ln\left(\frac{\binom{K}{M}^{m+1}}{\alpha^m(1-\alpha)^{n/l-m-1}}\right)\\
&\qquad + \frac{\eta l}{8} + \frac{c_\rho}{2K} + \frac{m(l-1)}{n}
\end{aligned}$$

where the first inequality follows from (32) via

$$|D^*_{\mathcal{Q},m,n}(x^n) - D^*_{\mathcal{Q}_K,m,n}(x^n)| \le c_\rho/(2K).$$

Now (33) follows from (34) and (35).

Next we show that the algorithm can be implemented with the claimed complexity by reducing the quantizer design algorithm to the problem of finding online the minimum-weight path in a weighted directed graph as discussed in Section IV. Consider the directed graph with vertices

$$\mathcal{V} = C^{(K)} \times \{1, 2, \ldots, M\} \cup (0, 0) \cup (1, M+1)$$

and edges

$$\begin{aligned}\mathcal{E} = \{((z, j-1), (\hat{z}, j)) : z,&\\ \hat{z} \in C^{(K)}, z < \hat{z}, j \in \{1, \ldots, M+1\}\}&\end{aligned}$$

such that at time instant $k$ the weight of edge $((z, j-1), (\hat{z}, j))$ is $\delta_k((z, \hat{z}))$, given at the bottom of the page, for all $j$, where $I_{\mathcal{B}}$ denotes the indicator function of the event $\mathcal{B}$. With a slight abuse of notation, let any path be described by the ordered sequence of its constituent vertices. Then it can be seen that for any $z_0 = 0 < z_1 < \cdots < z_M < z_{M+1} = 1$, the cost of a path $(z_0, 0), (z_1, 1), \ldots, (z_{M+1}, M+1)$ at time instant $k$ is the same as the cumulative distortion in the $k$th block of a nearest neighbor quantizer $Q$ with code points $\{z_1, \ldots, z_M\}$ (see Fig. 4 for an example). Moreover, any path from $s = (0, 0)$ to $u = (1, M+1)$ is of the form $(z_0, 0), (z_1, 1), \ldots, (z_{M+1}, M+1)$. Therefore, the random choice of a quantizer according to the probabilities given in the algorithm of Section III (see (20) and (21)) is equivalent to randomly choosing a path from vertex $s$ to vertex $u$ as in Section IV. Thus, as $|\mathcal{V}| = MK + 2$, $|\mathcal{E}| = (M-1)K(K-1)/2 + 2K$, and $T = n/l$, applying the algorithm described in Section IV, the random choice of $Q^{(k)}$ for $k = 1, \ldots, n/l$ can be performed in $O(MK^2n^2/l^2)$ time, provided the weights $\delta_k$ are known.

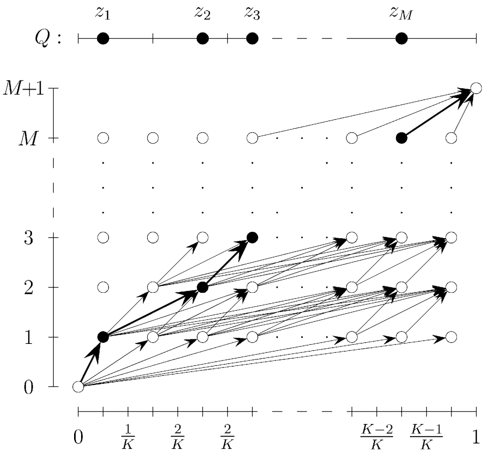

Fig. 4. An example of a scalar quantizer and the corresponding graph.

Now for each $k$, $\delta_k$ can be computed efficiently as follows: Let

$$\begin{aligned}a_{k,j} = |\{i : \bar{x}_i = (2j-1)/2K, (k-1)l+1 \le i \le kl\}|,&\\ j = 1, \ldots, K, \quad k = 1, \ldots, n/l.&\end{aligned}$$

(Computing the $a_{k,j}$ takes $O(n)$ time.) Then the weight $\delta_k(z, \hat{z})$ can be computed in $O(K)$ steps for each pair $(z, \hat{z})$ by running through $a_{k,1}, a_{k,2}, \ldots, a_{k,K}$. For example, for $0 < z < \hat{z} < 1$

$$\delta_k(z, \hat{z}) = \sum_{z < j \le \frac{z+\hat{z}}{2}} a_{k,j}\rho(|j - z|) + \sum_{\frac{z+\hat{z}}{2} < j < \hat{z}} a_{k,j}\rho(|j - \hat{z}|).$$

Thus, computing the weights in each block requires $O(K^3)$ time, resulting in a total computational time of $O(MK^2n^2/l^2) + O(K^3n/l) + O(n)$. (Note that the use of the finely quantized version $\bar{x}^n$ of the input sequence changed the total computational cost of the weights from $O(nK^2)$ to $O(K^3n/l)$. Although with certain choices of the parameters the latter quantity may be larger, it can be made linear in $n$ for large enough $l$, while $nK^2$ always grows faster than linearly when $K \to \infty$ as $n \to \infty$, a condition necessary for asymptotic optimality.) □

## VI. Online Multiresolution and Multiple Description Scalar Quantization

In this section, we generalize the online quantization algorithm to network quantization problems, such as multiresolution and multiple description quantization. Multiple description coding (e.g., [30]–[32]) makes it possible to recover data at a degraded but still acceptable quality if some parts of the transmitted data are lost. In this coding scheme, several different descriptions of the source are produced such that various levels of reconstruction quality can be obtained from different subsets of

$$\delta_k((z, \hat{z})) = \begin{cases} \sum_{i=(k-1)l+1}^{kl} I_{\{\bar{x}_i \le \hat{z}\}}\rho(|\bar{x}_i - \hat{z}|), & \text{if } z = 0\\ \sum_{i=(k-1)l+1}^{kl} I_{\{\bar{x}_i \in (z, \frac{z+\hat{z}}{2}]\}}\rho(|\bar{x}_i - z|) + I_{\{\bar{x}_i \in (\frac{z+\hat{z}}{2}, \hat{z}]\}}\rho(|\bar{x}_i - \hat{z}|), & \text{if } 0 < z < \hat{z} < 1\\ \sum_{i=(k-1)l+1}^{kl} I_{\{\bar{x}_i \ge z\}}\rho(|\bar{x}_i - z|), & \text{if } 0 < z \text{ and } \hat{z} = 1 \end{cases}$$

these descriptions. Multiresolution coding (e.g., [33]–[35]) is a special case of multiple description coding in which the information is progressively refined as more and more descriptions are received.

To simplify the notation, we consider only two-description systems, but the results can be generalized to several descriptions in a straightforward manner. As before, we restrict our attention to zero-delay coding schemes.

A fixed-rate zero-delay sequential two-description code of rate $(R_1, R_2)$ with $R_j = \log M_j, j = 1, 2$, is defined by an encoder–decoder pair connected via two discrete erasure channels having input alphabets $\{1, 2, \ldots, M_j\}$, $j = 1, 2$. The output alphabet of the $j$th channel is $\{0, 1, \ldots, M_j\}$ where the character 0 corresponds to an erasure. The channels are assumed to be memoryless and time invariant, but not necessarily independent. Let $r_j, j = 1, 2$, denote the (joint) probability that there is no erasure on channel $j$, and erasure occurs on channel $3 - j$, and let $r_0$ denote the probability that there is no erasure on either channel. Thus, only description $j$ is received with probability $r_j, j = 1, 2$, and both descriptions are received with probability $r_0$. As before, we assume that the encoder has access to a randomization sequence $U_1, U_2, \ldots$ of independent random variables distributed uniformly over the interval $[0, 1]$, and the input to the encoder is a sequence of real numbers $x_1, x_2, \ldots$ taking values in the interval $[0, 1]$. At each time instant $i = 1, 2, \ldots$, based on the observed input values $x^i$ and randomization sequence $U^i$, the encoder produces channel symbols $b_i^{(j)} \in \{1, 2, \ldots, M_j\}, j = 1, 2$, which are then transmitted over the corresponding channels. The decoder receives the (possibly erased) symbols $\hat{b}_i^{(j)}$, and outputs the reproduction $\hat{x}_i$ based on the channel symbols $(\hat{b}_1^{(1)}, \ldots, \hat{b}_i^{(1)}, \hat{b}_1^{(2)}, \ldots, \hat{b}_i^{(2)})$ received so far. Note that if an erasure occurs on channel $j$ then $\hat{b}_i^{(j)} = 0$, otherwise $\hat{b}_i^{(j)} = b_i^{(j)}$.

The code is formally given by a sequence of encoder–decoder functions $\{f_i, g_i\}_{i=1}^{\infty}$, where $f_i = (f_i^{(1)}, f_i^{(2)})$ with

$$f_i^{(j)} : [0,1]^i \times [0,1]^i \to \{1, 2, \ldots, M_j\}, \quad j = 1, 2$$

and

$$g_i : \{0, 1, \ldots, M_1\}^i \times \{0, 1, \ldots, M_2\}^i \to [0, 1]$$

so that $b_i^{(j)} = f_i^{(j)}(x^i, U^i), j = 1, 2$, and $\hat{x}_i = g_i(\hat{b}_i^{(1)}, \hat{b}_i^{(2)})$, $i = 1, 2, \ldots$, where

$$\begin{aligned}
\mathbb{P}\left\{\hat{b}_i^{(1)} = b_i^{(1)}, \hat{b}_i^{(2)} = b_i^{(2)}\right\} &= r_0 \\
\mathbb{P}\left\{\hat{b}_i^{(1)} = b_i^{(1)}, \hat{b}_i^{(2)} = 0\right\} &= r_1 \\
\mathbb{P}\left\{\hat{b}_i^{(1)} = 0, \hat{b}_i^{(2)} = b_i^{(2)}\right\} &= r_2 \\
\mathbb{P}\left\{\hat{b}_i^{(1)} = 0, \hat{b}_i^{(2)} = 0\right\} &= 1 - r_0 - r_1 - r_2.
\end{aligned}$$

Again, the distortion of the scheme is measured using a nondecreasing difference distortion measure $d(x, \hat{x}) = \rho(|x - \hat{x}|)$, defined in (28), satisfying the Lipschitz condition (29). Thus, the normalized cumulative distortion of the sequential scheme at time instant $n$ is again given by

$$\frac{1}{n} \sum_{i=1}^{n} \rho(|x_i - \hat{x}_i|).$$

The expected normalized cumulative distortion is

$$\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \rho(|x_i - \hat{x}_i|)\right]$$

where, in contrast to the single-description (scalar quantization) case, the expectation is taken with respect to both the randomizing sequence $U^n$ and the channel randomness.

An $(M_1, M_2)$-level multiple description scalar quantizer is given by two index mappings $\alpha_j : [0, 1] \to \{1, \ldots, M_j\}$, $j = 1, 2$, decoder functions

$$\beta_j : \{1, \ldots, M_j\} \to \{\hat{x}_1^{(j)}, \ldots, \hat{x}_{M_j}^{(j)}\} \subset [0, 1], \quad j = 1, 2$$

and

$$\beta_0 : \mathcal{C} \to \{\hat{x}_{i,j}^{(0)} : (i, j) \in \mathcal{C}\} \subset [0, 1]$$

where $\mathcal{C} = \{(\alpha_1(x), \alpha_2(x)) : x \in [0, 1]\}$. In addition, one must also specify a constant $\hat{x}^*$ to make the definition complete.

For each input $x$, the encoder assigns two index values $\alpha_1(x)$ and $\alpha_2(x)$ which are transmitted over two different channels. If the decoder receives both indices (descriptions), it outputs $q^c(x) = \beta_0(\alpha_1, \alpha_2) = \hat{x}_{\alpha_1(x), \alpha_2(x)}^{(0)}$; if only index $\alpha_j(x)$ is received $(j = 1, 2)$, the output is $q^{(j)}(x) = \beta_j(\alpha_j(x)) = \hat{x}_{\alpha_j(x)}^{(j)}$; if both indices are lost, that is, no description is received, then the output of the decoder is $\hat{x}^*$. Usually, $q^{(1)}$ and $q^{(2)}$ are referred to as the first and the second side quantizer, respectively, while $q^c$ is called the central quantizer. Let $\hat{x}$ denote the (random) reproduction of the multiple description quantizer $q$ when coding the input value $x$, and let $\alpha_j(x) = i_j$, $j = 1, 2$. Then the average distortion of $q$ is given by

$$\begin{aligned}
d_q(x) &= \mathbb{E}\rho(|x - \hat{x}|) \\
&= r_0 \rho\left(|x - \hat{x}_{i_1, i_2}^{(0)}|\right) + r_1 \rho\left(|x - \hat{x}_{i_1}^{(1)}|\right) \\
&\quad + r_2 \rho\left(|x - \hat{x}_{i_2}^{(2)}|\right) \\
&\quad + (1 - r_0 - r_1 - r_2) \rho(|x - \hat{x}^*|) \qquad (36)
\end{aligned}$$

where the expectation is taken with respect to the channel randomness. (For probabilistic stationary sources and the squared error distortion, $\hat{x}^*$ is optimally chosen to be the expectation of the source.)

The two-description code we defined can be viewed as special joint source–channel code for the erasure channel. In this sense, the basic system we described can be considered as a special case of joint source–channel codes for individual source sequences and stochastic channels considered by Matloub and Weissman [17], who showed that coding schemes devised for noiseless channels can be applied as source–channel codes if the distortion measure is replaced by its expectation with respect to the channel noise. Using this reduction method they constructed a universal zero-delay joint source–channel coding scheme for individual sequences and, based on [16], also constructed an efficient implementation for memoryless channels. The definition of our modified distortion measure in (36) is similar to the method of [17]. It can easily be seen that the approach of [17] would suffice to extend the general tracking problem in Section III to joint source–channel coding. However, our goal in this section is to achieve stronger results in a more special setup;

namely, we want to extend the tracking quantization scheme of Section V to obtain efficient algorithms tailored to the special structure of multiple description and multiresolution quantizers.

In contrast to traditional fixed-rate scalar quantization, the structure of optimal multiple description scalar quantizers is not well understood. In this direction, Vaishampayan [36] showed that the cells of optimal two-description scalar quantizers are unions of finitely many intervals. More precisely, he showed that the intersection of the $i$th cell of the first side quantizer and the $j$th cell of the second side quantizers (i.e., the set $\{x : \alpha_1(x) = i, \alpha_2(x) = j\}$) is either an interval or the empty set. In general, however, for an optimal quantizer the cells of the side quantizers (the sets $\{x : \alpha_j(x) = i\}$, $j = 1, 2, i = 1, \dots, M_j$) are not necessarily intervals. An example demonstrating this is given in [37].

Since the optimal side quantizers can have a very complex structure, finding these for a given source distribution may be computationally hard for quantizers with moderate or large rates. To avoid this problem, the restriction that the side quantizers have interval cells has recently been introduced in [28], [38]–[40]. These works use graph-theoretic and/or dynamic programming frameworks to construct algorithms with reasonable complexity to find optimal (entropy-coded or fixed-rate) multiple description or multiresolution quantizers (with interval cells) for a given discrete probabilistic source. While the performance loss that results from the assumption of interval cells has not yet been quantified, some heuristic arguments exist [41] that indicate that this loss may not be significant at high rates. In our online multiple description quantization problem, we also make the assumption that the cells of the side quantizers are intervals. Let $\mathcal{Q}^{\mathrm{MD}}$ denote the collection of all $(M_1, M_2)$-level multiple description scalar quantizers $(\alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2, \hat{x}^*)$ such that the side quantizers have interval cells, and all reproduction points belong to their respective cells.

In the special case of two-level multiresolution quantization, one has $r_0 + r_1 = 1$, that is, either the first description or both descriptions are received. Accordingly, there is no need to specify the reproduction points corresponding to the second side quantizer or for the case when both descriptions are lost. Hence, a multiresolution quantizer is defined by the quadruple $q = (\alpha_1, \alpha_2, \beta_0, \beta_1)$. For multiresolution quantizers we assume that $q^{(1)}$, the first side quantizer, has interval cells, and $q^{(2)}$, restricted to a cell of $q^{(1)}$, is a nearest neighbor quantizer. Thus, a cell of $q^{(2)}$ is a union of $M_1$ intervals, one subinterval from each cell of $q^{(1)}$. Let $\mathcal{Q}^{\mathrm{MR}}$ denote the set of all such quantizers.

As before, for any source sequence $x^n$, we want to compete with the best coding scheme which employs quantizers from $\mathcal{Q}^{\mathrm{MD}}$ (or $\mathcal{Q}^{\mathrm{MR}}$), and is allowed to change its quantizer $m$ times. The source sequence $x^n$ is unknown in advance, but we assume that the erasure probabilities $r_0$, $r_1$, and $r_2$ are known at the encoder.

### A. Adaptive Multiple Description Scalar Quantization

Our aim is to generalize the algorithm of Section V to obtain an adaptive online multiple description scalar quantization scheme of moderate complexity in the individual sequence setting. The class of codes $\mathcal{Q}^{\mathrm{MD}}_{m,n}$ we want to compete with is formally given by integers $1 < i_1 < \cdots < i_m < n$ and $(M_1, M_2)$-level multiple description scalar quantizers $q_0, \dots, q_m \in \mathcal{Q}^{\mathrm{MD}}$ such that $x_i$ is encoded by $q_j$ for $i_j < i \le i_{j+1}$ where $i_0 = 0$ and $i_{m+1} = n$. The minimum normalized cumulative distortion achievable by such schemes is

$$D^*_{\mathcal{Q}^{\mathrm{MD}},m,n}(x^n) = \frac{1}{n} \min_{1 \le i_1 < \dots < i_m < n} \sum_{j=0}^{m} \min_{q \in \mathcal{Q}^{\mathrm{MD}}} \sum_{i=i_j+1}^{i_{j+1}} d_q(x_i)$$

where $d_q(x_i)$ was defined in (36). As before, one would have to know the entire sequence in advance to find an optimal scheme achieving this minimum.

There are two main problems to overcome in constructing an efficient algorithm on the basis of the general coding scheme in Section III. The first is to find an efficient finite covering of $\mathcal{Q}^{\mathrm{MD}}$, and the second is to find an efficient implementation of Algorithm 2. We first deal with the covering problem.

Let $\mathcal{Q}^{\mathrm{MD}}_K \subset \mathcal{Q}^{\mathrm{MD}}$ denote the set of $(M_1, M_2)$-level multiple description scalar quantizers such that the cells of the side quantizers are right closed intervals with endpoints (called the decision thresholds of the side quantizers) that belong to the set

$$\hat{C}^{(K)} = \{1/K, 2/K, \dots, (K-1)/K\}. \tag{37}$$

In addition, we also specify that each quantizer in $\mathcal{Q}^{\mathrm{MD}}_K$ has all its reproduction points in $\hat{C}^{(K)}$, and that each reproduction point belongs to its corresponding quantization cell (except for the reproduction point for the case when both descriptions are lost). The following lemma shows that if $K$ is sufficiently large, $\mathcal{Q}^{\mathrm{MD}}_K$ provides a fine covering of $\mathcal{Q}^{\mathrm{MD}}$. The proof is relegated to the Appendix.

*Lemma 2:* For any $q \in \mathcal{Q}^{\mathrm{MD}}$ there is a $q' \in \mathcal{Q}^{\mathrm{MD}}_K$ such that the maximum difference of the average distortions are bounded as

$$\sup_{x \in [0,1]} (d_{q'}(x) - d_q(x)) \le \frac{6c_\rho}{K}.$$

The lemma implies that for all $x^n$

$$D^*_{\mathcal{Q}^{\mathrm{MD}}_K,m,n}(x^n) - D^*_{\mathcal{Q}^{\mathrm{MD}},m,n}(x^n) \le \frac{6c_\rho}{K}. \tag{38}$$

For any $q \in \mathcal{Q}^{\mathrm{MD}}_K$, the $j$th side quantizer is determined by $M_j$ reproduction points and $M_j - 1$ thresholds (note, however, that the reproduction points and the thresholds are not necessarily distinct); the central quantizer has at most $M_1 M_2$ reproduction points, while its cells are determined by the side quantizers. Therefore, the number of quantizers in $\mathcal{Q}^{\mathrm{MD}}_K$ is bounded as shown in (39) at the bottom of the page where the last term corresponds to the choice of the constant $\hat{x}^*$ in the definition of $q$.

$$N = |\mathcal{Q}^{\mathrm{MD}}_K| \le \binom{K}{M_1 - 1} \binom{K}{M_1} \binom{K}{M_2 - 1} \binom{K}{M_2} \binom{K}{M_1 M_2} K \tag{39}$$

As the next theorem shows, the general coding scheme of Section III applied to the base reference class $\mathcal{Q}_K^{\mathrm{MD}}$ provides an efficient solution to the problem of tracking the best multiple description quantizer. The general scheme must be slightly modified, however, since at the beginning of each block, the index of the randomly chosen quantizer is now transmitted over two unreliable (erasure) channels. Therefore, we will repeat the description of the quantizer several times to ensure that the corresponding index can be decoded with large enough probability. (We use this repetition code for the sake of simplicity, and to reduce encoding/decoding complexity. Alternatively, we could employ an optimal channel code as was done in [17], but in the theoretical analysis this would only improve the scheme's performance by a multiplicative constant term.)

We will use the first $h$ time instants of each block ($h$ will be specified later) to transmit the quantizer index. In the remainder of the block, for time instants $t = kl + h + 1, \ldots, (k+1)l$, the randomly chosen quantizer $Q^{(k)}$ is used to encode the source symbols $x_i$. While the index of $Q^{(k)}$ is transmitted, the decoder emits $\hat{x}_i = 1/2$. If the description of $Q^{(k)}$ can be reconstructed at or before the time index $t = kl+h$, $Q^{(k)}$ is used to decode the received channel symbols in the remainder of the block. Otherwise, the decoder emits $1/2$ in the entire block.

The distribution for the random choice of $Q^{(k)}$ is the same as in Section III with the modification introduced in Theorem 5. That is, when computing $Q^{(k)}$, the source sequence $x^n$ is finely quantized using a $2K$-level uniform quantizer as $\bar{x}_i = q_{2K}(x_i)$. For any $k' \le k$, let

$$Z_{k',k} = \sum_{Q \in \mathcal{Q}_K^{\mathrm{MD}}} e^{-\eta \sum_{j=k'}^{k} \sum_{i=(j-1)l+h+1}^{jl} d_Q(\bar{x}_i)} \tag{40}$$

where the distortion $d_Q$ is defined in (36). Furthermore, let $W_1 = 1$, and for $k = 2, \ldots, n/l$

$$W_{k+1} = \frac{\alpha}{N} \sum_{k'=2}^{k} (1-\alpha)^{k-k'} W_{k'} Z_{k',k} + \frac{(1-\alpha)^{k-1}}{N} Z_{1,k} \tag{41}$$

(recall that $N = |\mathcal{Q}_K^{\mathrm{MD}}|$). Then, according to Algorithm 2, we get (42) and (43) shown at the bottom of the page.

The performance and complexity of the above coding scheme is analyzed in the next theorem.

*Theorem 6:* Assume that $m, n, l, K, M_1, M_2, h$ are positive integers such that $2M_j - 1 \le K$, $j = 1, 2$

$$h = \min_{j \in \{1,2\}} \left\lceil -\frac{(M_1M_2 + 2M_1 + 2M_2 - 1)\log K}{\log M_j} \times \left( \frac{\log(n/\log M_j)}{\log(1 - r_0 - r_j)} + 1 \right) \right\rceil \tag{44}$$

$l$ divides $n$, and $h \le l$. Then for any $0 < \alpha < 1$, $\eta > 0$, the normalized cumulative distortion of the above coding scheme can be bounded for any sequence $x^n \in [0,1]^n$ as

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \rho(|x_i - \hat{x}_i|)\right] - D^*_{\mathcal{Q}^{\mathrm{MD}},m,n}(x^n) \\ &\quad \le \frac{h\rho(1/2)}{l} + \frac{\rho(1/2)(M_1M_2 + 2M_1 + 2M_2 - 1)\log K}{n} \\ &\qquad + \frac{1}{\eta n}\ln\left(\frac{|\mathcal{Q}_K^{\mathrm{MD}}|^{m+1}}{\alpha^m (1-\alpha)^{n/l-m-1}}\right) + \frac{\eta l^2}{8} + \frac{ml}{n} + \frac{13c_\rho}{2K}. \end{aligned} \tag{45}$$

The algorithm can be implemented with $O(M_1M_2K^5n^2/l^2) + O(M_1M_2K^6n/l) + O(n)$ computational complexity.

*Remark:* Optimizing the above bound with respect to $\eta$ and $\alpha$ as after Theorem 5, we obtain

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \rho(|x_i - \hat{x}_i|)\right] - D^*_{\mathcal{Q}^{\mathrm{MD}},m,n}(x^n) \\ &\quad \le C_1 \frac{\log K \log n}{l} + C_2 \frac{\log K}{n} \\ &\qquad + C_3 \sqrt{\frac{lm}{n} \log \frac{n}{lm}} + \frac{13c_\rho}{2K} + \frac{ml}{n} \end{aligned}$$

with suitable positive constants $C_1, C_2$, and $C_3$. From here the best possible rate achievable is $O((m/n)^{1/3}\log(n/m)\log n)$, when $l = c_1(n/m)^{1/3}$ and $K = c_2(n/m)^{1/3}$ (again, $c_1$ and $c_2$ are positive constants), which requires $O(M_1M_2n^3/m)$ computations.

On the other hand, if we set $l = c_1 n^{7/9}/m^{1/3}$ and $K = c_2 n^{1/9}/m^{2/15}$, then the computational complexity of the algorithm is $O(M_1M_2n)$ and the normalized distortion redundancy becomes $O(m^{1/3}\sqrt{\log n}/n^{1/9})$.

*Proof of Theorem 6:* The proof follows the lines of the proof of Theorem 5. However, the algorithm is more complicated, and in proving the performance bound we have to consider the problem that the description of $Q^{(k)}$ may not be received at the decoder.

Let $\epsilon$ denote the probability that the index of $Q^{(k)}$ cannot be decoded after receiving the first $h$ symbols of the $k$th block (note that this probability is the same for each block as the channels are memoryless, and $Q^{(k)}$ and $Q^{(k')}$ can be decoded independently for $k \neq k'$). Then the decoder emits $\hat{x}_i = 1/2$ in the

$$\mathbb{P}\{\tau_k = k'\} = \begin{cases} \frac{\alpha(1-\alpha)^{k-k'} W_{k'} Z_{k',k-1}}{NW_k}, & \text{for } k' = 2, \ldots, k \\ \frac{(1-\alpha)^{k-1} Z_{1,k-1}}{NW_k}, & \text{for } k' = 1 \end{cases} \tag{42}$$

and

$$\mathbb{P}\{Q^{(k)} = Q | \tau_k = k'\} = \begin{cases} \frac{e^{-\eta \sum_{j=k'}^{k-1} \sum_{i=(j-1)l+h+1}^{jl} d_Q(\bar{x}_i)}}{Z_{k',k-1}}, & \text{for } k' = 1, \ldots, k-1 \\ \frac{1}{N}, & \text{for } k' = k \end{cases} \tag{43}$$

entire block, and the per letter distortion is bounded by $\rho(1/2)$. Hence

$$\begin{aligned}\mathbb{E}&\left[\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\right]\\ &=\sum_{k=1}^{n/l}\mathbb{E}\left[\sum_{i=(k-1)l+1}^{kl}\rho(|x_i-\hat{x}_i|)\right]\\ &\le\sum_{k=1}^{n/l}\left(\mathbb{E}\left[\sum_{i=(k-1)l+1}^{kl}\rho(|x_i-\hat{x}_i|)\right.\right.\\ &\qquad\left.\left.\middle|\, Q^{(k)}\text{can be decoded}\right]+\epsilon l\rho(1/2)\right)\\ &=\mathbb{E}\left[\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\,\middle|\, Q^{(k)}\text{ can be decoded for all }k\right]\\ &\quad+\epsilon n\rho(1/2).\end{aligned}\tag{46}$$

If $Q^{(k)}$ can be decoded at the receiver for all $k$, then, similarly to (34) and (35), it can be shown (using the fact that all interval cells are closed from the right) that

$$\begin{aligned}&\left|\left(\frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\,\middle|\, Q^{(k)}\text{ can be decoded for all }k\right]\right.\right.\\ &\left.-D^*_{\mathcal{Q}_K^{\mathrm{MD}},m,n}(x^n)\right)\\ &-\left(\frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\rho(|\bar{x}_i-\hat{x}_i|)\,\middle|\, Q^{(k)}\text{ can be decoded for all }k\right]\right.\\ &\left.\left.-D^*_{\mathcal{Q}_K^{\mathrm{MD}},m,n}(\bar{x}^n)\right)\right|\le\frac{c_\rho}{2K}.\end{aligned}\tag{47}$$

Also, since the same quantizer encodes $x_i$ and $\bar{x}_i$ into the same channel symbols, if $Q^{(k)}$ can be decoded for all $k$, then the coding procedure is a special case of Theorem 3 with input $\bar{x}^n$ (note that the explicit value of $h$ is never used in the proof of Theorem 3). Therefore

$$\begin{aligned}&\frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\rho(|\bar{x}_i-\hat{x}_i|)\,\middle|\, Q^{(k)}\text{ can be decoded for all }k\right]\\ &-D^*_{\mathcal{Q}_K^{\mathrm{MD}},m,n}(\bar{x}^n)\le\frac{h\rho(1/2)}{l}+\frac{1}{\eta n}\ln\left(\frac{|\mathcal{Q}_K^{\mathrm{MD}}|^{m+1}}{\alpha^m(1-\alpha)^{n/l-m-1}}\right)\\ &\qquad+\frac{\eta(l-h)^2}{8l}+\frac{m(l-1)}{n}.\end{aligned}\tag{48}$$

Combining (46)–(48) with (38) implies

$$\begin{aligned}\mathbb{E}&\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\right]-D^*_{\mathcal{Q}^{\mathrm{MD}},m,n}(x^n)\\ &\le\frac{h\rho(1/2)}{l}+\frac{1}{\eta n}\ln\left(\frac{|\mathcal{Q}_K^{\mathrm{MD}}|^{m+1}}{\alpha^m(1-\alpha)^{n/l-m-1}}\right)\\ &\quad+\frac{\eta(l-h)^2}{8l}+\frac{m(l-1)}{n}+\epsilon\rho(1/2)+\frac{13c_\rho}{2K}.\end{aligned}\tag{49}$$

In order to complete the proof of (45) we need to bound the error probability $\epsilon$. Since on channel $j$ it takes $\lceil\log|\mathcal{Q}_K^{\mathrm{MD}}|/\log M_j\rceil$ symbols to transmit the index of a quantizer, in $h$ channel symbols the quantizer index can be repeated $\left\lfloor\frac{h\log M_j}{\log|\mathcal{Q}_K^{\mathrm{MD}}|}\right\rfloor$ times. Since the channel is memoryless and the probability that a symbol is not received is $1-r_0-r_j$, the probability that each symbol of the description of the quantizer is received at least once is

$$\begin{aligned}&\left(1-(1-r_0-r_j)^{\left\lfloor\frac{h\log M_j}{\log|\mathcal{Q}_K^{\mathrm{MD}}|}\right\rfloor}\right)^{\left\lceil\frac{\log|\mathcal{Q}_K^{\mathrm{MD}}|}{\log M_j}\right\rceil}\\ &\qquad\ge 1-\left\lceil\frac{\log|\mathcal{Q}_K^{\mathrm{MD}}|}{\log M_j}\right\rceil(1-r_0-r_j)^{\left\lfloor\frac{h\log M_j}{\log|\mathcal{Q}_K^{\mathrm{MD}}|}\right\rfloor}\end{aligned}$$

as $(1-x)^k\ge 1-kx$ for all $x>0$ and $k>0$. Now from (39) it follows that

$$\left\lceil\frac{\log|\mathcal{Q}_K^{\mathrm{MD}}|}{\log M_j}\right\rceil\le\frac{(M_1M_2+2M_1+2M_2-1)\log K}{\log M_j}$$

and

$$\left\lfloor\frac{h\log M_j}{\log|\mathcal{Q}_K^{\mathrm{MD}}|}\right\rfloor\ge\frac{h\log M_j}{(M_1M_2+2M_1+2M_2-1)\log K}-1.\tag{50}$$

Now for the $j^*$ realizing the minimum in the definition of $h$ given in (44), the right-hand side of (50) is $-\log(n/\log M_{j^*})/\log(1-r_0-r_{j^*})$ (note that this quantity is positive since $1-r_0-r_{j^*}<1$). Therefore, as $\epsilon$ is no more than the probability of not receiving the quantizer index on channel $j^*$

$$\epsilon\le\frac{(M_1M_2+2M_1+2M_2-1)\log K}{n}.$$

Combining this with (49) proves (45).

Next we consider the implementation complexity. Although somewhat more complicated than for traditional scalar quantization, it is still possible in the algorithm to reduce the random choice of a multiple description quantizer to the problem of finding a minimum-weight path in a directed graph. In a related work, Muresan and Effros [28] showed that the problem of optimal entropy coded multiple description scalar quantizer design can be reduced to the problem of finding a minimum-weight path in an appropriately defined graph. In the following, we modify this method to fit in our scheme of online design.

First, observe that the algorithm of Section IV for finding a minimum-weight path in a directed graph can be extended trivially to graphs with multiple edges (where each edge may have a different weight). This follows from the fact that the probability of choosing an edge from a given vertex depends only on the relative weight of the paths that go through this edge from the given vertex, and no other property of the edges is used. Therefore, the algorithm works for such graphs in exactly the same way, with no change in the redundancy and complexity (which, however, depend on the increased number of the edges that includes the multiple edges).

Also note that it is possible to choose the constant $\hat{x}^*$ of the quantizer independently of the side and central quantizers.

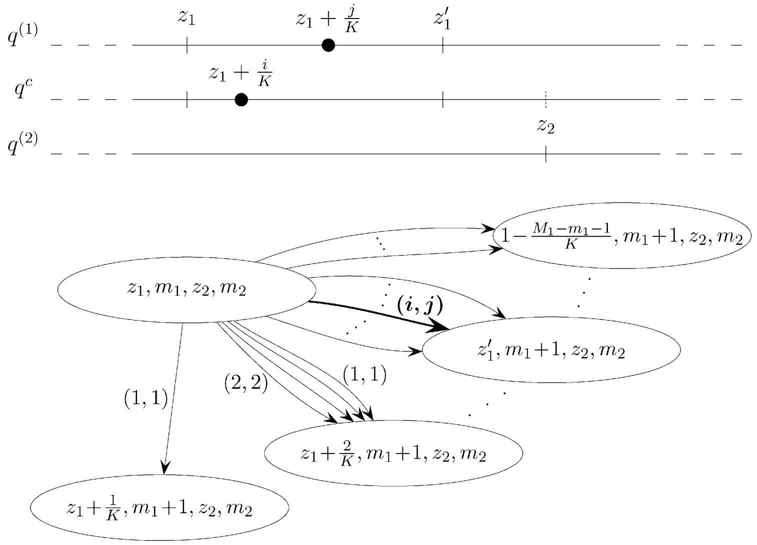

Fig. 5. A section of a multiple description scalar quantizer and the corresponding graph.

Choosing $\hat{x}^*$ from $C^{(K)}$ corresponds to a graph with two vertices, 0 and 1, with $K$ edges from 0 to 1 such that edge $e_j, j = 1, \ldots, K$, corresponds to $\hat{x}^* = (2j-1)/(2K)$, and the weight of $e_j$ at time $k$ (that is, in the $k$th block) is

$$\delta_k(e_j) = \sum_{i=(k-1)l+h+1}^{kl} \rho(|\bar{x}_i - (2j-1)/(2K)|).$$

Next we choose the central and side quantizers. For simplicity, we will refer to this triplet as a multiple description quantizer (and thus exclude the constant $\hat{x}^*$ from the problem). Consider a graph with vertices labeled $(z_1, m_1, z_2, m_2)$, where $z_j \in \hat{C}^{(K)} \cup \{0,1\}$ and $m_j \in \{0, \ldots, M_j\}$, $j = 1, 2$, such that $m_j = M_j$ if and only if $z_j = 1$. The vertex $(z_1, m_1, z_2, m_2)$ corresponds to the situation that the left endpoint of the $m_1$th cell of the first side quantizer is $z_1$ and the left endpoint of the $m_2$th cell of the second side quantizer is $z_2$. Following an edge from a vertex will correspond to adding a cell to the side quantizer $j$ whose $m_j$th cell lies more to the left (i.e., $z_j \le z_{3-j}$). Assume that $z_1 < z_2$ (note that in this case we necessarily have $m_1 < M_1$). Then there is an edge from $v = (z_1, m_1, z_2, m_2)$ to each vertex $v' = (z_1', m_1 + 1, z_2, m_2)$ such that $z_1 < z_1' < 1$ if $m_1 < M_1 - 1$, and $z_1' = 1$ if $m_1 = M_1 - 1$. If $z_1' \le z_2$, then an edge corresponds to the case that the next cell of the first side quantizer and that of the central quantizer is $(z_1, z_1']$ (except when $z_1 = 0$, in which case it is $[z_1, z_1']$). In the sequel, for simplicity, we do not consider $z_1 = 0$ or $z_1' = 1$; the definitions can be extended to this situation in a straightforward manner. The corresponding reproduction point in each quantizer can be any point of $\hat{C}^{(K)} \cup \{0,1\}$ which lies between $z_1$ and $z_1'$. Therefore, we need $K^2(z_1' - z_1)^2$ edges, such that edge $(v, v')_{i,j}$, $i, j \in \{1, \ldots, K(z_1' - z_1)\}$ corresponds to the situation that the new reproduction point of the central quantizer is $z_1 + i/K$, and that of the first side quantizer is $z_1 + j/K$ (see Fig. 5). Consequently, the corresponding weights in the $k$th block are the empirical distortions shown in (51) at the bottom of the page.

If $z_1' > z_2$, then the corresponding cell of the central quantizer is $(z_1, z_2]$, and the corresponding cell of the first side quantizer is $(z_1, z_1']$. Now the possible set of reproduction points for the central partition is $\{z_1+1/K, z_1+2/K, \ldots, z_2\}$ and for the first side quantizer $\{z_1 + 1/K, z_1 + 2/K, \ldots, z_1'\}$. Therefore, there are $K^2(z_2 - z_1)(z_1' - z_1)$ possible edges, and edge $(v, v')_{i,j}$, $i \in \{1, \ldots, K(z_2 - z_1)\}$, $j \in \{1, \ldots, K(z_1' - z_1)\}$, corresponds to central and side reproduction points $z_1 + i/K$ and $z_1 + j/K$, respectively, with corresponding weight given again by (51). The formula can be modified straightforwardly for $z_1 = 0$ and $z_1' = 1$. The edges and weights are similarly defined for $z_1 > z_2$.

If $z_1 = z_2$, then adding a cell to any side quantizer will determine the cell of only that quantizer but not any cell of the central quantizer. In this situation, we always choose to extend the first side quantizer, so from $v = (z, m_1, z, m_2)$ there are $K(z' - z)$ edges to $v' = (z', m_1 + 1, z, m_2)$ for every $z' > z$, and the weight of edge $(v, v')_i, i \in \{1, \ldots, K(z' - z)\}$ corresponds to reproduction point $z + i/K$ and has weight

$$\delta_k((v, v')_i) = r_1 \sum_{j=(k-1)l+h+1}^{kl} I_{\{z < \bar{x}_j \le z'\}} \rho(|\bar{x}_j - z - i/K|).$$

It is not hard to see that there is a one-to-one correspondence between paths from $(0,0,0,0)$ to $(1, M_1, 1, M_2)$ and multiple description quantizers from $\mathcal{Q}_K^{\mathrm{MD}}$ (with the constant $\hat{x}^*$ yet undefined), such that the weight of a path is the same as the distortion of the corresponding quantizer. Therefore, finding a quantizer according to the probabilities in (42) and (43) is equivalent to finding a path from vertex $(0,0,0,0)$ to

$$\delta_k\Big((v, v')_{i,j}\Big) = \sum_{t=(k-1)l+h+1}^{kl} I_{\{z_1 < \bar{x}_t \le z_1'\}} \Big(r_0 \rho(|\bar{x}_t - z_1 - i/K|) + r_1 \rho(|\bar{x}_t - z_1 - j/K|)\Big). \quad (51)$$

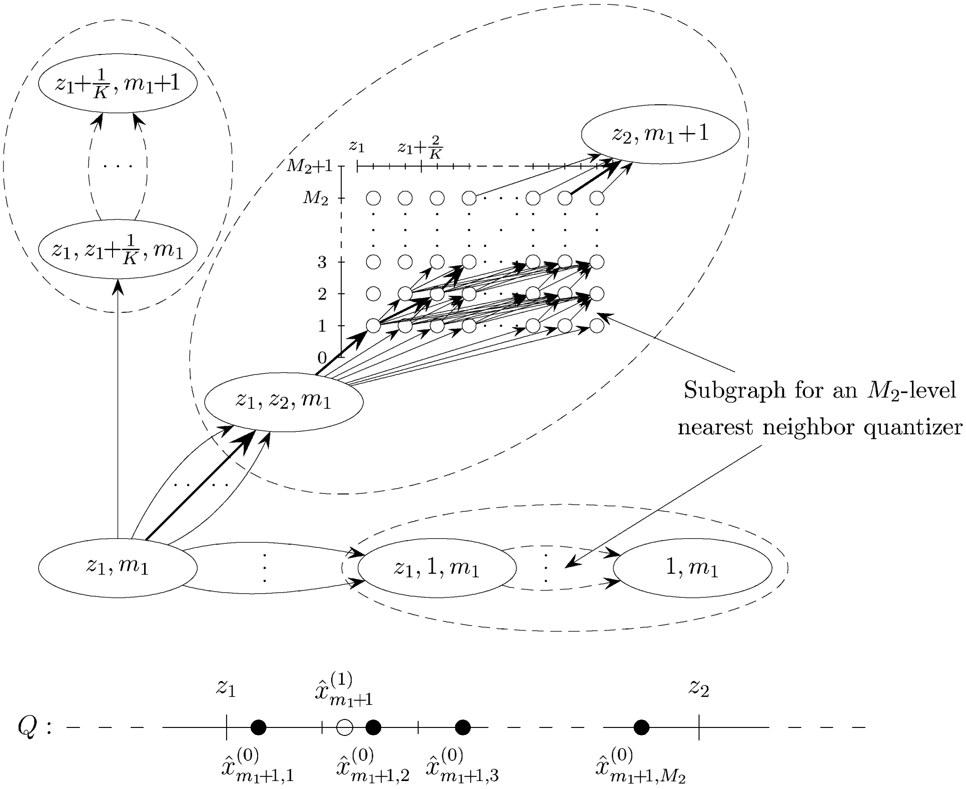
$z_1+\frac{1}{K}, m_1+1$
$z_1, z_1+\frac{1}{K}, m_1$
$z_2, m_1+1$
$z_1$
$z_1+\frac{2}{K}$
$M_2+1$
$M_2$
3
2
1
0
$z_1, z_2, m_1$
Subgraph for an $M_2$-level nearest neighbor quantizer
$z_1, m_1$
$z_1, 1, m_1$
$1, m_1$
$Q$ :
$z_1$
$\hat{x}^{(1)}_{m_1+1}$
$z_2$
$\hat{x}^{(0)}_{m_1+1,1}$
$\hat{x}^{(0)}_{m_1+1,2}$
$\hat{x}^{(0)}_{m_1+1,3}$
$\hat{x}^{(0)}_{m_1+1,M_2}$

Fig. 6. A section of a multiple description scalar quantizer and the corresponding graph.

vertex $(1, M_1, 1, M_2)$, and the algorithm of Section IV can be used to solve this problem. Since the number of edges in the constructed graph is $O(M_1 M_2 K^5)$, and the weight of each edge can be computed in $O(K)$ time (as in Theorem 5), the required time complexity of the algorithm is $O(M_1 M_2 K^5 n^2/l^2) + O(M_1 M_2 K^6 n/l) + O(n)$. □

### *B. Adaptive Multiresolution Scalar Quantization*

Multiresolution quantization is a special case of multiple description quantization with $r_0 + r_1 = 1$. However, recall that for multiresolution quantizers we assume that the second side quantizer $q^{(2)}$ restricted to a cell of $q^{(1)}$ is a nearest neighbor quantizer. Although this assumption is not compatible with our earlier assumption on the cell structure of multiple description quantizers, it allows, through similar methods, a simpler graph representation, and results in an algorithm with somewhat reduced complexity.

Similarly to the multiple description case, first we find a fine covering of the allowed multiresolution quantizers: Let $\mathcal{Q}_K^{\mathrm{MR}} \subset \mathcal{Q}^{\mathrm{MR}}$ denote the set of multiresolution scalar quantizers such that the cells of the first side quantizer (which is often referred to as the base quantizer) and the cells of the second side quantizer (the refinement quantizer) restricted to the cells of the base quantizer are right closed intervals with endpoints from $\hat{C}^{(K)}$, all the reproduction points are also from $\hat{C}^{(K)}$, and belong to the corresponding interval cell.

Then, similarly to Lemma 2, it can be shown that for any quantizer $q \in \mathcal{Q}^{\mathrm{MR}}$ there is a $q' \in \mathcal{Q}_K^{\mathrm{MR}}$ such that

$$\sup_{x\in[0,1]} (d_{q'}(x) - d_q(x)) \le 3c_\rho/K. \tag{52}$$

(see the footnote in the proof of Lemma 2).

This result allows us to compete with best code in $\mathcal{Q}^{\mathrm{MR}}$ by applying the coding scheme of Section III to the finite set of codes $\mathcal{Q}_K^{\mathrm{MR}}$. There are two differences compared to the general multiple description case: i) Since there is no loss on the first channel, it is enough to send the index of the chosen quantizer $Q^{(k)}$ only once in each block, requiring

$$\begin{aligned} h &= \left\lceil \frac{\log|\mathcal{Q}_K^{\mathrm{MR}}|}{\log M_1} \right\rceil \\ &\le \left\lceil \frac{1}{\log M_1} \log\left( \binom{K}{M_1-1} \binom{K}{M_1} \binom{K}{M_1 M_2} \right) \right\rceil \end{aligned} \tag{53}$$

time instants (recall that the second side quantizer restricted to a cell of the base quantizer is assumed to be a nearest neighbor quantizer). ii) The simpler structure of $\mathcal{Q}_K^{\mathrm{MR}}$ allows a smaller graph representation. Indeed, the graph describing quantizers from $\mathcal{Q}_K^{\mathrm{MR}}$ can be constructed as follows: Define vertices $(z, m_1)$ for each $z \in \hat{C}^{(K)} \cup \{0,1\}$ and $m_1 \in \{0, \dots, M_1\}$. The vertex $(z, m_1)$ corresponds to the case that the right endpoint of the $m_1$th cell of $q^{(1)}$ is $z$ (recall that the cells of $q^{(1)}$ are intervals). Now vertices $(z_1, m_1)$ and $(z_2, m_1+1)$ are connected via the following subgraph. The first vertex of the subgraph is $(z_1, z_2, m_1)$, and there are $K(z_2 - z_1)$ edges going from $(z_1, m_1)$ to $(z_1, z_2, m_1)$, each corresponding to a different possible code point of the cell from the set $(z_1, z_2] \cap \hat{C}^{(K)}$ with weight corresponding to the distortion of this cell in the $k$th block (if $z_1 = 0$, then there are $Kz_2 + 1$ edges, where the extra edge corresponds to the code point $z_1 = 0$, which is a valid code point only in this case). Then $(z_1, z_2, m_1)$ and $(z_2, m_1 + 1)$ are connected via a directed graph corresponding to an $M_2$-level nearest neighbor quantizer from $z_1$ to $z_2$ as in Theorem 5, but the weights of the edges are multiplied by $r_0$ (see Fig. 6). The number of edges of such a subgraph is

$O(M_2K^2)$; hence the total number of edges of the constructed graph is $O(M_1M_2K^4)$.

Therefore, the complexity of the algorithm is slightly reduced compared to the general multiple description case, as it requires only $O(M_1M_2K^4n^2/l^2) + O(M_1M_2K^5n/l) + O(n)$ computations. From here, similarly to Theorem 6, we obtain the following result.

*Theorem 7:* Assume that $m, n, l, K, M_1, M_2, h$ are positive integers such that $M_1M_2 \le K$, $l$ divides $n$, and $h \le l$, where $h$ is defined in (53). Then for any $0 < \alpha < 1$, $\eta > 0$, the normalized cumulative distortion of the above described coding scheme can be bounded for any sequence $x^n \in [0,1]^n$ as

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\right] &- D^*_{\mathcal{Q}^{\mathrm{MR}},m,n}(x^n)\\ &\le \frac{h\rho(1/2)}{l} + \frac{1}{\eta n}\ln\left(\frac{|\mathcal{Q}_K^{\mathrm{MR}}|^{m+1}}{\alpha^m(1-\alpha)^{n/l-m-1}}\right)\\ &\quad + \frac{\eta l^2}{8} + \frac{ml}{n} + \frac{7c_\rho}{2K}.\end{aligned}$$

The algorithm can be implemented with

$$O(M_1M_2K^4n^2/l^2) + O(n^3/l^3) + O(M_1M_2K^5n/l) + O(n)$$

computational complexity.

*Remark:* Optimizing the above bound as after Theorem 5, we obtain

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\rho(|x_i-\hat{x}_i|)\right] &- D^*_{\mathcal{Q}^{\mathrm{MR}},m,n}(x^n)\\ &\le C_1\frac{\log K}{l} + C_2\sqrt{\frac{lm}{n}\log\frac{n}{lm}} + \frac{7c_\rho}{2K} + \frac{ml}{n}\end{aligned}$$

where $C_1$ and $C_2$ are suitable positive constants. From here the best possible rate achievable is $O((m/n)^{1/3}\log(n/m))$, when $l = c_1(n/m)^{1/3}$ and $K = c_2(n/m)^{1/3}$ ($c_1$ and $c_2$ are arbitrary positive constants), which requires $O(M_1M_2n^{8/3}/m^{2/3})$ computations. On the other hand, if we set $l = c_1n^{3/4}/m^{1/2}$ and $K = c_2n^{1/8}/m^{1/4}$, then the computational complexity of the algorithm is $O(M_1M_2n)$, and the normalized distortion redundancy becomes $O(m^{1/4}\sqrt{\log n}/n^{1/8})$.

## VII. Conclusion

We presented a general scheme for limited-delay lossy coding of individual sequences. For any finite class of limited-delay, finite-memory reference coders, our scheme performs asymptotically as well as the best "tracking" scheme that can change its coder a given number of times (each time choosing from the reference class) while encoding an input sequence of finite length. In order to implement the method, we devised an efficient algorithm for online prediction that tracks the best expert even when the number of experts is exponentially large, provided the experts have a certain additive structure. The example of tracking the minimum-weight path in an acyclic graph was worked out in detail and used to construct efficient quantization schemes. In particular, we have constructed low-complexity schemes for tracking the best scalar quantizer, as well as the best multiple description or multiresolution quantizer (among the ones with interval cells). For all these schemes, analyses of distortion redundancy and computational complexity were provided.

The focus of this work was on implementable sequential lossy coding schemes based on ideas rooted in the theory of sequential prediction for individual sequences. While the considered combined (i.e., tracking) scalar (network) quantization schemes form more powerful classes than the base class of scalar (network) quantizers, these schemes are still less general than one would desire in certain applications. For example, efficiently implementable schemes for the class of coders with finite-state encoders and sliding-window (or finite-state) decoders would by desirable [2]. Also of interest would be the construction of linear-time scalar (network) quantization schemes with the same distortion redundancy rates as that of the more complex schemes presented here.

## Appendix A

*Proof of Theorem 1:* The proof of Theorem 1 is done through a sequence of lemmas. First observe that, denoting by $\mathbb{E}_t$ the expectation taken with respect to $U_t$ only, the sequence

$$\begin{aligned}V_t &= \ell(y_t,\hat{y}_t) - \mathbb{E}_t\ell(y_t,\hat{y}_t)\\ &= \ell(y_t,\hat{y}_t) - \mathbb{E}[\ell(y_t,\hat{y}_t)|U^{t-1}], \quad t = 1,\dots,T\end{aligned}$$

is a martingale difference with respect to $U_1, U_2, \dots, U_T$ such that with probability one

$$-\mathbb{E}[\ell(y_t,\hat{y}_t)|U^{t-1}] \le V_t \le -\mathbb{E}[\ell(y_t,\hat{y}_t)|U^{t-1}] + B.$$

Thus, it suffices to bound the difference

$$\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t) - \min_{\boldsymbol{t},\boldsymbol{e}} L(\mathcal{P}(T,m,\boldsymbol{t},\boldsymbol{e}))$$

since

$$\sum_{t=1}^{T} V_t = L_T - \sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)$$

and so by the Hoeffding–Azuma inequality [42] for sums of martingale differences, with probability at least $1-p$

$$L_T \le \sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t) + B\sqrt{\frac{T\ln(1/p)}{2}}$$

and

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)\right] = \sum_{t=1}^{T}\mathbb{E}\ell(y_t,\hat{y}_t).$$

*Lemma 3:* The cumulative loss of Algorithm 1 satisfies

$$\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t) \le -\frac{1}{\eta}\ln w^m_{T,i} + \frac{T\eta B^2}{8}$$

for all $i = 1,\dots,N$.

*Proof:* The proof follows standard arguments, see, e.g., Cesa-Bianchi and Lugosi [43]. From the definition of $W_t$ we have for all $t$

$$\begin{aligned}\ln\frac{W_{t+1}}{W_t} &= \ln\left(\frac{\sum_{i=1}^{N} w^s_{t,i}e^{-\eta\ell(y_t,\hat{y}_t^{(i)})}}{\sum_{i=1}^{N} w^s_{t,i}}\right)\\ &= \ln\left(\mathbb{E}_t e^{-\eta\ell(y_t,\hat{y}_t)}\right)\end{aligned}$$

$$\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)-\min_{\boldsymbol{t},\boldsymbol{e}}L(\mathcal{P}(T,m,\boldsymbol{t},\boldsymbol{e}))\le B\sqrt{\frac{T}{2}\left((m+1)\ln N+m\ln\frac{1}{\alpha}+(T-m-1)\ln\frac{1}{1-\alpha}\right)}$$
$$=B\sqrt{\frac{T}{2}\left((m+1)\ln N+(T-1)(D_b(\alpha^*\,\|\,\alpha)+H_b(\alpha^*))\right)}$$

$$=\ln\left(e^{-\eta\mathbb{E}_t\ell(y_t,\hat{y}_t)}\mathbb{E}_te^{-\eta(\ell(y_t,\hat{y}_t)-\mathbb{E}_t\ell(y_t,\hat{y}_t))}\right)$$
$$=-\eta\mathbb{E}_t\ell(y_t,\hat{y}_t)+\ln\left(\mathbb{E}_te^{-\eta(\ell(y_t,\hat{y}_t)-\mathbb{E}_t\ell(y_t,\hat{y}_t))}\right)$$
$$\le-\eta\mathbb{E}_t\ell(y_t,\hat{y}_t)+\frac{\eta^2B^2}{8}.$$

Note that the expectations are taken with respect to the random choice of $\hat{y}_t$, that is, with respect to the randomizing variable $U_t$, and the inequality holds by Hoeffding's inequality [42] for the moment generation function of bounded random variables. Summing the above inequality for all $t=1,\dots,T$, we obtain

$$\ln\frac{W_{T+1}}{W_1}\le-\eta\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)+\frac{T\eta^2B^2}{8}.$$

Since $W_1=1$ and $W_{T+1}\ge w_{T,i}^m$ for any $i$, we have

$$\ln w_{T,i}^m\le-\eta\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)+\frac{T\eta^2B^2}{8}$$

which implies the statement of the lemma. □

*Lemma 4:* For any $1\le t\le t'\le T$ and any $i=1,\dots,N$ we have

$$\frac{w_{t',i}^m}{w_{t,i}^s}\ge e^{-\eta L([t,t'],i)}(1-\alpha)^{t-t'}$$

where $L([t,t'],i)=\sum_{\tau=t}^{t'}\ell(y_\tau,\hat{y}_\tau^{(i)})$.

*Proof:* The proof is a straightforward modification of the one in [21]. From the definitions of $w_{t,i}^m$ and $w_{t+1,i}^s$ (see (2) and (3)) it is clear that for any $\tau\ge1$

$$w_{\tau+1,i}^s=\frac{\alpha W_{\tau+1}}{N}+(1-\alpha)w_{\tau,i}^m\ge(1-\alpha)e^{-\eta\ell(y_\tau,\hat{y}_\tau^{(i)})}w_{\tau,i}^s.$$

Applying this equation iteratively for $\tau=t,t+1,\dots,t'-1$, and (2) for $\tau=t'$, we obtain

$$w_{t',i}^m\ge e^{-\eta\ell(y_{t'},\hat{y}_{t'}^{(i)})}\prod_{\tau=t}^{t'-1}\left((1-\alpha)e^{-\eta\ell(y_\tau,\hat{y}_\tau^{(i)})}\right)w_{t,i}^s$$
$$=e^{-\eta L([t,t'],i)}(1-\alpha)^{t-t'}w_{t,i}^s$$

which implies the statement of the lemma. □

*Lemma 5:* For any $t\ge1$ and $1\le i,j\le N$, we have

$$\frac{w_{t+1,i}^s}{w_{t,j}^m}\ge\frac{\alpha}{N}.$$

*Proof:* By the definition of $w_{t+1,i}^s$ we have

$$w_{t+1,i}^s\ge\frac{\alpha W_{t+1}}{N}=\frac{\alpha}{N}\sum_{l=1}^{N}w_{t,l}^m\ge\frac{\alpha w_{t,j}^m}{N}.$$

This completes the proof of the lemma. □

*Proof of Theorem 1:* Let $\mathcal{P}(T,m,\boldsymbol{t},\boldsymbol{e})$ be an arbitrary partition. Then, by Lemma 3, the cumulative loss of Algorithm 1 can be bounded as

$$\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)\le-\frac{1}{\eta}\ln w_{T,e_m}^m+\frac{T\eta B^2}{8}\qquad\text{(A1)}$$

(recall that $e_m$ denotes the expert used in the last segment of the partition). Now $w_{T,e_m}^m$ can be rewritten in the form of the following telescoping product:

$$w_{T,e_m}^m=w_{t_0+1,e_0}^s\frac{w_{t_1,e_0}^m}{w_{t_0+1,e_0}^s}\prod_{i=1}^{m}\left(\frac{w_{t_i+1,e_i}^s}{w_{t_i,e_{i-1}}^m}\frac{w_{t_{i+1},e_i}^m}{w_{t_i+1,e_i}^s}\right).$$

Therefore, applying Lemmas 4 and 5, we have

$$w_{T,e_m}^m\ge w_{t_0+1,e_0}^s\left(\frac{\alpha}{N}\right)^m$$
$$\times\prod_{i=0}^{m}\left(e^{-\eta L((t_i,t_{i+1}],e_i)}(1-\alpha)^{t_{i+1}-t_i-1}\right)$$
$$=\frac{1}{N}e^{-\eta L(\mathcal{P}(T,m,\boldsymbol{t},\boldsymbol{e}))}(1-\alpha)^{T-m-1}\left(\frac{\alpha}{N}\right)^m.$$

Substituting this bound in (A1) proves (4) and (5), the first statements of the theorem.

To prove the second part, let

$$H_b(p)=-p\ln p-(1-p)\ln(1-p)$$

and

$$D_b(p\,\|\,q)=p\ln\frac{p}{q}+(1-p)\ln\frac{1-p}{1-q}.$$

Optimizing the value of $\eta$ in (4) gives the equation shown at the top of the page, where $\alpha^*=\frac{m}{T-1}$. For $\alpha=\alpha^*$, the bound becomes the equation shown at the top of the following page, where we used the fact that $\ln(1+x)\le x$ for all $x>-1$. □

## Appendix B

*Proof of Lemma 2:* In view of (32), the proof would be simple if only nearest neighbor quantizers were allowed as side quantizers. As this is not the case, a more involved construction is needed.

For $i=1,2$, let $0=t_0^{(i)}<t_1^{(i)}<\cdots<t_{M_i-1}^{(i)}<t_{M_i}^{(i)}=1$ denote the decision thresholds of $q^{(i)}$, the $i$th side quantizer of $q$ (that is, the cells of $q^{(i)}$ are intervals with endpoints $t_{j-1}^{(i)}$ and $t_j^{(i)}$, $j=1,\dots,M_i$). We will construct a sequence of two-description quantizers $q_j$, $j=0,\dots,K$, such that all the thresholds $t$ of the side quantizers $q_j^{(i)}$ that satisfy $t\le j/K$ are of the form $t=k/K$ for some integer $k$, the corresponding cells with right endpoints $t$ are right closed intervals, and $q_{j+1}$ and $q_j$ differ only in the interval $(j/K,(j+1)/K]$.

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}_t\ell(y_t,\hat{y}_t)-\min_{\boldsymbol{t},\boldsymbol{e}}L(\mathcal{P}(T,m,\boldsymbol{t},\boldsymbol{e}))\\
&\quad\le\frac{BT^{1/2}}{\sqrt{2}}\sqrt{(m+1)\ln N+m\ln\frac{T-1}{m}+(T-m-1)\ln\left(1+\frac{m}{T-m-1}\right)}\\
&\quad\le T^{1/2}\frac{B}{\sqrt{2}}\sqrt{(m+1)\ln N+m\ln\frac{T-1}{m}+m}
\end{aligned}$$

Let $q_0=q$, which clearly satisfies that all thresholds $t\le 0$ are of the form $0/K$. Assume that we have already constructed $q_j$ with the desired property for some $j\ge 0$. From $q_j$, we construct $q_{j+1}$ by modifying the thresholds of the side quantizers of $q_j$ in the interval $(j/K,(j+1)/K]$. Therefore, for all $x\notin(j/K,(j+1)/K]$

$$d_{q_{j+1}}(x)-d_{q_j}(x)=0. \tag{B1}$$

Furthermore, if $q_j$ has no threshold in the interval $(j/K,(j+1)/K]$, then let $q_{j+1}=q_j$ (so obviously $d_{q_j}=d_{q_{j+1}}$). If $q_j$ has exactly one threshold in $(j/K,(j+1)/K)$, then without loss of generality we can assume that this threshold $t$ belongs to the first side quantizer $q_j^{(1)}$. Let $q_{\rm left}$ and $q_{\rm right}$ be 2-description quantizers obtained from $q_j$ by replacing the threshold $t$ with $j/K$ and $(j+1)/K$, respectively (such that the new threshold is quantized to a smaller code point; that is, the corresponding cell is closed from the right), and $q_{j+1}$ will be the one with smaller guaranteed worst case distortion. Let $c_1^{(1)},c_2^{(1)}$ denote the code points of the side quantizer $q_j^{(1)}$ corresponding to the cells ending and beginning at $t$, respectively, and let $c_1^{(0)},c_2^{(0)}$ denote the corresponding code points of the central quantizer $q_j^{(0)}$. Now clearly, if $x\le j/K$ or $x>t$, then $q_j(x)=q_{\rm left}(x)$. For $x\in(j/K,t)$, we have

$$\begin{aligned}
&d_{q_{\rm left}}(x)-d_{q_j}(x)\\
&\quad=r_0\left(\rho(|x-c_2^{(0)}|)-\rho(|x-c_1^{(0)}|)\right)\\
&\qquad+r_1\left(\rho(|x-c_2^{(1)}|)-\rho(|x-c_1^{(1)}|)\right)\\
&\quad\le r_0\left(\rho(|t-c_2^{(0)}|)-\rho(|t-c_1^{(0)}|)+2c_\rho(t-j/K)\right)\\
&\qquad+r_1\left(\rho(|t-c_2^{(1)}|)-\rho(|t-c_1^{(1)}|)+2c_\rho(t-j/K)\right)\\
&\quad\le r_0\left(\rho(|t-c_2^{(0)}|)-\rho(|t-c_1^{(0)}|)\right)\\
&\qquad+r_1\left(\rho(|t-c_2^{(1)}|)-\rho(|t-c_1^{(1)}|)\right)+2c_\rho/K
\end{aligned}\tag{B2}$$

and similarly, for $x<t$ or $x>(j+1)/K$, $q_j(x)=q_{\rm right}(x)$, and for $x\in(t,(j+1)/K)$

$$\begin{aligned}
&d_{q_{\rm right}}(x)-d_{q_j}(x)\\
&\quad=r_0\left(\rho(|x-c_1^{(0)}|)-\rho(|x-c_2^{(0)}|)\right)\\
&\qquad+r_1\left(\rho(|x-c_1^{(1)}|)-\rho(|x-c_2^{(1)}|)\right)\\
&\quad\le r_0\left(\rho(|t-c_1^{(0)}|)-\rho(|t-c_2^{(0)}|)\right)\\
&\qquad+r_1\left(\rho(|t-c_1^{(1)}|)-\rho(|t-c_2^{(1)}|)\right)+2c_\rho/K.
\end{aligned}\tag{B3}$$

It is easy to see that for $x=t$, either $q_j(x)=q_{left}(x)$ (resp., $q_j(x)=q_{\rm right}(x)$), or the bound (B2) (resp., (B3)) holds. Now let $q_{j+1}=q_{\rm left}$ if the bound (B2) is smaller than (B3), and let $q_{j+1}=q_{\rm right}$ otherwise. Then, as the first two terms in (B2) and (B3) are the negative of each other

$$d_{q_{j+1}}(x)-d_{q_j}(x)\le 2c_\rho/K. \tag{B4}$$

for all $x\in(j/K,(j+1)/K]$.

If one of the side quantizers of $q_j$ has at least one cell that is contained in the interval $(j/K,(j+1)/K]$, then in $q_{j+1}$, all such cells are merged and enlarged into the larger cell $(j/K,(j+1)/K]$ with reproduction point $(j+1/2)/K$ (consequently, the neighboring cells may shrink). If the other side quantizer has no threshold in $(j/K,(j+1)/K]$, then no other modification is necessary. If it has exactly one threshold, then similarly to (B2) and (B3), that threshold can be moved either to $j/K$ or to $(j+1)/K$. Finally, if the other side quantizer also has at least two thresholds in the interval $(j/K,(j+1)/K]$, then merging all cells in this interval as for the other side quantizer results in a quantizer $q_{j+1}$ for which $(j/K,(j+1)/K]$ is a cell of both side quantizers, and consequently, it is also a cell of the central quantizer. In all cases, the bound (B4) holds for the distortion of $q_{j+1}$.

The last case to be considered is when both side quantizers have exactly one threshold in $(j/K,(j+1)/K]$. Let $t_1\le t_2$ denote these thresholds. If $t_1=t_2$ and there is no central partition cell $\{t_1\}$, then, similarly to (B2) and (B3), it can be shown that replacing the common threshold of the side quantizers with $j/K$ or $(j+1)/K$ results in a maximum increase in distortion of $3c_\rho/K$. Note that the factor 2 in (B4) is replaced with 3 because here all three quantizers (both side quantizers and the central quantizer) change.[2] If $t_1<t_2$, then similarly to (B4), it can be shown that $t_1$ can be replaced with $j/K$ or $t_2$ at the price of an increase of at most $2c_\rho/K$ in the pointwise distortion (such that the new central quantizer has no one-point cell $\{t_2\}$). Then, the threshold $t_2$ (which may be a single or a common threshold) can be quantized into $\hat{C}^{(K)}$ as before on the price of further increasing the pointwise distortion by at most $3c_\rho/K$. Finally, if $t_1=t_2$ and the central partition has a one-point cell $\{t_1\}$, then similarly to the case $t_1<t_2$, first we can replace this cell by the empty cell or by $(j/K,t_2]$, and then proceed as in the previous case. Based on the above, we have, for all $x\in(j/K,(j+1)/K]$

$$d_{q_{j+1}}(x)-d_{q_j}(x)\le 5c_\rho/K. \tag{B5}$$

[2] When proving the analogous covering result (52) for the multiresolution case, the factor remains 2 since there are only two quantizers to be considered, and $t_1=t_2$ is the only possible subcase. In that case, this results in the slightly better $3c_\rho/K$ final upper bound instead of $6c_\rho/K$.

From equations (B1), (B4), and (B5) we can see that for all $x \in [0,1]$

$$d_{q_K}(x) - d_q(x) \leq 5c_\rho/K.$$

Now let $q' \in \mathcal{Q}_K^{\mathrm{MD}}$ be a quantizer obtained from $q_K$ by replacing its code points by the corresponding closest points in the set $\hat{C}^{(K)}$ that also belong to the corresponding quantization cells. Then

$$d_{q'}(x) - d_q(x) \leq \frac{5c_\rho}{K} + \frac{c_\rho}{K} = \frac{6c_\rho}{K}$$

for all $x \in [0,1]$, as desired. □

## References

[1] T. Linder and G. Lugosi, "A zero-delay sequential scheme for lossy coding of individual sequences," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2533–2538, Sep. 2001.

[2] T. Weissman and N. Merhav, "On limited-delay lossy coding and filtering of individual sequences," *IEEE Trans. Inf. Theory*, vol. 48, no. 3, pp. 721–733, Mar. 2002.

[3] J. Hannan, "Approximation to Bayes risk in repeated plays," in *Contributions to the Theory of Games*, M. Dresher, A. Tucker, and P. Wolfe, Eds. Princeton, NJ: Princeton Univ. Press, 1957, vol. 3, pp. 97–139.

[4] D. Blackwell, "An analog of the minimax theorem for vector payoffs," *Pacific J. Math.*, vol. 6, pp. 1–8, 1956.

[5] V. Vovk, "Aggregating strategies," in *Proc 3rd Annu. Workshop Computational Learning Theory (Rochester, NY, Aug. 1990)*. San Francisco, CA: Morgan Kaufmann, 1990, pp. 372–383.

[6] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.*, vol. 108, pp. 212–261, 1994.

[7] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[8] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. New York: Cambridge Univ. Press, 2006.

[9] R. E. Schapire and D. P. Helmbold, "Predicting nearly as well as the best pruning of a decision tree," *Mach. Learn.*, vol. 27, pp. 51–68, 1997.

[10] F. Pereira and Y. Singer, "An efficient extension to mixture techniques for prediction and decision trees," *Mach. Learn.*, vol. 36, pp. 183–199, 1999.

[11] M. Mohri, General Algebraic Frameworks and Algorithms for Shortest Distance Problems AT&T Labs Research, Tech. Rep. 981219-10TM, 1998.

[12] E. Takimoto and M. K. Warmuth, "Path kernels and multiplicative updates," in *Proc. 15th Annu. Conf. Computational Learning Theory, COLT 2002*, J. Kivinen and R. H. Sloan, Eds. Berlin, Heidelberg: Springer-Verlag, Jul. 2002, vol. LNAI 2375, pp. 74–89.

[13] E. Takimoto and M. K. Warmuth, "Path kernels and multiplicative updates," *J. Mach. Learn. Res.*, vol. 4, pp. 773–818, 2003.

[14] A. György, T. Linder, and G. Lugosi, "Efficient algorithms and minimax bounds for zero-delay lossy source coding," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2337–2347, Aug. 2004.

[15] A. Kalai and S. Vempala, "Efficient algorithms for the online decision problem," in *Proc. 16th Annu. Conf. Learning Theory and the 7th Kernel Workshop, COLT-Kernel 2003*, B. Schölkopf and M. Warmuth, Eds. New York: Springer-Verlag, Aug. 2003, pp. 26–40.

[16] A. György, T. Linder, and G. Lugosi, "A "follow the perturbed leader"-type algorithm for zero-delay quantization of individual sequences," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2004, pp. 342–351.

[17] S. Matloub and T. Weissman, "Universal zero-delay joint source-channel coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5240–5250, Dec. 2006.

[18] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2210–2217, Nov. 1996.

[19] G. I. Shamir and N. Merhav, "Low-complexity sequential lossless coding for piecewise-stationary memoryless sources," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1498–1519, Jul. 1999.

[20] G. I. Shamir and D. J. Costello, Jr., "Asymptotically optimal low-complexity sequential lossless coding for piecewise-stationary memoryless sources—Part I: The regular case," *IEEE Trans. Inform. Theory*, vol. 46, no. 7, pp. 2444–2467, Nov. 2000.

[21] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Mach. Learn.*, vol. 32, no. 2, pp. 151–178, 1998.

[22] P. Auer and M. K. Warmuth, "Tracking the best disjunction," *Mach. Learn.*, vol. 32, no. 2, pp. 127–150, 1998.

[23] V. Vovk, "Derandomizing stochastic prediction strategies," *Mach. Learn.*, vol. 35, pp. 247–282, 1999.

[24] O. Bousquet and M. K. Warmuth, "Tracking a small set of experts by mixing past posteriors," *J. Mach. Learn. Res.*, vol. 3, pp. 363–396, Nov. 2002.

[25] M. Herbster and M. K. Warmuth, "Tracking the best linear predictor," *J. Mach. Learn. Res.*, vol. 1, pp. 281–309, 2001.

[26] X. Wu and K. Zhang, "Quantizer monotonicities and globally optimal scalar quantizer design," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 1049–1053, May 1993.

[27] A. Aggarwal, B. Schieber, and T. Tokuyama, "Finding a minimum weight $K$-link path in graphs with Monge property and applications," in *Proc. 9th Annu. Symp. Computational Geometry*, San Diego, CA, May 1993, pp. 189–197.

[28] D. Muresan and M. Effros, "Quantization as histogram segmentation: Globally optimal scalar quantizer design in network systems," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 302–311.

[29] A. György, T. Linder, and G. Lugosi, "Tracking the best of many experts," in *Proc. 18th Annu. Conf. Learning Theory, COLT 2005 (Bertinoro, Italy, Jun. 2005)*. Berlin, Germany: Springer-Verlag, 2005, pp. 204–216.

[30] L. H. Ozarow, "On the source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, pp. 1909–1922, 1980.

[31] A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 6, pp. 851–857, Nov. 1982.

[32] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[33] W. E. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.

[34] B. Rimoldi, "Successive refinement of information: Characterization of the achievable rates," *IEEE Trans. Inf. Theory*, vol. 40, no. 1, pp. 253–259, Jan. 1994.

[35] M. Effros and D. Dugatkin, "Multiresolution vector quantization," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3130–3145, Dec. 2004.

[36] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 821–834, May 1993.

[37] M. Effros and D. Muresan, "Codecell contiguity in optimal fixed-rate and entropy-constrained network scalar quantizers," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 312–321.

[38] S. Dumitrescu, X. Wu, and G. Bahl, "Fast algorithms for optimal two-description scalar quantizer design," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2004, pp. 42–51.

[39] S. Dumitrescu and X. Wu, "Algorithms for optimal multi-resolution quantization," *J. Algorithms*, vol. 50, pp. 1–22, Jan. 2004.

[40] S. Dumitrescu and X. Wu, "Lagrangian global optimization of two-description scalar quantizers," in *Proc. 2004 IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 301.

[41] C. Tiam and S. S. Hemami, "Universal multiple description scalar quantization: analysis and design," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 2089–2102, Sep. 2004.

[42] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, pp. 13–30, 1963.

[43] N. Cesa-Bianchi and G. Lugosi, "On prediction of individual sequences," *Ann. Statist.*, vol. 27, pp. 1865–1895, 1999.